



Decoupling the Modeling of Actions in Logistics Networks from the Underlying Simulation Data Model

Markus Rabe¹, Dominik Schmitt²(✉), Astrid Klüter¹,
and Joachim Hunker¹

¹ TU Dortmund, ITPL, Dortmund, Germany
{markus.rabe, astrid.klueter,
joachim.hunker}@tu-dortmund.de

² TU Dortmund, GSofLog, Dortmund, Germany
dominik.schmitt@tu-dortmund.de

Abstract. Actions in logistics networks are often determined and evaluated by logistics assistance systems. Typically, these actions are predefined by the logistics assistance system. In case of a data-driven logistics assistance system, these actions are depending on the underlying data structure. This paper describes an approach of decoupling an actions implementation from the underlying data model and, therefore, independent of a given data structure.

Keywords: Logistics assistance system · Decision Support System

1 Introduction

Maintaining a logistics network (LNW) in good conditions, under continuously changing internal and external demands, is a challenging task for a decision maker (DM). In order to adapt to the LNW's changing state, the DM needs to identify promising actions. An action may describe any modification or change to the logistics network, e.g., increasing the stock level of a stock keeping unit (SKU) or changing the transport structure of the logistics network (Werner 2013). These actions are considered to increase the network's performance. However, actions may have contrary effects on the network's performance, e.g., increasing the stock may improve the service level but it may also increase the stock costs. Thus, identifying the most promising actions to improve the overall performance of the LNW, e.g., defined by total costs or service level, is a complex task.

In order to address this issue, the authors have developed a logistics assistance system (LAS) that identifies the most promising actions and that suggests these actions to the decision makers (Rabe et al. 2017). The LAS is using a simheuristic approach (Juan and Rabe 2013), combining a data-driven discrete-event simulation with a metaheuristic. A simulation model, representing the LNW's state, is build, based on data from a database. The simulation is used for evaluating the LNW's performance. The metaheuristic identifies promising actions that are executed on the underlying database in order to alter the LNW's state.

A basic set of actions is predefined by the LAS. However, enterprises have enterprise-specific requirements for each action and, therefore, enterprises may need an individual set of actions. Thus, the authors developed an approach for integrating user-modeled actions into the LAS (Rabe et al. 2017). However, actions are closely related to the underlying database resulting in multiple issues, e.g., required knowledge of the database's structure when modeling actions or the necessity of adjusting actions in case of the database's structure is changing. To deal with these problems, the authors are investigating an approach for decoupling actions from the underlying simulation data model (SDM).

This paper is organized as follows: First, an overview about related work is given, including logistics networks in general, relevant data of logistics networks, data modeling, logistics assistance systems in general and a description of the author's LAS. After that, the approach for decoupling actions from the underlying data model is described, followed by an overview of the corresponding benefits that come along with this approach. Afterwards, an example of modeling and applying an action using the presented approach is given. The paper closes with a conclusion and an outlook.

2 Related Work

2.1 Logistics Networks

When looking at common definitions of logistics networks, it is noticeable that in the literature there is no clear distinction between logistics networks and supply chains and, therefore, between supply chain management (SCM) and logistics management. (Cordeau et al. 2006; Wang et al. 2016).

Larson and Halldorsson (2004) describe four perspectives on the relationship between logistics and SCM. One phenomenon is the one just described; the relabeling perspective that simply renames logistics; what was logistics is now SCM. Thus, the terms supply chain and logistics network might be used synonymously.

According to Rushton et al. (2010), there is a logistics network between the supply chain's supplier and the customers. Ma and Suo (2006) define a logistics network as the distribution part of a supply chain. These authors describe the distribution network as a critical part of managing a supply chain.

Our application case concerns a special type of logistics network, namely wholesale networks. A wholesaler is a legally and economically independent business which procures goods and sells them mainly to commercial users, commercial resellers, or bulk consumers unchanged or after customary manipulation (Seýffert 1972). By choosing the trade-level, wholesalers determine their position, within the trade chain, between original production and end customer (Barth et al. 2015).

In the following, we refer to wholesale logistics networks, as it is the internal company's terminology for the given use case. Regarding the exact technical terminology, since we handle data, it ultimately does not matter whether we formally collect data from logistics networks or supply chains, as they obviously do not differ significantly from each other. In the following, we will talk about data in logistics networks in general, knowing that this can also mean data from supply chains.

2.2 Data in Logistics Networks

In logistic networks, data are processed in various IT systems within a company as well as across companies. According to Folinas et al. (2006) a distinction is made between structured content (e.g., relational databases), unstructured content (e.g., plain text), or unstructured content with structured representation (e.g., HTML files) in many different forms (e.g., texts, diagrams, illustrations or tables).

Another distinguishing feature is the time reference. Data in logistics networks can be divided into master data and transaction data (Reuter and Rohde 2015). Master data are data that change only slightly over time and that are used to identify, classify and characterize entities of the logistics network. A classic example is address data as it is stored in logistics networks for suppliers and customers. Transaction data are data that undergo major changes over time. Information, required for planning the logistics network, is obtained from the transaction data that are created and changed by operational performance processes. Examples for transaction data are current inventories, current orders, availability of resources, or planned production quantities and stock levels (Reuter and Rohde 2015).

2.3 Database Systems

Since data occur in different kinds of formats, from different sources and in different relations, especially in the context of a logistics network, they need to be handled adequately. Therefore, data are typically stored in information systems, e.g., a database system. A database system consists of a database and a Database Management System (DBMS) (Elmasri and Navathe 2011).

A database is defined as a coherent, not random, collection of related data which represent an excerpt of the real world (Elmasri and Navathe 2011). Databases follow a database schema, describing the logical structure of a database. To describe such a schema, data models are used. A data model is a formal representation and description of the structure and the relations between data (Connolly and Begg 2015).

In this paper, the authors make use of an enterprise-specific data model that describes any data of the corresponding enterprise in a persistent way. The reader is kindly referred to West (2011), who identifies and distinguishes between different data models discussed in the literature and who highlights that in practice the enterprise-specific data model covers just most of the enterprise's data.

The process of creating and changing a data model is called data modeling. Different techniques for formally describing a data model can be distinguished in the literature, e.g., the entity-relationship-model or the unified modeling language.

For creating and maintaining a database and for manipulating its data, DBMSs are used. A DBMS is a software program that allows users to interact with the associated database. Connolly and Begg (2015) characterize the tasks of a DBMS by

1. handling access restrictions and providing multi-user access to the database,
2. offering a language to specify the database, called Data Definition Language and
3. a Data Manipulation Language to allow specific operations on the data like insert, delete, update, or retrieve.

One of the most-used languages regarding manipulation and querying of data, in the context of relational database systems, is the Structured Query Language (SQL) (Chamberlin 2017). SQL works on the data stored in the database through so-called statements. The data is, in case of SQL and due to the underlying relational data model, stored in the form of multiple tables (Chamberlin 2017).

2.4 Logistics Assistance Systems

To support the decision-making process of decision makers in logistics networks, so-called logistics assistance systems are used (Blutner et al. 2009). LASs are software systems that help the user to monitor a logistics network. Therefore, a LAS might be used to evaluate a specific logistics network's configuration or scenario, e.g. by calculating corresponding Key Performance Indicators. However, a LAS may also combine an optimization approach with an evaluation mechanism, e.g. heuristics and simulation (Juan and Rabe 2013), in order to determine and evaluate multiple logistics network's state configurations. The best solutions can be provided to the user. Each logistics network's configuration can be derived by applying one or more predefined actions to the current logistics network's state.

In the literature, the term Decision Support Systems (DSS) is often used to describe such a system (Kengpol 2008). In the context of logistics, DSS and LAS are used as synonyms. Publications, considering LASs or DSSs in the context of logistics, have been presented for example in the area of automobile production (Bockholt et al. 2011) or in combination with other methods like simulation (Heilala et al. 2010). Throughout this paper, the authors decided to use the term LAS in order to highlight the system's application domain.

2.5 Logistics Assistance System for Wholesale Logistics Networks

The authors have developed a logistics assistance system for managing wholesale logistics networks. This section provides a brief overview of the system's architecture (Fig. 1) and working principles. For a more detailed description of the LAS, the reader is kindly referred to previous publications of the authors (Rabe et al. 2017; Rabe et al. 2018a; Rabe et al. 2018b).

LASs are utilizing master data and transaction data from the real network. Companies use some kind of software in order to store this information, such as SAP R/3. This data is automatically extracted, transformed and loaded into the simulation's database. The LAS uses a data-driven discrete-event simulation combined with a heuristic, also called simheuristic (Juan and Rabe 2013). Therefore, the data from the database is used to instantiate the simulation model that represents the logistics network's state, e.g. the data base table "SiteHaveSKUs" combines the tables "SKU" and "Site" storing the stock for each combination of site and SKU. By running the simulation, the network's performance is evaluated. After each simulation run, the simulation results are fed into the heuristic unit for further investigation.

The logistics network's performance can be improved by altering the network's state, e.g. changing the stock or increasing the transport frequency. For each of these manipulations, a corresponding action needs to be applied to the logistics network.

The heuristic unit (HU) examines the search space, constructed by a collection of all possible actions in the LNW, searching for the most promising actions. Therefore, multiple algorithms are implemented, e.g. Evolutionary Algorithm or Reinforcement Learning. For a detailed description of the HU’s working principles, the reader is kindly referred to Rabe et al (2018a) and Rabe et al (2018c) regarding the Evolutionary Algorithm and to Rabe and Dross (2015) for the implementation of the Reinforcement Learning algorithm. Actions, selected by the heuristic unit are forwarded to the execution engine for further processing.

The execution engine transforms these actions into changes to the underlying database, in form of SQL-statements. Before applying these statements to the underlying database, additional predefined metadata will be evaluated. For instance, attributes’ default values are used to initialize of new entities. Additionally, attributes’ ranges and specific conditions or constraints are used to identify valid changes to the LNW, e.g., capacity constraints when shifting stocks. The actions’ effects are evaluated by instantiating a new simulation model, based on the changed data, followed by running the simulation.

This process is run iteratively until a predefined termination criterion is reached, e.g., a specific amount of iterations, a minimum quality of the logistics network’s performance or a continuous stagnation in the optimization process. The most promising actions are provided to the decision maker for further investigation.

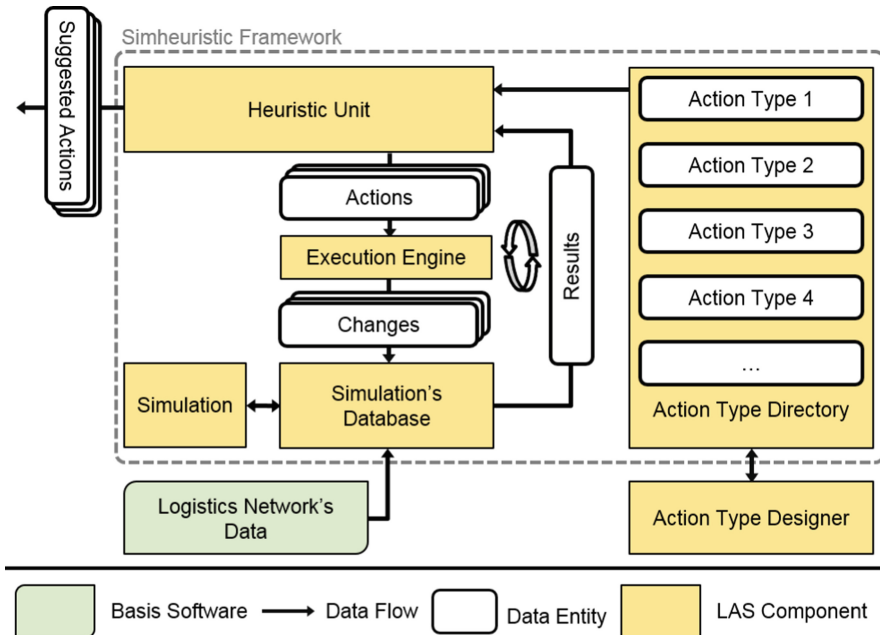


Fig. 1. Simplified architecture of the logistics assistance system, based on Rabe et al. (2018c).

In order to increase the LAS’s flexibility, the authors developed a method that allows the user to model action types and integrate them into the system (Rabe et al. 2017).

An action type is a generalization of multiple similar actions, e.g., an action type may describe the general instructions for changing the stock level of any SKUs in any site by any value, whereas a corresponding action describes the increase of stock level for a specific SKU in a specific site by a specific value. Thus, actions can be derived from an action type by specifying its parameter values.

For modeling user-created action types, the user has access to the action type designer, a graphical or textual interface. The action type designer has access to a domain-specific language that is specifically designed to ease the formal definition of action types in logistics networks (Rabe et al. 2017). Thus, the user is able to use predefined language constructs for this modeling process. From the action type designer the user can access all existing action types. Action types are structured as a module. Thus, the user can integrate any existing action type into the modeling process of new action types.

Action types are stored in the action type directory. The heuristic unit is connected with the action type designer to access all action types and, therefore, all corresponding actions in order to create the search space. In contrast to typical approaches for LASs the author’s approach for a logistics assistance system offers the possibility to extend the set of predefined actions by user-generated actions.

3 Decoupling Actions from the Underlying Data Model

An action defines specific changes to the logistics network and, therefore, to the underlying SDM (Rabe et al. 2017). Thus, when applying an action, the corresponding entities of the LNW need to be adapted accordingly. Considering a database as the underlying data source of the data-driven simulation, an entity class is defined as a table. A corresponding entry in the table is called entity. For identifying affected entities for a given action, the specific table and attribute names of the underlying simulation’s database are specified within the action’s definition (Fig. 2). For instance, an action that changes the stock level of an SKU in a site needs to address the specific area of the underlying database, where the corresponding changes are applied. Thus, the attribute “Initial_Stock” in the corresponding table “SiteHaveSKUs” in addition to the change of stock level, e.g., an increase of 20, must be defined.

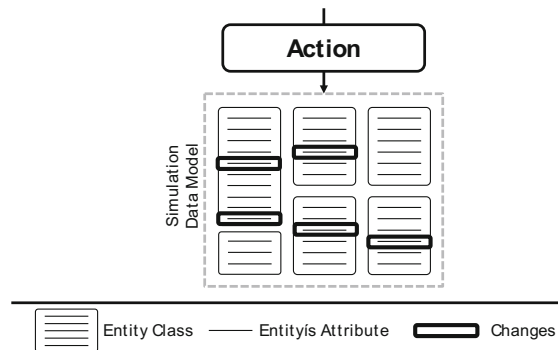


Fig. 2. Applying an action directly to the simulation’s database.

However, the SDM is typically predefined by the simulation tool and, thus, not easily adjustable. This aspect may result in multiple issues:

1. The modeling of actions is done in direct relation to the SDM's structure. Thus, users need deep knowledge about the underlying data structure, e.g., the entities' names and attributes, the relations between the entities, and the data that is stored in each entity class. However, users might not be familiar with the given data structure, because it may differ from the company's data structure. Therefore, training the users might become very time-consuming and costly.
2. Updating the simulation tool may change the simulation data model and, thus, actions may need to be adapted accordingly resulting in additional costs and potential downtime of the LAS. In addition, users may need to be trained to the changed data structure in order to model new actions.
3. Predefined metadata is specified against the EDM, e.g., entity classes or its attributes. Thus, changes to the EDM may require adaptations of the meta information leading to additional costs and potential downtime of the LAS.
4. Simulation tools typically have their own individual data structure. Thus, changing the current simulation tool used by the LAS may increase the problems mentioned in 1, 2, and 3.

To address these issues, the authors present an approach for decoupling actions from the simulation data model by adding an additional layer in between. The main part of the additional layer is a so-called enterprise-specific data model. The EDM represents the specific structure of the enterprise's LNW and, therefore, the EDM is typically determined by the company itself.

Each attribute of the SDM is distinctly linked to the corresponding attribute of the EDM. Each connection is defined in the so-called mapping model. The mapping model consists of a number of Json files, one for each entity of the EDM. Such a Json file contains the entity's name of the EDM and a number of tuples for the connections between each of the entity's attributes and the linked attributes of the underlying SDM. An example of such a Json file is pictured in Fig. 3.

Thus, actions will no longer be modelled against the SDM, but exclusively against the EDM instead. In addition, the predefined metadata will be specified against the EDM as well. Therefore, the EDM constitutes an abstract interface to the underlying SDM. When applying an action, the corresponding changes to the EDM are mapped to the SDM utilizing the mapping model and, therefore, the consisting Json files (Fig. 4).

The process of applying an action and transforming the action into changes to the SDM is pictured in Fig. 5. First, a compiler transforms an action into changes to the EDM, e.g., when increasing the stock level of an SKU in a site, the attribute "StockLevel" in the entity class "SitesStoreSku" of the EDM needs to be adjusted for all affected entities accordingly. A mapper transforms these changes to the SDM's structure utilizing the mapping model, e.g., the attribute "StockLevel" in "SitesStoreSku" is linked to the attribute "Initial_Stock" in "SiteHaveSKUs" of the underlying SDM. The utilization and mapping of any metadata is done in the same way. The mapped changes are applied to the SDM and, therefore, altering the LNW's state.

```

{
  "Entity EDM": "SitesStoreSku",
  "Attributes": [
    {
      "Attribute EDM": "SiteId",
      "Entity SDM": "SiteHaveSKUs",
      "Attribute SDM": "Site"
    },
    {
      "Attribute EDM": "SkuId",
      "Entity SDM": "SiteHaveSKUs",
      "Attribute SDM": "SKU"
    },
    {
      "Attribute EDM": "StockLevel",
      "Entity SDM": "SiteHaveSKUs",
      "Attribute SDM": "Initial_Stock"
    }
  ]
}

```

Fig. 3. Example of a Json file for mapping the attributes of entity “SitesStoreSku” from the EDM to the corresponding attributes of the SDM.

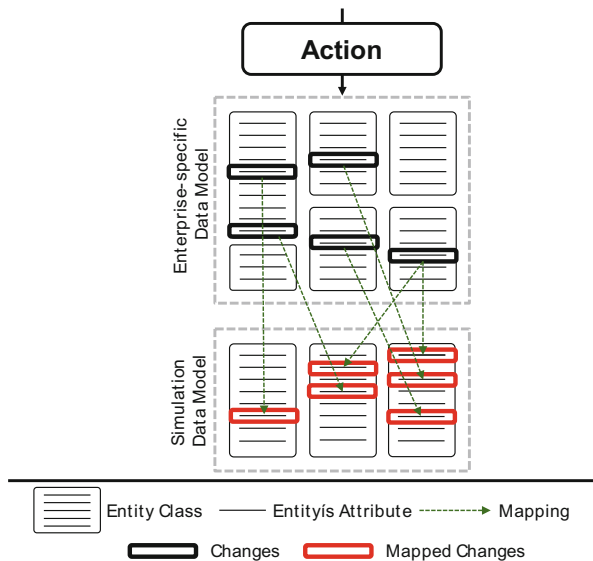


Fig. 4. Applying an action to the enterprise-specific data model and mapping the resulting changes to the simulation data model.

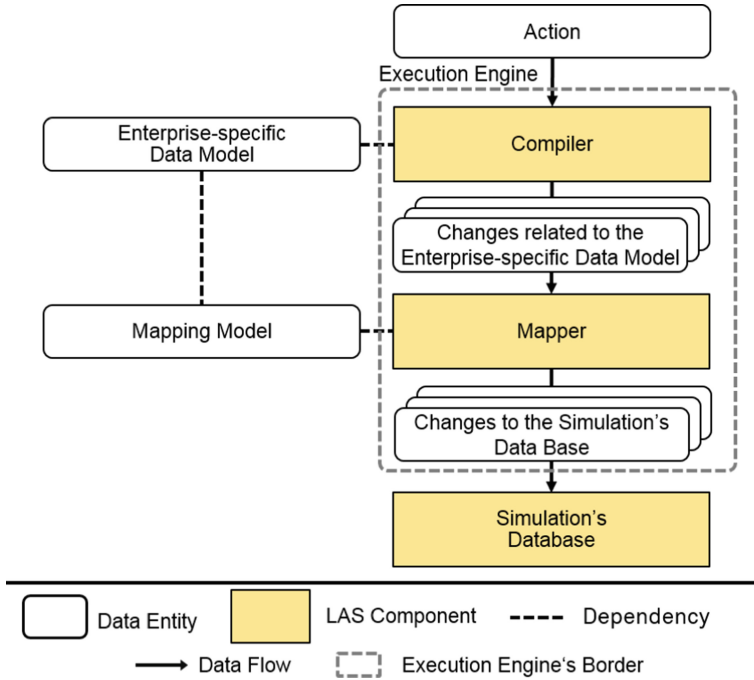


Fig. 5. Process of applying an action using an integrated enterprise-specific data model based on Rabe et al. (2017).

4 Implementation and Results

4.1 Nomenclature and Structure of the Enterprise-Specific Data Model

Typically, an EDM’s nomenclature and structure is individually defined by the enterprise itself. However, once specified, an EDM’s definition stays persistent. Thus, each of the SDM’s attributes need to be linked distinctly to a corresponding attribute of the EDM only once.

For example, an attribute “Initial_Stock” in “SitesHaveSKUs” of the SDM might be linked to an attribute “StockLevel” in “SitesStoreSku” of the EDM. However, the EDM’s attribute could also be named “Stock”. The only difference between both cases is that the corresponding Json file for the entity class “SitesStoreSku” defines either the attribute “StockLevel” or “Stock” of the EDM as being linked to the attribute “Initial_Stock” in “SitesHaveSKUs” of the SDM.

Information of an EDM’s entity class can also be split among several entity classes of the SDM. This information is specified by the attribute’s value of “Entity SDM” in the corresponding Json file. For instance, an EDM’s entity class may have two attributes “attribute1” and “attribute2”, whereas “attribute1” is linked to “attributeA” of an entity class “e1” in the SDM and “attribute2” is linked to “attributeB” of an entity class

“e2” accordingly. An extract of the corresponding Json file is displayed in Fig. 6. Consequently, the structure of the EDM is fully independent of the underlying SDM and, therefore, the modeling of action types as well.

```
{
  ...
  "Attributes": [
    {
      "Attribute EDM": "attributel",
      "Entity SDM": "e1",
      "Attribute SDM": "attributeA"
    },
    {
      "Attribute EDM": "attributel",
      "Entity SDM": "e2",
      "Attribute SDM": "attributeB"
    },
    ...
  ]
}
```

Fig. 6. Example of a Json file for the mapping between the EDM and the SDM.

4.2 Extensibility and Interchangeability of the Underlying Simulation Data Model

The presented approach opens up the possibility of extending the underlying SDM, e.g., by adding new entity classes to the SDM or by extending existing entity classes with new attributes. This supports and preserves the flexibility of the LAS and can be done intentionally and as needed. The use of an EDM allows for simply mapping an entity’s attribute of the EDM to the newly added ones in the SDM, as described before. The EDM remains as it is since it covers the whole representation of the enterprise’s LNW. For example, the logistics network might be divided into regions and, thus, the information could be stored in a corresponding attribute, e.g., “region” in the master data of the LNW for each of the network’s sites. However, the underlying SDM might not be able to represent this information in any way. Adding a new attribute to the SDM and linking it to the corresponding attribute “region” of the EDM opens up the possibility of using this attribute in an action type definition, e.g., for specifying which entities of the LNW are affected by corresponding actions.

Another advantage of this approach is that there is an independency from the underlying SDM and, therefore, from the used simulation tool. In general, simulation tools may have different data model characteristics with respect to their entity classes, attributes, and the relations between those entity classes. If another simulation tool is to be used, the already implemented action types and the EDM are retained in their entirety. Only a new mapping model needs to be created according to the data structure

of the new simulation model. Thus, switching the underlying simulation tool can be addressed by changing the mapping model that connects the EDM’s structure with the structure of the underlying SDM. The newly created mapping model will be used in the execution engine for further processing of any action. Figure 7 illustrates the procedure of switching the mapping model and, therefore, adapting to a new SDM.

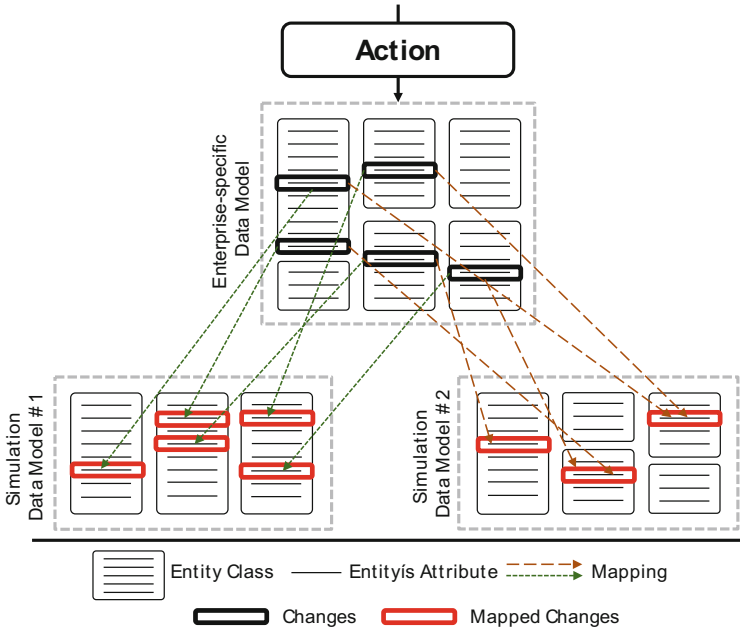


Fig. 7. Applying an action to the enterprise-specific data model and mapping the resulting changes to another simulation data model

5 Conclusion and Outlook

This paper demonstrates a concept for decoupling actions from the underlying SDM by adding an EDM as an additional layer in between the modeling of action types and the application of corresponding actions. The utilization of an EDM results in multiple advantages, e.g., using the EDM’s nomenclature and structure for the formal definition of action types in order to improve the LAS’s usability. Additionally, the possibility of composing information from multiple entities of the SDM into one EDM’s entity and vice versa is shown. Furthermore, an approach for integrating new entities or attributes to the EDM in order to improve the system’s quality is presented.

Currently, the authors are working on an object-oriented reference model (RM) for wholesale logistics networks. This RM could be used to derive specific EDMs for companies in that domain. Such EDMs would be linked to the corresponding RM. Following the object-oriented approach could lead to an additional increase of the LAS’s usability. Furthermore, the authors are investigating the approach of modeling

action types against the RM. These action types could be used in any EDM derived from the corresponding RM, reducing the enterprise's costs of modeling their own action types.

References

- Barth, K., Hartmann, M., Schröder, H.: Betriebswirtschaftslehre des Handels. Springer Gabler, Wiesbaden (2015). <https://doi.org/10.1007/978-3-8349-7184-5>
- Blutner, D., Cramer, S., Krause, S., Mönks, T., Nagel, L., Reinholz, A., Witthaut, M.: Assistenzsysteme für die Entscheidungsunterstützung. In: Buchholz, P., Clausen, U. (eds.) Große Netze der Logistik, pp. 241–270. Springer, Berlin, Heidelberg (2009)
- Bockholt, F., Raabe, W., Toth, M.: Logistic assistance systems for collaborative supply chain planning. *IJSPM* **6**(4), 297–307 (2011). <https://doi.org/10.1504/IJSPM.2011.048010>
- Chamberlin, D.: SQL. In: Liu, L., Tamer Özsu, M. (eds.) Encyclopedia of Database Systems, vol. 13, pp. 1–9. Springer, New York (2017)
- Connolly, T.M., Begg, C.E.: Database Systems. A Practical Approach to Design, Implementation, and Management. Pearson, Boston (2015). Always Learning
- Cordeau, J.-F., Pasin, F., Solomon, M.M.: An integrated model for logistics network design. *Ann. Oper. Res.* **144**(1), 59–82 (2006). <https://doi.org/10.1007/s10479-006-0001-3>
- Elmasri, R., Navathe, S.: Fundamentals of Database Systems. Addison-Wesley, Boston (2011)
- Folinas, D., Manikas, I., Manos, B.: Traceability data management for food chains. *Br. Food J.* **108**(8), 622–633 (2006). <https://doi.org/10.1108/00070700610682319>
- Heilala, J., Montonen, J., Jarvinen, P., Kivikunnas, S., Maantila, M., Sillanpaa, J., Jokinen, T.: Developing simulation-based decision support systems for customer-driven manufacturing operation planning. In: Dunkin, A., Ingalls, R.G., Yücesan, E., Rossetti, M.D., Hill, R., Johansson, B. (eds.) Proceedings of the 2010 Winter Simulation Conference, pp. 3363–3375. IEEE Press, Piscataway (2010)
- 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, pp. 641–649. Heinz-Nixdorf-Institut, Paderborn (2013)
- Kengpol, A.: Design of a decision support system to evaluate logistics distribution network in Greater Mekong Subregion Countries. *Int. J. Prod. Econ.* **115**(2), 388–399 (2008). <https://doi.org/10.1016/j.ijpe.2007.10.025>
- Larson, P.D., Halldorsson, A.: Logistics versus supply chain management: an international survey. *Int. J. Logistics Res. Appl.* **7**(1), 17–31 (2004). <https://doi.org/10.1080/1367556031001619240>
- Ma, H., Suo, C.: A model for designing multiple products logistics networks. *Int. J. Phys. Distrib. Mater. Manag.* **36**(2), 127–135 (2006). <https://doi.org/10.1108/09600030610656440>
- 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, pp. 2020–2032. IEEE, Piscataway (2015)
- Rabe, M., Schmitt, D., Dross, F.: Method to model actions for discrete-event simulations of logistics networks. In: Chan, W.K.V., D'Ambrogio, A., Zacharewicz, G., Mustafee, N., Wainer, G., Page, E. (eds.) Proceedings of the 2017 Winter Simulation Conference, pp. 3370–3381. IEEE Press, Piscataway (2017)

- Rabe, M., Ammouriouva, M., Schmitt, D.: Utilizing domain-specific information for the optimization of logistics networks. In: Rabe, M., Juan, A.A., Mustafee, N., Skoogh, A., Jain, S., Johansson, B. (eds.) Proceedings of the 2018 Winter Simulation Conference, Gothenburg, Sweden (2018a, to be published)
- Rabe, M., Schmitt, D., Ammouriouva, M.: Improving the performance of a logistics assistance system for materials trading networks by grouping similar actions. In: Rabe, M., Juan, A.A., Mustafee, N., Skoogh, A., Jain, S., Johansson, B. (eds.) Proceedings of the 2018 Winter Simulation Conference, Gothenburg, Sweden (2018b, to be published)
- Rabe, M., Schmitt, D., Ammouriouva, M.: Utilizing domain-specific information in decision support for logistics networks. In: Freitag, M., Kotzab, H., Pannek, J. (eds.) Proceedings of the International Conference on Dynamics in Logistics (LDIC), pp. 413–417. Springer, Cham (2018c)
- Reuter, B., Rohde, J.: Coordination and integration. In: Stadtler, H., Kilger, C., Meyr, H. (eds.) Supply Chain Management and Advanced Planning. Concepts, Models, Software, and Case Studies, 5th edn, pp. 241–256. Springer, Heidelberg, New York, Dordrecht, London (2015)
- Rushton, A., Croucher, P., Baker, P.: The Handbook of Logistics & Distribution Management. Kogan Page, London, Philadelphia (2010). <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10404161>
- Seöffert, R.: Wirtschaftslehre des Handels. Westdeutscher Verlag, Opladen (1972). <https://doi.org/10.1007/978-3-322-83523-9>
- Wang, N., Mi, Y., Gao, H., Liu, W.: Logistics network model based on matter element node. *Procedia Comput. Sci.* **91**, 351–356 (2016). <https://doi.org/10.1016/j.procs.2016.07.093>
- Werner, H.: Supply Chain Management: Grundlagen, Strategien, Instrumente und Controlling, vol. 5. Gabler, Wiesbaden (2013)
- West, M.: Developing High Quality Data Models. Morgan Kaufmann, Burlington (2011). <http://www.ebilib.com/patron/FullRecord.aspx?p=630008>