Model-driven logistics engineering - challenges of model and object transformation

Michael Lütjen*, Hans-Jörg Kreowskib, Marco Frankea, Klaus-Dieter Thobena, Michael Freitagc

aBIBA, Hochschulring 20, 28359 Bremen, Germany
bUniversity of Bremen, Linzer Strasse 9a, 28359 Bremen, Germany
cArcelorMittal, Carl-Benz-Straße 30, 28237 Bremen, Germany

Abstract

Production and logistics systems are part of a complex world with global sourcing and sales. In order to run production and logistics systems efficiently, plenty of different IT systems are needed. Thereby, production and logistics processes have to pass different systems with different purposes. In order to plan, control and coordinate such processes accurately, conceptual modeling techniques are needed, which overcome the gap between different domains and purposes as well as design and operational levels. Two main approaches of model-driven engineering (MDE) are suitable to face these problems; domain specific modeling languages (DSML) and model transformation methods. This paper develops the concept of model-driven logistics engineering by analyzing the domains of production and logistics. The objective is to identify characteristics of their inherent object transformation and to present aspects of model transformation. At least, the essentials of model-driven logistics engineering are defined and promising approaches of the field of model-driven engineering are presented.

© 2014 The Authors. Published by Elsevier Ltd.
Peer-review under responsibility of the Organizing Committee of ICIME 2014.

Keywords: Model transformation; system analysis; model-driven logistics engineering

* Corresponding author. Tel.: +49-(0) 421-218-50123; fax: +49-(0) 421-218-50003.
E-mail address: LTJ@biba.uni-bremen.de
1. Introduction

Today, business is complex, dynamic and fast with focus on production and logistics networks [1]. Global sourcing and sales demand for flexible business, production and logistics models, which are supported by powerful IT infrastructure. A simplified overview over the existing IT infrastructure is given by Hannus [2], see Fig. 1. Since 1996, the IT architectures aim a paradigm shift, from monolithic and central oriented systems to decentralized, loosely coupled and autonomous systems. Especially, the new concept of industry 4.0 requires flexible and cyber physical compatible IT systems, which allow the continuous monitoring of all relevant production and logistics processes as well as resources. The precondition is the availability of powerful software techniques, which helps to bridge the gap between real and cyber world [3].

![Fig. 1. IT-Infrastructure in the logistics [2].](image)

Model-driven engineering (MDE) is one of the most promising approaches, whereby the software development process follows the idea of standardising and simplifying the design process by use of graph-based modeling techniques. Thereby, the main challenge of the domain of production and logistics is to find an idea for the mismatch of standardisation and individualisation [4]. On the one hand, standards are needed to face growing data, more complex models and more detailed processes in order to plan precisely. But on the other hand, the production and logistics demands for individual and flexible process design in order to improve efficiency. Individuality and flexibility enlarge the options for decisions, but also lead to needs for very detailed process modeling. If individual process design should be considered for example in supply chain management, it will be necessary to model such processes very accurate in order to consider the specific system behaviour [5]. With respect to the domain of production and logistics this paper introduces the term of model-driven logistics engineering (Fig. 2).

The idea of model-driven logistics engineering is to address the characteristics and challenges of logistics by passing various modeling levels. In order to bring changes from the strategy level to the implementation level, efficient techniques and methods are needed, which help to link different kinds of models. By following the idea of model-driven business transformation, the information quantity will increase from strategy to implementation level. Thereby, the main challenge is to integrate model information from the higher to the lower levels and reversely. Besides this vertical view, the horizontal linkage and integration of models also have to be considered. Usually, each entity is interacting with a lot of other entities, for what reason communication between models of different entities have to be considered.
One essential technique to enable the integration of different kinds of models and the communication between them is transformation. Whereby, not only model transformation is analyzed but also object transformation. With respect to production and logistics systems, a lot of object transformations occur also inside the model, which have to be expressed [7]. These are serious modeling problems, because the correct modeling and transformation of semantics is still a serious problem [8]. In order to give an overview of the different ideas and problems of object and model transformation, the domains of production and logistics as well as informatics will be analyzed. Afterwards, essentials of model-driven logistics engineering will be outlined, whereas the last chapter will show promising approaches and methods of model-driven engineering (MDE). The paper will end with a conclusion and a short outlook to further activities.

2. Challenges of model and object transformation

2.1. Transformation as a concept of cybernetics

The cybernetics is an approach of control and communication regarding to complex systems, which has influenced many disciplines, like e.g. biology, engineering, informatics, management, sociology. By definition of Wiener the cybernetics is “the science of control and communication, in the animal and the machine” [9]. The purpose of cybernetics was to develop a language and techniques that would help to get an understanding of control and communication in general. Wiener introduced the idea of entropy, which describes the level of organization and structure of a system. He assumed that a system is not capable of organizing itself and naturally tends to increase the level entropy. By information processing, control can force a system to maintain the structure and organisation. This idea of closed-loop systems became the starting point of a new perspective, which finds entrance into many disciplines to explain system behaviours and structures. But cybernetics itself never became a discipline. Nevertheless, one of the most fundamental concepts in cybernetics is the idea of differences between objects and object property changes with time [10]. Thereby, the cybernetics uses a very clear understanding of the terms transition and transformation [10]:

- **Operand**: The property which is changed.
- **Operator**: The factor which acts upon the operand.
- **Transform**: The target value of the property which is changed.
• Transition: The transition is specified by the two states and the indication of which changed to which.
• Transformation: A set of transitions, on a set of operands. The operator can act on more than one operand.

This universal definition will help to understand the development of the domain-specific definition of production, logistics and informatics, which will be given at the following.

2.2. Object transformation in the domain of production

In the field of production the term of transformation is used for the effect of value creation processes. This means that the state of an object is changed by the production system. According to the black-box principle, the production system transform input to output [11]. Usually, the transformation process has defined goals to transfer the object from an existing state to a desired state [12]. This means that production material will be transformed to a product, whereby production resources and energy are used (Fig. 3). This leads to environmental effects, which occur and affect ground, air and water.

![Fig. 3. Value creation process of production systems, compare [13].](image)

The value creation processes of production systems can be classified into six different types of manufacturing processes by DIN 8580 (table 1). In context of the idea of transformation, it has to be considered that most of the manufacturing processes are not only the change of some properties of one object, but for example the joining of two objects or the dividing into two objects. For example, primary shaping is the creation of one object by an amount of other objects, which can be also in another phase.

The term of transition is not used in the domain of production engineering, but in the domain of business management and sociology with production as object of study. In these domains the transition means the changing of a production system by its structure and behaviour in relation to a new paradigm or technology [14]. Thereby, the term of transition is used in a meaning of minor transformation.
Table 1. Classification of manufacturing process (DIN 8580).

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary shaping</td>
<td>Primary shaping is the creation of an initial shape from the molten, gaseous or formless solid state.</td>
</tr>
<tr>
<td>Material forming</td>
<td>Manufacturing through the threedimensional or plastic modification of a shape while retaining its mass and material cohesion.</td>
</tr>
<tr>
<td>Dividing</td>
<td>Dividing is the local separation of material.</td>
</tr>
<tr>
<td>Joining</td>
<td>Joining is the assembly of individual workpieces to create subassemblies and also the filling and saturation of porous workpieces.</td>
</tr>
<tr>
<td>Coating</td>
<td>Coating means the application of thin layers on components, for example by galvanization, painting and foil wrapping.</td>
</tr>
<tr>
<td>Modifying material property</td>
<td>The purpose of modifying material property is to alter material characteristics of a workpiece.</td>
</tr>
</tbody>
</table>

2.3. Object transformation in the domain of logistics

The domain of logistics is characterized by transport of goods and cargo, which can take place in all fields of today’s business like procurement, production, distribution. In other words, the task of logistics is the processing of goods and cargo by use of logistics transformations [15]. There are three types of logistics transformation (Fig. 4). The spatial transformation describes the change of the position of the object, which means a movement of the object. The temporal transformation describes the passing time of an object until it is available again. The last transformation is the physical transformation. This kind of transformation changes physical properties of the object mainly in production. But also in logistics are some physical transformations, for example picking and packing. These transformations can be described by the logistical transformation LT as depiction of the logistics input object X with its properties \((x_1, x_2, \ldots, x_n)\) and the logistical output object Z with its properties \((z_1, z_2, \ldots, z_n)\).

![Fig. 4. Types of logistical transformations [15].](image-url)
Based on these three types of logistical transformations five different logistics processes can be defined [15], [16]:

- Transport process
- Shipment process
  - Transshipment
  - Allocation
  - Concentration
- Storage process
- Handling process
- Information process

The transport process is the local movement of an object, whereby the shipment can be classified into three subclasses; transshipment, allocation and concentration. For example, transshipment changes the belonging of the object from one truck to another truck, whereas allocation means to unload several boxes from a container. The concentration describes the opposite meaning and is used to load several boxes into one container, for example. The storage process is a temporal transformation. In a closer meaning the handling process means some physical transformations by having also some small spatial transformations. In the definition of VDI 2860 the term of handling can be nearly every process, which requires the movement of the object also for small ranges. Because logistics is motivated by having the correct object at the correct time at the correct quality at the correct quantity at the right place and information processes are needed for coordination.

2.4. Model transformation in the domain of informatics

Model transformation is part of model-driven engineering and performs the transformation of models. Mens and Van Gorp propose a taxonomy of model transformation [17]. At the following some essential ideas are reviewed briefly.

Kleppe et al. provide a definition of model transformation, which distinguish between transformation, transformation definition and transformation rule [18]:

“A transformation is the automatic generation of a target model from a source model, according to a transformation definition. A transformation definition is a set of transformation rules that together describe how a model in the source language can be transformed into a model in the target language. A transformation rule is a description of how one or more constructs in the source language can be transformed into one or more constructs in the target language.”

Additionally, it exists the idea of merging and splitting models by transformation. This means many-to-one model transformations have multiple source models and one target model. One-to-many model transformations have one source model and multiple target models. Regarding to their abstraction level it can be distinguished between horizontal and vertical model transformations. In case of a horizontal model transformation the source and the target model have the same level of abstraction, whereas a vertical model transformation have source and target models with different levels of abstraction. Another perspective of transformation is given by endogenous and exogenous model transformations. Endogenous transformations are transformations between models expressed in the same language. Exogenous transformations are transformations between models expressed using different languages.

Typical examples of exogenous transformation are [17]:

- Synthesis is the transformation of a higher-level specification (e.g. a design model) into a lower-level specification (e.g. an executable program), which is known as code generation for example. The inverse transformation from a lower-level specification into a higher-level specification is known as reverse engineering.
The concept of language migration describes the transformation from a program written in one language into another language, but keeping the same level of abstraction.

Typical examples of endogenous transformation are [17]:

- Refactoring is the transformation of the internal structure of a software to improve the quality without changing its observable behaviour.
- Simplification and normalisation is used to decrease the syntactic complexity, whereby formal refinement means to use first-order predicate logic or set theory. This can be gradually refined in a way that the end result uses exactly the same language.

3. Essentials of model-driven logistics engineering

By analyzing the different ideas of transition and transformation, it is shown that production and logistics are focused on object transformation while informatics are focused on model transformation. Nevertheless, there are some aspects, which have to be considered in the context of model-driven engineering. In reality, object transformation is not simple and requires technical, logistical as well as economical knowledge. Many challenges of model-driven logistics engineering exist, which still are not addressed or solved. We have identified the following features, which would be useful to evolve the idea of model-driven logistics engineering:

- different system views
- industry/section specific modeling
- process/system (re-)engineering
- scalable aggregation levels
- interoperability of business, production, logistics/transport models
- conceptual modeling and integrated simulation
- data driven generation of conceptual models
- readiness for industry 4.0

Production and logistics are very interdisciplinary. A lot of stakeholders have different interests and goals regarding to a production and logistics system. Therefore different views are needed in order to coordinate the interests of the stakeholder and to use the knowledge of them for efficient system design. Not only information, control and material flows are needed. Today, energy as well as mechanical and financial flows have to be part of such models. The common and most intuitive way is to use a process-centric 2D-modeling approach, but in production and logistics environment it is also necessary to integrate also 3D-virtual modeling environments. Because not all stakeholders are experts in information science, it is necessary to bring the model to the industry and/or section, where it belongs. This means domain specific modelling is a main challenge of production and logistics. Language, notations but also semantics have to be used, which are comprehensible for domain experts. Thereby, automatic classification of domain specific model parts would help to characterize domains and to give an idea of interoperability. Also computer-guided modelling could profit by this classification in order to lookup modeling examples. Intuitive procedures should help the domain experts to use different views in general, wherefore the current status of the modelling progress have to be detectable. Another aspect of model-driven logistics engineering is process/system (re-) engineering, which should help to identify semantic and syntactic inconsistency, but also optimisation potential for example by comparison to other process/system structures with similar functions. Thereby, key performance indicators are needed for evaluation and some options for automatic generation of process/system should be explored. Because modelling of production and logistics systems is very complex it is needed to get scalable aggregation levels, which allow scaling in spatial but also in temporal dimensions, like e.g. processes. But not only process and system views have to be scalable, also the calculation of key performance indicators should be scalable.

The interoperability of business, production, logistics/transport models is very needed because at the moment each division is viewing on their own goals and synergy effects could not be used, because there is no integrated planning and system design. Horizontal model connectivity and integrability should help to overcome these
problems. Therefore standard data exchange formats for models should be developed and it should be possible to
connect/integrate business process oriented models (e.g. BPMN) with physical models (e.g. MapleSim) in order to
get access to technical aspects of production and logistics. In general, it is very necessary to have an option of
integrated planning and simulation, which allow to generate simulation models by model transformation of
contceptual models. To be more visionary, it should be possible to generate real components and program structures
of production and logistics software. Thereby, source code should be generated but also the communication
interfaces and protocols should be possible. At least, this process should be reversible and conceptual models should
be generated out of log and program data. Another very important aspect is the readiness for industry 4.0. The world
of production and logistics is changing and it must be possible to model smart products, objects and infrastructures
as well as different aspects of autonomous control structures.

4. Promising approaches of model-driven engineering

The field of model-driven engineering comprises a lot of innovative research approaches, which could be used to
address the challenging requirements of model-driven logistics engineering. The main challenge is to overcome the
problem of standardisation and individualisation regarding complexity. It shows that MDE tries to become more
individual by use of domain-specific modelling languages (DSML), but also tries to standardize this process in a
way that DSMLs are described by metamodels, which can be used for model transformation [19]. Domain-specific
modeling languages (DSML) are modelling languages whose purpose is to efficiently represent concepts from a
particular domain, like financial services, production and logistics [20]. The description of domain specific
semantics by use of general purpose modelling languages, like UML, is sometimes too complex or just impossible.
The application of DSMLs means to simplify the use and to increase the model accuracy in the same way. Key
elements, semantics and constraints of the domain are defined precisely in the metamodel and can be used by
domain experts to express the system behaviour more intuitive [19]. Dekhinet et al. present an approach of defining
a DSML for production systems using the GME (Generic Modeling Environment) framework [21]. The modeller
can create a simple production system model and transform this DSML to a well-defined semantic model (Petri net,
abstract state machine, …) in order to perform certain operation such as formal verification. By use of such DSML
approaches the need of experienced software architects for simple scenario modelling will be reduced and the
modelling will get easier, e.g. by use of domain familiar graphics and icons. In DSML representation it can be
distinguished between “ad-hoc” representations and UML profiles, which add additional semantics to UML
elements via e.g. stereotypes and tagged values [20]. There are some approaches, which try to design DSMLs in an
automatic way by use of UML profiles [22]. This would allow improving the interoperability of such models.
Another approach is the collaborative and/or integrated modelling. Walter et al. presents a peer-to-peer
infrastructure, which allows the decentralised modelling in peers, which combine the models later on [23]. If the
models are different and DSMs, the approach of Vallecillo could be useful [24]. He combined different DSM
models, which would allow solving domain specific problems in an integrated way. Thereby, one basic requirement
is the efficient comparison and matching of models and ontologies [25]. In general, matching refers to elements
that represent the same idea or artifact. Thereby, Lin et al. developed a differentiation tool for domain-specific
models, which determines whether two models are syntactically equivalent [26]. The most approaches are focused
on such matching of metamodels, whereby the matching of semantics is still a big problem. Kappel et al. present an
approach, which is based on the user’s knowledge about the notation of the modeling language [27]. Such inter-
model mapping should allow the sufficient definition of model transformations regarding semantic correspondences
and should be more user-friendly then the direct specification of transformation rules. One of the most challenging
approaches is the idea of modeling language creation by demonstration [28]. In general, the variety of the
transformation approaches has led to the development of various model transformation tool [29]:

- General-purpose tools include AGG, PROGRESS and GrGent.NET
- Reengineering tools include FUJABA
- Model-To-Model transformation tools include GReAT, ATOM3 and MOLA
- Model checking and verification tools include VIATRA, GROOVE and CheckVML

A very fine attempt to the model-driven engineering of production and logistics systems has been made by
Kreowski et al. [30]. They propose a framework for the modeling of production and logistics systems with an emphasis on model transformation.

5. Conclusion

The idea of model-driven logistics engineering is motivated by the vision of flexible and powerful logistics systems, which can be designed virtually and realized effectively by adapted software and IT systems. Thereby, models of such systems should not be limited to one company or facility, but able to overcome the gap between different domains, modeling techniques and abstraction levels. At the moment, two main approaches of model-driven engineering (MDE) are suitable to face these problems; domain specific modeling languages (DSML) and model transformation methods. Domain specific modeling languages (DSML) can help to bring the models closer to real-world practices and model transformation can help to bring such models back to abstract layers and software engineering techniques. Therefore, the domains of production and logistics were analyzed by their object transformations functions and the essentials of model-driven logistics engineering defined. At least promising approaches of model-driven engineering were presented, which showed that the vision of model-driven logistics engineering is not a dream. Further research will define a framework for model-driven engineering, which allows to apply different modeling and simulation techniques in order to plan and evaluate logistics systems in an efficient way.

References


