CONFERENCE ON PRODUCTION SYSTEMS AND LOGISTICS
CPSL 2023

4th Conference on Production Systems and Logistics

Evaluation Of A Capacity-oriented, Agent-based Order Release For Matrix-structured Assembly Systems

Peter Burggräf[1,2], Tobias Adlon[1], Esben Schukat[1], Nina Katharina Weber[1]

[1] *Laboratory for Machine Tools and Production Engineering (WZL) of RWTH Aachen University, Aachen, Germany*

[2] *Chair of International Production Engineering and Management (IPEM), University of Siegen, Siegen, Germany*

Abstract

To address growing challenges in automotive assembly with ever shorter innovation cycles, increasing variant diversity and uncertain market development, innovative concepts for assembly systems are needed. As a response, the concept of matrix-structured assembly system was introduced. Matrix-structured assembly systems break up with the rigid line structure of assembly stations and replace the cycle time-bound and product-specific station assignment of line assembly. A major challenge in the design of matrix-structured assembly systems is the assembly control. While certain approaches, mostly decentral and agent-based, are already capable to assign orders to assembly stations based on the availability of production resources, order release as part of the assembly control has been largely neglected. This is because routing and sequence flexibility lead to temporal uncertainty in the prediction of station-specific capacity utilization. Accordingly, the authors' previous work includes a conceptual methodology for capacity-oriented order release in matrix-structured assembly systems. After implementing the previously introduced methodology, the actual benefit needs to be determined. For this purpose, the present paper suggests and applies a testing strategy based on the fundamentals of successful testing in software development domain. The testing aims to demonstrate the basic functionality of the implemented methodology as well as to compare it with other order release procedures that have been used for simulations in the context of matrix-structured assembly systems so far. It can be shown that the methodology for capacity-oriented order release in matrix-structured assembly systems achieves better adherence to delivery dates and lead times by anticipating bottlenecks compared to ConWIP control with a random order release. The knowledge gained from the testing strategy contributes to the improvement of order release in matrix-structured assembly systems.

Keywords

Matrix-structured assembly system; order release; multi-agent system; assembly control; factory planning

1. Introduction

Conventional assembly lines in automotive final assembly have been in place for 100 years. In recent decades, customized products, short innovation cycles and new vehicle concepts, resulting in high line balancing efforts and declining utilization of assembly stations, have challenged assembly lines [1–3]. Future assembly systems need to address these trends by being more flexible as well as adaptable, allowing economical production of smaller quantities and shorter ramp-up times [4]. As a potential solution, matrix structured assembly systems (MSAS) gained more and more attention in research [5]. MSAS break up with the rigid line structure of assembly stations [1,2,6]. Assembly stations are arranged in matrix form, allowing a cycle-time-independent and an order-specific flow of assembly objects [7]. By this, MSAS are supposed to reduce manufacturing costs in multi-variant production as well as to increase flexibility and efficiency in assembly [2]. An essential prerequisite for the operation of MSAS is an advanced assembly control that takes

DOI: https://doi.org/10.15488/13508
ISSN: 2701-6277
publish-Ing.

into account the increased degrees of freedom of assembly processes. The assembly control is responsible for the reactive and situational assignment of assembly operations and orders to multifunctional assembly stations [1,7–9]. In preliminary work on assembly control, order release as a sub-task of assembly control has been neglected so far. Corresponding simulation show randomly, alternating, or time-based order release, ignoring capacity constraints. Consequently, the authors developed and published a methodology for capacity-oriented, agent-based order release in MSAS [5].

The previously conceptual methodology for order release in MSAS has been recently implemented. Therefore, a qualified testing strategy is needed to scientifically validate the functionality and the actual benefit of the methodology for capacity-oriented, agent-based order release in MSAS. This includes a comparison to random order release. Accordingly, this paper presents a summary of essential properties for order release in MSAS as well as the aforementioned methodology. Afterwards, the evaluation environment used for testing and simulation is described. Based on that, a testing strategy specifying the development, planning and specification of test processes in the field of software development is applied. Subsequently, the testing strategy with its defined test cases are executed and results evaluated. Finally, the testing strategy is critically reviewed and an outlook for future research is given.

2. Order release in matrix-structured assembly systems

2.1 Properties to order release

The authors' previous work [5] specifies six requirements for order release in matrix-structured assembly systems. The **first property** is to perform a capacity analysis on operation and system level. In some cases, assembly stations can perform different kind of operations, so a capacity analysis at the operation and overall system level is necessary. The **second property** deals with the consideration of all possible sequences of operations and related capacity demands of an order when released and processed in MSAS. For this, orders that have not yet been released and orders that have already been released must be evaluated. The latter also includes a monitoring of the current processing progress. To meet the flexibility of MSAS, the **third property** asks for an event-oriented release logic. Order release should take place after defined events such as the completion of an operation. This reduces the frequency of calculations while latest information from MSAS is processed. The **fourth property** deals with the ability to set individual production targets. Possible production targets include minimizing lead times, reducing waiting times or compensating for fluctuations in the capacity situation of the individual assembly stations. The **fifth property** ensures the recognition of order-specific characteristics such as due dates or margins. These characteristics might be included in the release decision through weightings, allowing similar products of different relevance being released in the correct order. For this reason, orders should not be bundled as batches or lots. The last and **sixth property** for order release in MSAS refers to practicability. Order release should be scalable and feasible in reasonable computation time.[5] Scalability describes the ability of a system or process to easily handle extension as the number of elements and objects increases [10]. Reasonable computation time should always be significantly shorter than the shortest operation time so that order release can include the current capacity situation in its evaluation at any time [5].

2.2 Methodology for capacity-oriented, agent-based order release

To address these properties, a methodology for capacity-oriented, agent-based order release in MSAS has been developed. While a detailed description can be found in the authors' previous work [5], Figure 1 and the following explanations provide a summary of the developed methodology.

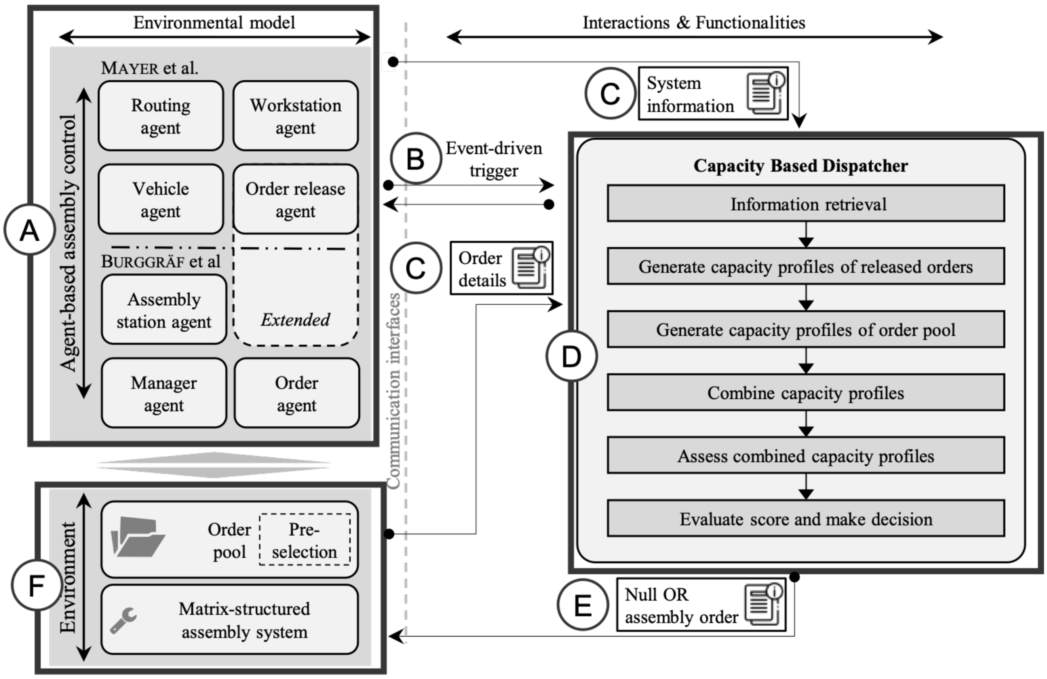

Figure 1: Methodology for order release in MSAS adapted from [5]

As shown in Section A, the methodology is embedded in a separate order release agent and can be part of an existing agent-based assembly control system as proposed by BURGGRÄF et al. [7] or MAYER et al [9]. In this methodology, the order release agent is referred to as Capacity Based Dispatcher (CBD). Section B shows that the CBD is triggered by a defined event, such as an order release or the completion of an operation. The CBD subsequently receives information from the agent-based assembly control system and order-specific details from the order pool, as presented in Section C.

Section D describes the functionality and procedure of the CBD. After the received information has been processed, next, to evaluate the capacity situation of the assembly system and takes release decisions, the CBD creates capacity profiles for each order already released and for each possible release candidate. A capacity profile describes the time-specific demand for assembly operations of an order. These are represented as a matrix, where each column represents a time step and each row represents the demand for an assembly operation as a binary variable. The number of columns covers the entire time horizon until the completion of an order. The sequence of assembly operations is based on the assembly precedence graph. A widely branched precedence graph offers more possibilities for different assembly sequences resulting in a higher number of capacity profiles per order. Thus, one capacity profile is generated for every valid sequence of assembly operations for each order. Capacity profiles also present the current state of the job. Accordingly, orders with completed assembly operations have fewer capacity profiles than unprocessed orders while also having fever number of filled columns representing as the time horizon until completion is shorter. In order to keep calculations between matrices possible, the size of every capacity profile is stretched to the longest possible sequence in the system and filled with zeroes in time steps that exceed the actual demand for assembly operations. Each capacity profile thus reflects a possible load of the assembly system by an order. Since all assembly sequences and thus all capacity profiles can occur in reactive assembly control, all capacity profiles are still considered. In order to map the current capacity load of the assembly system, all combinations of the capacity profiles of the orders currently being processed are added up. This results in an operation- and time-specific prediction of the system load. All these combinations are then extended by capacity profiles of unprocessed orders. Those matched capacity profiles must be compared with the available processing capacity of the assembly system. The available processing capacity is also described in a matrix. For this purpose, each assembly station receives a matrix in which the columns also show the time horizon and the entries in the rows show the ability to process a specific assembly operation. The overall available processing capacity can be determined by summing up the matrices of each workstation. Now, for each time unit, the overall available processing capacity can be compared column by column with the previously combined capacity profiles. If the demand for a certain operation exceeds the available processing capacity, the combined capacity profiles is rejected. Nevertheless, a different combined capacity profile of the same order may be accepted. The non-rejected profiles or rather their unprocessed orders are then evaluated in terms of best capacity fit. For this purpose, these orders are initially distinguished exclusively by the assembly operations they contain. Identical assembly operations of an order are interpreted as identical products and these orders are correspondingly equated at first. The capacity fit is the sum the free capacity of the system for all operations, calculated by subtracting the combined capacity profile from the available processing capacity over all time steps. The product which utilizes the system the most is then selected for further evaluation on order specific level.

In Section E, a score-based decision considering due dates or expected profit are considered to select a specific job for release is made. When no order was selected due to the overload constraint, then no order is dispatched and the agent goes into standby. Information about the order release decision is also passed back to the matrix-structured assembly system and order pool, which are shown in Section F.[5]

2.3 Evaluation Environment

To evaluate the implementation of the methodology in the CBD, an evaluation environment is required. In general, an evaluation environment is defined as a technology platform that serves the purpose of validation methodologies, models and theories developed in innovation projects. By integrating those into a common environment, the acceptance of new approaches can be increased. An evaluation environment allows multiple stakeholders to gain a common understanding of the program and the evaluation process [11,12]. In the present context, an evaluation environment must include an agent-based assembly control for MSAS, have an order pool, simulate the processes in MSAS and also record all results while making them available for output. Such an evaluation environment was developed accordingly. The used logic for assembly control is based on the preliminary work of the authors and includes the agents mentioned in Figure 1 [7]. The evaluation environment enables simulation of an MSAS, allowing users to enter their own assembly system configurations, products and production scenarios quickly and easily. The graphical user interface of the evaluation environment is shown in Figure 2.

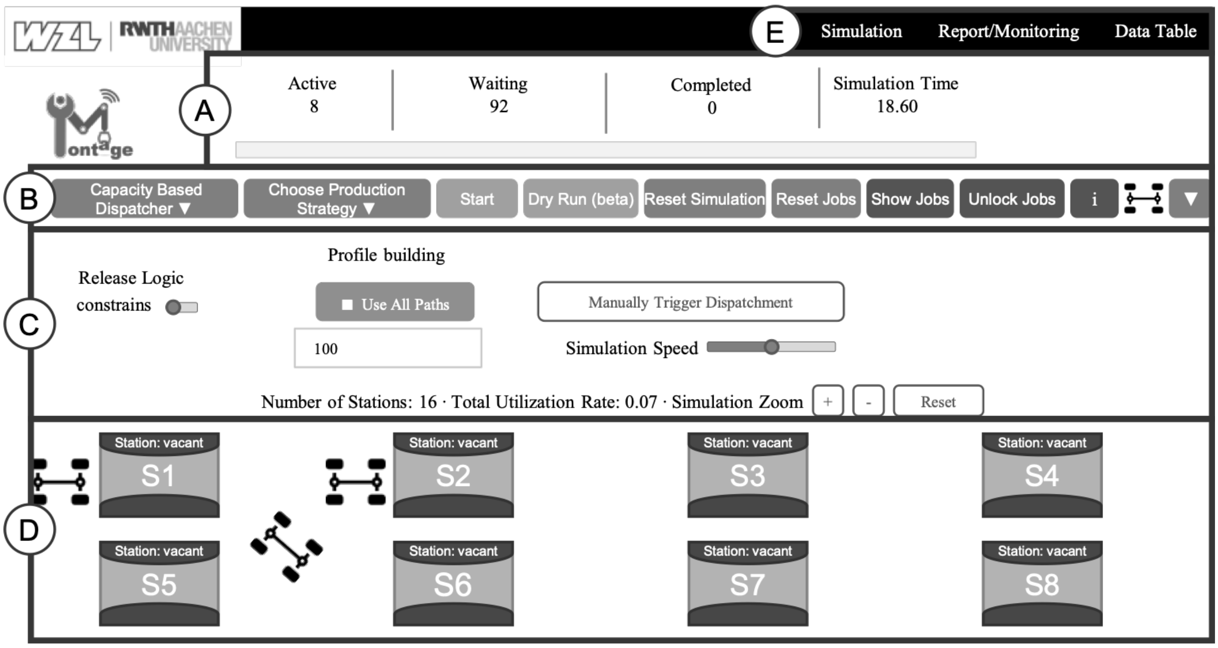

Figure 2: Dashboard of Evaluation Environment [13]

Section A shows a status display of the simulated assembly. The status display visualizes active, waiting and completed orders as well as the current time stamp since a simulation has been started. Buttons to control the simulation and set parameters for assembly control can be found in in **Section B**. This includes options such as start, reset or also an option to enter order-specific due dates. Furthermore, different order release agent types such as the Capacity Based Dispatcher (CBD), which includes the Methodology for capacity-oriented, agent-based order release in MSAS, can be selected. Alternatively, order release agent types for push- or pull-based order releases are possible. In this work, a Random Pull Dispatcher (RPD) is relevant in addition to the CBD. It releases orders randomly using a ConWIP control. **Section C** gives options to set order release parameters. These options differ between the order release agent types. The CBD allows to limit the number of capacity profiles considered for order release. This reduces the number of calculations this type of order release agent is performing when triggered. **Section D** shows a visualization of the entered layout with its assembly stations and orders. Different icons can be selected to visualize orders. **Section E** offers access to the Report and Monitoring as well as the Data Table of the evaluation environment. Report and Monitoring gives a detailed analysis of the system and tracking of individual orders can be done. Here, all information on orders and assembly stations such as station histories, operation histories or lead times can be seen and exported. Moreover, utilization rates of the assembly stations and the number of active orders is mapped. The Data Table includes information regarding products, assembly stations and their arrangement.

3. Testing strategy

3.1 Conception

Literature mostly evaluates the effectiveness of order release procedures simulation-based analysing key performance indicators (KPIs). Typical KPIs are lead times, adherence to delivery dates or capacity utilization [14–16]. The implementation of those in the evaluation environment build the fundament to test the implemented CBD. Before running any tests, a testing strategy to systemically validate the degree of fulfilment of mentioned properties is defined. A testing strategy answers the questions, which test cases should be used, which goals are pursued and which expectation the software application should fulfill [17]. To conceptually design the testing strategy, existing approaches in software development in accordance with WITTE, GRIMME and the standard published of the Institute Of Electrical and Electronics Engineers are combined. The characteristics of test organization, infrastructure and execution specified in the standard are made applicable in WITTE by a theoretical basis to the practical approach [17,18]. GRIMME gives additions to test procedures and environments [19]. This results in five phases and 11 steps, shown in Figure 3.

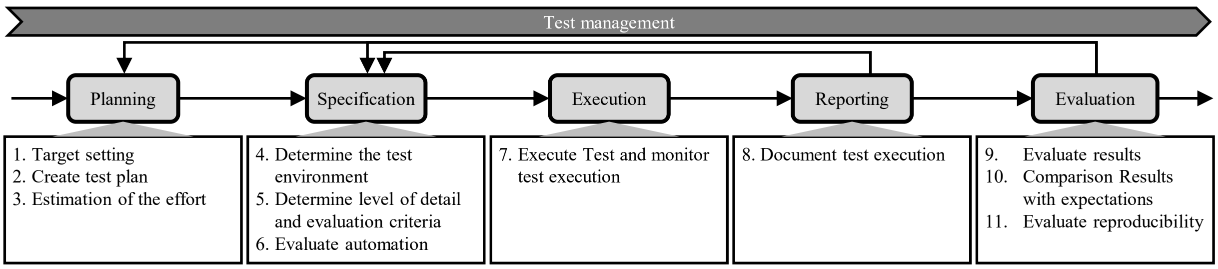

Figure 3: Testing strategy based on [17–19]

The eleven steps of the testing strategy are assigned to the phases of planning, specification, execution and reporting. The individual phases are not to be processed strictly one after the other and partly overlap in time. A continuous test management supports the phases. The planning includes the **target definition**, creation of the **test plan** and the **estimation of the effort**. Test effort and benefit should be plannable and predictable. The relationship between effort and benefit should be constantly optimized from the point of view of economic efficiency. In the specification, the **test environment** is described. When defining the test environment, the hardware, middleware and software used are specified, as well as properties of the organization, test data used, simulation, operational components and various configurations. These properties should be mapped under conditions close to production. A realistic environment, detailed documentation and the use of appropriate test tools improve test results. Next, the **level of detail as well as the evaluation criteria** are determined and **test automation** is evaluated. This is followed by the test **execution including monitoring**. The test is **documented** in the reporting. Finally, the evaluation takes place. In the evaluation, the **results are analysed** and **compared with expectations**. When comparing results to expectations, two questions will be answered. First, whether the objectives were achieved and second, what deviations occurred and how they can be explained. The **reproducibility** of the results for evaluating the suitability of the testing strategy is also examined. It allows conclusions about quality of the test strategy. The results should not be collected randomly or under unknown boundary conditions. The aim is to identify deviations in the results, explain their significance and make a final assessment.[17–19]

3.2 Specification for capacity-based order release in MSAS

The **target** of the test is to validate the functionality of the methodology for capacity-oriented, agent-based order release in MSAS. It shall be shown whether the methodology fulfils the underlying six requirements for order release in MSAS. Furthermore, the added value of the methodology compared to a random order release in MSAS shall be shown. The **test plan** includes the definition of six different test cases to validate each of the requirements for the methodology. The six test cases are executed sequentially. During the execution, data is collected on the production characteristics and, depending on the test case, the lead time, the waiting and transport time or the utilization of the assembly stations are considered and compared. Data is processed and presented in diagrams. To do justice to **effort** in comparison to benefit, the black-box testing method is used in this testing strategy. Black-box testing is a simple and widely used method. It tests the essential functionality of the application without going into the implementation level details [19]. The goal is to determine if the user's original requirements are met and to identify faulty functionality.

For the design of the **test environment** in a matrix-structured system, the number of assembly stations is described in advance. These determine the assembly layout. It should also be specified how many orders will be fed into the system and how many product types will be produced. Variations in these characteristics should be considered during testing. The **level of detail and the evaluation criteria** are set up in form of six test cases. The evaluation is based on KPIs. Depending on the test case process times or the capacity utilization of the assembly stations is analysed. The **execution** of the individual test case**s** is performed in the evaluation environment. This is done automatically. However, data analysis and evaluation are performed manually. A complete implementation for **test automation** in this use case would significantly exceed the effort compared to the benefit. Consequently, automation is not required here. The test cases are implemented in the evaluation environment to perform multiple simulations. The evaluation environment is also used for Monitoring and provides the required productivity metrics for data evaluation. The **test documentation** is ensured by a structured data export of KIPs generated during simulation via the agents representing orders and assembly stations. Simulation aborts or misbehaviour are manually logged. The data export is used to **evaluate the results**. The results are **compared with the expectations** defined in advance and the deviations are examined. It is expected that the developed order release methodology can meet the properties for order release in MSAS which have been stated before. By using the agent-based, capacity-oriented approach different sequences of assembly operations should be anticipated, bottlenecks should be avoided and different weighting factors for order release should be considered. The **reproducibility** is verified by the applicability in the evaluation environment. Nevertheless, the tests are performed three times to support the significance of the data generated during simulation.

4. Application and results

4.1 Test cases

For each property, one specific is defined. The **first test case** examines capacity analysis at the operation and system levels. For this purpose, the various process times of the average lead time, processing time, transport time and waiting time of all orders are examined as a first step. The obtained data is compared with the different types of order release agents as CBD and RPD during simulation execution. For RPD, there are consistently 16 orders in system according to the number of assembly stations to model a ConWIP control. Second, this test case examines the workloads of the assembly stations and the overall workload for overloads. The **second test case** examines the extent to which the order release methodology considers different paths an order takes through the assembly system by evaluating the corresponding sequences of assembly operations. Therefore, the number of sequences and capacity profiles calculated and considered by the CBD are analysed. To validate the requirement of the event-based release logic, test environment is adapted to create a bottleneck at an assembly station for the **third test case**. In addition, there is a disorder in the system. Here, the responsiveness of the agent is evaluated. The **fourth test case** is to validate the adjustability of individual manufacturing goals. As an individual manufacturing goal, for example, the focus can be set on a certain operation to prioritize the processing of this operation. To validate this feature, weighting shall be shifted to a specific operation and an analysis of the utilization of the assembly stations shall be performed. Individual properties are also considered in the order release methodology. The properties can be the weighting of a delivery date, a margin, or a product. The **fifth test case** is used to test the property that an urgent delivery date of an order leads to an earlier order release due to the higher weighting. Margins or product weightings are neglected here at first. Release times and delivery dates of the orders are compared for evaluation. The practicability of the order release methodology in MSAS is evaluated using an acceptance test for the **sixth test case**. The acceptance test verifies the scalability of the CBD. Therefore, one parameter of each test case is incrementally increased during the execution, presenting larger problem instances. Secondly, the sixth test case evaluates the added value of the CBD. Similar to the first test case, KPIs are accessed after certain simulation runs using CBD and RPD. The added value is highlighted by comparing the process times.

To run the test cases, a notional use case is chosen. The individual precedence graphs of each product are presented in Figure 4.

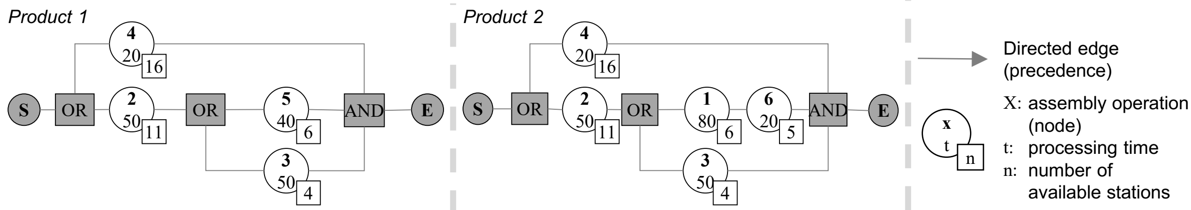

Figure 4: Assembly precedence graphs of Product 1 and Product 2

The use case contains two fictional product types. For assembly, six different operations are performed. Based on possible sequences for assembly, eight sequences can be derived for Product 1 and 15 possible sequences for Product 2. The layout contains 16 assembly stations (see Figure 5).

Each station has specific capabilities, resulting in multiple assembly operations that can be conducted. The numbers shown in each assembly station represent the possible operations. The total simulation time is determined by the processing time of a set 50 orders of each product type. The number of capacity profiles to be considered was limited to a number of 100. This is to circumvent the known NP-hardness of the CBD that is caused by the consideration of all assembly sequences an order can take leading to an exponential growth of the number of capacity profiles and corresponding calculations. The limitation ensures a stable test environment. If promising results are obtained despite this limitation, functionality and added value can still be demonstrated.

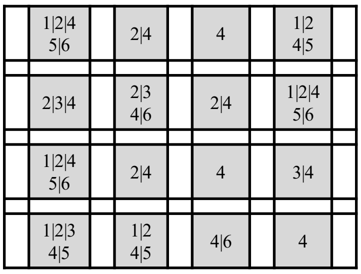

Figure 5: Assembly station layout and possible operations

4.2 Results

Functionality of the Capacity Based Dispatcher can be fully validated in almost all aspects. The **first test case** has shown that, as expected, no bottlenecks occur in the assembly system when performing a capacity analysis at operation and system level. The stations are utilized according to available capacity. In the **second test case**, it was found that the order release methodology considers different assembly sequences according and includes 100 capacity profiles in the evaluation. In addition, the CBD responds in real time to different situations in the system. For example, in the **third test case**, a corrected disruption in the assembly system and the resulting increase in available capacity leads to an additional order release. The **fourth test case** was also able to meet almost all expectations. Depending on how utilized the assembly system already is, certain assembly stations can be utilized to a greater extent by weighting individual operations. In addition, the **fifth test case** confirms the realization of early release of orders with more urgent delivery dates. Only in terms of scalability the expectations could not be met to the full extent in the **sixth test case**. Increasing certain parameters leads to memory leaks and deadlocks in simulation. As a possible improvement of this deficit in application of CBD, use of higher computing power could be mentioned. Moreover, reproducibility of the results is fully achieved for all test cases. There are no significant deviations in the generated data of the individual test runs. By meeting these requirements, the use of the CBD already provides added value in this assembly system. In addition, better process times can be achieved by using the CBD instead of the RPD (see Figure 6).

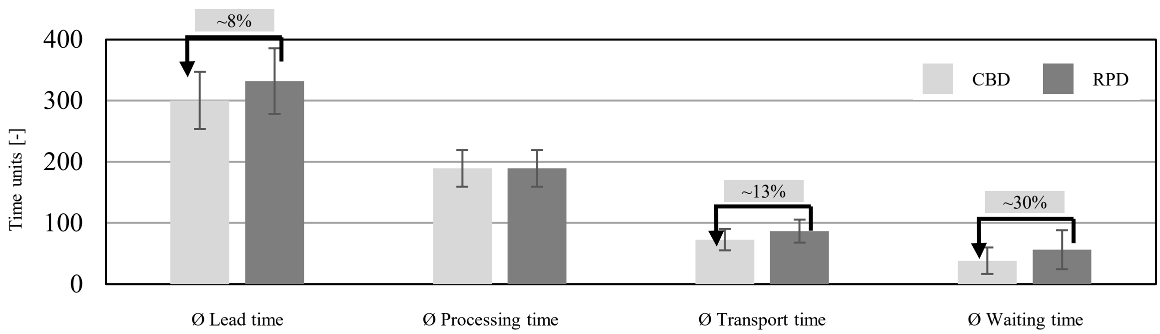

Figure 6: Process time distribution

Using CBD, an average decrease of 13% in transport times and 30% in waiting times can be achieved. This results in an 8% lower value for the total lead time. Since the CBD allows orders to be released only if they can be processed by the system in terms of capacity, shorter waiting times result. In addition, weighting is carried out according to transport routes. All simulation runs were performed three times with an Intel® Core™ i7-2620M CPU@2.70 GHz and 4 GB of RAM. No interruptions, deadlocks or errors were detected.

5. Discussion

The individual test cases were executed without any interruptions. In almost all cases, the results reflect the expectations defined in advance. Nevertheless, it should be critically considered that only one notional use case is used for execution so far. It should be investigated if other use cases e.g., with different arrangements of assembly stations or a higher variety of products lead to a significant different behaviour. Furthermore, the number of capacity profiles considered by the CBD was limited. Although results are promising, the correlation between obtained benefits and the considered capacity profiles can be further investigated. This, however, requires an improvement of the algorithms used for capacity profile generation and evaluation. A renewed execution with higher computing power should also be considered to examine if better results can be archived.

To sum up, the goal of this work was achieved. After developing and applying a testing strategy can validate the methodology for capacity-oriented, agent-based order release in MSAS, results deliver evidence for functionality and added value.

6. Summary and Outlook

Assembly control and its subtask order release is a major challenge in matrix-structured assembly systems. To address this, the authors presented a methodology for order release in a previous work. The suggested methodology performs a capacity analysis at operation and system level specifically. By this, different sequences resulting from sequence and routing flexibility can be considered in capacity analysis. The methodology suggests an event-oriented release logic. In addition, individual production targets as well as individual orders can be set. After implementing the former conceptual approach in an order release agent, this paper aimed to validate its functionality and added value. Thus, this paper introduced a testing strategy based on the fundamentals of successful testing in the field of software development. The testing strategy includes 11 steps that were applied to six test cases. It can be shown that all properties to successfully run a capacity-orientated order release for matrix-structured assembly systems can be widely fulfilled. Especially in comparison to random order release, also used by many researchers in the context of MSAS, the capacity-orientated order release improves lead times.

Further research should repeat the testing with larger problem instances and high-performance computing power. In addition, the implemented methodology is currently limited by the efficiency of its algorithms when calculating all possible sequences and matched capacity profiles to evaluate the capacity constraints, resulting in NP-hardness. Thus, the considered sequences and capacity profiles were limited for this work. Consequently, new solutions need to be found to ease this limitation.

Acknowledgements

We would like to thank the BMWK, DLR-PT and our partners for their kind support. This work is part of the research project "AIMFREE", which is funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) within the "Directive on a joint funding initiative to support research and development in the field of electro mobility" (funding code: 01MV19002A) and supported by the project management agency German Aerospace Center (DLR PT). The authors are responsible for the content. More information about the current research activities within the project AIMFREE – Agile assembly of electric vehicles can be found at www.aimfree.wzl.rwth-aachen.de

References

[1] Greschke, P., 2020. Matrix-Produktion: Konzept einer taktunabhängigen Fließfertigung, Norderstedt.

[2] Bebersdorf, P., Huchzermeier, A., 2021. Variabler Takt. Springer Berlin Heidelberg, 275 pp.

[3] Kern, W., 2021. Modulare Produktion. Dissertation.

[4] Kampker, A., Kreisköther, K., Burggräf, P., Dannapfel, M., Bertram, S., Wagner, J., Fischer, A., Knau, J., Grams, J., Bartl M., 2017. Agile Low-Cost Montage.

[5] Schukat, E., Rachner, J., Göppert, A., Adlon, T., Burggräf, P., Schmitt, R.H., 2022. Agent-based Order Release In Matrix Structured Assembly Systems. publish-Ing.

[6] Schönemann, M., Herrmann, C., Greschke, P., Thiede, S., 2015. Simulation of matrix-structured manufacturing systems. Journal of Manufacturing Systems 37, 104–112.

[7] Burggräf, P., Dannapfel, M., Adlon, T., Kahmann, H., Schukat, E., Holtwiesche, L., 2020. Multiagentensysteme in der Agilen Montage.

[8] Kern, W., Rusitschkab, F., Kopytynskib, W., Keckla, S., Bauernhansl, T., 2015. Alternatives to assembly line production in the automotive industry.

[9] Mayer, S., Arnet, C., Gankin, D., Endisch, C., 2019. Standardized Framework for Evaluating Centralized and Decentralized Control Systems in Modular Assembly Systems.

[10] Bondi, A.B., 2000. Characteristics of scalability and their impact on performance, 195–203.

[11] Koch, R., 2011. Systemarchitektur zur Ein- und Ausbruchserkennung in verschlüsselten Umgebungen. Zugl.: München, Univ. der Bundeswehr, Diss., 2011. Books on Demand, Norderstedt, 490 pp.

[12] Fynn, J.F., Hardeman, W., Milton, K., Jones, A.P., 2020. A scoping review of evaluation frameworks and their applicability to real-world physical activity and dietary change programme evaluation. BMC public health 20 (1), 1000.

[13] Werkzeugmaschinenlabor WZL der RWTH Aachen. Agile Assembly. https://www.produktionssysteme.rwth-aachen.de/agile-montage/.

[14] Corsten, H., Sartor, C., 2005. Simulationsuntersuchungen zur Auftragsfreigabe bei Werkstattproduktion. Zeitschrift für Planung & Unternehmenssteuerung 16 (1), 27–52.

[15] Fernandes, N., Thürer, M., Pinho, T., Torres, P., Carmo-Silva, S., 2020. Workload control and optimised order release: an assessment by simulation. International Journal of Production Research, 3180–3193.

[16] Hendry, L., Wong, S., 1994. Alternative order release mechanisms: a comparison by simulation. International Journal of Production Research 32 (12), 2827–2842.

[17] Witte, F., 2016. Testmanagement und Softwaretest. Springer Fachmedien Wiesbaden, 274 pp.

[18] Institute of Electrical and Electronics Engineers. IEEE Standard for Software and System Test Documentation, Piscataway, NJ, USA, 132 pp.

[19] Grimme, C., 2011. Strategien für den Aufbau und Betrieb von Testumgebungen.

Biography

Prof. Dr.-Ing. Peter Burggräf (*1980) is scientific head of the Department of Factory Planning at the Laboratory of Machine Tools and Production Engineering WZL of RWTH Aachen University and has been head of the Institute of International Production Engineering and Management at the University of Siegen since 2017.

Tobias Adlon (*1989) has been head of department of Factory Planning at Laboratory of Machine Tools and Production Engineering WZL of RWTH Aachen University since 2020.

Esben Schukat (*1993) has been research associate in the Department of Factory Planning at Laboratory of Machine Tools and Production Engineering WZL of RWTH Aachen University since 2019. In 2022 he took over the group lead of the research group 'Assembly Planning' while continuing his research activities on matrix-structured assembly systems.

Nina Katharina Weber (*1999) has been student research assistant at the Laboratory of Machine Tools and Production Engineering WZL of RWTH Aachen University since 2022.