Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 81 (2019) 198–203

www.elsevier.com/locate/procedia

52nd CIRP Conference on Manufacturing Systems

Determination of the abstraction level in production network models

Martin Benfer*[a], Moritz Ziegler[b], Andreas Gützlaff[a], Bastian Fränken[a], Sven Cremer[a], Jan-Philipp Prote[a], Günther Schuh[a]

[a]Laboratory for Machine Tools and Production Engineering (WZL) at RWTH Aachen University, Campus-Boulevard 30, Aachen 52074, Germany
[b]Chair of Operations Management at University of Stuttgart, Keplerstraße 17, Stuttgart 70174, Germany

* Corresponding author. Tel.: +4917643685605; *E-mail address:* martin.benfer@outlook.com

Abstract

In recent years the importance of production network modeling has increased significantly, as companies face rising complexity and dynamics in their environment. Quantitative models help to gain understanding and support strategic decision making in production network management. Choosing the right abstraction level by determining the trade-off between model accuracy and effort for modeling is crucial for developing viable and applicable models. While experienced model builders can identify the right abstraction level intuitionally, a structured process would lead to results which are more consistent. This paper presents an approach to structure and streamline the determination of the abstraction level. A basic concept is presented, that enables systematic definition of the abstraction level for a given model framework. A process implementing the concept is proposed, that helps to identify the right abstraction level based on case-specific requirements and restrictions and tailor the model to solve the examined problem. The process is tested in a case study at a globally operating tooling machine company.

Keywords: Abstraction level; Model framework; Model granularity; Modeling; Production network model; Supply chain model;

1. Introduction

Enabled by the reduction of transport costs and advances in communication technology, companies have created large interconnected global production networks to realize cost benefits and to conquer new markets [1]. Planning, optimizing and managing production networks is crucial to achieve those goals and survive in a fierce market environment. However, managing these networks is also increasingly difficult, due to the complexity created by sheer size and interconnectivity. To manage this complexity, quantitative models have been developed in cooperation between industry and research institutions [2-6]. These models help to gain a better understanding of the examined systems, conduct experiments and predict their behavior. The generated insights are used to systematically design and optimize existing structures.

Aside from studying which models are suited for which general application, the process of creating and adapting such models for a specific use case is important. Multiple concepts framing this modeling process exist. Those include technical norms like VDI 3633 [7] and practical guidelines, for example [8,9]. Although various modeling process concepts exist, most agree on explicitly distinguishing *conceptual* and *executable models* [9,10-13]. As Fig. 1 shows, a conceptual model is obtained by abstraction of a problem entity, i.e. simplifying a system to achieve a specific goal. Whereas the conceptual model is only a theoretical concept, the executable model is one specific implementation of it.

The abstraction process is the foundation of any implementation and experimentation and thus crucial for the success of the overall model based problem-solving process. The process requires a sound understanding of the problem as well as creativity. The goal of the abstraction process is to create conceptual models with the right balance between the level of detail and simplicity. Whereas the former ensures validity in terms of the studied objectives, the latter decreases

10.1016/j.procir.2019.03.035

complexity and thus enables better understanding [9,14]. One approach to simplify and standardize abstraction in part is to use generalized *model frameworks*. Model frameworks comprise a set of building blocks and rules of behavior based on which specific models are created. While the use of model frameworks limits and simplifies the abstraction process, they can still be used for models with varying levels of detail. Model builders need to understand the requirements of the specific application and the restrictions the framework and the examined system pose to depict the examined systems as desired. Here application comprises both the original system that is examined as well as the specific problem that needs to be solved.

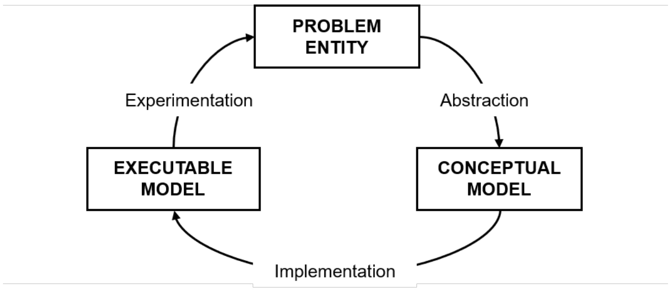

Fig. 1. Modeling process for simulation based on [12].

2. State of the art

The definition of the *abstraction level* or *level of detail* is an integral part of the conceptual modeling process [15]. The two most relevant types of contributions in the literature are:

- Conceptual modeling approaches, that organize the modeling process and determine the abstraction level.
- Specific model frameworks and the way they are implemented in case studies.

2.1. Conceptual modeling approaches

Authors have studied different facets of the model creation process in different contexts. For the purposes of this work, only the modeling process of quantitative symbolic models are examined in detail. For such models, several design guidelines exist.

VDI 3633 describes the modeling process from problem identification to the solution in very general terms [7]. It acknowledges the level of detail of models as critical for their successful application, yet it offers no advice on how to choose the right level of detail systematically. ZEIGLER et al. present an approach to creating a family of models of the same system with varying complexity for different tasks [14]. Multiple simplification methods such as aggregation, omission, linearization, etc. are outlined. The concept of aggregation is explored in detail. However, no specific process is defined to correctly identify the level of detail for a given application. A less technical conceptual modeling method is illustrated by ROBINSON [8]. The method consists of five parts: understanding the problem, determining the objective, identifying the outputs, identifying the inputs, and determining the model content. The model content is based on two elements, scope, and level of detail. Field and model experts determine whether to include components of the examined system in the model and thus determine the scope and level of detail. While this is a very helpful guideline for model building, the process of defining the abstraction level is still relatively vague. BANKS et al. analyze a wide range of mathematical model types and characterize the model conceptualization "[…] probably as much art as science." They only offer some basic guidelines on the modeling process [11].

2.2. Specific model frameworks

Within the space of production network and more broadly supply chain modeling, several contributions study specific model frameworks. They provide interesting insight regarding the challenge of finding the right abstraction level:

FRIESE presents a model framework to plan flexibility and capacity of production networks of the automotive industry [3]. It is designed to be able to solve different types of problems within this field. For every identified type of problem, corresponding aggregation levels for the relevant entity types are proposed. The model-based planning approach shown by BUNDSCHUH can solve multiple problems that typically occur in strategic production network planning [4]. The presented model framework can be adapted and - similar to [3] - different *aggregation levels* are suggested for different problem types by the author. Both contributions demonstrate that the same model framework can be used on different abstraction levels for different problems. ULSTEIN et al. present a model for the optimization of the supply chain in the silicon production industry [6]. In this model, customers are aggregated into groups based on their attributes to simplify the model. The attributes 'geographic location', 'demand', 'prices', and 'transport cost' are used as *aggregation criteria*. With this technique, the model is simplified while retaining a high degree of accuracy. The same idea is shown by VILA et al. who aggregate products in a similar fashion [5].

2.3. Deficits of existing approaches

While several approaches to conceptual modeling exist, they do not specifically explore the idea of utilizing model frameworks. Therefore, those methods are not completely applicable to the abstraction process based on such model frameworks. The contributions proposing model frameworks provide some useful insight on how to implement them. However, these insights are not organized in a generally applicable structured process to determine the abstraction level. Thus, the abstraction process is largely left to the creativity and the expertise of the modeler. This can lead to an unnecessary number of iteration loops, that increase the effort for modeling or worse, to models that are not capable of solving the problem. A systematic approach is needed to adapt and tailor model frameworks for specific systems and to solve specific problems.

3. Approach

The contribution of this paper is a systematic method to abstract a production network and to generate a tailored, specific model based on a given general model framework. It builds upon the general modeling process presented in [8] and draws inspiration from both the aggregation levels shown in [3,4] and the aggregation criteria from [5,6] and consolidates them to a structured method. The method was created through theoretic considerations and has been applied in a case study on the global production network of a tooling machine manufacturer.

3.1. Definitions, context & general concept

A *model* can generally be understood as a purposefully simplified representation of a real system [16]. Fig. 2 shows a real-world system consisting of several *objects* of different types that interact with each other. A model intentionally strips these objects of some irrelevant properties and represents them with *entities* corresponding to the type of object. It also mirrors the relevant interactions between objects. While they may be used differently in other contributions, the term *entity* will always refer to the model representation of a real-world *object* within this paper.

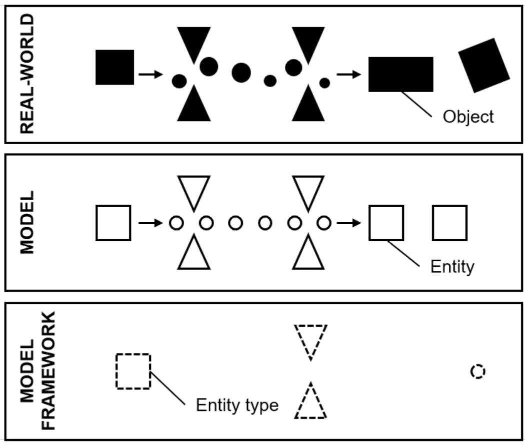

Fig. 2. Real-world systems, models and model frameworks.

A majority of the modeling work consists of identifying objects that are relevant to the application and including properties of these objects in the model [8]. This part of the modeling process is not the focus of this paper and assumed to be given by a *model framework.*

However, another important aspect exists, concerning the *level of detail* in which model entities represent real objects. To optimize model complexity and capability, it is critical to decide which objects can be aggregated in one entity and which need to be differentiated.

When imagining an *object type* as a combination of multiple properties, an *object* is then one possible combination of values of those properties. For example, the object type ‘production site’ may contain the properties ‘location’, ‘associated subdivision’, and ‘technological capabilities’. One object of this type, i.e. a ‘production facility’ can then be characterized by the values of these properties. In mathematical terms, each property can be understood as a dimension and an object as a tuple of the dimensions characterizing an object type.

Following the previous notions, the abstraction level of a model can be determined by identifying all properties relevant for the examined application and create model entities for every unique combination of those properties.

Fig. 3 shows how this concept is applied. Multiple objects are displayed as value tuples of three dimensions, e.g. properties of the examined object type. Only two of these dimensions form the model plane on which entities that represent the objects found in the space exist.

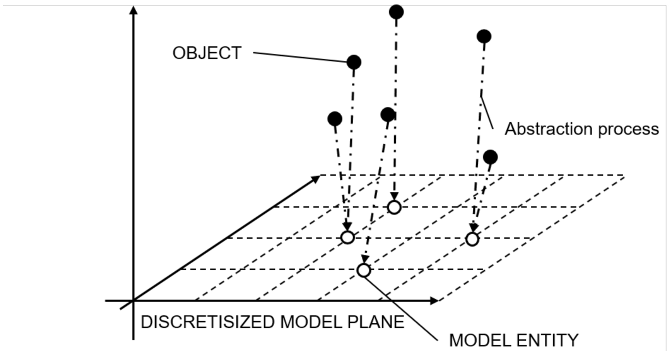

Fig. 3. Representing examined objects with model entities.

A good example of this concept is the entity type ‘production facility’. In an exemplary model framework, the main function of ‘facilities’ is to enable the creation of ‘products’ using ‘sub-products’ and resources of the ‘facility’ such as ‘machines’ and ‘personnel’.

In principle, this process can be viewed in several levels of detail. For example, with high detail when considering every production line as a ‘facility’ or on a very broad level when determining a ‘production facility’ only by its geographical location. This idea has already been explored by some authors in their respective model frameworks [3,4].

In the following, the properties selected for differentiation will be referred to as *abstraction criteria*. Based on the selection of the right abstraction criteria for every object type and associated entity type, the abstraction level of the entire model can be determined.

Fig. 4 portrays a method to structure the process of abstraction level determination. This three-phase method consists of problem decomposition, concept application and model composition which are described in the following sections.

3.2. Phase 1 – Requirements & restrictions

The first phase serves to increase understanding of the task in a structured way. To do so all factors contributing to the model are closely examined. Those are, as shown in Fig. 2, the *application* of the model, the *model framework* as well as the examined system and the *data* used to describe it. Fig. 5 shows how these requirements and restrictions can be viewed as limitations to the available modeling freedom.

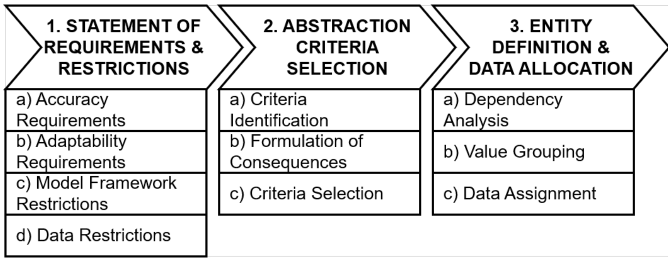

Fig. 4. 3-phase process to create a model.

A possible way to structure requirements of the application is to differentiate between *accuracy* and *adaptability* requirements. These two differ significantly in their implications and the way their fulfillment can be validated. Accuracy requirements can be identified with the fundamental question: *Which parts of the studied system need to be represented with which error margin?* The first part of the question is qualitative. Examples of answers to this question in supply-chain models could be: a) 'Stocks need to be represented!' b) 'Single unit costs do not need to be represented!' c) 'Accurate comparison with intra-firm calculations is relevant!' After these qualitative requirements are identified, the second part of the initial question asks to quantify the requirements. For example, a) 'Unit costs have to be within ±5% of internal calculations!' b) 'Transport volumes have to be within ±10% of internal records!'

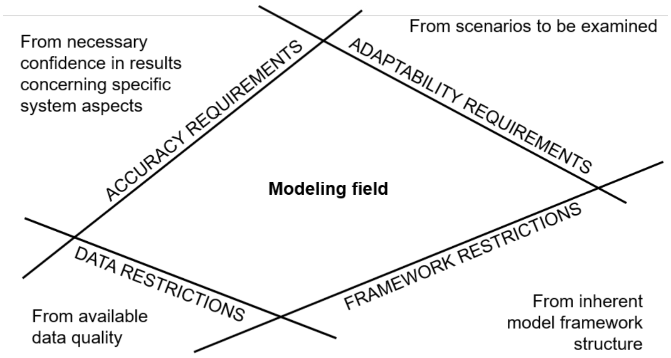

Fig. 5. Overview of requirements & restrictions limiting the field of modeling.

Whereas accuracy requirements are mainly aimed at the static replication of the current system, adaptability requirements point towards the necessary flexibility of the model. The lead question is: *Which changes to the system does the model need to replicate?* For a supply chain model such requirements could be: a) 'The model needs to be able to replicate consolidation of a site!' b) 'Production machine transfers need to be possible!' A good process to identify these requirements is to study all the scenarios the model is supposed to replicate and categorize implied changes to the system. In case of both accuracy and adaptability requirements, it is beneficial to already include negative statements, i.e. requirements the model does not need to meet and even prioritize requirements. Additional techniques to determine objectives for the model and derive requirements are for example described in [10]. Whereas the application implicates requirements, both the used model framework and the data impose restrictions on the modeling freedom. While various models can be derived from a single model framework, the number of possible models is still limited. For example, a strategic supply chain model framework is probably ill-suited to solve factory layout problems. The specific restrictions of the used framework should be investigated and documented. A helpful tool when doing so are entity relationship maps, as described in [17]. ER-maps show the data structure of the model framework. Especially interesting here are dependencies between entity types, as they enforce a minimal level of detail. These *positive restrictions* can also be understood as mandatory abstraction criteria. E.g. an entity type A that is dependent on another entity type B must at least have as many instances as B. In addition to positive restrictions, model frameworks also impose *negative restrictions*. They limit the maximal amount of detail a model can feature. The ER-maps are also used to determine all the necessary types of data. Based on this potential data sources to obtain the data for all entity types. These sources are then examined in terms of their quality. Aspects of data quality and their assessment are sufficiently covered in the literature, for example in [18]. Within these aspects, the granularity of the data is specifically interesting for the examined problem. Essentially the model can only be as exact as the data used to create it. By identifying the dimensions which are aggregated in the data, it is possible to select the final sources and formulate resulting restrictions.

After completion of the data restrictions, a *comprehensive statement of both requirements and restrictions* is obtained. It is later in phase 2 used to assess and evaluate modeling options.

3.3. Phase 2 – Selecting abstraction criteria

In the following second phase, the concept to define the level of detail through abstraction criteria is utilized and implemented. First, available abstraction criteria, i.e. object dimensions need to be identified for every object type-entity type pair. This is a creative task without a single generally true solution. A model builder who confronts this problem needs to have a good understanding of the application, i.e. the modeled system and the problem to be solved as well as the model itself.

After identifying the potential dimensions, the consequences of using or not using each of these dimensions as abstraction criteria need to be formulated. Fig. 6 shows how the properties of each object type-entity type pair in a model framework are identified and the corresponding consequences for the entire model are considered. By matching these with the requirements and restrictions defined before, properties are defined to be abstraction criteria.

The process of identifying potential abstraction criteria and their corresponding consequences contains the main creative part of conceptual modeling with a given framework. However, as this task is not specific to one application of the model framework but rather generalizable for every application of it, these two steps can also be generalized. Hence, a model framework owner could generate an *entity catalog* with

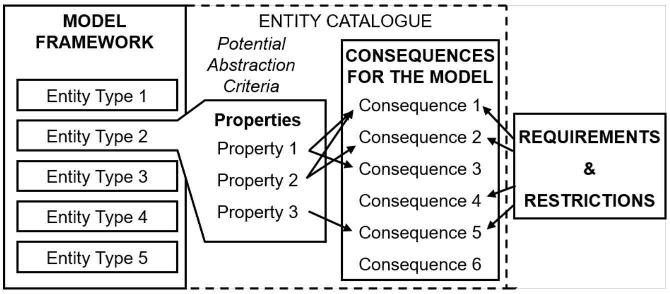

Fig. 6. Selecting abstraction criteria by matching resulting consequences for the model with requirements and restrictions.

potential abstraction criteria and their respective consequences on a finished model for all entity types of the framework.

A model builder using the framework and the catalog only needs to select the right abstraction criteria by matching the consequences of using or foregoing a criterion with the requirements and restrictions from phase 1. To simplify this, the consequences of foregoing specific abstraction criteria can be formulated as restrictions towards the application. Similarly, consequences of using criteria can be classified as requirements towards the context, i.e. data sources and model framework. By choosing exactly the criteria necessary to fulfill the application requirements and checking for violations of the restrictions, the right criteria can be selected. If restriction violations occur, either data sources, model framework or the application goals should be adapted accordingly.

After the abstraction criteria have been selected, the actual implementation is still required, which can be subdivided into *dependency analysis, value grouping,* and *data assignment*. To create entities efficiently, the right order of conduct is important. *Dependencies* between distinct entity types need to be considered first. For example, when the object type 'product' is investigated and the associated 'production facility' is chosen as an abstraction criterion, it is crucial to first define the 'facilities' before the 'products', as the number of distinct 'product' entities is influenced by the number of 'product facility' entities. To integrate this problem into the existing concept, properties of objects can be classified as *dependent* and *independent*. Dependent dimensions reference either the property of another object type or even objects of another type themselves. By starting with entity types without dependencies and then step by step defining entity types only dependent on already defined ones, the entity definition process can be carried out efficiently. In the case of circular dependencies, an iterative procedure is necessary.

3.4. Phase 3 – Defining entities and assigning data

Using object dimensions which contain a large number of different values as abstraction criteria may lead to a very high number of model entities. For example, when the dimension location is selected as an abstraction criterion for the entity type customer, a big number of entities may be required. By *grouping* values into sensible sections, this problem can be solved. Several techniques to create groups of values exist, from simple fixed interval groups to multidimensional cluster techniques. All of these techniques can significantly reduce the required number of entities and thus the complexity of the model. However, the related loss of representation accuracy should also be considered.

Finally, additional data for entity properties must be assigned to the entities. As the dimensions which are distinguished in data sources may deviate from the dimensions distinguished in the model, *assignment techniques* are necessary. These depend on the property that is assigned and the constellation of model and data dimensions.

With the conclusion of this final step, a model is created that is tailored to its specific application. It features a specific and standardized abstraction level and has been populated with data efficiently. Whereas the modeling process is presented as being linear for simplicity here, several iterations will usually be required. To ensure the validity of the model the assumptions and results made should be validated after each step. Several quantitative and qualitative techniques to do so are presented in the literature [9,11,13].

4. Application

The presented method was used in a case study at the global tooling machine manufacturer TRUMPF. The model framework *OptiWo* created at the Laboratory for Machine Tools and Production Engineering (WZL) was utilized to develop strategies to strengthen the market position of TRUMPF in Asia and North America. The goal of the project was to identify potential costs savings in the production network for TRUMPF's high volume products for laser-based sheet metal cutting and punching. While the company held a comfortable market position in Europe, it was struggling in the more cost sensitive Asian market. The question was more specifically, which products or components were best suited for local production and which strategic manufacturing network configuration was most cost-efficient.

OptiWo is a model framework designed for the analysis of the internal production network. Products are modeled in a bill of materials fashion, i.e. a product requires sub-product that can be manufactured and shipped from different facilities. A product 'car' could, for example, require sub products like an 'engine', a 'chassis', a 'suspension', and an 'interior'. Every product then requires a certain processing time of technologies like 'assembly', 'milling', 'cleaning', or 'painting'. The accumulated processing time of all products at a location is used to determine the number of required machines of each technology as well as employees operating them. By accumulating material costs, machine operation costs, employee costs, machine, and facility depreciation, etc. the total production costs of a specific network configuration can be determined. The cost efficiency of different network configurations in multiple scenarios can then be compared [19, 20].

Following the presented approach, the major requirements and restrictions were identified. Among the most important requirements was the ability to represent recognizable components in the model and calculate production transfers accurately. One important restriction of the framework was the specify of 'products' to one 'facility'. Based on these requirements and restrictions the abstraction criteria 'product function', 'production technology type', and 'production site' were used. Whereas the other two abstraction criteria may be self-explanatory, 'production technology type' was used out of the idea that a product could be produced wherever similar production technologies were available which is a good model of reality in this context. Other available criteria like production process where found to be too specific without offering significant benefits in terms of the requirements. While the requirements where fulfilled, the bill of material, BOM, which contained approximately 4000 'product' object entries on average was narrowed down to about 40 'product' entities. Thus, significantly reducing overall model complexity. This selection of abstraction criteria for 'products' required 'facilities' and 'production technologies' to be defined prior to the 'products'. The 'products' then had to be defined in an iterative process to ensure the recognizability of components across multiple product types. Finally, additional computation data like 'transport size' and 'raw material costs' were allocated to the product based on several data sources. Similar steps were taken for the other entity types of the framework resulting in an optimized model fulfilling the requirements with manageable complexity.

5. Conclusion & future work

In comparison with existing approaches, the here presented concept and method marks a step toward systemizing the conceptual modeling process. For cases where an already existing model framework is used to solve a problem, significant advantages in terms of both efficiency and effectiveness of modeling can be seen. Those advantages mainly derive from structuring the given problem and the solution space. By creating *entity catalogs* for frequently used frameworks the efficiency can be increased even further. In addition, less experienced model builders would be enabled to easily create good models.

In the future, additional case studies of different modeled systems, application focusses and model frameworks could be used to further test the presented method and identify weaknesses and shortcomings. Expanding the method to incorporate the aspect of *system boundary definition* would also be most valuable, as it would complete the entire process of model creation based on a preexisting framework. Very interesting and undoubtedly useful for the practical application would be the creation of an algorithm to automate the model creation process. Such an algorithm would have to be adapted to every organization individually as data systems vary from company to company. It would also have to overcome problems with data quality and non-numerical data interpretation. However, it would allow organizations to reduce the costs of model creation significantly and use models of production networks as standby tools for strategic and tactical network planning.

Acknowledgment

The authors wish to acknowledge that the research presented in this paper was supported and funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project number: SCHU1495/116-1

References

[1] F. Jacob, G. Strube. Why Go Global The Multinational Imperative. In: Global Production, E. Abele, T. Meyer, U. Näher, G. Strube, R. Sykes. Berlin-Heidelberg: Springer-Verlag; 2008. p. 2-33.
[2] W. Klibi, A. Martel, A. Guitouni. The design of robust value-creating supply chain networks: a critical review. European Journal of Operational Research 2010;203:283-93.
[3] M. Friese. Planung von Flexibilitäts- und Kapazitätsstrategien für Produktionsnetzwerke der Automobilindustrie; 2008.
[4] M. Bundschuh. Modellgestützte strategische Planung von Produktionssystemen in der Automobilindustrie. Hamburg: Verlag Dr. Kovač; 2008.
[5] D. Vila, R. Beauregard, A. Martel. The strategic design of forest industry supply chains. INFOR: Information Systems and Operational Research 2009;47:185-202.
[6] N. L. Ulstein, M. Christiansen, R. Grønhaug, N. Magnussen, M. M. Solomon. Elkem Uses Optimization in Redesigning Its Supply Chain. Interfaces 2006;36:314-25.
[7] VDI 3633 Part 1: Simulation of systems in materials handling, logistics and production – Fundamentals. VDI-Handbuch Produktionstechnik und Fertigungsverfahren. Band 1: Grundlagen und Planung; 2014.
[8] S. Robinson. Conceptual modelling for simulation Part II: a framework for conceptual modelling. Journal of the Operational Research Society 2007;59:291-304.
[9] A. M. Law. How to build valid and credible simulation models. Proceedings of the 2008 Winter Simulation Conference; p. 39-47.
[10] K. Kotiadis, S. Robinson. Conceptual Modelling : Knowledge Acquisition and Model Abstraction. Proceedings of the 2008 Winter Simulation Conference, p. 951-958.
[11] J. Banks, J.S. Carson II, B. L. Nelson, D. M. Nicol. Discrete-Event System Simulation. 4th ed. Upper Saddle River, NJ: Pearson Education Internat.; 2005.
[12] R. G. Sargent. Validation and verification of simulation models: Proceedings of the 2004 Winter Simulation Conference; p. 17-28.
[13] O. Balci, Verification, validation, and accreditation: Proceedings of the 1998 Winter Simulation Conference; p. 41-48.
[14] B. P. Zeigler, H. Praehofer, T. G. Kim. Theory of modeling and simulation. 2nd ed. Academic Press; 2000. p. 329-364.
[15] S. Robinson. Conceptual modelling for simulation Part I: definition and requirements, Journal of the Operational Research Society 2007;59:278-90.
[16] D. Brade. A generalized process for the verification and validation of models and simulation results. 2004.
[17] P. P.-S. Chen. The Entity-Relationship Model – Toward a Unified View of Data. ACM Transactions on Database Systems 1976;1:9-36.
[18] L. L. Pipino, Y. W. Lee, R. Y. Wang. Data Quality Assessment. Communications of the ACM 2002;45:211-8.
[19] G. Schuh, T. Potente, R. Varandani, T. Schmitz. Global Footprint Design based on genetic algorithms – An Industry 4.0 perspective. CIRP Annals 2014;63:433-6.
[20] G. Schuh, J. Prote, B. Fränken, A. Gützlaff. Assessment of production network scenarios based on economic value added. DEStech Transactions on Engineering and Technology Research icpr 2017;116-21.