Available online at www.sciencedirect.com
ScienceDirect
Procedia CIRP 17 (2014) 475 – 480
www.elsevier.com/locate/procedia

Variety Management in Manufacturing. Proceedings of the 47th CIRP Conference on Manufacturing Systems

Simulation-Based Performance Analysis of a Miniload Multishuttle Order Picking System

Mustafa Güller[a,] *, Tobias Hegmanns[a]

[a]*Chair of Factory Organization, TU Dortmund, Dortmund, 44227, Germany*
* Corresponding author. Tel.: +49-231-755 5773; fax: +49-231-755 5772. *E-mail address:* guller@lfo.tu-dortmund.de

Abstract

In today's fluctuating business environment, flexibility, responsiveness, and reconfigurability in the field of warehousing systems are key characteristics as well as the level of automation, cost effectiveness and maximum throughput. The miniload multishuttle system represents relatively a new scalable automated storage and retrieval solution which provides considerable flexibility for adapting throughput capacity to meet market needs. The miniload multishuttle system comprises autonomous vehicles (shuttles), lifts, and a system of rails that facilitate movement of the vehicles in the x and y dimensions in a tier. In this new technology, the shuttles can be moved between different tiers by means of shuttle lifts. Especially in large and complex material flow systems, the modeling problem arises with the interactions between transactions and the collisions between shuttles. An agent-based simulation approach that differs from discrete-event simulation (DES) offers an alternative way to model the autonomous control of the multishuttle system. This research develops a simulation model for the miniload multishuttle order picking system using the multi-agent modeling approach. The main objective of this paper is to create a detailed simulation model and to evaluate the performance of the system in order to support in design process of miniload multishuttle order picking systems.

Selection and peer-review under responsibility of the International Scientific Committee of "The 47th CIRP Conference on Manufacturing Systems" in the person of the Conference Chair Professor Hoda ElMaraghy"
Keywords: Flexibility; Miniload multishuttle; Multi-agent simulation

1. Introduction

Intralogistics systems are defined as a combination of organization, controlling, execution and optimization of the in-house material and information flow [1]. Classical criteria in the design process of intralogistics systems are minimization of costs (by means of e.g. maximal utilization of resources), quality of service (order lead time, timeliness, error levels), flexibility (ability to cope with changing process parameters, such as order structure or article structure) and scalability (ability to grow with increasing system load) [2]. All of these design criteria in the field of intralogistics are key challenges for the industry, especially in dynamic and uncertain environments. This dynamic environment is characterized by a wide variety of products, fluctuations in demand, and increased customer expectations in terms of quality and delivery time. To deal with these increasingly challenging issues, firms need to develop strategies that provide the flexibility to succeed in uncertain environments. However, it seems to be difficult to achieve within the rigid, specific and specialized world of automated material handling systems [3] which offers static and inflexible hardware solutions for the main functions of intralogistics systems. At this point, autonomous vehicles have been widely adopted as a key component to intralogistics systems in order to improve adaptability by physical flexibility in disposition, routing and space consumption.

Automated storage and retrieval systems (AS/RSs) are major material handling systems that have been widely used in warehousing for storing and retrieving finished products and parts [4]. In its most basic depiction, the basic components of an AS/RS are storage racks, a crane (or, equivalently, S/R machine), and input/output (I/O) stations. The benefits of AS/RSs include low labor cost, enhanced space exploitation, improved material tracking and high system throughput [5]. Although AS/RS technology can

2212-8271 © 2014 Elsevier B.V. This is an open access article under the CC BY-NC-ND license
(http://creativecommons.org/licenses/by-nc-nd/3.0/).
Selection and peer-review under responsibility of the International Scientific Committee of "The 47th CIRP Conference on Manufacturing Systems" in the person of the Conference Chair Professor Hoda ElMaraghy"
doi:10.1016/j.procir.2014.01.041

achieve high throughput and fast response times in many material handling applications, classic AS/RS systems have limitations, such as limited flexibility and autonomy by the physical build-up of the system. The traditional design is a single shuttle that can carry a pallet or handling unit. In order to increase the throughput capacity of the system, multishuttle AS/RSs have been developed in recent years [6]. The miniload multishuttle system represents relatively a new scalable AS/RS solution which provides considerable flexibility for adapting throughput capacity to meet market needs. While there is usually just a handful of an automated crane in conventional AS/RS, there are a higher number of coordinated vehicles in the multishuttle system to fulfil the storage and retrieval orders.

Especially in large and complex material flow systems, numerous questions arise both on the design and control levels. From a design perspective, the system configuration is often complicated by large varieties of products needing storage, varying areas of required storage space and drastic fluctuations in product demand [7]. Large-scale, complex or highly dynamic environments make the systems too complex to be evaluated analytically. Especially, automated storage and retrieval systems are difficult to model analytically because they incorporate interactions of many subsystems [8]. For this reason, simulation is used as an important decision support tool to analyze all the processes and interactions of the warehouse that are complex, dynamic and stochastic in nature. Most of research on warehouse simulation ([9], [10], [11]) uses discrete-event simulation approach to model the operations of a warehouse or the dynamic behaviors of the system are expressed using mathematical techniques. However, using such models, it is difficult to model the autonomous vehicle's control. Multi-agent systems [12], which are composed of different interacting computing entities called agents, offer an alternative way to design and implement simulation of intralogistics systems based on multishuttles and autonomous control.

In this paper, we develop a multi-agent simulation model to evaluate the performance of a miniload multishuttle order picking system. The main objective of our research is to introduce the application of a statechart-based model for the new generation multishuttle systems in which the shuttles can be moved between different tiers. The remainder of this paper is organized as follows. In Section 2 the architecture of the system and control structure is presented. We describe the multi-agent simulation model of miniload multishuttle order picking system in Section 3. Finally, details of the implementation and evaluation results are given in Section 4.

2. System description and control structure

Since throughput capacity is a concern with AS/RS, multishuttle systems have been developed for automated storage and retrieval of cartons or small parts in order to increase the throughput capacity of a system. This system comprises autonomous vehicles (shuttles), lifts, and a system of rails that facilitate movement of the vehicles in the x and y dimensions in a tier. In this new technology, the shuttles can be moved between different tiers by means of shuttle lifts. Figure 1 illustrates an example of a multishuttle with lift and one aisle rack system used in the storage area.

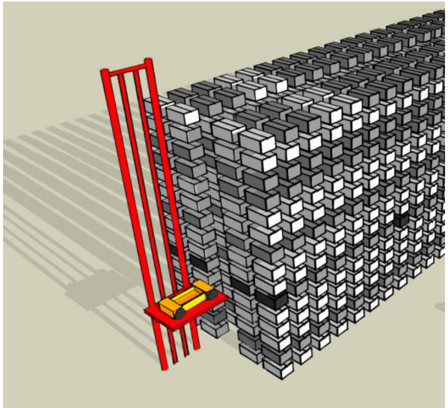

Fig. 1. The multishuttle automated storage/retrieval system

Figure 2 shows the structure of the miniload order picking system under study (mentioned in [13] and [14]). The order picking system can be classified as a part (or product)-to-picker system. Three main areas can be distinguished, namely the storage area, workstations, and conveyor. At the workstation, the picker picks a number of required items from the product tote based on customer order and puts them in an order tote. A conveyor loop is used to transport the product totes between the storage area and workstations.

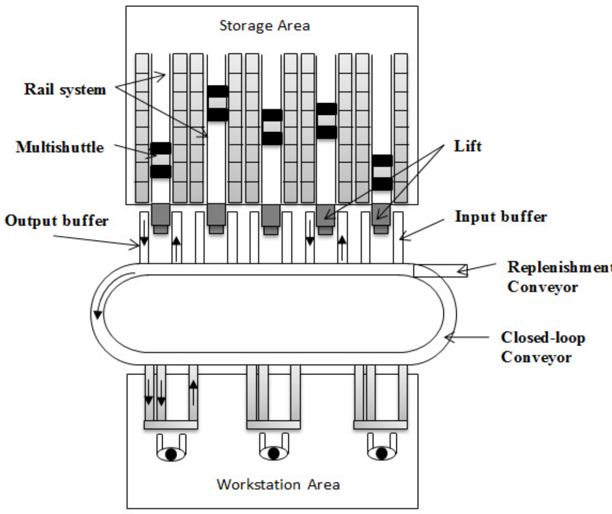

Fig. 2. Miniload multishuttle order picking system

The storage area consists of multiple levels of racking, shuttles, and buffer conveyors. It is essentially automated with multishuttles to serve two main functions, namely the storage and retrieval of product totes. Multishuttle system is an important goods to person material handling technology. Currently, one storage crane works all vertical levels within the storage aisle in conventional AS/RS. However, in next generation of multishuttle technology, each vertical level in the storage system is concurrently serviced by an autonomous vehicle or vehicles can be moved between different levels by means of shuttle to process transactions across all levels [15]. Besides, the system can be expanded with changing the number of vehicles in a fixed storage configuration based on as throughput requirements.

The retrieved totes are put on the output buffer in the storage area, waiting to get access to the central conveyor loop to be sent. The central conveyor loop transports product totes to the picking stations. To ensure that there will be sufficient storage spaces at the rack system for the returning product totes, the replenishment product totes are given less priority than returning product totes in the enter point of conveyor.

Each of the five workstations in the system consists of two input buffers and one output buffer as shown at the bottom of Figure 2. Once product totes arrive at the order picking workstation, they form queues on buffer conveyors. In our case, a picker always completes an order before starting to pick items for the next order. After the required product tote is available, the picker then picks a number of required items from the product tote and put the item(s) into an order tote when a product tote reaches its assigned picking station. After item picking, it is possible that the product totes becomes empty or the number of product in a tote may be under the safety level. In that case, the empty product tote will be put on the take-away conveyor along with the finished order totes to be sent to the replenishment area. If the product tote still contains any items left, the tote will be returned to the storage area using the central conveyor loop. The destination storage point for a returning product tote is the same point from which it was retrieved. After the product totes reach the input buffer of the assigned storage area, they wait for the multishuttle to store them into the miniload racks.

3. Proposed agent based modelling for the miniload multishuttle system

Especially in large and complex material flow systems, numerous questions arise both on the design and control levels. Furthermore, highly dynamic environments make systems too complex to be evaluated analytically. For this reason, the simulation is used as an important decision support tool to analyse all processes of the warehouse that are dynamic and stochastic in nature. They are commonly used to model decentralized controls where a large number of heterogeneous objects with individual behaviours and their interactions with other objects have to be considered [16]. A typical agent-based model has three elements [17]:

- A set of agents, their attributes and behaviours.
- A set of agent relationships and methods of interaction: An underlying topology of connectedness defines how and with whom agents interact.
- The agents' environment: Agents interact with their environment in addition to other agents.

Developing an agent-based simulation (ABS) requires identifying, modelling, and programming these elements. In ABS, the model consists of a set of agents that are autonomous and self-directed. The main features of agents are the capability to make independent decisions, their ability to perform flexible actions in a dynamic and unpredictable environment, as well as their pro-activeness depending on motivations generated from their internal states [18]. One of the fundamental principles of these simulations is to design the agent's behaviour, which is described by simple rules or more abstract representations [17]. An agent's behaviour relates the information that comes through interactions with other agents and with the environment.

Agent-based modelling (ABM) can be performed either by using programming languages or by using a readily available simulation software packages that address the special requirements of the agent modelling. Several ABM tools has been developed presenting different functionalities, graphical interfaces and also programming languages such as Repast, Swarm, NetLogo and Mason. We refer the reader to the surveys in [19] and [20] for a much more detailed information. In this research, the simulation model has been implemented in AnyLogic™, which is Java based simulation software that support discrete-event, system dynamic and agent-based modelling approaches [21]. One major advantage of AnyLogic is that it allows the user to create a simulation model combining differential equations, discrete events and agent based modelling. These combination possibilities make it a very interesting tool for simulation of complex systems. Another reason to use AnyLogic is the possibility to use Java code at any place of the program and object oriented structure. This allows great freedom to expand the feature set of the tool.

The developed simulation model is composed of a set of agents that communicate with each other and with the environment via messages. In order to design agents and specification of their behaviours, statecharts are used. Statecharts are basically directed graphs where different kinds of nodes represent states and edges characterise transitions [22]. A state can be considered as a set of reactions to external events that determine the object's situation. The transitions within the statecharts are triggered by certain events, such as a message, an arrival, a condition, or a timeout. When a transition is triggered, the agent leaves its current state, initiates the actions specified for that transition, and enters a new state [23].

3.1. Simulation model

Figure 3 shows a screenshot of developed simulation model including the storage area with lifts, the closed loop conveyor, and order picking stations. To model the multishuttle miniload order picking system, a combined Discrete Event (DES) and Agent Based Simulation is used. The different developed agents consist of *multishuttle* agent, *lift* agent, and *order manager* agent. All these agents have their own characteristics and behaviours. The agent implementation in the model is based on the statecharts, the set of rules, and functions. Figures from 4 to 6 depict the statecharts modelling respectively the behaviours of order manager, multishuttle and lift agents.

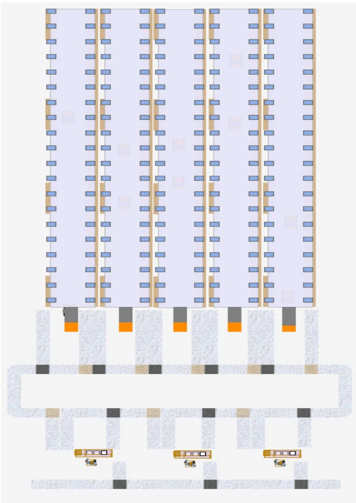

Fig. 3. The screenshot of 2D simulation model developed in AnyLogic for the miniload multishuttle order picking system

In Fig. 4 the statechart modelling for order manager agent is illustrated. Boxes in this figure show the possible states and the arrows show the possible transitions. The square shapes in the chart represent branches (decision nodes). The *order manager agent* can be one of three states: "initial" state, "assignBox" state and "selectBox" state. There are three branches between three states to check the order, the task type and free MSM agents. The transitions between states are modelled by the timeout, the message and the condition. The "initial" state represents the state where the *order manager* agent waits for a new customer order. A new order will cause the statechart to follow the first branch. At the second branch, the *order manager* agent checks the multishuttle availability. Then it controls the task like retrieval or storage. Once the task is defined, the *order manager* agent communicates with the *multishuttle* agents by sending a message and assigns the order to a free multishuttle.

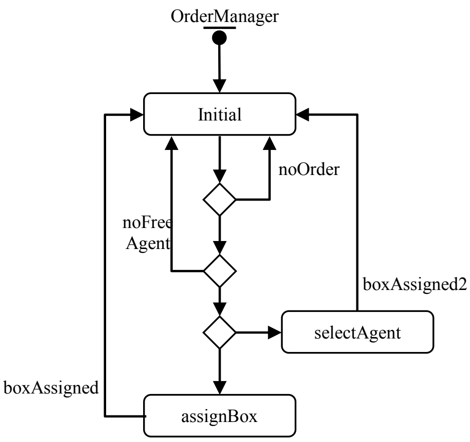

Fig. 4. State transition model of the order manager agent

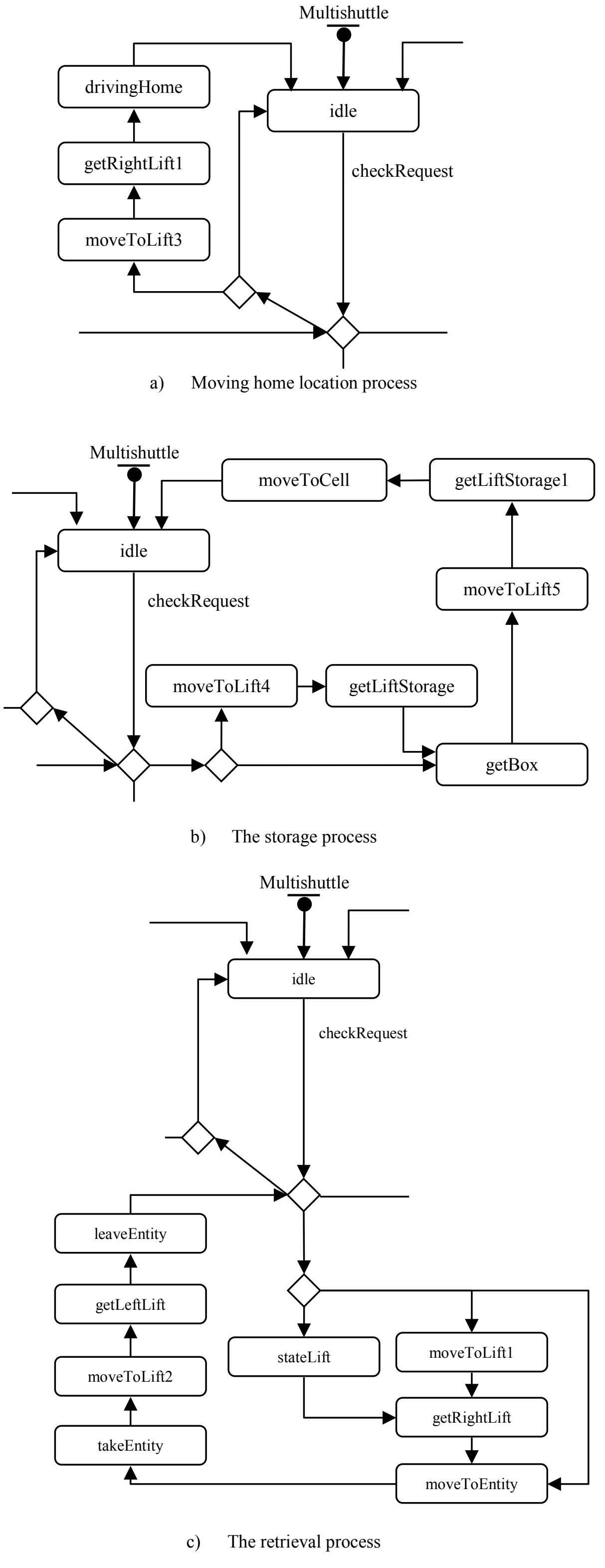

Fig. 5. State transition model of the multishuttle agent

The *multishuttle* agent statechart is depicted in Fig. 5. The *multishuttle* agent template consists of three main blocks for the retrieval process, the storage process, and moving home location process. As the block depicted in Fig. 5-a is used for moving home location, the block in Fig. 5-b is used for the storage process. The *multishuttle* agent starts at their home location. While in the state idle, it waits for a new order request. Each multishuttle agent checks the order pool with the "Check Request" timeout transition. If there is a request, they change their state based on the location of the requested box. If the box is at a different level, *multishuttle* agents change its state to the "move to lift" state and then they move through the *lift* agent. The decision of where to drive next is made when the current task is completed and the next state is triggered by the arrival transition. Some of the state changes are connected by passing messages. As soon as the *multishuttle* agent reaches the lift, it communicates with the *lift* agent by sending a message. The transition from the "move to lift" to the "get right lift" state is triggered by the arrival transition and the agent moves to the destination tier vertically. The state of *multishuttle* agent is changed to "take entity" state with a message received from the *lift* agent. Once the requested order is picked from the storage area, the agent executes the "move to lift" state again and the *multishuttle* agent moves to the lift. When the agent finishes the retrieval transaction, the multishuttle executes the "driving home" state if there is not a storage transaction and moves back to his base location.

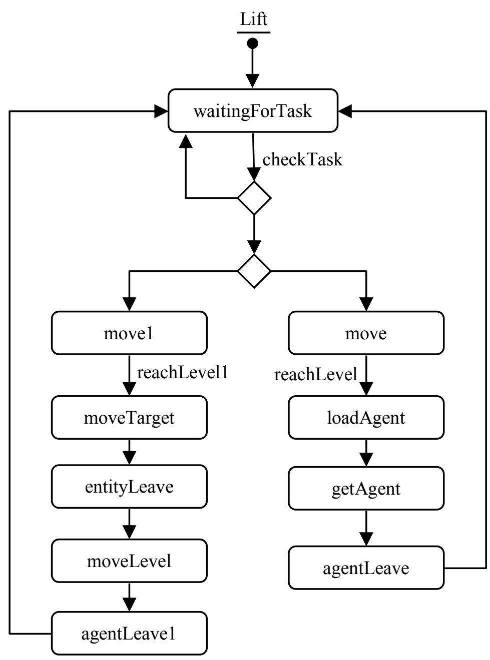

Fig. 6. State transition model of the lift agent

In Figure 6 is shown the statechart for modelling the behaviours of the *lift* agent. A queuing system has been implemented for the *lift* agent. The *lift* agent template consists of two main blocks which use a very similar logic. In the right block, the *multishuttle* agent will be moved to the target tier to pick the requested order. The left block is used to move the multishuttle from its level to the input buffer of an aisle. The *lift* agent is in the "waitingForTask" state by default and transitions into the "move" state when a multishuttle is ready to move to the destination tier. If the lift is not in the "waitingForTask" state, the *multishuttle* agent is added to a queue list of the lift. When the moving destination level task is completed, the next state is triggered by the arrival transition. The transition between the "agent leave" and the initial state is triggered by timeout. Other related warehouse operations are modelled by using DES.

4. Simulation experiments and results

In this section, we will use the developed agent-based simulation model in order to analyze the performance of the multishuttle order picking system under different scenarios. In particular, we are interested in system's throughput (d) and average cycle time per order. The miniload-workstation order picking system is triggered by orders that enter the system at any time. An order is composed of order lines, where each order line consists of a particular item type. In other word, an order line represents a Stock Keeping Unit (SKU) type and the required amount of items for that SKU. It is assumed the warehouse stocks a total of 600 SKUs, which are provided in boxes.

Table 1. The order type structures used in the experiment

Order Structure	Single Order Line	Two Order Line	Three Order Line	Four Order Line
A	% 40	% 40	% 10	% 10
B	% 40	% 20	% 20	% 20
C	% 25	% 25	% 25	% 25

Customer orders are modelled as independent Exponential arrival processes with 80 order/hour and the target throughput per hour is assumed 80 orders. The order size varies between 1 and 6 units and generated from a uniform distribution. To show the effect of different size of order on the performance of the system, the order structure is varied in the number of order lines as shown in Table 1. The exemplary order picking system consists of a shelving system with 5 tiers, 5 aisles and specially developed 3 workstations. Each aisle is served by 2 multishuttles. Number of buffer places per buffer lane at one workstation and the aisle is 4. For each scenario, 15 replications are run with a random seed and simulation results represent the mean of these replications.

Table 2. Performance comparisons of different simulation scenarios

Order Structure	A	B	C
Average cycle time	106 (sec)	120 (sec)	126 (sec)
Orders per hour	94	84	79

Fig. 7 shows the cycle time and order lines for 50 sample orders under the order structure A. One interesting result obtained for the cycle time against the number of order lines is that the cycle time of single line order can be higher than the cycle time of four line order. This is because all multishuttles can be busy if there are boxes waiting to be stored on the buffer of the storage area. Table 2 shows the influences of different order structure on the expected cycle time and throughput performance of the system. As it can be seen in Table 2, the order structure impacts the mean order throughput time. As the probability of three and four order line increases, the throughput per hour decreases. Under order structure A and B, the system achieves the target throughput.

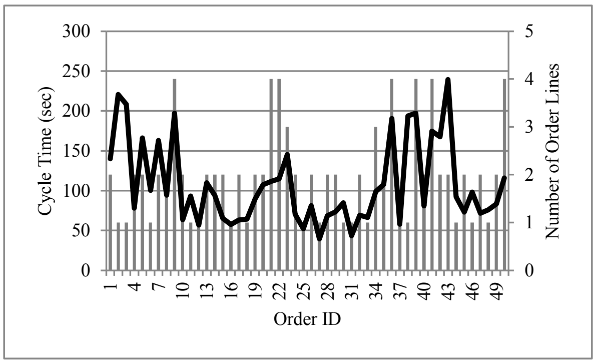

Fig. 7. Number of order line and cycle time per order for order structure A

5. Conclusion

The evolution in intralogistics systems put forward new challenging requirements. Today, flexibility, reconfigurability and high availability are important as well the level of automation, cost effectiveness, and maximum throughput. In recent years, the new generation multishuttle systems have been developed for automated storage and retrieval of cartons or small parts in order to increase the flexibility and throughput capacity of a miniload order picking system. Discrete-event simulation is one of the methods commonly used to represent the real situation that occurs in a system. However, discrete event simulation is not suitable to be used in order to analyse the model of autonomous behaviours.

In this paper, we propose an agent-based simulation model for the miniload-multishuttle order picking system to evaluate the performance. The results show that the order structure has a significant impact on the performance of the system. Furthermore, agent-based simulation approach proposes a powerful tool in modelling of complex multishuttle systems that takes advantages of agent properties. The simulation in this study has assumed a simplified control strategy. This study can be extended in many directions by considering different control policies such as storage location assignment. The rack design and the optimization of number of multishuttles are the focus of future work.

References

[1] D. Arnold and K. Furmans, Materialfluss in Logistiksystemen.: Springer, 2006.

[2] T. Schmidt and F. Schulze, "Future Approaches to Meet Complexity Requirements in Material Handling System," FME Transactions, vol. 37, no. 4, pp. 259-166, 2009.

[3] L.F. Marin and R. Carrasco-Gallego, "A Design Framework for Flexible Automated Warehouses," in 7th International Conference on Industrial Engineering and Industrial Management, Spain, 2013, pp. 463-471.

[4] Russel D. Meller and Anan Mungwattana, "MultiShuttle Automated Storage-Retrieval Systems," Department of Industrial Engineering, Auburn University, Auburn, Technical Report 95-06 1996.

[5] M. Dotoli and M. P. Fanti, "Modeling of an AS/RS serviced by rail-guided vehicles with colored Petri nets: a control perspective," in Proceeding of: Systems, Man and Cybernetics, 2002 IEEE International Conference, Italy, 2002, pp. 3-6.

[6] Shin-Ming Guo and Tsai-Pei Liu, "Simulation Evaluation of Single Shuttle and Twin Shuttle AS/RS," International Journal of Electronic Business Management, vol. 6, no. 2, pp. 106-115, 2008.

[7] Andrew Kusiak, Computational Intelligence in Design and Manufacturing. New York, USA: John Wiley & Sons, Inc, 2000.

[8] J.P. Gagliardi, A. Ruiz, and J. Renaud, "A simulation modeling framework for multiple-aisle automated storage and retrieval systems," Journal of Intelligent Manufacturing, 2012.

[9] Choong-Yeun Liong and S.E. Loo Careen, "A Simulation Study of a Warehouse Loading and Unloading Ssystems Using Arena," Journal of Quality Measurement and Analysis, vol. 5, no. 2, pp. 45-56, 2009.

[10] V Colla and G Nastasi, "Modelling and simulation of an automated warehouse for the comparison of storage strategies," in Modelling Simulation and Optimization., 2010.

[11] Banu Y. Ekren and Sunderesh S. Heragu, "Simulation based performance analysis of an autonomous vehicle storage and retrieval system," Simulation Modelling Practice and Theory, vol. 19, pp. 1640–1650, 2011.

[12] M. Wooldridge, An Introduction to Multiagent Systems. England: John Wiley & Sons Inc., 2002.

[13] J. Verriet, R. Hamberg, J. Caarls, and B. Wijngaarden, "Warehouse simulation through model configuration," in 27th European Conference on Modelling and Simulation, ECMS 2013, Alesund, Norway, 2013.

[14] R. Andriansyah, W.W.H. Koning, R. Jordan, and L.F.P. Etman, "Simulation study of miniload-workstation order picking system," Eindhoven University of Technology, The Netherlands, SE Report ISSN: 1872-1567, 2008.

[15] S. S. Heragu, X. Cai, and A. Krishnamurthy, "Analysis of Autonomous Vehicle Storage and Retrieval System by Open Queueing Network," in 5th Annual IEEE Conference on Automation Science and Engineering, India, 2009.

[16] R. Siegfried, A. Lehmann, R. A. Khayari, and T. Kiesling, "A reference model for agent-based modeling and simulation," in SpringSim '09 Proceedings of the 2009 Spring Simulation Multiconference, San Diego, CA, USA, 2009.

[17] C. M. Macal and M. J. North, "Tutorial on agent-based modelling and simulation," Journal of Simulation, vol. 4, pp. 151–162, 2010.

[18] P. Sebers, A. Uwe, C. Helen, and C. Clegg, "A multi-agent simulation of retail management practices," in Proceedings of the 2007 Summer Computer Simulation Conference, 2007.

[19] R. Tobias and C. Hofmann, "Evaluation of free Java-libraries for social-scientific agent based simulation," Journal of Artificial Societies and Social Simulation, vol. 7, 2004.

[20] C.J.E Castle and A.T. Crooks, "Principles and Concepts of Agent-Based MOdelling for Developing Geospatial Simulations," Center of Advanced Spatial Analysis, University Collage London, UK, Techincal Report 2006.

[21] http://www.anylogic.com/.

[22] H. Alexander, H. Tom , and B. Yolande, "Testing AGVs in dynamic warehouse environments," Lecture Notes in Computer Science, 2006.

[23] M. Glinz, "An Integrated Formal Model of Scenarios Based on Statecharts," in Software Engineering – ESEC'95, Berlin, 1995, pp. 254-271.