

# An Approach for Deadlock Prevention in Decentralized Controlled Pre-storage Zones with Dynamic Behavior and Time Windows

Lukas Kopecki

**Abstract**—This paper introduces an approach for deadlock prevention of container handling in decentralized organized pre-storage zones. Decentralization in pre-storage zones lack the ability of global state information, time invariants and non-deterministic behavior by algorithm dependency on communication with other entities in the system. Therefore, an approach is introduced which is capable of handling container over transport units with the possibility of inverse flow detection and securing system consistency by deadlock prevention.

**Index Terms**—Industrial engineering, optimization of systems, automation, design and manufacturing.

## I. INTRODUCTION

Internet of Things, cyber-physical systems or Industry 4.0 are expecting to offer different solution for the operation of many existing industrial systems as transportation or manufacturing systems [1].

Current approaches focus on centralized control which lacks in scalability and flexibility, caused by the strict hierarchical order of control in pre-storage zones in warehouses. The building blocks for controlling each element in a pre-storage zones are already modularized, in hardware and software specifying the behavior of one element. But the interaction in one working group (several elements combined) has to be implemented manually due to the heterogenic project specific structure.

Furthermore, logistic systems like the pre-storage zones facing different challenges, e.g. complexity in product individuality, dynamic in transportation and interaction between the goods, information's causing a higher communication effort due to the dynamics in transportation and hierarchical aspect each level solution or computation is depended on other levels [2].

To overcome the dynamics in transportation in decentralized organized pre-storage zones an opportunistic routing algorithm with discrete-time slot reservation is needed to approximate the expected arrival time and react on time shifts local while transporting the container on one transportation unit (TU). Furthermore, this routing algorithm has to ensure the system stability with a deadlock prevention algorithm to enable continuous routing of container.

This paper focus is the deadlock prevention in

opportunistic routing and is organized as follows. Section II introduces the background and the field of work. Section III introduce the used method for deadlock prevention in time-discrete opportunistic routing and Section IV gives a conclusion and further challenges in research.

## II. BACKGROUND AND CURRENT STATE OF THE ART

### A. System Structure

Current decentralized controlled warehouse systems are using distributed small logical units, which are capable to compute simple tasks to fulfill an overall goal.

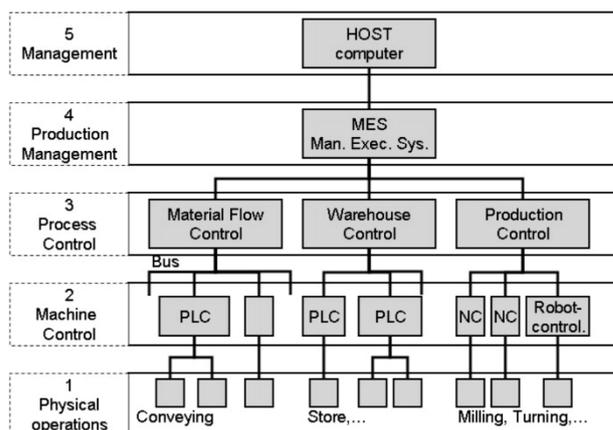


Fig. 1. Current structure in automatized control [3], [4].

Fig. 1 shows the current automatized structure in hierarchical systems used in warehouses specialized in distribution. For transportation control of container, level three, the material flow, control (MFC) down to level one for physical operations are mainly used for transportation.

Level three determines the route and organization of the route execution, capable to communicate with programmable logical controller (PLC) on level two.

The route determination is based on the underlying algorithm which focus lays on a predefined factor during the project acquisition phase or planning phase in interaction between the warehouse provider and customer. The interface is a bus (for the interested reader see [5], [6] or [7]) system which enables the communication between the MFC and the PLC on Level two.

In Fig. 2 devices are displayed for controlling the physical level e.g. conveyor, milling, turning, with PLCs (Control Level) by using external devices (Remote I/Os) to send and receive the analog / digital signals for the motor or the sensors to compute the actions in the next operational step.

Manuscript received May 15, 2015; revised July 23, 2015.

Lukas Kopecki is with University of Paderborn, Business Computing, CIM, Paderborn, Germany (e-mail: kopecki@hni.uni-paderborn.de).

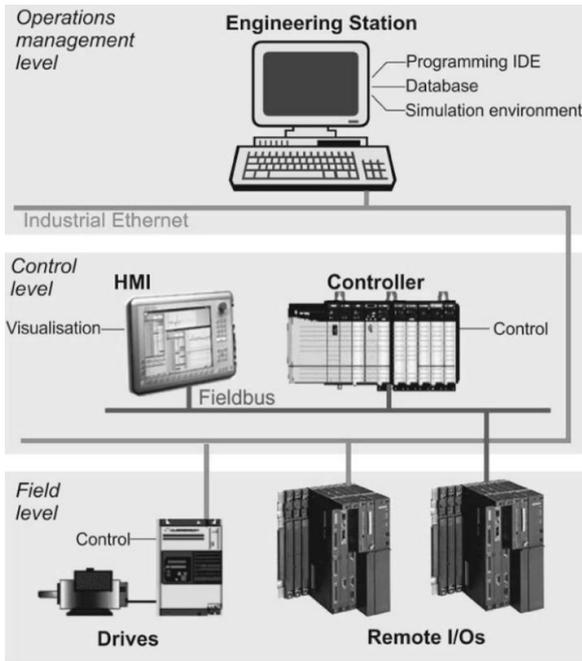


Fig. 2. Hardware architecture of factory floor automation systems on machine control and physical operation level [8].

This approach is strong hierarchical and needs high effort in customization in the planning phase and afterwards in build phase extensions.

Therefore, the current focus in research is the decentralization of the control at the PLC and MFC level to flatter the organization structure to increase the scalability.

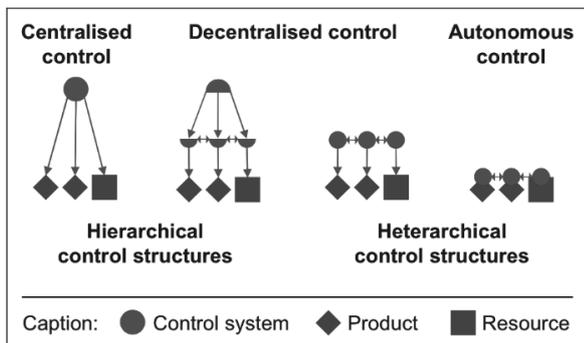


Fig. 3. Harmonization of control structures [8].

Fig. 3 shows the harmonization of the control structure from centralized hierarchical control structures to autonomous heterarchical control structures.

In contradiction to a hierarchical control structure in heterarchical control structures the system and his subsystems are autonomous. To ensure operability each subsystem has a specific degree of self-organization and communication with his surrounding environment to enable the execution of higher task interacting with others.

In warehousing especially in the distribution of goods each subsystems consists of one control unit and one specific physical hardware for transporting, sorting or distributing the goods.

**B. System Structure in Practice**

For transporting only continuous conveyor with motor and binary sensors based on light are considered for controlling. The control unit is capable to parameterize the motor based on the sensor data and information's given by other TUs. The

sorting and distribution is done by right angle decks, capable to shift the goods in a 90-degree manner to other TUs.

A PLC is the main control unit in industrial automation system. They are connected with the sensors and actuators via direct electric wiring to their input / output I/O ports or via remote modules [9].

In heterarchical structures the PLC is directly mounted on the TU and is wired with the I/Os, no remote modules are needed. The communication is done by real-time Ethernet capable protocols. Therefore, the PLCs are connected with regular industry network cables to reduce the overall cost.

The control software of PLCs are function oriented and guaranteed a scanning cycle of up to 50  $\mu$ sec. Their advantage is a dedicated hardware or software reading the inputs, applying the control logic and writing the outputs [10].

**C. Time-Discrete Routing in Pre-storage Zones**

Warehouses focused on distribution with a small dwell time of goods are related to a fast routing algorithm. Such types of warehouses have a high throughput and a high dynamic behavior in transportation of goods. The dynamic comes from interaction between the transportation goods by blocking each other's. This blocking cumulates over the goods and can dramatically decrease the service grade of a warehouse.

The operative cost of a decentralized system is depended on the running cost of the TUs, the waiting time and transportation of the units. Which can lead to a stop and go behavior with increased power consumption.

The routing algorithm used for determining the path during execution is a time-discrete decentralized opportunistic route algorithm. This routing algorithm is separated in three different parts: routing, communication and cost tracking, displayed in Fig. 4.

The routing part (Fig. 5) is a decentralized scheduler storing the arrival, estimated execution time and end time. The scheduler receives the needed data from the path finding algorithm which is based on the communication and cost estimation modules. The time slots in the scheduler are giving an expected time a container could arrive and dependencies between other container to estimate deadlocks and for preventing deadlocks at a TU.

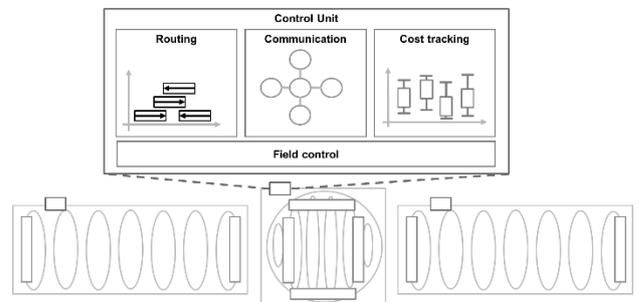


Fig. 4. Routing algorithm in one decentralized PLC unit controlling a right angle deck.

The execution of a container is done in a FiFo manner. When a container arrives the TU are checked if a deadlock could occur if this container is handled immediately. If a deadlock free situation exists, the container will be handled. Otherwise a dependency to this specific timeslot will be stored and after all dependencies are resolved the container

will be handled.

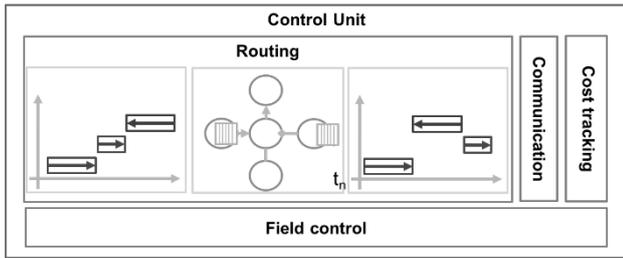


Fig. 5. Routing module with decentralized Scheduler for container arrival estimation.

The communication module (Fig. 6) handles the communication to other PLC units in the system. Data packages are being forward to the destination or handled by the PLC. The main task is path finding for a container entering the system.

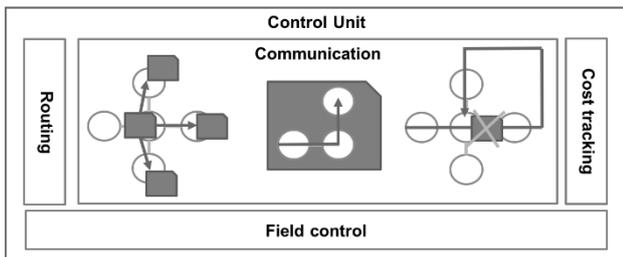


Fig. 6. Communication module for handling messages and for path finding.

This is done by multicasting a message with the path already visited. Each TU adds his current estimated costs when using this specific TU. If the message arrives at one sink which is not the destination the message will be dropped. If the TU already computed the costs for one container, the message will be dropped too, to ensure a loop free path. If the message arrives at the destination it will be send back to the source, based on the stored path. The source itself waits for a determined time in prior for a path and chooses the one with the lowest transportation cost.

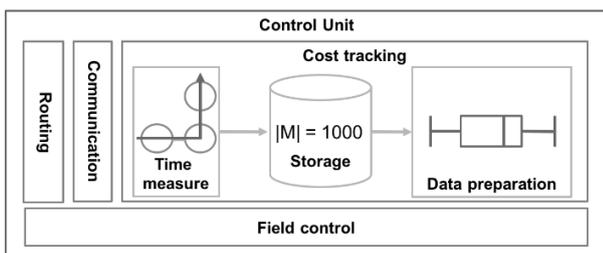


Fig. 7. Cost tracking for dynamic cost adjustment based on the handling time.

The Cost tracking shown in Fig. 7 module determines the costs for one transportation task done by the TU. This is done by measuring the time over the lifecycle of the TU in the system handling the container. If a cost estimation is needed for path finding the cost tracking module computes the lower quintile based on all stored data. The lower quintile ensure that the shortest path will be preferred instead of detours.

### III. METHOD

This section describes the used method to prevent deadlocks in opportunistic routing in decentralized organized

pre-storage zones.

Opportunistic routing (OR) uses the underlying behavior as an opportunity instead of a limitation [11]. Therefore, the method is considered in taking advantage of the current situation at a specific time period at a TU by analyzing based on a simplified model of the situation and to make a decision in a proper time window.

In OR a set of candidates will forward the message to the destination given by a metric determining the candidate for forwarding [12]. In contradiction to packet transportation in wireless network the transportation of containers is more crucial given by the fact that a container cannot be physically destroyed, duplicated or resent like packets in wireless network communication. Therefore the OR metric has to consider deadlocks or prevent deadlocks during the runtime to ensure the system consistency.

This chapter describes a way to ensure the system consistency during runtime given by defined routes given by the OR algorithms. This is done first by estimating and evaluate the local information a TU is able to access. And based on TUs local information making a decision for a container to wait or to transport.

#### A. Situation Estimation

When a container arrives at a TU a decision has to be made. Therefore the current local situation based on the available information's at the specific TU has to be made.

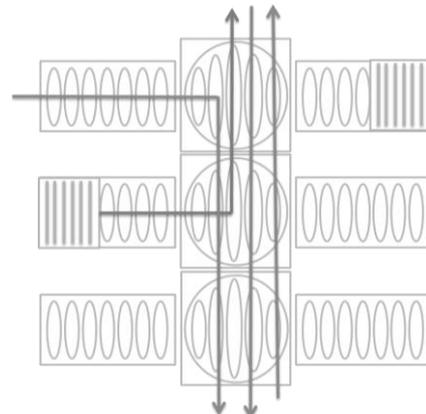


Fig. 8. Local route information at one TU.

Fig. 8 shows the local information one TU stores for decision making. All schedules routes are stored with their handling time representing by a time window.

If a container arrives at a TU able to distribute the container in the system (every TU with more than 1 output), the Situation estimation is divide as follows:

- 1) First inverse flows will be searched.
- 2) If no inverse flow exists the container will be transported, otherwise the following steps will be computed.
- 3) The union digraph of the flow and the inverse flow will be calculated.
- 4) Flows with time scheduled between the current and the inverse flows will be approximated.

The first and second step will be done pragmatically by a regular search engine, searching the scheduler for any not handled task with inverse flows.

The third step is determining the digraph representing the inverse flow.

This is done by a sequential edge comparison of both digraphs.

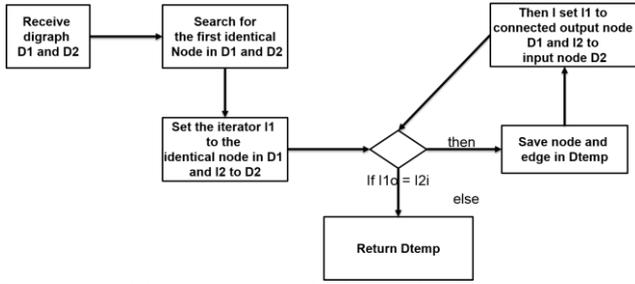


Fig. 9. Sequential edge comparison for finding digraph representing critical flow.

Fig. 9 shows the structure of finding the critical flow (inverse flow between to transportation tasks). After receiving both digraphs D1 and D2 the search engine looks for the first identical match in both graphs. After finding one two iterators I1 and I2 are set to the identical node. The next step is the comparison between I1 and I2 if both are the same both iterators are set to the next node for checking. If I1 and I2 are different the temporary build critical flow digraph will be return and the computation aborted.

### B. Time-Window for Decision Making

After determining the critical flow the TU has to make the decision for the container to wait for the other container or block this partial part of the route and execute the transportation. The decision will be made based on the approximated transportation time of both container over the critical path. The container with the lower absolute transportation time over the critical path will be executed first and other containers will be blocked.

This will be done by calculating the approximated transportation time of the container over that partial part in the routing.

$$C_1 = \sum_{i=0}^n \Delta D_{1i} \text{ with } \Delta D_1 = D_1 \cap D_{temp} \quad (1)$$

$$C_2 = \sum_{i=0}^n \Delta D_{2i} \text{ with } \Delta D_2 = D_2 \cap D_{temp} \quad (2)$$

Eq. (1) and (2) are calculating the expected costs for transporting one container. If  $C_1 < C_2 + \Delta_t$ , with  $\Delta_t$  as the time when the other container at the critical path arrives, the TUs will be blocked to transport the container over the critical path.

The cost for one particular node in the digraph, e.g.  $\Delta D_1$  or  $\Delta D_2$  is calculated by the lower quartile (Q25) in the OR metric in prior when determining the next hops for the container over the TUs.

If two containers arrive at the same time at the TU for transportation with an inverse flow, the underlying network scheme decides which one will be served first. This is due to the determinism behavior of network communication given by static message parsing. This means, the message arriving first for the handshake between two TUs has a higher priority

and the later message for the handshake will be dropped.

During transportation (while blocking another container) an inverse flow with a third container can happen. A nested calculation will be performed for the new container with the new inverse flow. A deadlock free behavior between nested container cannot occur due to the fact that only container with a transportation task at a specific time are able to block other containers. Therefore, there must exist at least one container with a deadlock free partial route through the critical flows which is able to be handled by the TUs.

The nested blocking situation will be dissolve by recursively transporting the next container. When the container arrives at the TU at the end of the critical path it will release the next container with the inverse flow. If the freed container arrives at the last TU at his critical path he will release the blocking container. In this sequential manner the nested blocking will be resolved.

## IV. CONCLUSION AND FURTHER WORK

The field of controlling decentralized TUs and ensuring a deadlock free transportation is of vital importance. Since decentralization is already implemented in current systems to increase their dynamic and flexibility in transportation or production. But still a general approach is missing in current research for routing containers or goods in the system controlled by PLCs.

The simplicity allows the implementation in industrial PLCs with limited computation and storage which enables a higher acceptance in the industry. And a higher throughput compared to currently routing algorithm for container transportation.

Further work focus on improving the decision making algorithm to increase the overall optimization, instead of a decision making based on First in Last out manner. Therefore a simulation framework is needed capable of simulating the communication and organization of the controller in a decentralized control system. This provides more insight in the behavior of PLCs in decentralized controlled systems.

## REFERENCES

- [1] L. D. Xu *et al.*, "Internet of things in industries: A survey," *IEEE Trans on Industrial Informatics*, vol. 10, no. 4, Nov. 2014.
- [2] H. Rekersbrink, "Methoden zum selbststeuernden routing autonomer logistischer objekte," Ph.D. Dissertation, Dept. Productionstechnology, Bremen Universtiy, Bremen, December 2012.
- [3] D. Arnold and K. Furmans, *Materialfluss in Logistiksystemen*, Heidelberg, Berlin: Springer-Verlag, 2009, p. 9.
- [4] S. Mayer, "Development of a completely decentralized control system for modular continuous conveyor systems," Ph.D. Dissertation, Dept. Machine Engineering, Karlsruhe Univ., Karlsruhe, 2009.
- [5] B. Galloway and G. Hancke, "Introduction to industrial control networks," *IEEE Communications Survey & Tutorials*, vol. 15, no. 2, 2015.
- [6] L. Durkop *et al.*, "Using OPC-UA for the autoconfiguration of real-time ethernet systems," *Industrial Informatics*, 2013.
- [7] G. Reinhart *et al.*, "Automatic configuration (plug & produce) of industrial ethernet networks," *IEEE/IAS International Conference on Industry Applications*, 2010.
- [8] B. Scholz-Reiter and M. Freitag, "Autonomous processes in assembly systems, CIRP annals," *Manufacturing Technology*, Januar 2007.
- [9] V. Vyatkin, "Software engineering in industrial automation: State-of-the-art review," *IEEE Trans. on Industrial Automation*, 2013.
- [10] P. Ferreira, S. Doltsinis, A. Anagnostopoulos, F. Pascoa, and N. Lohse, "A performance evaluation of industrial agents," *Industrial electronics Society IECON*, 2013.

- [11] N. Chakchouk, "A survey on opportunistic routing in wireless communication network," *IEEE Communications Surveys & Tutorials*, IEEE, 2015.
- [12] A. Boukerche and A. Darehshoorzadeh, "Opportunistic routing in wireless networks: Models, algorithms and classifications," *ACM Computing Surveys*, 2015.



**Lukas Kopecki** is a research fellow with the International Graduate School in Paderborn and is now researching at the University of Paderborn in the Department Business Computing especially CIM.

He started his studies in Iserlohn at the University of Iserlohn in the field computer science with the specialization in computer architecture. After an abroad study at the California State University in Hayward with the focus electronic engineering he

moved to Paderborn for graduate studies at the University of Paderborn in computer science for his master degree in embedded systems. He is currently a Ph.D. student at the University of Paderborn at the Department Business Computing especially CIM with focus on simulation and decentralization of control software/hardware in logistics.

Lukas Kopecki is member of the IGS fellowship in cooperation with lodige industries GmbH focusing the interdisciplinary research between university and the company in planning air cargo terminals worldwide.