ISSN 1846-6168 (Print), ISSN 1848-5588 (Online)
https://doi.org/10.31803/tg-20230511165415

Preliminary communication
Received: 2023-05-11, Accepted: 2023-05-15

Towards Logistics 4.0: A Skill-Based OPC UA Communication between WMS and the PLC of an Automated Storage and Retrieval System

Andreas Lober*, Lisa Ollinger, Sven Voelker, Hartwig Baumgaertel

Abstract: In order to bring intralogistics systems to the same level of interoperability as today's modern production systems, logistics must take the essential steps towards Industry 4.0. This requires an increasing abstraction level of control logic as an enabler for horizontal and vertical integration. The abstraction will lead to the interconnection of manufacturing and logistics control with the production planning and warehouse management systems (WMS). A main enabler for these communication paths are service-oriented architectures (SoA). OPC UA has established itself as a widely used and already adopted SoA-based communication standard in industry. The paper describes the realization of an OPC UA-based approach for the communication between a WMS and a PLC of an automated storage and retrieval system (ASRS). The conceptual basis of communication design are skills of the ASRS. The work is supported by an architectural design with a subsequent prototypical implementation.

Keywords: Control Logic; Industry 4.0; Logistic System; OPC UA; Skill-based Engineering

1 INTRODUCTION

The field of intralogistics is experiencing considerable progress in terms of flexibility and adaptability addressed by Industrie 4.0. Replacing individual interfaces with standardized interfaces is a central goal of Industrie 4.0 to enable a holistic network architecture. It has become apparent that production and logistics still have considerable hurdles to overcome for reaching this target [1]. Resources from different system providers are often used for logistics subsystems, which means that a large number of different interfaces must be managed. To overcome this interface problem, an increasing degree of abstraction of the control logic is required as an enabler for horizontal and vertical integration. Increased abstraction enables a linking of technical manufacturing and logistics control with production planning and warehouse management systems. Service-oriented architectures (SoA) are an important enabler for a flexible and adaptive communication pattern.

Concerning the vertical dimension of communication, there is still a noticeable gap between the control layer and higher automation functions for monitoring the production plan [1]. This discrepancy is reflected in different requirements in terms of timing constraints and synchronous/asynchronous communication patterns. OPC Unified Architecture (OPC UA) has emerged as a widely used and already adopted SoA-based communication standard in industry. Traditional communication approaches between Operations Technology (OT) and IT systems cause high development effort, long commissioning times, and inefficiencies, as they require individual arrangements by programmers on both sides for each individual interface. Despite extensive scientific work and trade show demonstrations over the last decade, general migration concepts have not been established, and experts who can perform such a transformation are rare. The migration of new approaches in legacy technical systems into cyber-physical systems poses various challenges, such as non-standard control logic or the use of different programming languages and end devices.

In order to be able to simulate such a scenario in an industry-oriented way and to implement a skill-based control logic via OPC UA, this paper considers the technical implementation of the conversion of an automated rack storage system (ARSS) for small receptacles according to the concept of skill-based engineering (SBE). For this purpose, the original control system, consisting of an industrial PC with Soft PLC, ProfiBus PCI card, and a proprietary Warehouse Management System (WMS) was replaced by a PLC and a commercial WMS. In the next step, a concept for the conversion and transition to a modern, networked architecture was developed. In this concept, OPC UA is used as the communication and information standard. Hence, a Siemens S7-1500 PLC was chosen as the central control unit, as this PLC type is equipped with an OPC UA server. The information model for this server was designed based on a skill model for the ARSS. The last step, for now, is the implementation of a new WMS, which communicates with the ARSS' PLC exclusively via OPC UA. User activities are translated by this WMS into calls of OPC UA methods at the PLC. The paper presents a successful implementation of OPC UA based on the SBE approach, which is not based on OPC UA programs like many existing approaches, but on OPC UA methods. As these methods are applied to automation functions, a so-called skill-based control concept is realized. This allows to use digital twins (DTs) from the beginning of the engineering process [2]. Those DTs can proactively negotiate an optimal production mode that choreographs production capabilities within a market scenario [3]. In addition, the advantages and disadvantages of implementing this approach are discussed.

After discussing related work in Section 2, the authors present the use case of an automated rack storage system in Section 3. After presenting the skill model and communication architecture in Section 4, the implementation of this architecture is described in Section 5. In Section 6, the presented approach is discussed and the advantages and disadvantages of the implementation are presented. Finally, a conclusion is drawn in Section 7 and an outlook on future work is given.

2 RELATED WORK

Research has shown that several concepts built on a service-oriented architecture (SoA) are promising for representing production systems knowledge and capabilities [4]. The SoA concept is characterized by higher flexibility in contrast to the traditional component-based hierarchical and less complex approaches [5]. SoA is a type of possible software design that modularizes previously monolithic IT systems. It is based on the concept of callable services. These services must provide access to the capabilities of a software or hardware system through a predefined interface through a service description [6]. SBE is described as a possible implementation for the implementation of SoA callable services for industrial applications [7]. SBE covers an extensive research area, and therefore approaches dealing with the skill description of production machines are discussed in the following.

2.1 Skill-based Engineering in Production

The research area of automation technology shows that the approach of skill-based engineering is becoming increasingly important. Many approaches focus on mapping capabilities in such a way that possible interfaces are simplified and communication technologies can be easily integrated.

Zimmermann et al. [7] describe three prerequisites for the successful introduction of skill-based engineering in the creation of PLC programs. For this, the machine manufacturer provides SBE automation components and the application engineer orchestrates the provided skills and creates the final process and program flow. In addition, a cross-manufacturer standardization of skills is critically considered. Dorofeev [8] shows a generic interface with specific functionalities in terms of the details of SBE. A metamodel of capabilities represents semantic information. Perzylo et al. [9] describe a knowledge-based digital SBE concept for semantic integration of various data and information along the value chain of manufacturing companies. In Köcher et al. [10], a method for automating and supporting the development of machine capability and capability models is outlined. The presented method enables mechanical engineers and machine developers to create formal models of machines and their abstract capabilities without additional effort. A further work that builds on this by Köcher et al. [11] derives semantic skill models from existing PLC code. However, a holistic approach would derive a PLC code hierarchy from a semantic skill model to derive the skill-based PLC code directly.

According to the literature, a skill-based control logic structure that uses all aspects of OPC UA offers the greatest advantages in the implementation of production systems. Therefore, further consideration will focus on approaches that combine SBE with OPC UA.

2.2 OPC UA & Skill-based Engineering

The combination of the skill-based engineering approach and OPC UA technology has already been used in previous work. To be able to classify the present work, the best-known approaches of SBE and OPC UA are presented below.

Dorofeev and Zoitl [12] demonstrate an SBE approach for OPC UA programs using a state machine to describe the execution logic of skills. As a possible future work, the authors mention the implementation of a skill as a pure OPC UA program. Dorofeev and Wenger [5] compare a SBE approach with a classical hierarchical control logic approach. Despite the capability-based architecture using nonreal-time communication via OPC UA, the performance is high compared to the traditional approach according to IEC 61131-3. Zimmermann et al. [7] do not only focus on the use of SBE for technical purposes but use a SBE control and show that such an architecture can, in principle, be realized with OPC UA. The paper highlights that the current implementation is mainly based on the client/server model of OPC UA and the use of methods over TCP/IP limits the application possibilities due to the lack of real-time synchronization between different components.

Spitzer et al. [13] propose an approach in which the core concept of SBE programming is implemented via capabilities, variables, configurations, and messages. In doing so, the functionality of a device is moved to an abstract level. This abstraction supports a clear separation between orchestration and the subsequent execution of production processes. Profanter et al. [14] present a generic system architecture for a plug & produce system. OPC UA is chosen as the basis for this system. Using a combination of OPC UA's decentralized discovery mechanism and a skill detector, the system automatically detects newly added components and their skills. Jhunjhunwala et al. [15] show that building a SBE architecture using the digital twin concept defined in IEC 61499 with OPC UA is a feasible approach as an enabler for the capability structure of a production system.

A feature in [13] is the use of a specific OPC UA part, the OPC UA programs. Other approaches are using OPC UA methods based on services that are offered by the devices. These approaches call services via OPC UA, whereby the complexity of the control logic remains implemented on the PLC, since the services are programmed as methods on the PLC. In contrast to those approaches, this article will focus on implementation without OPC UA programs and complete services as OPC UA methods, and shows that it is possible to successfully implement a control system for the ARS system based on so-called basic skills that are implemented as OPC UA methods. Basic skills represent basic capabilities of the device that are needed, for example, to offer a certain service. These basic skills are put into a specific order and are executed sequentially or in parallel. The cited work describes the application of the approaches in scenarios where the work of three types of experts is required. The work presented here offers a different approach that has been implemented in a student project. However, as preparation, the capabilities had to have been implemented as callable OPC UA methods prior to the student project. Because otherwise it would not have been possible for the student group to implement their solution, since they had no knowledge about PLC programming and were not supposed to work on the PLC.

3 USE CASE – AUTOMATED RACK STORAGE SYSTEM

An automated rack storage system (ARS) for small receptacles is the considered use case of this paper (Fig. 1). The ARS system consists of one row of racking with 11 rack columns and 9 rack levels, where each compartment can store one small standard bin. The bins are stored and retrieved using an automated stacker crane (ASC), which has a load carrying device (LCD) mounted on its mast. Two belt conveyors connect an operator workplace with the ARS, one for the bins to store and one for the retrieved bins. On the operator workplace side, a transfer carriage connects the two conveyors. Between the conveyor belts, several airlift pumps are installed that can lift boxes from the belt and fix them in a specific position each. This realizes the functionality of accumulating conveyors for the belt conveyors. After the introduction of the new Siemens PLC at the operations technology level, the ARSS was operated at the user level by a commercial WMS. This WMS and the PLC exchanged commands and data by ASCII files called telegrams, which are transmitted via the TCP protocol. Both systems used a Siemens software library to handle telegrams. This library contains routines for buffering, parsing, and interpreting incoming telegrams, and creating and sending telegrams to the other party. Typical action orders are "store a bin to a specified bin location", "retrieve a bin from a specified bin location", and "read the ID barcode of the bin at a specified bin location". A WMS user initiates these orders via the WMS graphical user interface. A problem of the used standard telegram software library is that it offers many parameters for each command whose semantics is not obvious in many cases. This causes high effort at commissioning as experts for PLC programming and WMS programming need to agree on the usage and semantics of parameters for all used telegram types.

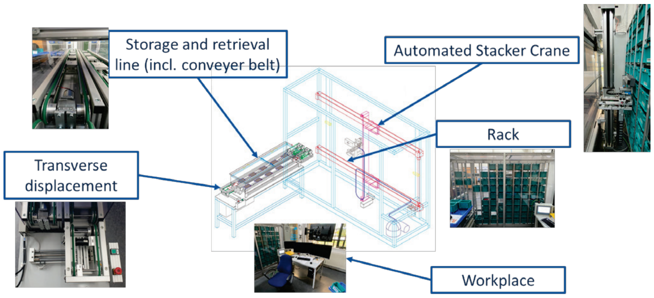


Figure 1 Automated Rack Storage System

In most cases, the PLC sends telegrams to the WMS as responses to received telegrams, e.g. orders or status inquiries. However, the PLC sends a telegram proactively when a bin arrives at the store-in location. Then, the WMS reacts with corresponding action order telegrams. The action orders from the WMS are split by the PLC into several basic activities, like moving the ASC in *X* and *Z* directions, moving the load carrying device towards the rack or backward (*Y* direction), grasping a bin by the LCDs fork, and releasing a bin from the fork when it is placed. Some of these actions can be performed in parallel, like movements in the *X* and *Z* direction. Others require an exactly planned sequence of actions, such as grasping a bin as a combination of movements in the *Y* direction, the *Z* direction upward to lift the bin from its location, and back in the *Y* direction to pull the bin out of the rack. The sequence of actions needed to store a bin is distributed over PLC and WMS. After a new bin is detected at the store-in location by an occupation sensor, the ASC has to move to this location, read the bin ID from a barcode at the front of the bin and pick the bin to its fork.

Then, PLC and WMS exchange telegrams with the bin ID and, if valid, the target bin location. After that, the ASC moves to the target location, checks if it is free, and, if yes, finally puts the bin to this location. Depending on the operation strategy, the ASC can then proceed with retrieving a bin or moving back empty to the store-in location. In most cases, the decision on where to put a bin and what to do as the next step is made by the WMS. However, other parts of the action sequence could be either hard-coded in the PLC or initiated by the WMS. The PLC and WMS programmers agree on how this is organized during the commissioning of the ARSS. As more than one order to the ARSS may be due at a time, the WMS maintains an order queue, as the ASC can process only one order at a time and needs time for that.

To demonstrate the skill-based OPC UA approach to communication between the WMS and the PLC of the ARSS described above, a new WMS was designed and developed by the project team. For this development, the following requirements for the WMS were derived:

- An interface for OPC UA must be established between the WMS and the ARS that enables various services such as "putaway", "retrieval" and "relocation" based on skills.
- The WMS communicates with the ARS exclusively via OPC UA. No other communication channel may be used.
- The WMS provides an inventory function, updates its database accordingly, and sends all non-readable containers to the operator workstation.
- The WMS provides an identification point, i.e., functionality that allows a bin ID to be associated with a material ID of the bin contents.
- The WMS focuses on the operation of the ARS and does not cover other process areas in real warehouses such as goods receipt or picking.
- The data model of the WMS must be as simple as possible, but as realistic as necessary. It must separate the master data of the rack, storage bins, and material from the movement data. It should be possible to change master data via special GUI views restricted to authorized users.

The following chapter describes in detail the underlying skill model of the ARSS and the corresponding communication architecture.

4 SKILL MODEL AND COMMUNICATION ARCHITECTURE

In order to be able to handle a skill-based control logic via an OPC UA communication architecture, the ARS must first be defined in terms of its services and skills. For the definition of individual skills, a distinction must first be made between processes and tasks (or services).

In this case, the ARS presented has three tasks: Storing, Retrieving, and Relocating. These different tasks can be realized by a specific sequence of skills. In the Storing task example, the following skills are called first:

1) Convey bin
2) Check occupation of storing location
3) Move Stacker Crane
4) Identify bin
5) Pick bin
6) Move Stacker Crane
7) Check occupation of target location
8) Put bin.

In that case, the third and sixth skills are so-called combined skills, which are a combination of two skills performed in parallel. The skill model thus distinguishes combined skills from simple ones. A combined skill corresponds to skills that execute different capabilities in parallel, e.g. to realize a new type of skill. In the example shown, this corresponds to the diagonal movement of the stacker crane, which is realized by the parallel execution of the movement in the *x*- and the *z*-direction. By combining these eight standard skills, the three services mentioned can be realized in the use case presented. The services differ in the orchestration of the skills. For the realisation of a service it is possible that certain skills are not required or are called in a different order compared to another service. The complete skill model is shown in Fig. 2. Based on this skill model, the control logic in the PLC of the ARS system was adapted. For this purpose, the OPC UA server was set up on the PLC and the information model was built based on the skill model.

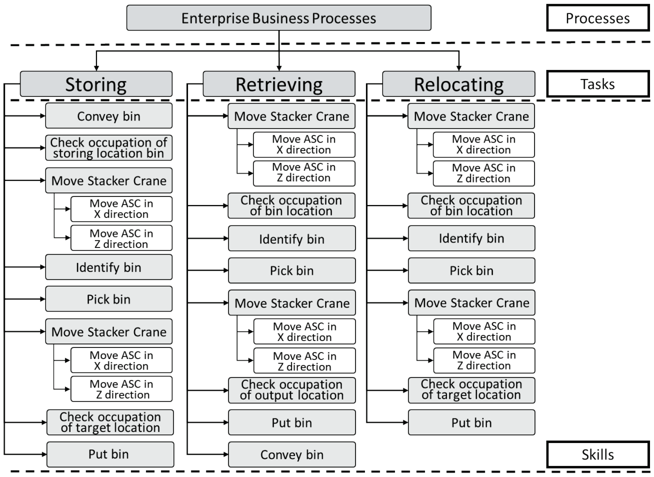


Figure 2 Skill model of the ARS

While the connection of PLC variables with variable nodes in the IM is simple and straightforward, the integration of callable methods requires the implementation of specific wrapper function blocks, whose main task is to transfer the OPC UA method parameters to the PLC code and vice versa. Based on the skill model and system constraints, finally, sixteen callable OPC UA methods are implemented. They can be classified into five groups:

- Methods to be executed by the ASC (including the LCD), e.g. "Put bin to", "Pick bin from", "Read barcode of bin at position"
- Method to be executed by the conveyor at the bin retrieval position, "Move bin to outbound conveyor"
- Methods to be executed by the conveyor and the airlift pumps in bin retrieval direction, e.g. "Move bin from pos L to pos L+1"
- Methods to be executed by the conveyor and the airlift pumps in store-in direction
- Methods to be executed by the transfer carriage at the end of the conveyors, e.g. "Pick bin and move in front of storing conveyor", "Put bin".

The OPC UA methods and the skills from the skill model are not identical. The ASC methods aggregate several basic skills. For example, the method "Pick bin from" consists of the basic skills Move Stacker Crane, Identify bin and Pick bin, and the method "Put bin to" of Move Stacker Crane, Check occupation of target location and Put bin. These aggregated skills occur in all three task processes of the skill model.

That is, they can be seen as reusable building blocks for the operation workflows. On the other hand, the skills from the skill model regarding the conveyors needed to split into several OPC UA methods, as the accumulating behavior of the conveyors was not considered. The skill Check occupation of storing location is not realized as an OPC UA method but can be mapped to a subscription to the corresponding variable by OPC UA clients.

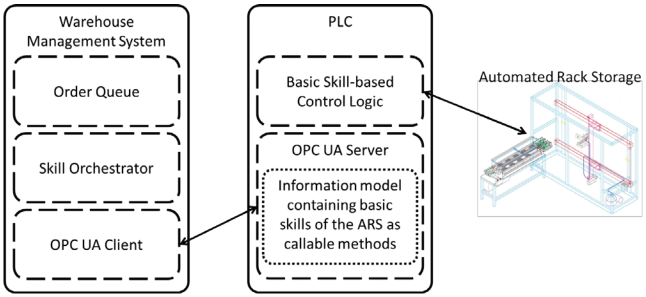

Warehouse Management System
Order Queue
Skill Orchestrator
OPC UA Client
PLC
Basic Skill-based Control Logic
OPC UA Server
Information model containing basic skills of the ARS as callable methods
Automated Rack Storage

Figure 3 Communication Architecture

Based on these two adjustments to the system, the following communication architecture between the WMS and the PLC can be built (see Fig. 3). The implemented control logic within the PLC can be divided into two areas. First, the basic skill-based control logic is implemented in the PLC in the code, which ensures that the PLC only executes functions based on basic skills. Second, the PLC contains the OPC UA server with an integrated information model that integrates various skills of the ARS system as callable methods. The PLC is connected to the physical ARS via a standard Ethernet port. Its counterpart is the WMS, which has a total of three components: the Order Queue, the Skill

Orchestrator and the OPC UA Client. The order queue manages the orders that are placed with the ARSS and stores them temporarily so that the orders can be processed according to a certain logic.

Based on the information model of the ARSS, the Skill Orchestrator can split the tasks into individual skills and send them as method requests via the OPC UA Client to the PLC.

Fig. 4 shows an example communication flow of the WMS with the PLC via OPC UA between the frontend, backend and PLC. If certain tasks, such as a "store-in bin", are to be managed via the WMS, a command is first transmitted to the back end via the front end through REST. The process request is started in the back end by transmitting a request ID to the front end and triggering the necessary skills via OPC TCP. By issuing commands via the backend of the WMS, the PLC starts the physical processing of the request by controlling the actuators. After a certain time, the backend iteratively checks the status of the request by issuing a command to the PLC via OPC TCP to check the status. The PLC passes the current status of the skills to the backend, which in turn updates the status in the WMS. The frontend can iteratively query the status via the request ID or receives a response about the current status when the process is fulfilled or accepted.

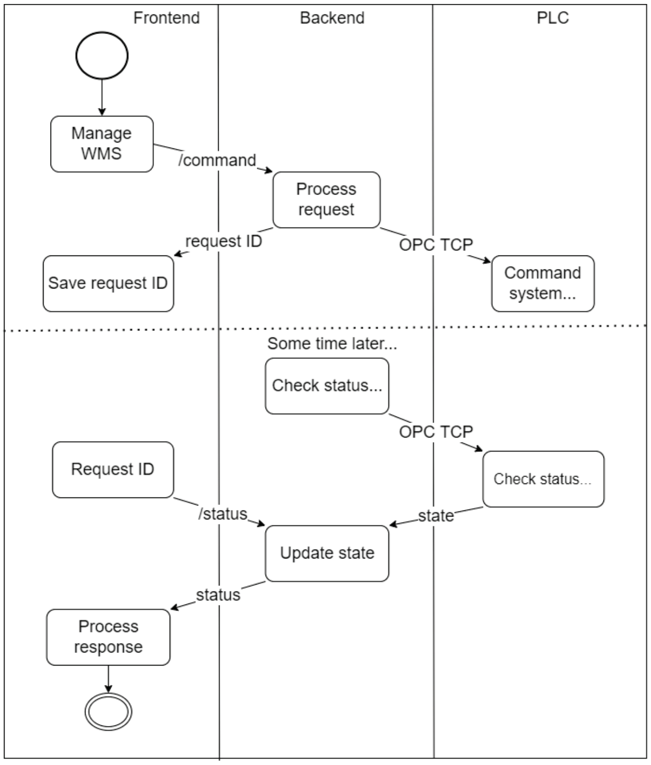


Figure 4 Basic Communication principle

5 IMPLEMENTATION

The WMS system of the ARS is divided into a backend and a frontend part (see Fig. 5). The frontend realizes the Graphical User Interface (GUI) as a thin web client written in VueJS. A Flask web server serves it, which is part of the backend system. The server offers a RESTful API to the web client. The backend system is written in Python and contains, beneath the Flask server part, a database interface to a SQLite database, an OPC UA client. The core logic of the program contains an order queue for user orders from the GUI, procedures that translate user orders into sequences of skills, and the basic methods to call OPC UA methods on the PLC that realize single skills. The Graphical User Interface (GUI) consists of a grid part on the left-hand side and a table and activity part on the right-hand side (see Fig. 6).

The grid part offers several views of the rack content and interacts with the table view, which offers functionalities like

- search for bins or material,
- user actions like retrieving, relocating, or storing bins,
- ARS maintenance like banning or releasing storage locations, assigning material numbers and amounts to bins (the so-called identification point or I-point)
- assigning storage locations to material numbers for fixed position storing strategy, and
- settings of the WMS like the choice of a storing strategy (like fixed position storing, chaotic storing, storing in ABC zones according to the turnover frequency of the material, and manual determination of storing locations).

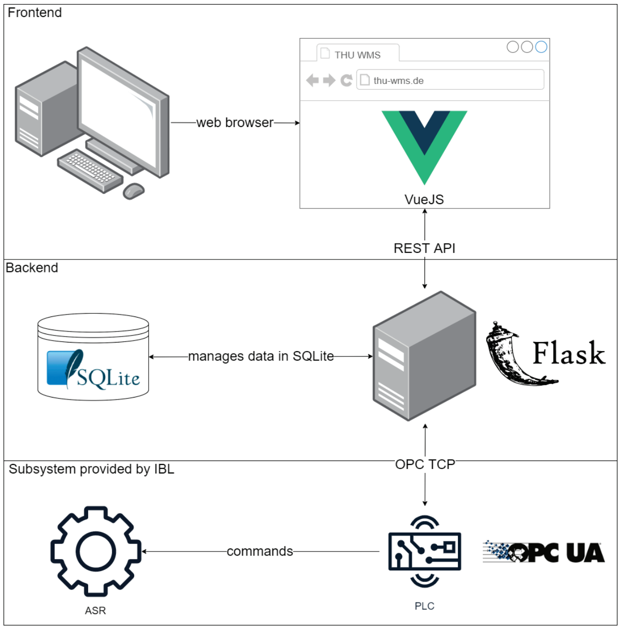
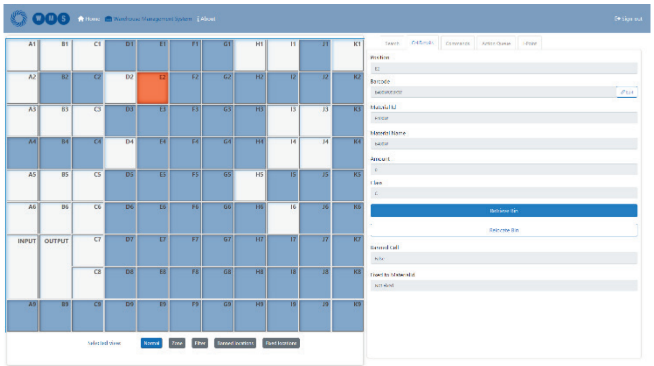


Figure 5 System architecture of the WMS, overview

The grid views contain visualization of the results of the current search, in which matching locations are highlighted, visualisations of the rack zones for A, B, and C classified material, banned locations, and more. An important insight from the tests of the first version of the software in the lab was that the skill orchestration needed to be split for the several technical components of the storage system, like the stacker crane, the conveyor belts, and the transfer carriage, and let these components work in parallel. This was not explicitly required from the beginning, because it was taken for granted by the laboratory staff, as they were used to it from the previous system. In the current system, the PLC takes care of the parallel execution.



A1	B1	C1	D1	E1	F1	G1	H1	I1	J1	K1
A2	B2	C2	D2	E2	F2	G2	H2	I2	J2	K2
A3	B3	C3	D3	E3	F3	G3	H3	I3	J3	K3
A4	B4	C4	D4	E4	F4	G4	H4	I4	J4	K4
A5	B5	C5	D5	E5	F5	G5	H5	I5	J5	K5
A6	B6	C6	D6	E6	F6	G6	H6	I6	J6	K6
INPUT	OUTPUT	C7	D7	E7	F7	G7	H7	I7	J7	K7
		C8	D8	E8	F8	G8	H8	I8	J8	K8
A9	B9	C9	D9	E9	F9	G9	H9	I9	J9	K9

Position: E2
Barcode: [illegible] Edit
Material Id: [illegible]
Material Name: [illegible]
Amount: 0
Class: C
Retrieve Bin
Relocate Bin
Banned Cell: false
Fixed to MaterialId: Not fixed

Figure 6 GUI of the WMS, grid view (left), command view (right)

The first version of the WMS software conducted all skill method calls sequentially. This led to the observation that the stacker crane was blocked as long as a conveyor belt conveyed a bin. As this is obviously not a desired behavior, an additional requirement was stated that the system components should work in parallel.

This caused a serious re-design of the skill orchestration part of the backend software. In the first version, only two threads were used: one for the web server to be responsive to the GUI client, and one for the whole ARS operation. The second approach used separate threads for each technical component of the ARSS, like the stacker crane, the two conveyor belts, and the transfer carriage. For each thread, a separate action queue was introduced. Tasks in the queues are either stated by skill orchestration for user commands or in case of the handover of bins from one technical component to another (see Fig. 7). The introduction of these threads leads to the expected behavior of the ARSS components as known from pure PLC control: different components can operate in parallel as long as their activities are independent and synchronized when they have to coordinate their activities, e.g. in the case of bin hand over from one component to another.

Fig. 7 shows an example of the coordination of two threads, one of the ASC and one of a conveyor belt.

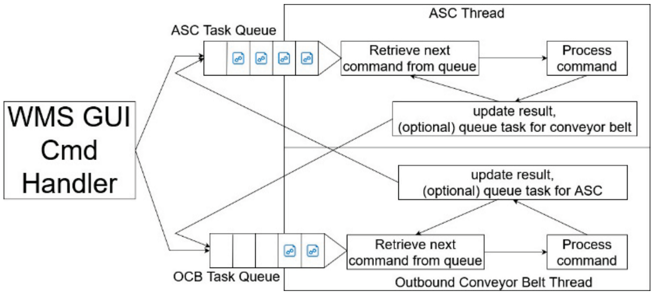

ASC Thread
ASC Task Queue
Retrieve next command from queue
Process command
WMS GUI Cmd Handler
update result, (optional) queue task for conveyor belt
update result, (optional) queue task for ASC
Retrieve next command from queue
Process command
OCB Task Queue
Outbound Conveyor Belt Thread

Figure 7 WMS Skill Orchestrator

The design and implementation of the new WMS was conducted as a study project of Computer Science students in their 7th semester. Based on the requirements and additional specifications and recommendations, the students designed and implemented the software up to a certain level of maturity according to an agile software development approach.

A description of this student project and its evaluation and students experiences is reported in [16]. After some further development, the new WMS is now stable and used for student laboratory courses with the ARSS.

6 DISCUSSION & CONCLUSION

The presented approach shows the application of a skill-based control logic that can execute different skills of an ARSS via an OPC UA communication architecture. The WMS makes the skill requests to the PLC based on the OPC UA methods implemented in the PLC. The skill orchestrator implemented in the WMS takes over the task of aggregating the skills into an executable service. Through the use case presented here, it could be shown that some of the skills in the different services only had to be "re-orchestrated" to enable a new service. Based on the information model of the ARSS, the Skill Orchestrator can split the services into individual skills and send them as method requests to the PLC via the OPC UA client.

The advantages over other approaches are the extended reconfigurability and the implementation of new services based on existing implemented basic skills. Due to the skillset, new services can be implemented more easily via the Skill Orchestrator based on an information model of the resource. Conventional approaches that implement their services as OPC UA methods in the PLC also require an adaptation in the PLC program when creating new services. In the PLC program, these approaches require new control sequences to be implemented as OPC UA methods. In the approach presented here, an adaptation in the PLC program is only necessary if the new service cannot be realized by the basic skills already implemented. A new service based on the already known basic skills can be implemented faster by an adaptation in the Skill Orchestrator. The unfavourable situation by applying the standard telegram software library for calling services implemented on the PLC whose semantics are not unique in many cases, cannot occur with this approach. This problem, as with the services implemented as OPC UA methods, causes a high effort during commissioning and adaptations, since experts for PLC and WMS programming have to agree on the use and semantics of the parameters for all telegram types used. This coordination problem would not be necessary with a PLC based on a basic skill control logic, since the skills could be implemented directly as OPC UA methods and the necessary services for the WMS could be compiled from them. However, since such an approach requires the WMS programmer to understand which basic skills make up a swap service. The research on this approach will be continued so that, on the one hand, an automatic generation of OPC UA methods from the basic skill-based control hierarchy in the PLC is implemented, and, on the other hand, an automatic orchestration of these OPC UA methods into a service based on the OPC UA information model.

7 OUTLOOK

The project provided many insights into different topics, from skill definition via the creation and implementation of the OPC UA information model on the Siemens PLC to the necessary design of the client software to operate the technical system properly.

For the skill definition, it turned out that the initial skill design was not complete. The skill set of the conveyor belts needed to be extended by skills that allow conveying bins over longer distances without lifting them at every single location. These skills should be used if no other bins are on the belts that force a bin to wait. For the ASC, a skill was missing that checks whether a storage location is free or not. The skill for reading a bin ID barcode is not suitable for this as in the case that no valid bin ID could be read it cannot distinguish if the reason for this was that the location was empty or the barcode of an existing bin was not readable. It will be analyzed in a further step of the project, how these mistakes in the skill definition could have been avoided in the design phase. However, the PLC program is currently updated, and the missing skills will be added thereby.

In a further branch of the project, a digital twin of the ARSS will be created that can be controlled by the same WMS as the original one. The basis for that digital twin is a simulation and 3D animation model of the ARSS that already exists. To extend this model to a digital twin, it has to be enhanced by control logic of the simulation elements that corresponds to the behavior of the real system components and accompanied by a new software module as interface to the WMS. This new module has to include an OPC UA server like the ARSS' PLC, so that the WMS can connect to it as an OPC UA client, and needs to be linked to the simulation model elements control units to steer them like the PLC steers the physical components of the ARSS. Finally, the simulation model has to create values of parameters that are mapped to variables of the OPC UA information model so that the WMS can monitor these variables in the same way as it monitors variables on the PLCs OPC UA server.

A further extension of the physical ARSS was started recently: the ASC will be extended by an RFID reader and its communication infrastructure to accompany or as alternative to the existing barcode reader. All receptacles of the system are already equipped with RFID tags according to the ISO 15693 standard, i.e. with a frequency of 13,56 MHz, and a readable and writable memory of 112 Bytes. The RFID reader will be mounted on the load carrying device. It can then read and write the tags of the bin in front of it. This offers several advantages compared to the barcode reading, which is sometimes unreliable, since it needs good light conditions for successful reading. The reader will be connected to a communication module. This module could be connected to the PLC via ProfiBus, but this would require an adaptation of the PLC program as well. Nevertheless, the communication module can also be connected to a computer network via an Industrial Ethernet connection and an OPC UA server with an information model for AutoID devices on board. This offers the opportunity to connect it directly to the WMS without affecting the PLC. As the WMS is an OPC UA client, it can establish connections to different OPC UA servers at the same time by instantiating the OPC UA client object more than once. As soon as this RFID system is commissioned and the TCP/IP connection is established between the communication module and the WMS machine, the WMS code can be extended to access to this second OPC UA server and the skill orchestration can be extended by the skills of the RFID system. An OPC UA client will use this as a proof of concept for the orchestration of skills of different technical systems into common services. This can generalized to a concept for easy integration of new technical components into Service-oriented control architectures for production and logistics systems.

8 REFERENCES

[1] Schuh, G., Anderl, R., Dumitrescu, R., & Krüger, A. (2020). Using the Industrie 4.0 Maturity Index in Industry - Current challenges, case studies and trends. acatech COOPERATION. http://www.acatech.de/wp-content/uploads/2020/04/aca_KOOP_MatInd_en_Web.pdf

[2] Lober, A., Lehmann, J., Hausermann, T., Reichwald, J., & Baumgartel, H. (2022). Improving the engineering process of control systems based on digital twin specifications. In *the 4th IEEE International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM)*. https://doi.org/10.1109/ELECOM54934.2022.9965259

[3] Lehmann, J., Lober, A., Rache, A., Baumgärtel, H., & Reichwald, J. (2022). Collaboration of Semantically Enriched Digital Twins based on a Marketplace Approach. *International Conference on Wireless Information Networks and Systems*, SciTePress. https://doi.org/10.5220/0011141200003286

[4] Leitao, Colombo, K. (2016). Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges. Emerging ICT concepts for smart, safe and sustainable industrial systems. *Computers in Industry, 81*, 11-25. https://doi.org/10.1016/j.compind.2015.08.004

[5] Dorofeev, K. & Wenger, M. (2019). Evaluating skill-based control architecture for flexible automation systems. In *the 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. https://doi.org/10.1109/ETFA.2019.8869050

[6] OASIS. (2006). Reference Model for Service Oriented Architecture 1.0: Committee Specification 1. http://www.oasis-open.org/committees/download.php/19679/soa-rm-cs.pdf (Accessed 27 March 2023)

[7] Zimmermann, P., Axmann, E., Brandenbourger, B., Dorofeev, K., Mankowski, A., & Zanini, P. (2019). Skill-based engineering and control on field-device-level with OPC UA. *The 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Zaragoza, Spain, 1101-1108. https://doi.org/10.1109/ETFA.2019.8869473

[8] Dorofeev, K. (2020) Skill-Based Engineering in Industrial Automation Domain: Skills Modeling and Orchestration. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering: Companion Proceedings (ICSE '20)*. Association for Computing Machinery (ACM). https://doi.org/10.1145/3377812.3381394

[9] Perzylo, A., Kessler, I., Profanter, S., & Rickert, M. (2020). Toward a knowledge-based data backbone for seamless digital engineering in smart factories. *The 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. https://doi.org/10.1109/ETFA46521.2020. 9211943

[10] Köcher, A., Hildebrandt, C., Caesar, B., Bakakeu, J., Peschke, J., Scholz, A., & Fay, A., (2020). Automating the development of machine skills and their semantic description. *The 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, Vienna, Austria, 1013-1018.

https://doi.org/10.1109/ETFA46521.2020. 9211933

[11] Köcher, A., Jeleniewski, T., & Fay, A. (2021). A method to automatically generate semantic skill models from PLC code. *The 47th Annual Conference of the IEEE Industrial Electronics Society (IECON 2021)*. https://doi.org/10.1109/IECON48115.2021. 9589674

[12] Dorofeev, K. & Zoitl, A. (2018). Skill-based Engineering Approach using OPC UA Programs. *The 16th IEEE International Conference on Industrial Informatics (INDIN)*. https://doi.org/10.1109/INDIN.2018.8471978

[13] Spitzer, F., Lindorfer, R., Froschauer, R., Hofmann M., & Ikeda, M. (2020). A generic Approach for the Industrial Application of Skill-based Engineering using OPC UA. *The 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. https://doi.org/10.1109/ETFA46521.2020.9211872

[14] Profanter, S., Perzylo, A., Rickert, M., & Knoll, A. (2021). A generic plug produce system composed of semantic OPC UA skills. *IEEE Open Journal of the Industrial Electronics Society*, 128-141. https://doi.org/10.1109/OJIES.2021.3055461

[15] Jhunjhunwala, P., Atmojo, U. D., & Vyatkin, V. (2021). Applying skill-based engineering using OPC-UA in production system with a digital twin. *The 30th IEEE International Symposium on Industrial Electronics (ISIE)*. https://doi.org/10.1109/ISIE45552.2021.9576342

[16] Baumgärtel, H., Lober, A., Som, J., Völker, S., & Ollinger, L. (2023). Design of a WMS with OPC UA-based Control of an Automated Rack Storage as Study Project in Computer Science and Proof-of-Concept for Skill-Based Engineering. In *ICLF 13th Conference on Learning Factories, Reutlingen University*.

Authors' contacts:

Andreas Lober, M. Eng.
(Corresponding author)
Institute of Industrial Engineering and Supply Chain Management,
University of Applied Science Ulm,
Prittwitzstraße 10, 89075 Ulm, Germany
Andreas.Lober@thu.de

Lisa Ollinger, Prof. Dr.-Ing.
Institute of Production Technology and Material Science,
University of Applied Science Ulm,
Prittwitzstraße 10, 89075 Ulm, Germany
Lisa.Ollinger@thu.de

Sven Völker, Prof. Dr.
Institute of Industrial Engineering and Supply Chain Management,
University of Applied Science Ulm,
Prittwitzstraße 10, 89075 Ulm, Germany
Sven.Voelker@thu.de

Hartwig Baumgärtel, Prof. Dr.-Ing.
Institute of Industrial Engineering and Supply Chain Management,
University of Applied Science Ulm,
Prittwitzstraße 10, 89075 Ulm, Germany
Hartwig.Baumgaertel@thu.de