Proceedings of the 2015 Winter Simulation Conference
L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, and M. D. Rossetti, eds.

A REINFORCEMENT LEARNING APPROACH FOR A DECISION SUPPORT SYSTEM FOR LOGISTICS NETWORKS

Markus Rabe

Department of IT in Production and Logistics
Technical University Dortmund
Leonhard-Euler-Straße 5
44227 Dortmund, GERMANY

Felix Dross

Graduate School of Logistics
Technical University Dortmund
Leonhard-Euler-Straße 5
44227 Dortmund, GERMANY

ABSTRACT

This paper presents the architecture and working principles of a Decision Support System (DSS) for logistics networks. The system relies on a data-driven discrete-event simulation model. A brief introduction to Reinforcement Learning (RL) and an explanation of the adoption of RL to the concepts of the DSS is given. An illustration of the realization is presented using a specific aspect of a logistics network. The logistics network is described in a data model which is represented by database tables. The tables are used to dynamically instantiate the simulation model. The authors describe how SQL queries can be used to model actions of an RL agent. A Data Warehouse can be used to measure Key Performance Indicators on the simulation output data of the simulation model, which can be used as a reward criterion for the RL agent. The paper presents a basis for the ongoing development of an RL agent.

1 INTRODUCTION

Logistics networks are very complex socio-technical systems which operate in an environment of uncertainty (McGinnis 2005). In order to cope with the complexity of these systems, many companies have developed dedicated logistics departments. They strive to provide accurate business reports to their managers to help them to decide about the right measures in the logistics network. The generation of reports is usually supported by Data Warehouse (DWH) technology.

DWHs provide data structures in order to support tools for analytical decision making. In contrast to common databases, DWHs store data in redundant and aggregated ways, speeding up interactive analysis and providing data at sufficient aggregation levels. (Ehmke et al. 2011)

Online Analytical Processing (OLAP) software provides a fast, flexible and interactive access to the data in a DWH and enables the organization, aggregation and visualization of information. Data are presented in terms of hypercubes depicting multidimensional structures. The cubes visualize system performance measures (the cube's cells) in context of their dimensions (the cube's borders) and therefore enable the flexible and multidimensional analysis of the data. (Bauer and Günzel 2013; Jarke et al. 2003)

OLAP technology is often used to realize Performance Measurement Systems (PMS). A PMS unites different performance measures that relate to each other in a hierarchical form and typically culminate in one key performance indicator (KPI). KPIs are typically used by the management to measure, control and steer the logistics systems. Furthermore, some companies have developed dedicated KPI Monitoring Systems (KPIMS), which are designed to ensure a constant improvement of a logistics network regarding the monitored KPIs (Dross and Rabe 2014). Each KPIMS constantly monitors one KPI and sends an individually composed KPI Alert to a responsible manager if the KPI leaves certain predefined corridors. A KPI Alert generally consists of two parts: A list of facts that caused the KPI to deteriorate and a set of possible actions that could be performed by the addressed manager in order to improve the KPI.

978-1-4673-9743-8/15/$31.00 ©2015 IEEE

Nevertheless, even with state-of-the-art DWHs, OLAP technology and smart KPIMS, the outcomes of certain actions in a logistics network are very hard to predict. In many situations, the managers are groping in the dark when it comes to decide about the right correcting changes in their network. It gets even more difficult if a manager tries to predict the consequences of a change regarding multiple KPIs at once, including the temporal development of the network. As a logical consequence, businesses are demanding for better solutions to plan possible changes in their logistics networks.

The authors are currently developing a smart Decision Support System (DSS), which uses a Discrete-Event Simulation (DES) model to predict the consequences of possible changes in the logistics network. Therefore, a mechanism has been developed to measure real world DWH KPIs on the simulation data (Dross and Rabe 2014). With the concept of a central Heuristic Unit (HU), the system is designed to automatically suggest smart changes in different areas of the network and predict their temporal effects regarding the overall development of the network. Cooperating with a large, international trading company, the authors are able to test the system with data from over 100 warehouses in different countries. In this paper, the authors present how Reinforcement Learning (RL) techniques are used to implement and test a novel approach to the realization of the internal HU in form of an RL agent.

The paper is structured as follows: Section 2 presents an overview of the related work regarding simulation-based DSS and the use of RL techniques in the context of logistics networks. Section 3 provides a description of the architecture and the general working principles of the DSS described in this paper. Section 4 provides an illustration of the realization of the system with RL techniques. It briefly describes the logistics network of the company providing the data for the study. A specific aspect of the logistics network and possible actions are presented. A brief introduction to RL and an explanation of the adoption of RL concepts to the DSS is given. Section 5 closes the paper with an outlook on future research.

2 RELATED WORK

2.1 Simulation-based Decision Support Systems

Liebler et al. (2013) presented a simulation-based approach for gaining insight in global supply networks and explained its use for Logistic Assistance Systems (LAS). LAS are defined as 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) are further publications that described LAS for planning and decision support in supply chains, especially in the automotive sector. In general, the concepts of LAS have been described by Kuhn et al. (2008) and Blutner et al. (2007). The terms LAS and DSS for logistics networks are mostly used synonymously in the literature. For this paper, we decided to consistently use the term DSS, although the system described here could also be referred to as a LAS.

Heilala et al. (2010) presented a simulation-based DSS which can be used to help planners and schedulers organize production more efficiently. Although designed for a different problem domain, they explained the major challenges for a DSS, which are the data integration, the automated simulation model creation and the visualization of results for interactive and effective decision making.

The combination of heuristics and meta-heuristics with simulation techniques in order to efficiently solve stochastic combinatorial optimization problems has been described as SimHeuristics by Juan and Rabe (2013). A SimHeuristic Framework as a DSS designed around this approach has been described by Dross and Rabe (2014). A good general introduction to the combination of simulation and optimization techniques can be found in März et al. (2011).

2.2 Reinforcement Learning in the Context of Logistics Networks

RL is the area of machine learning concerned with learning the actions that a software agent (RL agent) should take in a particular environment in order to maximize its rewards. It has attracted a considerable amount of interest recently. Classical textbooks on this topic are Sutton and Barto (1998), Bertsekas and

Tsitsiklis (1996) and Gosavi (2015). A general book on artificial intelligence with a good section about RL is Poole and Mackworth (2010). One of the most popular RL algorithms, also described in this paper, is the Q-Learning algorithm, which was developed by Watkins (1989).

The usage of RL techniques in the logistics domain date back to 2002. Pontrandolfo et al. (2002) proposed an approach to study global supply chain management problems using RL techniques. They explained how the RL framework allows the management of global supply chains under an integration perspective. Giannoccaro and Pontrandolfo (2002) presented an approach to manage inventory decisions at all stages of a supply chain in an integrated manner. They described how RL is used to determine a near optimal inventory policy under an average reward criterion. Stockheim et al. (2003) presented a decentralized supply chain management approach based on RL. They showed that an RL solution outperformed a simple heuristic for all their training states. Subramaniam and Gosavi (2004) described how RL can be used to solve problems related to replenishing inventories at retailers in distribution networks operated under the paradigm of Vendor Managed Inventory (VMI). Another RL approach to determine a replenishment policy in a VMI system with consignment inventory has been presented by Zheng Sui et al. (2010). Qiu et al. (2007) described an approach where an RL algorithm is used to obtain the decision policies and system costs regarding different business service modes in distribution systems. Chaharsooghi et al. (2008) described how they considered supply chain ordering management as a multi-agent system and formulated it as an RL model. They proposed a Q-learning algorithm to solve the RL model. Further explanations of the use of Q-Learning in the supply chain context can be found in Zhang and Bhattacharyya (2007) and Tim van Tongeren et al. (2007).

To the best of our knowledge, there exists no description of a DSS for logistics networks which uses RL techniques to evaluate different action alternatives in the background, before suggesting concrete actions to the decision maker.

3 THE DECISION SUPPORT SYSTEM

The DSS described in this paper is build around the concept of a SimHeuristic Framework, which has been previously introduced by Dross and Rabe (2014). In this section, the authors will briefly summarize the most important parts. Additionally to Dross and Rabe (2014), further figures are introduced to clarify the technical implementation and the general working principles of the proposed DSS.

3.1 Architecture

The architecture of the DSS is shown in Figure 1. First, the data of the company, which is planning to implement the system, are used to create a DES model of the logistics network. This includes the process data, stock data and structure data. The process data and stock data are regularly collected by the transactional systems and are regularly transferred to the DWH. This process contains the steps Extract, Transform and Load and is therefore abbreviated with ETL. Once the data are in the DWH, they can comfortably be analyzed using OLAP technology. This can be especially useful when it comes to the parameterization of the simulation model. Structure data, e.g. warehouse capacities, might also be available in the DWH. If not, they need to be obtained from separate databases or files. Ideally, the process of simulation model parameterization should be automated or semi-automated, so that the simulation model can be automatically updated with the latest data from the logistics system under consideration. The implementation presented in this paper uses a data-driven simulation tool with a generic simulation model, which makes it easy to update the data of the simulation model. A more detailed description of the simulation tool will be given in the remainder of this section.

In the second step, the KPI logic of the DWH has to be copied to a Shadowed Data Warehouse (SDWH). This virtual DWH ensures that no simulation output data are mixed with data from the actual logistics system. In order to work properly, the data model of the DWH and the exact queries for each KPI have to be transferred to the SDWH. This logic shadowing is another process which should ideally be automated, as well as the model parameterization process.

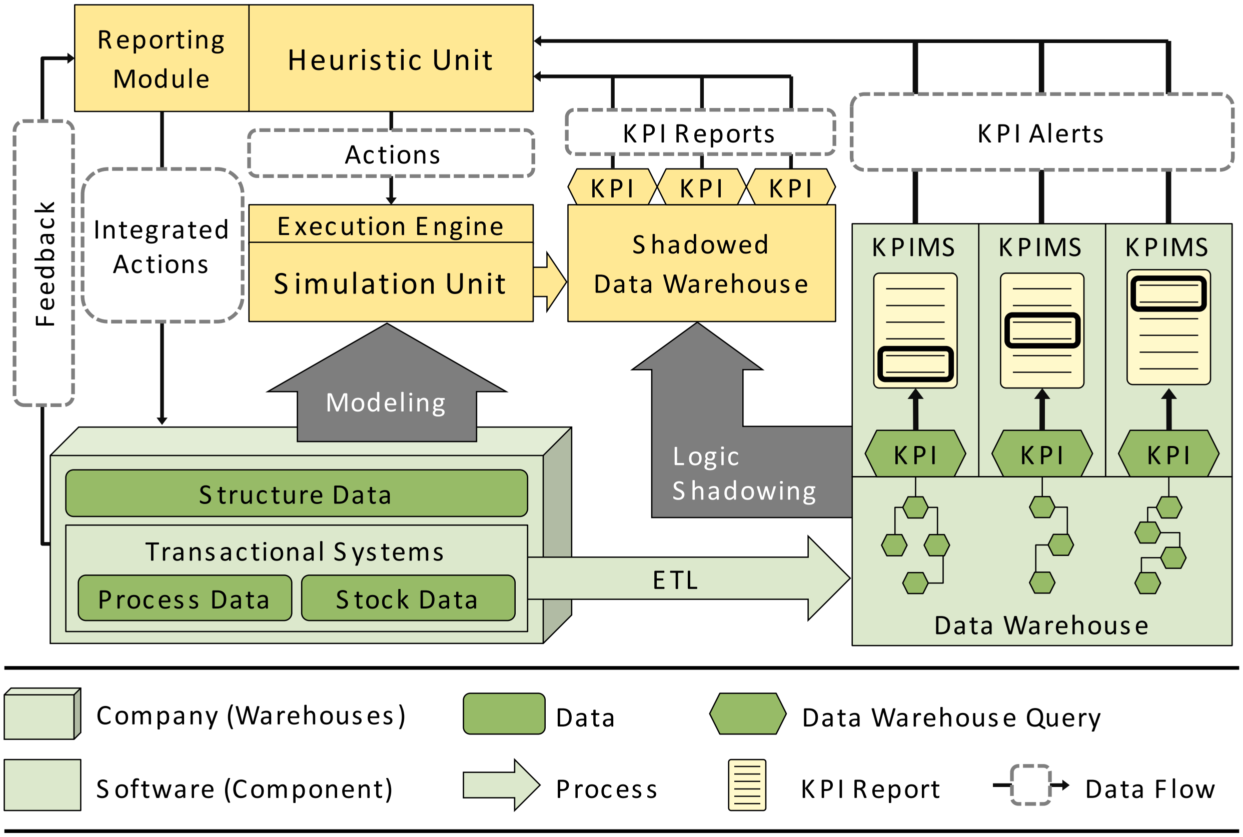

Figure 1: The architecture of the Decision Support System based on Dross and Rabe (2014).

As briefly described in the introduction, DWH technology is often used to realize complex PMS. In Figure 1, the hierarchical structure of each PMS is illustrated by a tree of DWH queries, each of them culminating in one KPI. For each KPI, there exists one KPIMS, which periodically generates a KPI Report and examines it on the basis of specific, predefined criteria. Within this analysis, there are certain conditions which can trigger a KPI Alert, e.g. if a KPI has passed a predefined, fixed nominal value or if the development of a KPI shows a negative trend. The resulting KPI Alert typically consists of two parts: The facts that triggered the KPI Alert and a list of possible actions that could be performed by the manager in order to improve the KPI, respectively remove the conditions which triggered the KPI Alert.

Without the DSS presented here, the KPI Alerts would be sent directly to the responsible manager in the company. With multiple, decoupled KPIMS, this could lead to a situation where one KPIMS would strive to improve its own KPI while possibly deteriorating one or more other KPIs. The concept could lead to discontent among the managers and deteriorations instead of improvements of the logistics network. Therefore, automated interdependence analyses of the different suggested actions are proposed to be performed by the HU. The results of these interdependence analyses are integrated actions, which are supposed to improve the overall network situation instead of just one KPI.

Conceptually, the HU is able to execute the different action possibilities on the simulation model with the help of an Execution Engine. Regarding this aspect, the realization with a data-driven simulation model becomes very useful. Once an action has been executed on the simulation model, it is necessary for the HU to evaluate the effects of the action. In order to get an estimate of the effects caused by an action, a simulation experiment has to be conducted with the modified simulation model. After the simulation experiment has been conducted, the traces of the simulation experiment are extracted, transformed and loaded into the SDWH using an abstracted form of the ETL process. Applying the KPI logic in the SDWH, the simulated effects of the actions on the KPIs can be evaluated. Using this framework, the HU is able to test different actions on the simulation model before suggesting integrated actions to the responsible managers in the company. The Reporting Module is planned to generate the final reports, which are sent to the managers. By means of a feedback mechanism, the HU should be able to learn about important structural changes in the network, which cannot be automatically derived from the DWH. Thinking further, the HU should also be able to learn from real world effects of previously suggested actions. The feedback could be received manually from the executing managers or automatically from the surveillance of actually occurring changes in the data of the logistics system.

The described framework has already been extensively described in Dross and Rabe (2014) and the interested reader is referred to this paper for more detailed descriptions and explanations. In the remainder of this paper, the authors will explain their first implementation results of the described SimHeuristic Framework as well as their experiences using RL techniques for the realization of the HU as an RL agent.

3.2 The Simulation Tool

The simulation tool used for the prototypical implementation of the described SimHeuristic Framework is SimChain (SimPlan AG 2015). It consists of a generic supply chain simulation model for Siemens Plant Simulation and a corresponding data model stored in a MySQL database. The actual supply chain simulation model is dynamically instantiated from the data model, which describes the concrete configuration and parameterization of the generic building blocks. The structure and modeling approach of SimChain has been described in Gutenschwager and Alicke (2004). SimChain was chosen, because of its suitability and the authors' experience using it. It has been recently used in the e-SAVE project, which was funded by the European Commission (e-SAVE 2015; Rabe et al. 2012; Rabe et al. 2013).

Figure 2 conceptually visualizes the realization of the SimHeuristic Framework with the simulation tool SimChain. The data model of SimChain contains database tables for all information necessary to describe a supply chain. The database tables are divided into basis and configuration tables. With the basis tables, the basic structure or layout of the logistics system is described. This includes e.g. the geographical locations of the sites, suppliers and customers. The configuration tables are used for the detailed specification of the dynamics of the simulation model. This includes for example the allocation of Stock Keeping Units (SKUs) to sites or the customer demands for SKUs at a site. The configuration tables allow for using configuration indices and the configuration indices can be used to define scenarios. This reference structure enables the modeler to easily specify different scenarios, constructed out of different configuration indices. It enables for example the description of scenarios with different SKU demands, but with the same replenishment configuration in each scenario. The output data of the simulation experiments is written back into specific statistics tables in the database. The granularity of the output data can again be specified with corresponding configuration tables.

Following the idea of the SimHeuristic Framework, the raw data for the simulation data model are automatically derived from the DWH in predefined periods. As shown in Figure 2, an Input Data ETL Module is responsible to periodically extract the data from the DWH and transform and load it into the data structures of the simulation data model. The step of conceptually modeling the logistics network and defining the Input Data ETL processes is illustrated by the grey Modeling process in Figure 1. The data from the DWH can be combined with forecast scenarios, which have to be specified upfront by the domain experts. This may translate into different configurations for the dynamic behavior of the logistics system within the simulation data model. The transforming steps have to be periodically validated and verified with automated procedures and by the responsible persons. Once the automated procedures and the business processes for validation and verification are set up, the process of updating the simulation data model can be automated to a high extend.

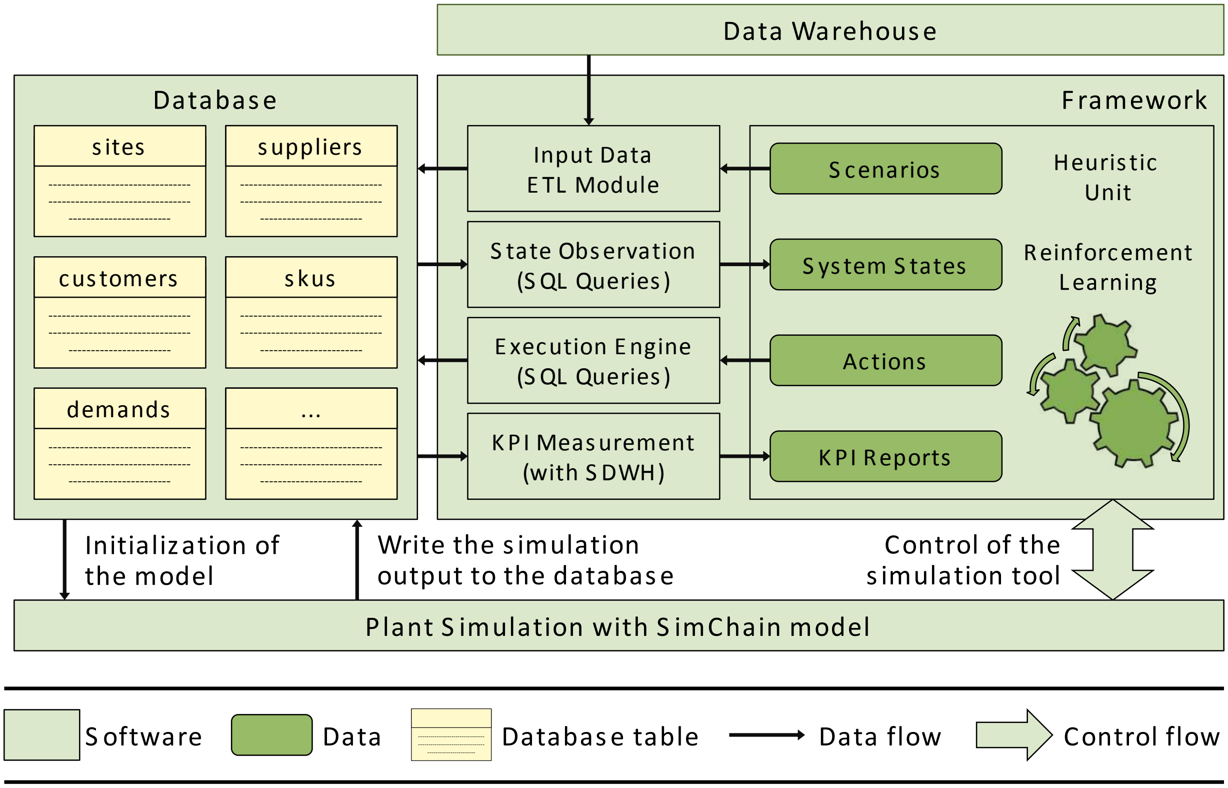

Figure 2: Concept for the realization of the SimHeuristic Framework with the simulation tool SimChain.

As shown in Figure 2, the HU can receive the current system state with a State Observation module. This module uses SQL queries to extract the attributes necessary to describe the current system state and stores them accordingly. Action executions are realized with changes in the data model, not with changes in the instantiated simulation model. Thus, the Execution Engine performs SQL transactions to execute actions. Possible actions have to be defined as descriptions of changes in the simulation data model. Each SQL transaction has to fulfill the ACID requirement (Atomicity, Consistency, Isolation, Durability), leaving the data model in a consistent state (Coronel and Morris 2014). After a simulation experiment has been conducted, SimChain writes the simulation output data to the statistics tables in the database. From there the data can be extracted, transformed and loaded into the SDWH. The shadowed KPI logic is then used to generate KPI reports from the simulation output data. This setup builds the foundation for the realization of the HU as an RL agent.

3.3 General Working Principle

The general idea behind the realization of the DSS with the help of RL techniques is that it should be able to learn from experiments with the simulation model. At the time where the decision maker asks the system for a recommendation, it should already have knowledge about the possible consequences of actions, because it already has experience in applying actions to the simulation model of the logistics system. This is a fundamental difference to a classical simulation-based optimization approach, which can be explained with a simple example.

Let S^r be the set of all possible system states of the logistics system under consideration. Let S be the set of all equivalent, possible system states of the simulation model of the logistics system. Let the system state $s_t^r \in S^r$ be the state of the real logistics system at the time t, for which the logistics manager needs a recommendation from the DSS. Let further be $s_t \in S$ the equivalent system state in the simulation model of the logistics system. Let $A(s_t)$ be the set of possible actions in a state s_t and $n = |A(s)|$ the number of possible actions in the state s_t. If n is a large number, the problem to decide which action combination should be taken becomes a NP-hard combinatorical problem (Juan and Rabe 2013). In a classical simulation-based approach a heuristic or meta-heuristic would be used to solve this problem. At the time t where the decision maker would ask for a recommendation from the DSS, the system would take the simulation model at system state s_t and from there on intelligently try different actions $a_t \in A(s_t)$. The computation therefore typically starts at the time t. Furthermore, a temporally shifted execution of actions and the consequential development of the model are usually not considered.

Using an RL approach, it is possible to train the system in advance, using the system state s_{t-1} to learn a policy for this particular system state s_{t-1}. A policy π defines which action a_t should be taken in a particular state s_t. An approximation for the best action a_t in state s_t can be obtained with an approximation of the so-called action-value function for state s_t using a function approximation technique. An explanation of the underlying principles will now be given using a concrete example.

4 ILLUSTRATION OF THE REALIZATION

4.1 The Simulation Model

As briefly mentioned in the introduction, the authors are cooperating with a large, international trading company with over 100 warehouses in different countries and are thus able to test the system with different data sets. The company has an inventory of around 150,000 items on permanent stock and operates a large, complex and heterogeneous logistics network. It is organized as a decentralized multi-echelon network with central, regional and local warehouses. Each warehouse has a sales division that receives customer orders on a consistent basis. A special characteristic of the logistics network are certain warehouses that can perform value-added services, for example cutting, drilling or milling. Shuttle transfers between the warehouses and the shipment of the goods to the customers are performed using separated fleets with variable amounts of vehicles.

In the following the authors will use one very basic, but specific aspect of the logistics network in order to illustrate the working principles of the described DSS. An illustration is given in Figure 3. In this SimChain simulation model, three sites with their respective customers have been modeled. Each customer is allocated to one delivery route. There exist only one article (SKU) in the model and only one plain supplier, from which the sites can potentially replenish the article. Only one future customer demand scenario is considered in order to keep the complexity to a minimum. The decision a logistics manager has to make for each site is whether a site should stock keep an article and regularly replenish it from the plain supplier, or if the article should be stock kept at another site and the article should be called on demand. If a site is replenishing from a plain supplier, a dynamic, demand-oriented safety stock calculation is used to calculate the reorder point. The authors will use the term system configuration to describe the combination of site configurations. Here, a site configuration could be represented with one bit, indicating whether the site is stock keeping or not. The system configuration could thus be described using a binary vector $\vec{c} = \{1, \ldots, x\}$, where x is the number of sites.

Figure 4 shows two exemplary system configurations out of the set of all possible system configurations. The manager has the task to choose the actions which will lead to the the most promising configuration. The decisions about the right actions in the logistics system involve the consideration of the stochastic behavior of the system, e.g. the customer demands at each site, the transportation cycles between the sites, the replenishment time of the plain supplier, the transportation costs, the inventory costs and the resulting service level at each customer. Of course, capacity restrictions have to be considered as well. Thus, even for this very basic decision problem with only one article, the use of a DES for the evaluation of the system behavior is justified.

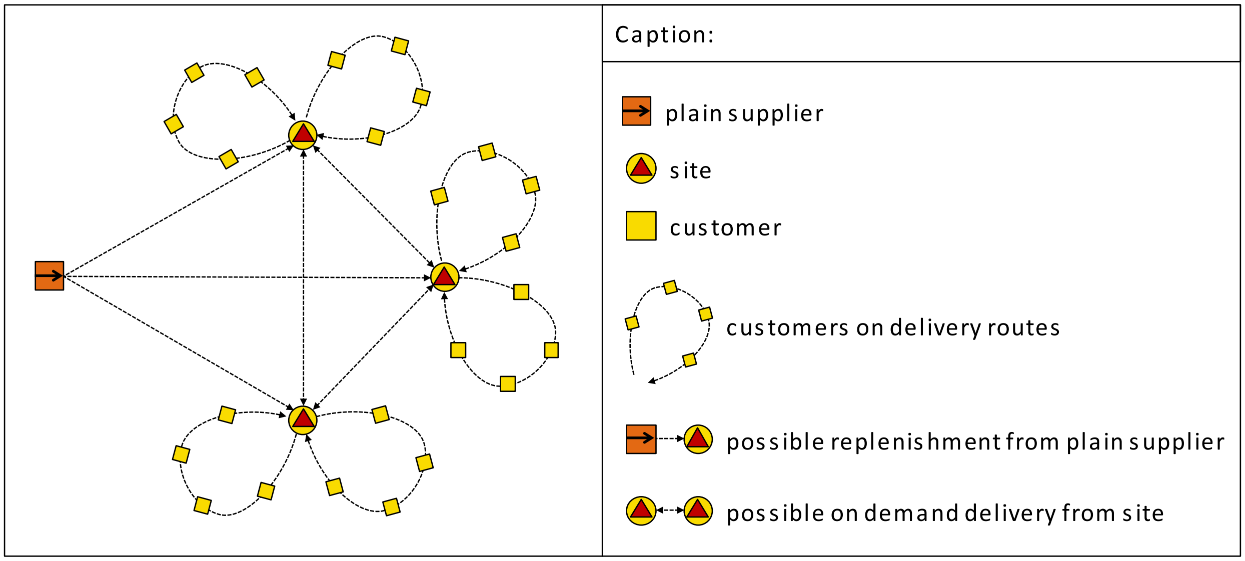
Caption:
plain supplier
site
customer
customers on delivery routes
possible replenishment from plain supplier
possible on demand delivery from site

Figure 3: The logistics system under consideration.

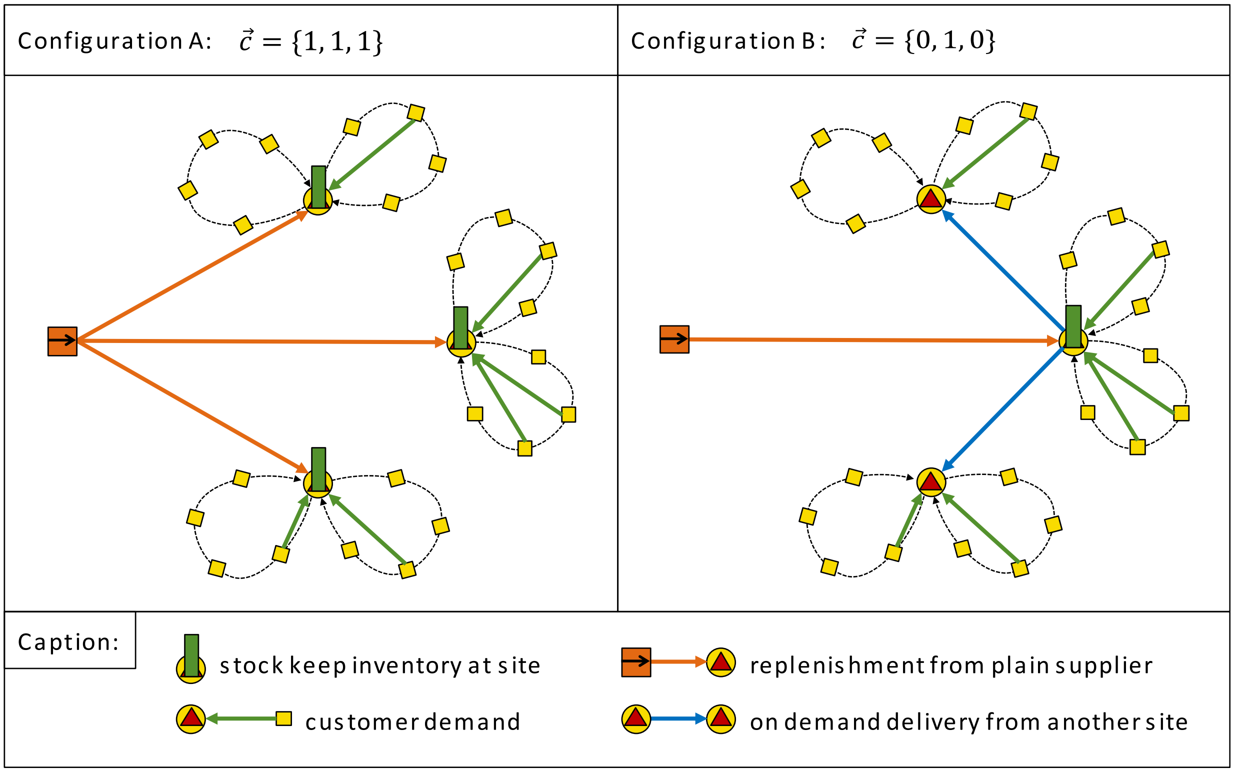
Configuration A: $\vec{c} = \{1, 1, 1\}$
Configuration B: $\vec{c} = \{0, 1, 0\}$
Caption:
stock keep inventory at site
replenishment from plain supplier
customer demand
on demand delivery from another site

Figure 4: Visualization of two exemplary system configurations A and B.

4.2 Application of Reinforcement Learning

The decision problem a manager might be facing as described above can also be formulated as a Markov Decision Problem (MDP). An MDP consists of

- a set of possible states s,
- a set of possible actions a,
- a specification of all the transition probabilities $P(s'|s,a)$, specifying the probability of transitioning to state s' if action a is taken in state s and
- a specification of all the expected immediate rewards $r(s,a,s')$, which is the expected reward from doing action a in state s and transitioning to state s'.
  (Poole and Mackworth 2010; Sutton and Barto 1998)

Thus, MDP provide a mathematical framework for decision problems where the consequences of decisions are partly random and partly in control of the decision maker. The solution to an MDP is a policy π, which defines which action a_t should be taken in a particular state s_t. Given a reward criterion, rewards are combined to a cumulated return R. A policy has an expected value $V^\pi(s)$ for each state, which is the expected return of following π starting in state s. V^π is called the state-value function for policy π. Another important function is the action-value function Q^π. $Q^\pi(s,a)$ is the expected return of applying action a in state s and then following policy π. There always exists at least one optimal policy π^*, which is better than or equal to all other policies. All optimal policies share the same optimal state-value function, which is defined as $V^*(s) = \max_\pi V^\pi\,(s)$ for all states $s\ \in S$ and the optimal action-value function, which is defined as $Q^*(s,a) = \max_\pi Q^\pi\,(s,a)$ for all states $s\ \in S$ and all actions $a\ \in A(s)$.

If the transition probability model of an MDP is known, Dynamic Programming methods can be used to solve the MDP (Gosavi 2015). If the transition probabilities are too difficult to find because of the complex stochastics of the underlying system, which is the case in the system described in this paper, RL can be an approach to solve the MDP. One of the most often used algorithms in RL is the Q-learning algorithm. In its simplest form, one-step Q-learning, is defined by

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

where α is the step-size parameter ($0\ < \alpha \leq 1$) and γ is the discount-rate parameter ($0\ < \gamma \leq 1$). The choice of the step-size parameter α is basically controlling how fast the algorithm is learning and the discount-parameter γ is controlling the extend to which an estimation of the future value is added to the return. The agent maintains a table of $\mathrm{Q[S,A]}$, where the rows are the possible states S and the columns are the possible actions A. $Q[s,a]$ represents the agent's current estimate of $Q^*(s,a)$.

In the context of the presented DSS, the algorithm works as follows:

```
Initialize Q[S,A] with 1.0 for all s and a
Repeat
   Observe state s
   Execute an action a
   Perform simulation experiment
   Receive reward r and state s'
   Update Q[s,a]
Until termination
```

The system states description for the example above are assembled from the system configuration and the manifestation of the system variables needed to learn a useful policy. An illustration is given in

Table 1, which resembles an illustrative state of the table $Q[S, A]$. Each system state can be described using a vector $\vec{s}$. The vector consists of different sections, described by $\vec{c}$, $\vec{\imath}$ and $\vec{d}$. $\vec{c}$ stands for the system configuration. For each site it is stored 1 if a site is stock keeping and 0 if not. $\vec{\imath}$ stands for the inventory for the considered article at each site. $\vec{d}$ stands for the demand forecast for the article at the respective site for the period under examination in the DSS run.

Table 1: Illustration of the $Q[S, A]$ table for the exemplary decision problem.

STATES			**ACTIONS**				
$\vec{c}$	$\vec{\imath}$	$\vec{d}$	**A1**	**A2**	**A3**	**A4**	**…**
[1,1,1,	80,42,10,	12,3,8]	1.0	0.2	0.5	0.8	…
[1,0,1,	80,0,52,	12,3,8]	0.8	0.9	1.0	0.2	…
…	…	…	…	…	…	…	…

The actions are abbreviated with A1, A2, et cetera. They are essentially functions in the Execution Engine, which can be called by the RL agent. The functions perform SQL queries which change the data in the simulation data model. The signatures of the functions used in the example described above are:

```
instructWarehouseToStockkeepArticle(WarehouseID)
instructWarehouseToOrderArticleOnDemand(WarehouseID)
```

The reward for an action is measured through the SDWH, as shown in Figures 2. For the minimal example presented here, it is a weighted sum of the decline of logistics costs and increase in delivery performance. The transportation costs are currently not considered in this problem instance. The raw data for the KPIs are written to the statistics tables by SimChain and are subsequently combined in the SDWH.

5 CONCLUSION AND OUTLOOK

A first prototype of the described system is currently developed using an RL library. First experiments have been conducted to obtain an estimate on how well the DSS might perform with larger logistics problems. The first experiments have shown that RL techniques seem to be a suitable approach to construct the internal principles of the proposed DSS. It can be used for solving small problem instances, as the one described in this paper. Furthermore, there already exists some literature on the successful use of RL techniques to solve control problems in the logistics context.

In its conception, RL offers everything that is needed to realize the proposed DSS. Still, it will be interesting to research how the DSS behaves for larger problem instances. Since it is not feasible to store the whole action-value table for problems with large state spaces, it will be mandatory to use a value-function approximation, such as an artificial neural network, which learns the action-value function from samples.

The key question in the subsequent research will be how to use the experience of an agent with a limited subset of the state space to successfully synthesize a useful policy for the rest of the state space. This form of generalization will especially be important in the application presented in this paper, since it cannot be guaranteed that the system will visit all relevant states, for which a decision maker might need a recommendation, in advance.

Finally, the system should be able to learn from experience with the simulation model in order to provide useful action recommendations for states in the real logistics system that are similar to the ones it has interacted with in the simulation model.

REFERENCES

Bauer, A., and H. Günzel. 2013. *Data-Warehouse-Systeme*: *Architektur, Entwicklung, Anwendung.* 4th ed. Heidelberg, Germany: dpunkt.

Bertsekas, D., and J. Tsitsiklis. 1996. *Neuro-Dynamic Programming.* Nashua, NH, USA: Athena.

Blutner, D., S. Cramer, S. Krause, T. Mönks, L. Nagel, A. Reinholz, and M. Witthaut. 2007. "Assistenzsysteme für die Entscheidungsunterstützung" Technical Report No. 06009, Dortmund.

Bockholt, F., W. Raabe, and M. Toth. 2011. "Logistic Assistance Systems for Collaborative Supply Chain Planning". *International Journal of Simulation and Process Modelling* 6/4: 297–307.

Chaharsooghi, S. K., J. Heydari, and S. H. Zegordi. 2008. "A Reinforcement Learning Model for Supply Chain Ordering Management: An Application to the Beer Game". *Decision Support Systems* 45/4.

Coronel, C., and S. Morris. 2014. *Database Systems*: *Design, Implementation, and Management.* 11th ed. Boston, MA, USA: Cengage Learning.

Deiseroth, J., D. Weibels, and M. Toth. 2008. "Simulation-based Decision Support System for the Disposition of Global Supply Chains" In *Advances in Simulation for Production and Logistics Applications*. *Proceedings of the 13th ASIM Conference on Simulation in Production and Logistics,* edited by M. Rabe, 41–50. Stuttgart, Germany: Fraunhofer IRB.

Dross, F., and M. Rabe. 2014. "A SimHeuristic Framework as a Decision Support System for Large Logistics Networks with complex KPIs" In *Proceedings of the 22nd Symposium Simulationstechnik (ASIM 2014),* edited by J. Wittmann and C. Deatcu. Vienna, Austria: ARGESIM / ASIM.

Ehmke, J. F., D. Großhans, D. C. Mattfeld and L. D. Smith. 2011. "Interactive Analysis of Discrete-event Logistics Systems with Support of a Data Warehouse", *Computers in Industry* 62/6: 578–586.

e-SAVE. 2015. e-SAVE. Accessed on March 17. http://www.e-save.eu.

Giannoccaro, I., and P. Pontrandolfo. 2002. "Inventory Management in Supply Chains: a Reinforcement Learning Approach" *International Journal of Production Economics* 78/2: 153–161.

Gosavi, A. 2015. *Simulation-Based Optimization*: *Parametric Optimization Techniques and Reinforcement Learning.* 2nd ed. Berlin and Heidelberg, Germany: Springer.

Gutenschwager, K., and K. Alicke. 2004. "Supply Chain Simulation mit ICON-SimChain" In *Logistik Management,* edited by T. Spengler, S. Voß, and H. Kopfer, 161–78. Heidelberg, Germany: Physica.

Heilala, J., J. Montonen, P. Järvinen, S. Kivikunnas, M. Maantila, J. Sillanpää, and T. Jokinen. 2010. "Developing Simulation-based Decision Support Systems for Customer-driven Manufacturing Operation Planning" In *Proceedings of the 2010 Winter Simulation Conference (WSC),* edited by B. Johansson, S. Jain, J. Montoya-Torres, and E. Yücesan. IEEE Press.

Jarke, M., M. Lenzerini, Y. Vassiliou, and P. Vassiliadis. 2003. *Fundamentals of Data Warehousing.* 2nd ed. Berlin and Heidelberg, Germany: Springer.

Juan, A. A., and M. Rabe. 2013. "Combining Simulation with Heuristics to Solve Stochastic Routing and Scheduling Problems" In *Simulation in Produktion und Logistik 2013*. *Proceedings of the 15th ASIM Conference on Simulation in Production and Logistics,* edited by W. Dangelmaier, C. Laroque, and A. Klaas, 641–649. Paderborn, Germany: Heinz-Nixdorf-Institut.

Kuhn, A., B. Hellingrath, and H. Hinrichs. 2008. "Logistische Assistenzsysteme" In *Software in der Logistik*. *Weltweit sichere Supply Chains*, 20–26. München, Germany: Huss.

Liebler, K., U. Beissert, M. Motta, and A. Wagenitz. 2013. "Introduction OTD-Net and Las: Order-to-delivery Network Simulation and Decision Support Systems in Complex Production and Logistics Networks" In *Proceedings of the 2013 Winter Simulation Conference (WSC),* edited by Pasupathy, R., S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl, 439–451. IEEE Press.

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

McGinnis, L. F. 2005. "Technical and Conceptual Challenges in Organizational Simulation" In *Organizational simulation,* edited by W. B. Rouse and K. R. Boff, 273–98. Hoboken, NJ, USA: Wiley.

Pontrandolfo, P., A. Gosavi, O. G. Okogbaa, and T. K. Das. 2002. "Global Supply Chain Management: A Reinforcement Learning Approach" *International Journal of Production Research* 40: 1266–1317.

Poole, D., and A. Mackworth. 2010. *Artificial Intelligence*: *Foundations of Computational Agents.* Cambridge, U.K: Cambridge University Press.

Qiu, M., H. Ding, J. Dong, C. Ren, and W. Wang. 2007. "Impact of Business Service Modes on Distribution Systems: A Reinforcement Learning Approach" In *IEEE International Conference on Services Computing (SCC 2007),* 294–9. IEEE Press.

Rabe, M., K. Gutenschwager, T. Fechteler, and U. M. Sari. 2013. "A Data Model for Carbon Footprint Simulation in Consumer Goods Supply Chains" In *Proceedings of the 2013 Winter Simulation Conference (WSC),* edited by R. Pasupathy, S.-H. Kim, A. Tolk, R. Hill, and M. E. Kuhl. IEEE Press.

Rabe, M., S. Spieckermann, A. Horvath, and T. Fechteler. 2012. "An Approach for Increasing Flexibility in Green Supply Chains Driven by Simulation" In *Proceedings of the 2012 Winter Simulation Conference (WSC),* edited by C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A. M. Uhrmacher, 3144–55. IEEE Press.

SimPlan AG. 2015. SimChain. Accessed on March 17. http://www.simchain.net.

Stockheim, T., M. Schwind, and W. Koenig. 2003. "A Reinforcement Learning Approach for Supply Chain Management" In *EUMAS: 1st European Workshop on Multi-Agent Systems*, edited by M. Inverno, C. Sierra, and F. Zambonelli.

Subramaniam, G., and A. Gosavi. 2004. "Simulation-Based Optimization for Material Dispatching in a Retailer Network" In *Proceedings of the 2004 Winter Simulation Conference,* 351–6.

Sutton, R. S., and A. G. Barto. 1998. *Reinforcement learning*: *An introduction.* Cambridge, MA, USA: MIT Press.

van Tongeren, T., U. Kaymak, D. Naso, and E. van Asperen. 2007. "Q-learning in a Competitive Supply Chain" In *2007 IEEE International Conference on Systems, Man and Cybernetics*, 1211–6.

Watkins, C. 1989. "Learning from Delayed Rewards." Ph.D. thesis, Kings College, Cambridge, England.

Zhang, Y., and S. Bhattacharyya. 2007. "Effectiveness of Q-learning as a Tool for Calibrating Agent-based Supply Network Models" *Enterprise Information Systems* 1/2: 217–233.

Zheng Sui, A. Gosavi, and Li Lin. 2010. "A Reinforcement Learning Approach for Inventory Replenishment in Vendor-Managed Inventory Systems With Consignment Inventory" *Engineering Management Journal* 22/4: 44–53.

AUTHOR BIOGRAPHIES

MARKUS RABE is full professor for IT in Production and Logistics at the Technical University Dortmund. Until 2010 he had been with Fraunhofer IPK in Berlin as head of the corporate logistics and processes department, head of the central IT department and member of the institute direction circle. His research focus is on information systems for supply chains, production planning and simulation. Markus Rabe is vice chair of the “Simulation in Production and Logistics” group of the simulation society ASIM, member of the editorial board of the Journal of Simulation, member of several conference program committees, has chaired the ASIM SPL conference in 1998, 2000, 2004, and 2008, and was local chair of the WSC’2012 in Berlin. More than 160 publications and editions report from his work. His e-mail address is markus.rabe@tu-dortmund.de.

FELIX DROSS is a doctoral candidate at the Graduate School of Logistics at the Technical University Dortmund. He holds a B.Sc. in Applied Computer Science from the Ruhr-University Bochum and a M.Sc. in Software Engineering from the University of Augsburg, Technical University Munich and

Ludwig-Maximilians-University Munich. He graduated with a master thesis on a hybrid-heuristic approach to solve the Inventory Routing Problem. Since 2013 he focuses his research on simulation-based optimization techniques for logistics networks. His e-mail address is felix.dross@tu-dortmund.de.