*Simulation in Produktion*
*und Logistik 2017*
Sigrid Wenzel & Tim Peter (Hrsg.)
kassel university press, Kassel 2017

# Decision Support for Logistics Networks in Materials Trading Using a Simheuristic Framework and User-generated Action Types

***Entscheidungsunterstützung für Logistiknetzwerke des Werkstoffhandels unter Einsatz eines Simheuristic Framework und benutzergenerierten Maßnahmentypen***

Markus Rabe, Felix Dross, Dominik Schmitt, Majsa Ammouriova, TU Dortmund, Dortmund (Germany), markus.rabe@tu-dortmund.de, felix.dross@tu-dortmund.de, dominik.schmitt@tu-dortmund.de, majsa.ammouriova@tu-dortmund.de

Carola Ipsen, thyssenkrupp Materials Services GmbH, Essen (Germany), carola.ipsen@thyssenkrupp.com

**Abstract:** This paper presents recent extensions made to a previously presented decision support system (DSS) for logistics networks in materials trading. The system uses a simheuristic framework to compute integrated action sets from actions that could possibly be realized in a logistics network. In order to increase the usability and flexibility of the DSS, a concept for extending the system with user-generated action types is presented in this paper. Furthermore, different implementations of the systems' heuristic unit have been tested. In this respect, the paper presents promising results and gives an outlook regarding further research.

## 1 Introduction

Logistics networks are complex systems that can be very hard to manage. With the aim to develop methods and systems that provide decision support for logistics networks, the authors are cooperating with an international materials trading company. The company operates a large, complex and heterogeneous logistics network with over 100 warehouses and an inventory of around 150,000 items on permanent stock. The company uses a data warehouse to measure different key performance indicators (KPI) for different aspects of the logistics network, e.g., to measure inventory productivity or the transportation utilization. Furthermore, different KPI monitoring systems (KPIMS) are used to help the decision makers cope with the complexity of the network (Dross and Rabe 2014). Each KPIMS constantly monitors one KPI and sends an individually composed KPI alert to a responsible decision maker, if the KPI leaves certain predefined corridors. An alert generally consists of two parts: the performance measure that caused the KPI to deteriorate and a set of possible actions that could be performed by the addressed decision maker in order to improve the KPI. The different KPIMS are not connected to each other and each alert is sent to the decision makers of the company independently. As a result, the actions suggested by one KPIMS could improve its own KPI while possibly worsen one or more other KPIs. Therefore, the initial setup could lead to a decline of the overall network performance instead of an improvement (Dross and Rabe 2014; Rabe and Dross 2016). The overall research goal is to develop a decision support system (DSS) for logistics networks that can automatically find action sets that have the potential to improve the overall network situation regarding all KPIs, called integrated action sets (Dross and Rabe 2014).

In order to solve the problem described above, Dross and Rabe (2014) proposed to develop a simheuristic framework as the basis for a decision support system (DSS). A simheuristic approach combines a simulation model with a meta-heuristic (Juan and Rabe 2013). In recent years, a prototype, realizing the concept, has been continuously developed (Rabe and Dross 2015; Rabe and Dross 2016; Rabe et al. 2017). For the DSS, action types define which types of actions can be applied to the simulation model. Therefore, action types influence the search space for the meta-heuristic. In order to extend the functionality of the DSS, a concept to integrate user-generated action types was necessary. This concept and its implications on the DSS components are presented in this paper.

The paper is structured as follows: Section 2 gives an overview of related work. Section 3 introduces the development of a modelling language as the prerequisite for the integration of new action types. Section 4 describes the usage concept for integrating new action types into the system. Section 5 gives an outlook on expected implications on the heuristic unit. Section 6 presents the authors' current approaches for the heuristic unit. Section 7 closes the paper with a conclusion and an outlook.

## 2 Related Work

This section presents related work with an emphasis on DSS for logistics networks and supply chain simulation approaches. Furthermore, the architecture of the DSS is presented.

### 2.1 Decision Support for Logistics Networks

Logistics assistance systems (LAS) have been described as systems providing decision support for logistics networks by Blutner et al. (2007) and Kuhn et al. (2008). LAS are systems which assist planners to quickly identify critical situations and objectively evaluate consequences of possible decision alternatives. Deiseroth et al. (2008) and Bockholt et al. (2011) describe LAS for planning and decision support in supply chains, especially in the automotive sector. Liebler et al. (2013) present a simulation-based approach for gaining insight in global supply networks and explain its use for LAS. The terms LAS and DSS for logistics networks are used synonymously in the literature. In general, DSS is a widely accepted term in the international literature (Shim et al. 2002; Kengpol 2008). For this research, the authors decided to consistently use the term DSS, although the system could also be referred to as a LAS.

## 2.2 Supply Chain Simulation and Simulation-based Optimization

In order to support the research described in this paper, a corresponding prototype has been developed using the supply chain simulation tool SimChain (SimPlan AG 2017). The concept of SimChain has been described by Gutenschwager and Alicke (2004). It is a supply chain simulation tool especially suitable for the analysis of distribution and production networks. SimChain is a data-driven simulation tool. It consists of two components: a set of generic building blocks for a logistics network simulation in Siemens Plant Simulation and a corresponding data model stored in a MySQL database. The data model holds the complete parameterization for the generic building blocks, including the structure of the network. The actual simulation model is dynamically instantiated from the data model at run time. Therefore, a possible action can be applied as a change to the data model, and is then reflected in the instantiated simulation model. The approach, therefore, offers a data-level interface to the simulation model. Another simulation tool with a comparable approach, but targeted more towards distribution networks in the automotive sector, is OTD-NET (Fraunhofer IML 2017). SimChain has been chosen because the authors had good experiences using the software in the E-SAVE project in a comparable application (Rabe et al. 2013).

A good overview of possible combinations of simulation and optimization techniques is provided by März et al. (2011). In case a simulation is used as the evaluation function of an optimization algorithm, this is defined as a "Category D" approach by the German VDI (VDI 3633 Part 12 2016). Juan and Rabe (2013) have proposed to use the term simheuristics when a meta-heuristic is combined with a simulation model to solve stochastic optimization problems. Therefore, this term is also consistently used in this paper. Previous researches have proposed to solve combinatorial optimization problems using genetic algorithms or evolutionary algorithms (EA) (Osaba et al. 2014; Zhang et al. 2015; Cabrera et al. 2016). The EA approach is also the one followed in the research presented here.

## 2.3 Decision Support System for Logistics Networks with a Simheuristic Framework

The simheuristic framework approach for the DSS has already been presented in Dross and Rabe (2014), Rabe and Dross (2015), Rabe et al. (2015), Rabe and Dross (2016) and Rabe et al. (2017). Therefore, this paper only provides a brief overview of the system. An illustration of the system architecture is given in Figure 1.

A model builder software is used to automatically transform the raw data drawn from the data warehouse into a simulation data model. Once the simulation data model has been loaded into the database of the simulation tool, the simulation can be executed, as described in section 2.2. A shadowed data warehouse is used to gather the simulation output data and calculate the relevant KPIs. A credible measurability of the data warehouse KPIs on the simulation output data is ensured by a specifically developed procedure model (Rabe et al. 2015). Once the heuristic unit has registered the initial simulation results, it can experiment by applying actions to the simulation database. An action is resolved into changes by an execution engine. After an action is applied, a new simulation can be instantiated, the simulation can be run and the potential KPI effects of an action can be evaluated by the heuristic unit.

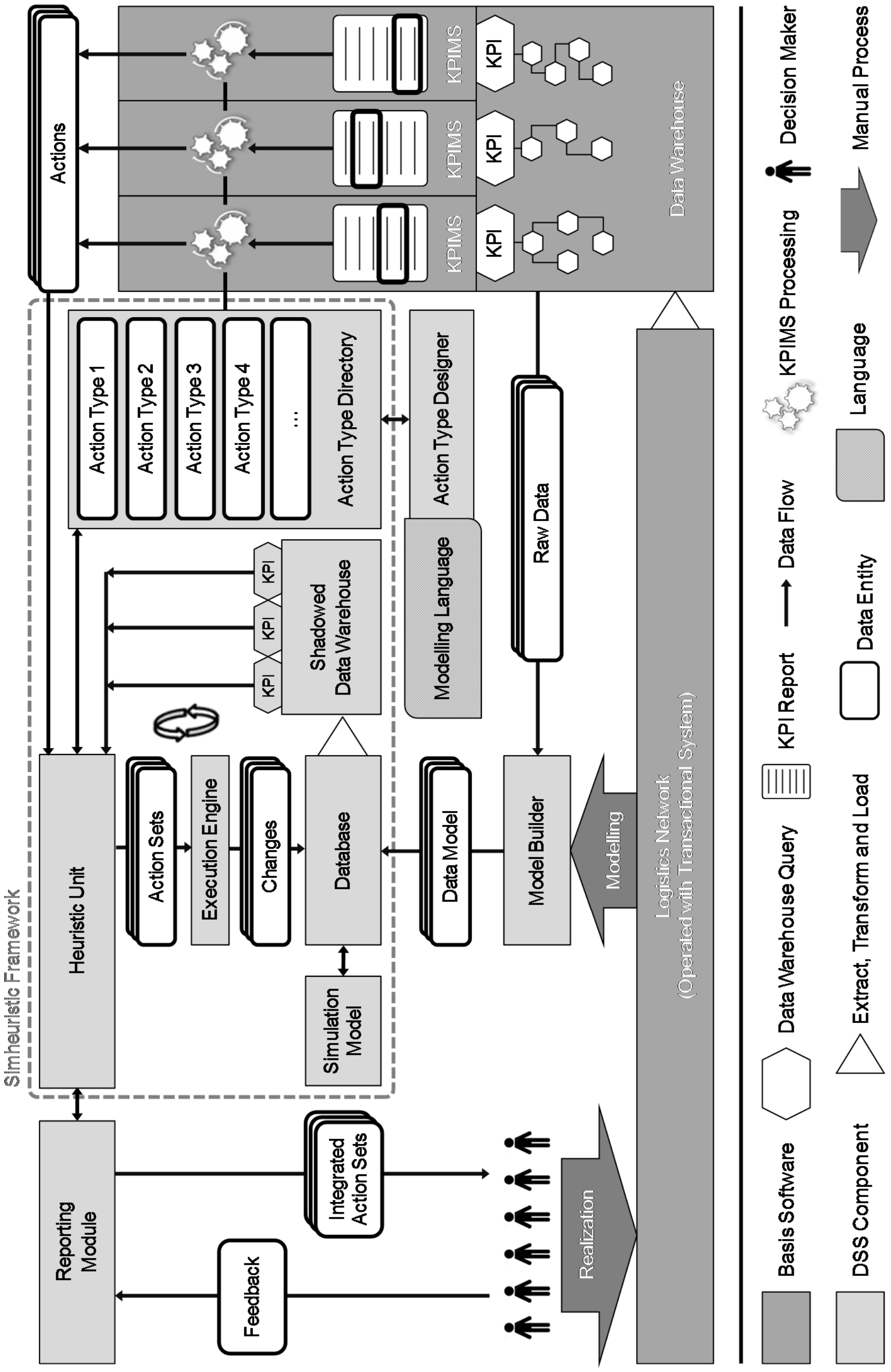


***Figure 1:*** *Architecture of the DSS (Based on Dross and Rabe 2014)*

The heuristic unit can evaluate single actions as well as complete action sets. Integrated action sets can be reported to the decision makers through the reporting module. From there, the decision makers can also provide feedback to the system. The prototypical implementation of the reporting module has been realized as a web-based, responsive user interface (Rabe and Dross 2016). The prototype serves as a proof of concept, and the system is currently under further development. As an important abstraction level to actions, the concept of action types has been introduced (Rabe and Dross 2015; Rabe et al. 2017). An action type describes possible actions in a generic way. E. g., an action type might generally describe the possibility of changing the replenishment of a stock-keeping unit (SKU) in a site to a different supplier. A corresponding action could describe the change of a specific replenishment of a specific SKU in a specific site. Available action types are stored in an action type directory. In order to increase the flexibility of the DSS, a concept for extending the system with user-generated action types has been developed.

## 3 Modelling Language and Action Type Designer

Changing the simulation model means performing changes on one or more parts of the underlying database. In consequence, actions modifying the logistics system are directly correlated with changes in the database. These relations, however, may be very complex and have to be implemented by the user. Understanding the relations between actions and the database is inevitable for extending the system with new action types, but also for an insight into the actions' effects. In order to provide the user with an integrated and consistent way of modeling action types, the authors propose to use a domain-specific modeling language (DSML). A DSML is a language that is easy to understand by a human, but can be executed on a computer. It is focused on a clear and small domain using a bare minimum of features in order to support solving domain-specific problems (Fowler 2011). The DSML is related to multiple areas and components of the system. Therefore, the modeling language is just indicated below the simheuristic framework in Figure 1.

Action types can be created and edited in the action type designer, which is depicted as a square below the action type directory in Figure 1. The action type designer is the bridge between the modeling language and the user's view to the logistics system. All user-generated action types are saved in the action type directory. The user has access to all action types through the action type designer. He can manipulate and update them. Additionally, the user can use these action types to create new action types, e.g., by using them as a basis or by concatenating them.

Eventually, using a feasible interface to access the DSML in order to model action types should provide an easy-to-learn approach for extending the DSS with new actions for the modeled logistics network.

## 4 Usage Concept for Integrating New Action Types

An action type may be related to one or more areas of the underlying data model, depending on its structure. Thus, changing the data model may also lead to adjustments for corresponding action types. In order to reduce this dependency and to provide a more generic approach, the authors propose to use a reference model for the simulation data. With this approach, action types are related to the reference model instead of being directly related to the data model. Information regarding the mapping between the reference model and the data model will be given by a mapping specification. Regarding the action type designer, the authors propose to integrate an access to the structure of the reference model from within the DSML. Thus, the modelling language provides a way of mapping actions for the modeled logistics network to corresponding changes to the reference model. Regarding the user roles, action types may be created and edited by simulation experts or KPIMS experts. A corresponding illustration is given in Figure 2. The user of the action type designer can model new action types by specifying different kinds of information, e.g., references to the corresponding tables in the reference model, required input parameters and data for identification or for increasing the understanding of the action type, such as an ID or a description. Additionally, the action types may contain domain-specific information which can be used by the heuristic unit to increase the search for integrated action sets. The user may be able to define a list of correlations between action types that can be used by the heuristic unit.

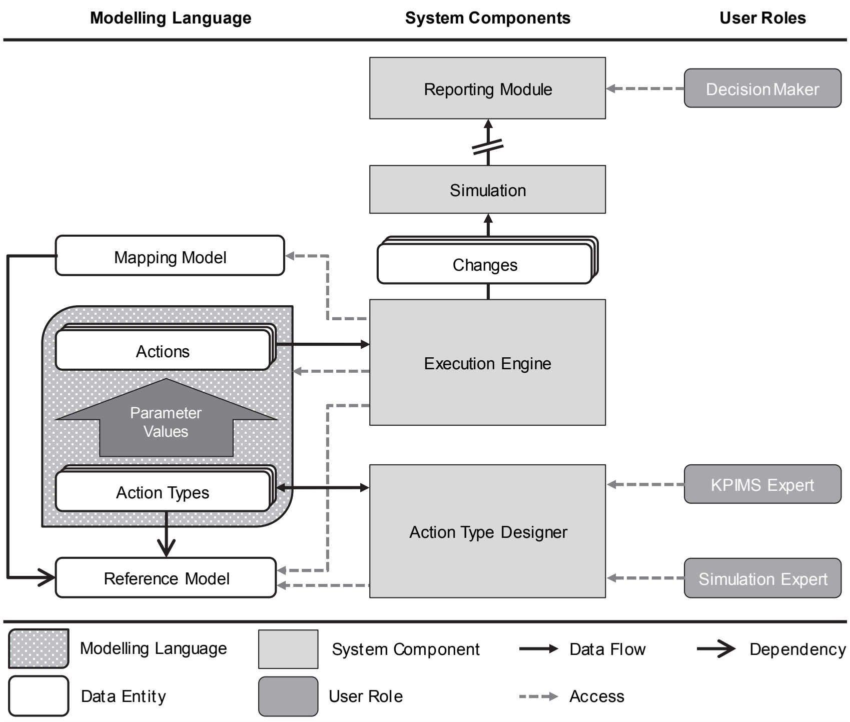


***Figure 2:*** *Modelling Language, System Components and User Roles*

By analyzing correlated action types, promising candidates for further changes to the modeled system can be derived after applying an action. Additionally, a factor for the expected impact regarding its influence to the overall performance of the modeled logistics network can be given. Applying actions might be restricted by constraints, such as the requirement of a specific technique for storing certain SKUs. Complex action types, such as shifting an assortment to a central warehouse may consist of several sub-action types, e.g., adjusting the minimum order quantity or minimum stock of corresponding SKUs, as well as removing or adding transport relations. Thus, the DSML must provide adequate constructs for concatenating action types in order to model these hierarchical structures. Instantiating an action type with parameter values will result in a concrete action. The execution engine can process this action to derive corresponding changes to the data model. Obviously, the execution engine needs access to the mapping model in order to transform actions into changes. The effects of the applied actions are evaluated by running a simulation experiment based on the modified data model. The results of the experiment are provided to the reporting model. The decision maker can access the simulation results through the reporting model.

## 5 Implications on the Realization of the Heuristic Unit

The heuristic unit needs to have access to actions types stored in the action type directory, in order to add corresponding possible actions to the solution space. Accordingly, decision variables define the selected actions. The heuristic unit has to determine the concrete actions and search for the most promising integrated action sets. With the new concept, the heuristic unit will also get access to all domain-specific information of action types. Hence, it can possibly utilize this information in its search. From this perspective, it can be expected that the heuristic unit can possibly fasten the selection of action sets from the solution space. Accordingly, the learning ability of the heuristic unit is expected to increase, and the search process may become faster and more effective. Furthermore, the link between the heuristic unit and the reporting module is expected to be improved. This might increase the learning ability of the heuristic unit. The heuristic unit will report its findings, most promising integrated action sets, to the decision makers. The decision makers can modify the action sets by deleting actions or suggesting actions. Additionally, feedback regarding the effects of action set realizations in the real system can be reported. As an additional benefit, the new concept could lead to a better understanding of the proposed action sets and support the search for suitable actions.

## 6 Approaches for the Realization of the Heuristic Unit

As shown in Figure 1, the heuristic unit interacts with the simulation model in an iterative manner. Currently, the authors implemented two different approaches for the heuristic unit, an EA implementation and a Deep Reinforcement Learning (DRL) implementation. EA has been chosen, because it has previously been successfully used in similar kinds of research (see section 2.2). On the other hand, DRL has been selected since an RL agent has the ability to learn based on rewards. This should enable the DSS to speed up the search of integrated action sets based on previous simulation runs. Experiments have been run with these prototypes to compare the performance of the selected approaches. A small-size logistic network has been used for the experiment consisting of 5 sites, 103 customers, 30 SKUs and 176 orders.

The EA mimics the biological evolution. In this approach, a possible action set is represented as an individual in a population, and discrete-event simulation is used to evaluate the fitness of the individuals. Cost and β-service level have been used as fitness values to be minimized and maximized, respectively. New individuals in the successive generations are formed by selecting individuals from the population based on their fitness, and using cross-over or mutation. EA algorithms require the definition of parameters: cross-over probability, mutation probability, generation size, and initial individual. In addition, terminating conditions should be defined: maximum experiment time, maximum number of generations and stagnation condition. For the experiments, the following parameter values have been set: cross-over probability 70 %, mutation probability 25 %, maximum number of generations 50, maximum experiment run time 2 hours, and the NSGA2 algorithm for individual selection. The remaining parameters have been iterated according to Table 1.

DRL combines RL with a convolutional neural network. RL utilizes an agent that searches the solution space and selects actions based on their previously recorded rewards. Each action and state combination has an associated expected return $Q^{\pi}(s, a)$, resulting from applying action $a$ in state $s$ while following policy $\pi$. The Q-learning algorithm is applied in the RL as follows:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\left[r_{t+1} + \gamma \max_a Q(s_{t+1}, a_t) - Q(s_t, a_t)\right] \tag{1}$$

Where $\alpha$ is a learning rate parameter, $r$ is the expected immediate reward, and $\gamma$ is the discount rate parameter at time $t$. The learning rate has been set to 0.00025, while the discount rate parameter has been varied as in Table 2.

Tables 1 and 2 show cost reduction, β-service level improvement, the experiment run time, and the number of simulation runs during the experiments. The EA converges faster than the DRL. The EA terminates within 1.5 hours or less, depending on the parameter setting. On the other hand, DRL requires more time, but it has the ability to learn and build experience. Therefore, DRL is able to deliver an acceptable solution for a specific situation without conducting new simulation runs, just based on its experience from past experiments.

***Table 1:*** *EA experiments' parameters and performance*

| Initial individual size | Generation size | Cost reduction [%] | Service level improvement [%] | Run time [minutes] | Simulation runs |
|---|---|---|---|---|---|
| 1 | 4 | 0.38 | 0.11 | 7 | 10 |
| 1 | 16 | 2.73 | 0.24 | 17 | 32 |
| 4 | 4 | 5.48 | 0.34 | 87 | 158 |
| 4 | 16 | 5.79 | 0.45 | 90 | 151 |

***Table 2:*** *DRL experiments' parameter and performance*

| Discount rate parameter | Cost reduction [%] | Service level improvement [%] | Run time [minutes] | Simulation runs |
|---|---|---|---|---|
| 0.1 | 2.00 | 0.02 | 630 | 600 |
| 0.5 | 5.00 | 0.08 | 510 | 600 |

## 7 Conclusion and Outlook

The DSS presented in this paper uses a simheuristic framework. The framework allows for experimenting with different components. Since the complete interaction between the modules is done on a data base, experiments with different modules can be conducted. Thus, the simheuristic approach can be tested with different implementations for the heuristic unit. The simulation kernel could possibly be also exchanged by alternative simulation tools in the future. Therefore, the simheuristic framework already provides a very modular research approach for building a DSS for logistics networks. Furthermore, the system will now be extended with the possibility to integrate user-generated action types, which will again increase the flexibility of the system. How the heuristic unit can handle these new action types will be of special interest regarding further research. Besides this aspect, the authors plan to generally extend their research in DRL and simheuristics. Possible combinations will be considered in the future. Regarding this, a special focus will be on considering larger logistics networks, and its impact on the total simulation time.

## Acknowledgement

Special thanks to thyssenkrupp Materials Services GmbH for supporting this research. The work was also supported by the German Jordanian University.

## References

Blutner, D.; Cramer, S.; Krause, S.; Mönks, T.; Nagel, L.; Reinholz, A.; Witthaut, M.: Technical Report 06009 „Assistenzsysteme für die Entscheidungsunterstützung“. ISSN 1612-1376, Dortmund, 2007.

Bockholt, F.; Raabe, W.; Toth, M.: Logistic assistance systems for collaborative supply chain planning. International Journal of Simulation and Process Modelling 6 (2011) 4, pp. 297-307.

Cabrera, G.; Niklander, S.; Cabrera, E.; Johnson, F.: Solving a distribution network design problem by means of Evolutionary Algorithms. Studies in Informatics and Control 25 (2016) 1, pp. 21-28.

Deiseroth, J.; Weibels, D.; Toth, M.: Simulation-based decision support system for the disposition of global supply chains. In: Rabe, M. (Eds.): Advances in Simulation for Production and Logistics Applications. Stuttgart: Fraunhofer IRB 2008, pp. 41-50.

Dross, F.; Rabe, M.: A SimHeuristic framework as a decision support system for large logistics networks with complex KPIs. In: Wittmann, J.; Deatcu, C. (Eds.): Proceedings of the 22nd Symposium Simulationstechnik, Berlin (Germany), 3rd-5th September 2014, pp. 247-254.

Fowler, M.: Domain-specific languages. Upper Saddle River, NJ: Addison-Wesley 2011.

Fraunhofer IML: OTD-NET. 2017. http://www.iml.fraunhofer.de/de/themengebiete/supply_chain_engineering/produkte/otd-net.html. Last access 15.05.2017.

Gutenschwager, K.; Alicke, K.: Supply Chain Simulation mit ICON-SimChain. In: Spengler, T.; Voß, S.; Kopfer, H. (Eds.): Logistik Management. Heidelberg: Physica 2004, pp. 161-178.

Juan, A.A.; Rabe, M.: Combining simulation with heuristics to solve stochastic routing and scheduling problems. In: Dangelmaier, W.; Laroque, C.; Klaas, A. (Eds.): Simulation in Produktion und Logistik 2013. Paderborn: HNI-Verlags-schriftenreihe 2013, pp. 641-649.

Kengpol, A.: Design of a decision support system to evaluate logistics distribution network in Greater Mekong subregion countries. International Journal of Production Economics 115 (2008) 2, pp. 388-399.

Kuhn, A.; Hellingrath, B.; Hinrichs, H.: Logistische Assistenzsysteme. In: Software in der Logistik. München: Huss 2008, pp. 20-26.

Liebler, K.; Beissert, U.; Motta, M.; Wagenitz, A.: Introduction OTD-Net and LAS: Order-to-delivery network simulation and decision support systems in complex production and logistics networks. In: Pasupathy, R.; Kim, S.-H.; Tolk, A.; Hill, R.; Kuhl, M.E. (Eds.): Proceedings of the 2013 Winter Simulation Conference. Piscataway: IEEE 2013, pp. 439-451.

März, L.; Krug, W.; Rose, O.; Weiger, G., Hrsg.: Simulation und Optimierung in Produktion und Logistik: Praxisorientierter Leitfaden mit Fallbeispielen. Heidelberg: Springer 2011.

Osaba, E.; Carballedo, R.; Diaz, F.; Onieva, E.; La Iglesia, I. de; Perallos, A.: Crossover versus mutation. The Scientific World Journal, 2014, DOI 10.1155/2014/154676.

Rabe, M.; Dross, F.: A Reinforcement Learning approach for a decision support system for logistics networks. In: Yilmaz, L.; Chan, W.K.V.; Moon, I.; Roeder, T.M.K.; Macal, C.; Rossetti, M.D. (Eds.): Proceedings of the 2015 Winter Simulation Conference. Piscataway: IEEE 2015, pp. 2020-2032.

Rabe, M.; Dross, F.: Intelligente Entscheidungsunterstützung für Logistiknetzwerke des Werkstoffhandels. Werkstoffe in der Fertigung (2016) 5, pp. 24-25.

Rabe, M.; Dross, F.; Vennemann, A.: A procedure model for the credible measurability of data warehouse metrics on discrete-event simulation models of logistics systems. In: Rabe, M.; Clausen, U. (Eds.): Simulation in Produktion und Logistik 2015. Stuttgart: Fraunhofer 2015, pp. 167-176.

Rabe, M.; Dross, F.; Wuttke, A.: Combining a discrete-event simulation model of a logistics network with deep reinforcement learning. In: Duarte, A.; Juan, A.A.; Lourenço, H.R. (Eds.): Proceedings of the 12th Metaheuristics International Conference (MIC), Barcelona (Spain) 2017, pp. 438-447.

Rabe, M.; Gutenschwager, K.; Fechteler, T.; Sari, U.M.: A data model for carbon footprint simulation in consumer goods supply chains. In: Pasupathy, R.; Kim, S.-H.; Tolk, A.; Hill, R.; Kuhl, M.E. (Eds.): Proceedings of the 2013 Winter Simulation Conference. Piscataway: IEEE 2013, pp. 2677-2688.

Shim, J.P.; Warkentin, M.; Courtney, J.F.; Power, D.J.; Sharda, R.; Carlsson, C.: Past, present, and future of decision support technology. Decision Support Systems 33 (2002) 2, pp. 111-126.

SimPlan AG, 2017: SimChain. http://www.simchain.net/. Last access 15.05.2017.

VDI 3633 Part 12: Simulation von Logistik-, Materialfluss- und Produktionssystemen – Simulation und Optimierung. Berlin: Beuth 2016.

Zhang, T.; Xie, S.; Rose, O.: Flexible job-shop scheduling with dynamic stochastic machine sets. In: Rabe, M.; Clausen, U. (Eds.): Simulation in Produktion und Logistik 2015. Stuttgart: Fraunhofer 2015, pp. 21-28.