A Motivation and Evaluation of Hierarchical Data Structures for Application in Automotive Demand and Capacity Management

Konrad Pawlikowski
Faculty of Business Economics
Bochum University of Applied Sciences
Bochum, Germany
email: konrad.pawlikowski@hs-bochum.de

Daniel Fruhner
Institute IDiAL
Dortmund University of Applied Sciences and Arts
Dortmund, Germany
email: daniel.fruhner@fh-dortmund.de

Katja Klingebiel
Faculty of Business Studies
Dortmund University of Applied Sciences and Arts
Dortmund, Germany
email: katja.klingebiel@fh-dortmund.de

Michael Toth
Faculty of Business Economics
Bochum University of Applied Sciences
Bochum, Germany
email: michael.toth@hs-bochum.de

Axel Wagenitz
Faculty of Business & Social Sciences
Hamburg University of Applied Sciences
Hamburg, Germany
email: axel.wagenitz@haw-hamburg.de

***Abstract*— The demand and capacity management (DCM) is an essential component of the automotive supply chain management. Resource requirements in the automotive supply chain result from future or already realized market demands. DCM synchronizes these requirements with capacities and restrictions of the supply chain and production system. Demand uncertainty and volatility are especially challenging for DCM. Product variety and supply chain complexity intensify this problem. Here, an efficient product data management may increase transparency and support the DCM processes effectively. This contribution analyses and evaluates the benefits of an integration of distributed product data into a hierarchical tree structure and its applications in DCM against the background of complexity reduction. Moreover, the underlying optimization algorithms are described. The results of this study prove that a hierarchical integrated information model provides a significantly improved basis for a scenario-based DCM planning process. Data from a German automotive manufacturer (OEM) has served as basis for this evaluation.**

Keywords- product structure; automotive production; demand and capacity management; optimization; complexity; BOM rules.

I. Introduction

This contribution is an extended version of work published in [1]. The previous work has been extended, e.g., by evaluation of a full product spectrum of an OEM to provide greater insights into the effects of the integration of the distributed product data into a hierarchical tree structure. In addition, an elaborated overview of different types of product structures has been integrated.

To compete in international markets automotive manufacturers, i.e., original equipment manufacturers (OEMs), tend to offer their customers a huge variety of models which can be individualized by several hundred options. These options comprise design elements (i.e., colors), functional components (i.e., climate control system) and nowadays assistance systems (i.e., navigation and driver assistance systems). Furthermore, OEMs constantly update their product range with increasing frequency [2].Though customers have to deal with the rapidly changing variety of models, they tend to expect that their vehicle orders can be re-customized anytime, i.e., changed even shortly before actual production, and that the produced car is rapidly delivered on the formerly planned date [3][4].

In this context, logistics plays an important role. The core competence of a car manufacturer has shifted to product marketing, the coordination of suppliers, assembly of supplied parts, and the distribution of the end product [5]. Nowadays, suppliers do not only produce simple components, but also develop complex modules [6]. They also have to manage product complexity and variety and need to know in time if the OEM revised the production program for a specified model and market. Hence, the effective integrated management of the automotive production and supply chain is critical. The anticipation of the future market demand, the timely derivation of resource and component requirements as well as the integrated and coordinated capacity planning are indispensable prerequisites [7]. Most critical, resource requirements resulting from anticipated or realized market demands need to be synchronized with resource capacities and restrictions of the production and supply chain by an effective demand and capacity management (DCM). DCM processes identify demand- and capacity-asynchronies and implement appropriate countermeasures in a timely manner. DCM acts as an essential interface between market, production and supply chain processes [8][9].

Caused by the increasing number of variants and options the complexity of this process has continually increased over the past decades. A typical car consists of about 3000 to 6000 material items. If different variants and their parts are considered, it results in about 15000 to 20000 items per car (as of 2010) [10]. The planning systems have been extended continuously to manage the resulting complexity. Nevertheless, the relevant DCM data is typically kept in a highly fragmented system landscape. Data fragments are handled by different systems and thus overall transparency is limited.

For example, within DCM processes part demand is typically gradually derived from sales figures in a number of sequential processes taking into account a variety of systems [11][12][13]. These systems consider historical sales data (e.g., car rentals), retailers' annual model requests, companies knowledge about the local customer preferences and on marketing capabilities to influence customer demand [12]. Since automated processes only allow the identification and reporting of formal inconsistencies, additionally a human planner has to review the process. But, due to the increasing variety, it is more and more difficult to review the product information manually. The success of the overall process is highly dependent on the planner's experience.

Even more so, variants are often quite similar and a significant amount of information is redundant. For example, typically several car series of one OEM are based on the same vehicle platform. Additionally, the common-part-strategy supports the installation of identical modules, e.g., navigation systems, in numerous car models of different series [14][15]. If a common part is changed or even not deliverable, this effects several series. These possibly wide-spread bottlenecks are difficult to identify and control in a fragmented system landscape.

As it is easily understood, an integrated information base could reduce the complexity and increase transparency of the DCM processes immensely. The advantage of this integration is the faster and easier access to relevant data and its innermost dependencies, as well as the reduction of redundancies. Therefore, an innovative system's concept ought to integrate all related data from sales to supply chain data into a consistent and integrated information structure. Only this information model may provide the essential basis for a continuous and effective DCM process.

In general, several types of data structure types are offered by literature and practice, which may form the basis to realize such an information model. Especially, graph structures and here tree structures are an intuitively attractive approach because of their proximity to car design principles. In this context, this paper analyses and evaluates the benefits of a hierarchical tree-based data structure for the integration of DCM relevant distributed product data. The evaluation is performed against the background of complexity reduction and transparency increase. In comparison to [1] not only two car series, but a full product spectrum of a German OEM has been analyzed.

In the next section, the state of the art of automotive DCM processes is given. Afterwards promising information structures are presented in section III and the tree structure is chosen for further analysis. In section IV, an introduction to tree based data optimization methods is given, whereas section V analyses the complexity reductions gained by the application of an integrated and optimized information model for the DCM process. A conclusion including a summary and a perspective on future research and development is given in section VI.

II. State of the Art in Automotive DCM Processes

Before product data and information models may be discussed, an illustration of the state of the art in automotive planning processes and the embedded DCM process is necessary.

Today, the typical DCM planning cascade is initiated by the sales department with forecasting and planning of medium-term future market demand [16]. In this step, model volumes (e.g., number of VW Golf Trendline 2.0 TDI.) and option quotas (e.g., ratio of models with LED light or a certain navigation system) for worldwide sales regions are being planned for a horizon of 12 to 24 months. A model is typically defined by a specific series, body type, engine and gear type. The underlying forecast is based on current information about the automotive market (market shares, economic forecasting), but also on current and historical orders [17][18][19]. Furthermore, sales quotas for options are influenced for example by the sales region, technical restrictions, strategic decisions or customer preferences.

In a second step, the production planning integrates these figures with existing order volumes which may already be available for closer time periods. Next, the sales plan is translated into a production program for all sites [17].

The planning complexity of both steps is tremendous due to the variety of products. For example, a typical mid-class series (e.g., VW Golf, BMW 1 Series, Audi A3) offers about 30 to 50 different car models with up to 200 options. This results in several thousand volumes to be planned for one car models in all sales regions over a specific granular time period (e.g., month, week or day depending on planning granularity) and some 10 million related option quotas. To illustrate this complexity an easy example shall be given: Assuming three options are valid for a sound system in 40 different car models (e.g., standard radio system, comfort radio system and full navigation system). In result, for this small planning fragment 216,000 option quotas for assumed 12 months and 150 worldwide sales regions need to be planned.

The compatibility of options for a respective model is described by a highly complex set of technical rules, while the relationship between the fully-configured car order and the corresponding parts is described by the bill of material (BOM) (see Fig. 1). Technical rules define the technical feasibility of customer selectable options and further OEM-internal technical options. These rules equal restrictions and may prohibit or force options for specific models (e.g., no sunroof for convertibles), force specific combinations of options (e.g., LED head light only in combination with LED back lights) or prohibit combinations (e.g., a navigation system rules out all other radios). In addition, sales constraints (e.g. not all options are available for all models or regions) and customer preferences need to be included. So, even the planning of consistent volumes and option quotas is complex in itself and requires the integration of human experience and intuition (cf. [20]).

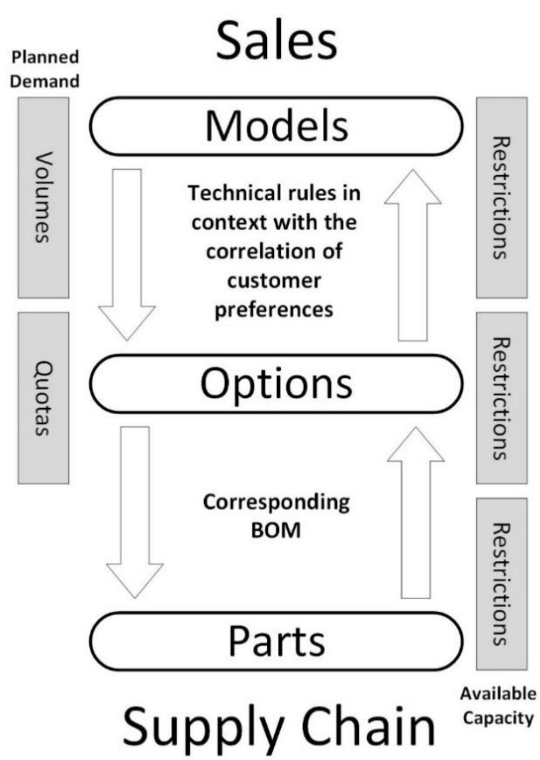


Figure 1. Bridging the gap between demand information and capacity information

The resulting sales plan needs to be balanced with production and supply chain restrictions. Capacity constraints (e.g., the maximum number of leather seat coverings per month) are present on sales level, production level and supply chain level. To balance volumes and resource requirements with constraints and restrictions in order to identify possible bottlenecks, it is necessary to bridge the gap between demand information and capacity information [13][16][11].

If fully specified orders are available, the gap is easily bridged. Fully specified orders allow to derive part demands by BOM explosion; possible bottlenecks can be determined by comparison of capacity restrictions with capacity demands. Nevertheless, because of short order-to-delivery lead times in contrast to partly long supply chain lead times, DCM process have to work on forecasts and plans rather than orders to a great extent. And it is obviously impossible to predict the exact future vehicle orders, as customers can choose from billions of possible configurations for each car type [21][10]. Forecast uncertainty, demand volatility, rapid product changes, as well as changes in the supply chain complicate this task significantly.

Even more so, a huge number of the resulting resource requirements for production or logistics depend not only on single model volumes and quotas for options, but on a particular combination of model, options and sales region (e.g. the BMW 3 series with 143hp, option = “sun roof” is anticipated to be sold 1000 times in the sales region Germany in February 2005). Therefore, some part volumes are harder to predict than others until the exact configuration of the vehicle, i.e., the order, is known. Nevertheless, as replenishment lead times in global supply networks can be long, a certain number of vehicle parts has to be ordered long before customer orders are known (cf. [16]).

In summary, the DCM process is challenging. Because of market dynamics, complexity in car configurations and correlations among models, options, and parts, the planning itself is complex [12]. But even more so, it is also characterized by conflicting goals. Sales departments are forced to react to volatile markets, increased global competitions and changing customer requirements: flexibility and reactivity is requested. Production is interested in a stable production program, which guarantees both high capacity utilization and optimal operating results. Material planning wants to fix part requirements as early as possible to avoid bottlenecks proactively as well as to negotiate the flexibility of suppliers appropriately.

This conflict can be named the dilemma of automotive DCM (cf. [22]). Typically, it is solved by planning cycles of four to six weeks, which are based on numerous workshops and committee meetings between sales, program- and material planning [17] [12] . The consequence of this long planning cycle is insufficient flexibility in reaction to market changes. To counteract this, the program is adjusted manually between program approvals and even after program freeze, i.e., within the so-called frozen period [17]. However, these adjustments cause a lack of program stability and poor transparency on future demand for parts on the supply side. The probability of bottlenecks increases and induces additional internal costs, as well as deterioration of the delivery service to the customer.

To overcome these problems, there are two theoretical approaches for the integration of these sequential planning processes in an effective holistic DCM process.

The first approach focusses on the early inclusion of resource restrictions into the sales and program planning. This requires to trace back restrictions on all levels to the decision variables, i.e., to planned volumes and quotes. As discussed, model volumes and option quotas typically include several million variables. Furthermore, technical rules and BOM rules relate these planning variables amongst each other and to part demands and thus capacity restrictions.

For example, a capacity restriction may limit the installable volume of a specific powerful battery. As a matter of fact, the installation of this battery may depend on several combinations of options, e.g., the battery is only selected if specific but several electronical options are chosen in combination. To derive the resulting limitations on model volumes and option quotas all BOM rules and technical rules that relate directly or indirectly to that battery have to be analyzed. In case of a mid-class model this may amount to a significant proportion of the overall number of rules which amount to about 15,000 technical and 600,000 BOM rules. Even more so, between option quotas and model volumes partially unmanageable correlations exist. These result not only from technical restrictions, but also from product strategy, customer preference, and marketing strategies. A customer preference for the combination of navigation systems and seat heating modules shall be given as an example for such correlations: these two options may be independent from the viewpoint of the customer, but historical data has shown that most customers who chose the navigation system also selected the seat heating; customers who do not select the navigation system rarely choose the seat heating. Consequently, when planning option quotas for navigation systems and seat heating, the high correlation of these two options and the resulting relation to the powerful battery as a part restriction needs to be integrated (based on [23]).

In result, not all restrictions may be deterministically traced back to the decision variables. This is aggravated by ramp-up and run-out processes (continuous change in options, models, etc.), dynamic changes in capacity information, multiple use of parts, commonality strategies and other restrictions that may change daily. The complete derivation of restrictions on planning variables harbors an immense complexity and is not deterministically feasible. Even if such a complete deterministic derivation process would be possible, no planner would be able to comprehend or verify the results. Hence, the early inclusion of resource restrictions into the sales and program planning only allows to focus on selected, historically critical restrictions. But of course any limitation is problematic against the background of an effective and holistic DCM process.

Consequently, the most promising perspective of an effective holistic DCM is seen in the second approach, the iterative scenario-based planning process, which is outlined in Fig. 2. Starting with a planning scenario, resources and part demands are derived by propagation of volumes and quotas by application of the full product structure. Typically, planned orders are applied here to transform planning scenarios into explodable orders. In a third step, capacity bottlenecks are identified and disclosed by backtracking to the point of origin in the planning scenario and revision of the plan.

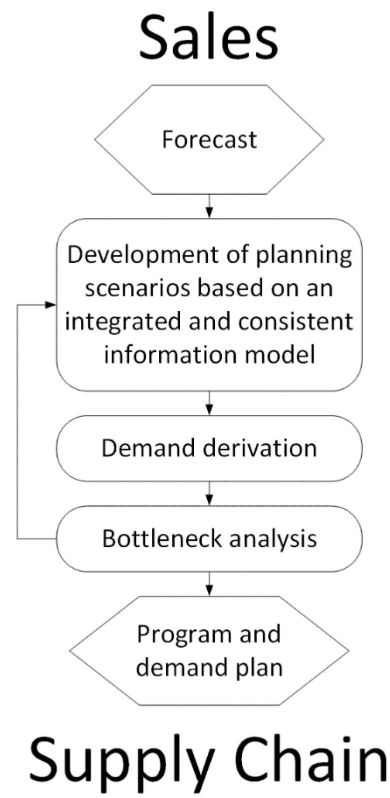


Figure 2. Integrated, scenario-based DCM process

The basis for this DCM planning process is a consistent and holistic information model, which comprises all relevant information. This information may be divided into three data partitions: the planning scenarios, the resource information (restrictions) and the product structures. Whereas planning scenario and restrictions are structured in a simply way, an efficient product representation is critical to provide transparent holistic information and allow for an efficient backtracking mechanism.

Typically, the relevant product information is complex and distributed over several systems, i.e., different data fragments as technical rules, volumes and option quotas, BOM are not integrated in a common information base.

But to efficiently support a scenario-based DCM planning process, all relevant information needs to be integrated in one common data structure. Though enhanced technologies and database infrastructures have been introduced in the last decades, an extensive data preprocessing and reduction is necessary to provide a compact yet comprehensive information basis for the later analytical data processing.

After the DCM process has been described, a general overview of different suitable product structures will be described in the next section.

III. PRODUCT STRUCTURES IN THE FIELD OF AUTOMOTIVE INDUSTRY

According to Schuh [24], a product structure is generally a structured composition of the product and its components. Typically, structure levels are introduced to represent assemblies, which bundle components in the product structure. Product structures support the multiple use of assemblies and parts. Another important objective is seen in the reduction of production data and the support of the information flow [24].

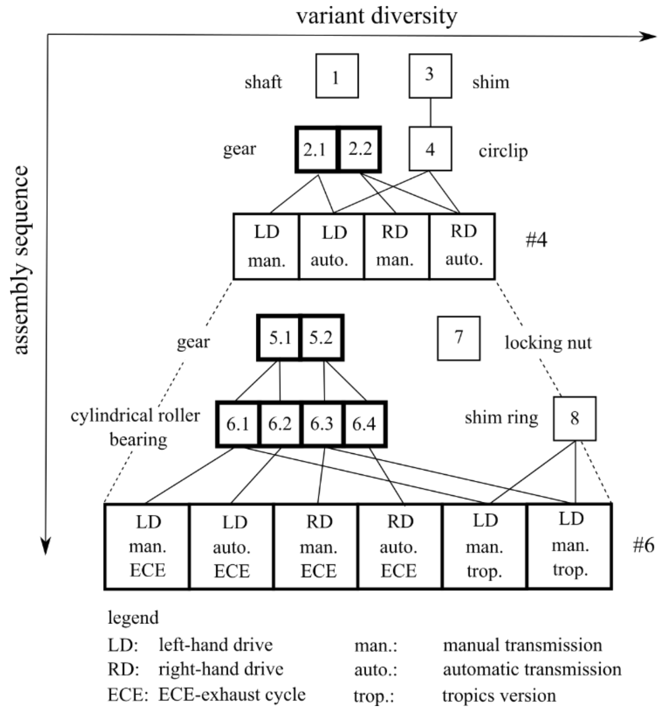


Figure 3: Variant tree [based on [30][32]]

From a technical perspective, product representations are product knowledge composed into its elementary components [25]. A component can be either a physical or a non-physical artifact (service and software components) [26]. The next larger unit are called modules. Klug [10] defines a module as an assembly of several components or assembly units. The module may comprise a variety of functions. The modules can generally be replaced (e.g., door, seat, cockpit, power pack, roof). They are used within a so-called modularization to subdivide a system. Aspects of the product life phase such as development, procurement, production, distribution, utilization and disposal may lead to modularization [27][28][29].

Especially, tree structures as product representation have proven to be promising for various applications. Kesper [30] differs between feature trees and variant trees, whereby the widely propagated feature tree is often incorrectly also referred to as a variant tree. The trees differ in their representation and the integrated information. While feature trees illustrate the variety resulting from the combinatorial of characteristics and their properties, the variant tree represents the variety of semi-finished products arising during the assembly process [30][31]. Thus, the variant tree forms the basis for the reduction of variants by means of product structure optimization or assembly sequence optimization.

The variant tree is often used to graphically represent component and product diversity, that arises in assembly processes [30][32]. Schuh [32] identifies variant trees as important means to design and evaluate product variants. The different components are symbolized by different boxes (see Fig. 3) [30]. According to Schuh and Schwenk [31], variant trees are constructed in defined steps. First, the product characteristics and their properties are captured. Then, the prohibitions of combination and other constraints on combinations of properties are defined. Thereupon, variants are generated. After integration of part information and allocation of part usage, the assembly sequence is determined. As the last step, the variant tree may be depicted graphically [30].

The feature tree is an instrument for visualizing variants or spectra with a focus on their characteristics and properties. Usually, the feature tree is started with a "root". The tree is then branched from left to right (see Fig. 4).

Each vertical level represents precisely one feature. One branch of the tree corresponds exactly to one variant. The extent and shape of the feature tree depends on the order of features. A different order alters the total number of the feature expressions to be displayed [30]. This kind of tree is not only used to depict features, it also allows the visualization of the diversity resulting from the combinations of characteristics and properties.

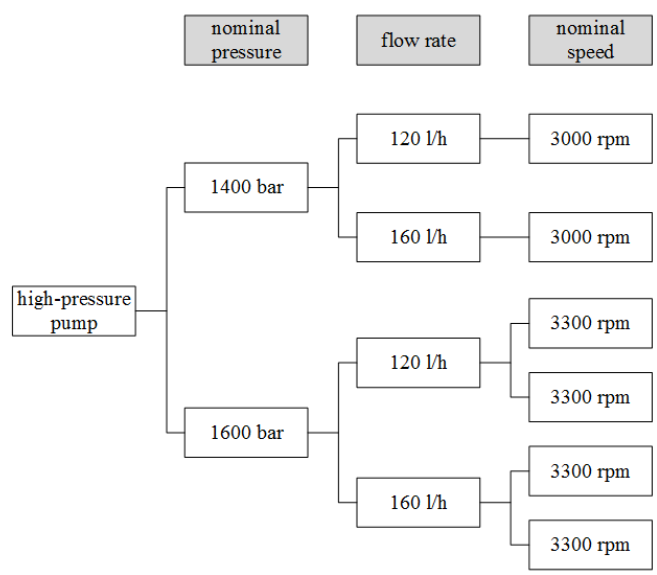


Figure 4: Feature tree [based on [30]]

Not just the product structure itself, also relationships between a product, its components and the relevant assembly tasks have to be considered within a product representation. These can be described in an extended product tree structure originally proposed by Zeng and Gu [33]. In an extended product tree, two types of nodes are distinguished. A component node represents a product or component while an assembly task node represents simplified assembly information included in the product structure. The connection between two component nodes is a parent-child relationship. A parent component (or assembly) consists of all its child components. A connection between components and an assembly task signals that the respective component is assembled by the appropriate assembly task. All nodes together form a recursive product structure tree [25]. Different end products can share the same modules as long as functional requirements and cost-effectiveness persist [34].

Also, a bill of material (BOM) is typically part of the product structure. Parts and components that constitute the product in the context of an assembly, subassembly, or model are listed within a BOM [35]. In the automotive industry, BOMs are considered an integral part of the product representation. Information on components (e.g., compressor, cable, etc.) that are necessarily installed in a product to implement a function (e.g. climate control system) can be looked up in BOMs [20]. A similar, but more detailed specification of BOMs can be found in [36]: Brière-Côté describes a BOM as a list of subassemblies, components, parts, and raw materials which is applied to construct higher-level assemblies. This list illustrates the type and quantities of each to build a finished product.

Newer approaches to depict product structures are based on ontologies and semantic networks. An ontology defines a uniform vocabulary for researchers who need to exchange information in a particular field; an ontology allows inter alia reuse and analysis of knowledge [37]. In contrast, a semantic network is a graphical representation of knowledge. Semantic networks are realized with the aid of nodes and arcs [38]. An example presents Vegetti et al. [39]. Here, two hierarchies are applied to handle product variants from different angles. The abstraction hierarchy allows to represent product data on various granularity levels to efficiently deal with a high number of variants. The structural hierarchy organizes knowledge related to structural product information and to the BOM.

From development perspective, "Design Structure Matrices" (DSMs) denote a compact representation of product element contexts [25]. DSMs allow a comprehensive presentation of information (elements of any type, i.e., components or process steps) and are therefore suitable for models with many variant features. The DSM is illustrated as a square matrix with the same number of rows as columns to map the relationships of parameters between components. In general, only one type of relationship (e.g., "…is linked to…") per DSM can be defined. Furthermore, for larger systems with several hundred elements, it is difficult to keep an overview and ensure the manageability of the matrix representations [26].

To conclude, a variety of types of product representations is available today. As mentioned before, especially graph structures and here tree structures are an intuitively attractive approach because of their proximity to car design principles. Nevertheless, ontologies and semantic networks offer a perspective for integration of additional information. To allow for a real-time analysis of the feasibility of a planning scenario, an integrated DCM requires the application of smart quantitative methods on a holistic information model to derive future resource requirements from market requirements. Innovative processes and methods for DCM (e.g., approaches of [13][40][41][20] have been evaluated in [42]. None of those approaches unites the criteria of part demand calculation from market predictions, realistic lead time assumptions between market demand and resource demand and process based description for at least a part of DCP (Demand and Capacity Planning).

Based on these findings, a product representation has been developed which merges the concepts of variant trees and ontologies into a holistic information model concept. Furthermore, algorithms, which base on the generation of planned orders to derive part demands have been implemented for this product representation and have been validated in combination at several German OEMs. The respective tool suite is known under the name of OTD-DCM, where OTD refers to the basic instrument OTD-NET (order-to-delivery and network simulator, cf. [20]). The next section presents the underlying concepts and optimization methods that are applied in this approach to reduce the data complexity.

IV. Hierarchichal Product Structure and Optimization Methods used in the DCM

It is necessary to assure consistency and avoid redundancy in and between all data entities when integrating data into one information model. Inconsistencies occur for example when subsets of technical rules or BOM rules contradict each other so that orders cannot be specified fully. In the development cycle of a car, rules are added and revised within different IT systems, thus rules are partially redundant and sometimes even contradictory. Hence, it is necessary to process planning-relevant information regarding structural requirements (syntax and semantics) and to verify their consistency.

As a result, the implemented data processing in OTD-DCM has been based on the principle of generating a hierarchically-linked structure of variant clusters (cf. [43]). Here, a variant cluster contains by definition a subset of allowed vehicle variants (typically car models), that have common properties (example: sales region = Germany, fuel type = diesel, gear type = automatic). Each variant cluster is characterized by its temporal validity, the technical rules and the lists of allowed and forced options which apply for all in the cluster included vehicle variants.

Within the hierarchy-linked structure of clusters, each variant cluster inherits all characteristics of his parent cluster.

The first pre-optimization of the product structure comprises the generation of a hierarchical tree where tree levels are based on subsequently detailed variant cluster specifications. The initial tree structure can be derived from automotive engineering principles: Tree levels, which de facto group models, may be based on model characteristics like car series, fuel type, sales region and many more (see Fig. 5). It should be mentioned that the level sequence influences directly the later optimization results and the optimal structure may differ for different applications or different OEM or car series. The sequence applied in this paper has been developed in close communication with all involved departments of the OEM providing the example presented. It is subject to confidentiality and shall not be detailed in this contribution.

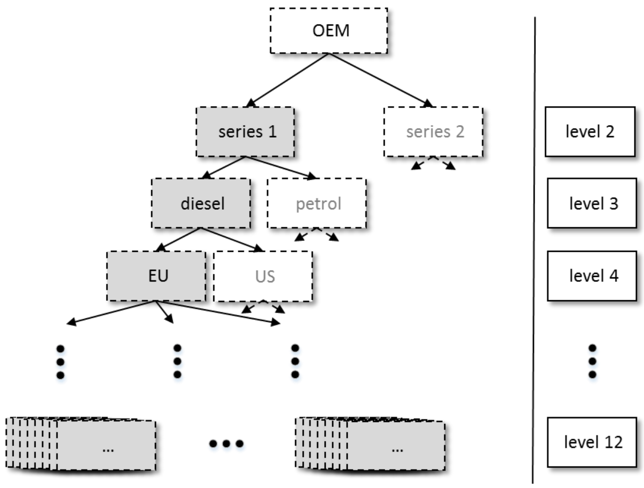


Figure 5. Extract of the generated tree structure

Nevertheless, each tree level can have one or many nodes, depending on the level and type of models clustered (e.g., 90 kW or 150 kW for the engine nodes). Options, technical and BOM rules but also volumes, quotas and restrictions are related to the tree nodes in a last step to obtain a holistic DCM information model.

Among all integrated information, rules are identified as the most complex data fragment. Technical rules represent the technical feasibility by Boolean expressions, e.g., "if engine = 90 kW then transmission = 6-speed manual gearbox". BOM rules follow the same Boolean schema but link options to part demands, e.g., "if engine = 90 kW and radio = "Radio Basic" then parts 5678973 and 5678974". Further, a free definable period is typically specifying the validity of a specific technical or BOM rule. It should be noted that the temporal validity of all data fragments has to be handled within this tree structure [42]. Also, all algorithms working on the product tree have to process the temporal validity.

In the following, the algorithm integrated in this approach will be presented. After initial tree generation all rules are listed on the lowest level; the nodes on this level relate to models one to one. As described in section II, the possible number of BOM rules for a fully specified car amounts to over 600,000 and the number of technical rules to 15,000 per series. The optimization of rules has been subdivided into three subsequent optimization steps. The call sequence of these steps is stated in the following algorithmic code: The procedure starts to identify all points in time within the planning interval where the validity of any rule may change (see Pseudocode 1). Next, the function for the reduction of the number of properties as well as the function for the reduction of the number and length of rules are proceed.

```
// optimize allowed properties and rules
FUNCTION optimizationSteps(variantcluster, originalData)
    LIST timePoints =  get points in time of any changes
                       in originalData
    FOR EACH timePoint IN timePoints
        ARRAY of allowedProperties FROM originalData
        LIST originalRules =  get original rules from
                              originalData
        // function to reduce the amount of properties
        reduceProperties(allowedProperties)
        // function for reduction of number and length
        newRules = reduceNumberAndLength(originalRules)
    END FOR
    RETURN allowedProperties, newRules
END FUNCTION
// recursively move rules upwards
CALL function pullRulesUP WITH masterVariantCluster
               as argument
```

Pseudocode 1: Overview call sequence

The objective of the first optimization step is to identify all forced options, i.e., the options that have necessarily to be chosen for a specific variant cluster (e.g., every car for the German market has necessarily a specific exhaust system). Therefore, principally allowed options for one variant cluster are reduced by excluding non-feasible options.

```
FUNCTION reduceProperties (allowedProperties)
    BOOLEAN reduceAllowedProperties = TRUE
    WHILE (reduceAllowedProperties)
        LIST forcedProperties = calculate forced
                                properties of
                                possible allowed
                                properties
        FOR EACH property IN allowedProperties
            SET property IN forcedProperties
            BOOLEAN valid =  check temporary
                             Configuration against
                             all technical rules
            IF (NOT valid)
                REMOVE property FROM allowedProperties
            ELSE
                reduceAllowedProperties = FALSE
            END IF
        END FOR
    END WHILE
END FUNCTION
```

Pseudocode 2: Reduce properties

This is done by checking intelligently selected, partly specified theoretical configurations against all applicable technical rules (see Pseudocode 2). In a sub-step, fixed options are set, i.e., the ones that define the variant cluster. Afterwards all currently available options are temporarily added. If a contradiction occurs, the option will be deleted from the set of allowed options. When this process leads to only one possible option from a set of alternative options, this option is set as forced.

An inner inconsistency is identified if an identified forced property violates a technical rule. An outer inconsistency is identified if a positive demand quota for an option has been planned, but the option itself is technically not allowed. Another outer inconsistency is identified, if the sum of all planned quotas for all allowed options within a subset of alternative options in a specified time period does not equal 100%.

The second optimization step reduces the number and the length of rules by, e.g., application of the Identity Law of the Boolean algebra (cf. [44]). In a first sub-step, the so-called negative normal form is constructed, where negation operators are only occurring directly at variables and not at brackets. Factorization is achieved by counting the occurrences of each sub-term in a current term. Afterwards, the sub-term will be factored out (see Pseudocode 3).

```
FUNCTION reduceNumberAndLength(originalRules)
    LIST newRules
    FOR EACH rule IN originalRules
        BUILD negation normal form of rule
        REMOVE tautologies
        DO WHILE (successful optimizations)
            REMOVE redundant structures and expressions
                    from the tree
            FACTOR identical subterms out
            REMOVE redundant expressions in child nodes
            REMOVE constants
        END DO
        ADD newRule TO newRules
    END FOR
    RETURN newRules
END FUNCTION
```

Pseudocode 3: Reduce number and length

It should be noted that these steps are valid only for one variant cluster and a specified, fixed time period. Consequently, these steps need to be executed for each variant cluster and all relevant time periods. In a next sub-step, the OTD-DCM implementation shortens rules by merging similar rules that belong to more than one resource, i.e., workstations, assembly lines and more [42][23]. Next, the algorithms aim to further reduce the actual length of all rules by Boolean simplification of terms. If the optimized length of the rule is shorter than the original one, it is replaced by the new representation. Example: The Boolean expression "$\neg$ ($\neg$A $\wedge$ B $\wedge$ $\neg$C)" will be reduced to "$\neg$B $\vee$ A $\vee$ C". This simplification does not only reduce the amount of data, but allows the following step to identify identical rules.

The third and last optimization step tries to identify commonalities for nodes in the hierarchical product structure. For example, the rules that are valid for each child node of one variant cluster are moved upwards to the parent node, i.e. variant cluster, and deleted from all children. The preliminary condition for this step is that all derived variant clusters share this rule over the same time period. Example: The forced option "Owner's manual in German language" may be valid for all variant clusters within the sales region = Germany. Hence, it can be transferred upwards to the variant cluster "variants - German" [42]. The last pseudocode (Pseudocode 4) represents this step including a method to limit intervals of the rules: If the interval of a rule contains the whole validity interval of the variant cluster, the interval of the rule will be adapted.

```
FUNCTION recursivelyMoveRulesUpwards(variantcluster)
    FOR EACH child OF variantcluster
        recursivelyMoveRulesUpwards(child)
    END FOR
    //move identic rules upwards
    GET valid interval for variantcluster
    LIMIT interval of rules
    LIST rulesToMoveUpwards
    GET rulesOfChildren FROM all rules
    FOR EACH rule IN rulesOfChildren
    ADD rule TO rulesToMoveUpwards
        FOR EACH child OF variantcluster
            GET allRules for child
            IF allRules CONTAINS rule
                // rule can be pulled up
            ELSE
                REMOVE rule FROM rulesToMoveUpwards
            END IF
        END FOR
    END FOR
    FOR EACH child OF variantcluster
        REMOVE rulesToMoveUpwards from rules
        of the child
        REMOVE rules with invalid interval
    END FOR
    GET valid interval of variantcluster
    REMOVE rules with invalid interval
    from rulesToMoveUpwards
    ADD rulesToMoveUpwards TO rules of variantcluster
END FUNCTION
```

Pseudocode 4: Recursively move rules upwards

Concluding, the described optimization process eliminates redundancies and identifies inconsistencies within the integrated information model. In the next section will be shown that this leads to a significant reduction of data complexity in relation to the data entities and thus dependencies.

V. ANALYSIS OF COMPLEXITY REDUCTIONS

The evaluation of the previously described optimization steps has been analyzed in a first step for real data of one middle class series [1] of a German OEM. In addition, to provide greater insights into the effects of the optimization steps, the full product spectrum of this OEM, which consists of currently 54 series has been analyzed in a second step. As BOM rules follow the same principles as technical rules, the illustration in this contribution is limited on BOM rules only.

In the following, a tree node represents a variant cluster as described in the previous section. The parameter $n(l)$ is defined as the number of tree nodes on a level (as mentioned before, e.g., fuel type). The respective sum of BOM rules before optimization is defined as $r^{pre}(l)$ and after optimization as $r^{post}(l)$. The number of average rules per tree node within a level is defined as

$$a^{pre}(l)= r^{pre}(l) / n(l) \quad (1)$$

and

$$a^{post}(l)= r^{post}(l) / n(l). \quad (2)$$

A null-entry rule characterizes a rule without condition, i.e., this rule is valid for the whole variant cluster. The total number of null-entry rules on a specific level l before optimization is defined as $v^{pre}(l)$ and on a specific level l after optimization as $v^{post}(l)$.

As described before, the first evaluation is based on one middle class series, which is mirrored in $n(2)=1$, i.e., the number of nodes on level 2 relate to this series. The built hierarchy structure consists of 12 levels and 819 nodes on all levels.

Table I illustrates in the third column that the lowest level of the hierarchical tree structure contains all existing BOM rules $r^{pre}(l)$ before all optimization steps. Levels 1 to 11 do not contain any rules because these levels have been generated artificially in the first pre-optimization step in order to construct the primary tree structure. After optimization, a significant proportion of BOM rules has been hoisted to higher levels resulting in $r^{post}(l)$.

Furthermore, the overall number of rules is reduced from 3,688,514 to 284,219, which amounts to a reduction of 92.3% in relation to the original number.

The reduction as well as the average ratio of rules per node are comparable by columns $a^{pre}(l)$ and $a^{post}(l)$. The weighted average considers the number of nodes of the whole tree per level, where the reduction in this case also results in 92.3%.

TABLE I. INDICATORS WITHOUT OPTIMIZATION (PRE) AND WITH OPTIMIZATION (POST)

level l	$n(l)$	$r^{pre}(l)$	$r^{post}(l)$	$a^{pre}(l)$	$a^{post}(l)$	$v^{pre}(l)$	$v^{post}(l)$
1	1	0	1,568	0	1,568	0	1,208
2	1	0	0	0	0	0	0
3	2	0	918	0	459	0	126
4	2	0	0	0	0	0	0
5	9	0	8,525	0	948	0	1,867
6	9	0	0	0	0	0	0
7	14	0	2,918	0	208	0	572
8	26	0	13,767	0	530	0	2,399
9	31	0	4,745	0	153	0	1,691
10	35	0	4,418	0	126	0	555
11	54	0	6,154	0	114	0	1,856
12	635	3,688,514	241,206	5,809	380	965,379	18,537
	sum	**sum**	**sum**	**weighted average**	**weighted average**	**sum**	**sum**
	819	3,688,514	284,219	4,504	347	965,379	28,811

This analysis of one middle class series illustrates the immense complexity reduction by application of the OTD-DCM hierarchical tree structure.

When a specific variant cluster at lowest level is regarded (for example, for generation of fully specified planned orders) it is necessary to take into account all valid rules for this specific node, because rules at parent nodes are valid for all child nodes. Thus, the rules on the upper levels need to be propagated downwards to all child nodes when evaluating the total number (sum) of valid rules for one variant cluster.

TABLE II. PROPAGATED RULES PER VARIANT CLUSTER AT LOWEST LEVEL (LEVEL 12)

propagated rules - level 12	pre-optimization	post-optimization
sum	3,688,514	2,672,905
average ratio	5,806	4,215
median	6,671	4,424
minimum	0	2,408
maximum	7,620	5,943

Table II shows the number of propagated rules on the lowest level. As a positive side effect, the reduction of the overall number of rules for the car series in focus of this analysis amounts to 27.5%.

Since only a small information model of one car series has been considered here, an analysis of a full product spectrum is of interest to provide greater insights into the effects of the optimization steps. The full product spectrum of the analyzed OEM consists of 54 car series. The results of the full spectrum analysis are presented in Table III and generally confirm the scale of the reduction. The overall number of rules could be reduced from 67,668,544 to 6,575,089. Concluding, this results in a reduction of 90.3%.

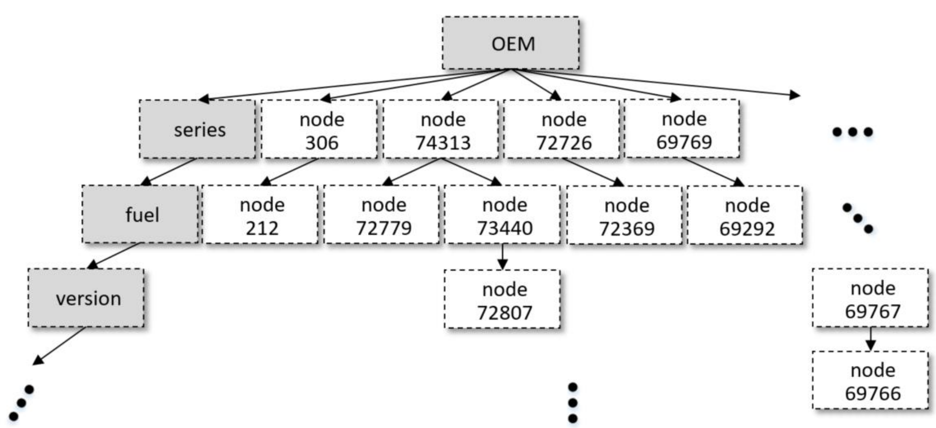
Compared to the first results, it is interesting to note that there are no rules, which could be moved upwards to the first level. Nevertheless, it is of course not surprising that no rules exist, which are valid for every car in the spectrum of the OEM. This again reflects the significance of chosen hierarchy levels and their respective order.

TABLE III. INDICATORS WITHOUT OPTIMIZATION (PRE) AND WITH OPTIMIZATION (POST) FOR FULL SPECTRUM

level l	$n(l)$	$r^{pre}(l)$	$r^{post}(l)$	$a^{pre}(l)$	$a^{post}(l)$	$v^{pre}(l)$	$v^{post}(l)$
1	1	0	0	0	0	0	0
2	54	0	92,017	0	1704	0	59,811
3	98	0	63,145	0	644	0	26,328
4	150	0	97,377	0	649	0	30,102
5	360	0	309,627	0	860	0	105,810
6	360	0	0	0	0	0	0
7	510	0	145,359	0	285	0	37,739
8	823	0	434,216	0	528	0	69,239
9	1083	0	196,330	0	181	0	68,823
10	1246	0	175,603	0	141	0	32,884
11	1307	0	17,953	0	14	0	5,226
12	11,810	67,668,544	5,043,462	5,730	427	15,022,405	677,528
	sum	**sum**	**sum**	**weighted average**	**weighted average**	**sum**	**sum**
	17,802	67,668,544	6,575,089	3,801	369	15,022,405	1,113,490

To gain more insights into this effect a door hinge as a part (named here P24139) has been traced through the optimization steps. Upon start, the rules that refer to this part are linked to nodes on the lowest level (level 12) of the product tree. There are 2711 variant clusters referencing this part. After execution of all optimization steps, the number of rules is reduced to 40. An extract of the results is mapped in Fig. 6, which displays those nodes (named N306, N74313,…) that contain the corresponding part for the first five levels of the tree. On levels seven to nine four more door hinge rules exist which are not illustrated here. Referring to the rules that are hoisted upwards, the situation that both – parent and child – nodes refer to the same rule (cf. node 306 on level 2 and node 212 on level 3 in Fig. 6) is valid, because time period for parent and child may differ. The specific rule on child level is so only valid for this specific variant cluster. This example demonstrates the effectiveness of the complexity reduction.

Also, for the full product spectrum, a reduction in propagated rules on the lowest level may be noted as a positive side effect. Whereas the overall reduction amounts to 7.7%, this effect varies widely among the car series: for a high-class car series a reduction of 49% is realized, but for other car series the number of propagated rules have increased slightly caused by the interval split. This indicates a starting point for further optimizations on OEM side as well as algorithmically.



level	No. of nodes referring to door hinge (P24139) rules
1	0
2	13
3	5
4	5
5	13
⋮	⋮

Figure 6. Trace of door hinge as a part after execution of optimization steps

VI. Conclusion and Future Work

An integral component of the automotive supply chain management is DCM, where resource requirements resulting from future or already realized market demands are synchronized with capacities and restrictions of the supply chain and production system. Because it is impossible to predict the exact future vehicle orders, part demand is typically gradually derived from sales forecasts in a number of sequential processes involving a variety of systems as well as experienced human planners. A transparent, lean but holistic product representation plays a key role for the effective facilitation of this DCM process.

This paper has given an overview of different types of product structures. Especially, the tree structure is an intuitively attractive approach because of its proximity to car design principles. Newer approaches based on ontologies and semantic networks complementary benefits. Thus, the integration of distributed DCM data into an extended hierarchical tree structure has been analyzed against the background of complexity reduction.

This study did not only perform the analysis for one middle class series, but moreover for the full product spectrum of an OEM. For a better understanding of the results, the applied algorithms have been described in form of pseudo code. It has been demonstrated that by choosing a hierarchical tree structure the total number of BOM rules could be reduced by a factor of 10 (reduction of nearly 90%) whereas the number of BOM rules related to a specific variant cluster (i.e., propagated rules) decreases as well. In contrary, this number can often be decreased massively in parallel by elimination of surplus information. As a door hinge has been traced to visualize the result of the optimization steps and to demonstrate how rules are reduced, merged and hoisted upwards within the tree structure.

In summary, the hierarchical integrated information model provides more transparency as redundant and surplus information is dramatically reduced. Thus, it proves to be an enhanced basis for a scenario-based DCM planning process for the automotive industry, which relies on transparent and consistent data. A sound DCM process will increase program stability and transparency on future part demand. Bottlenecks and the resulting deterioration of delivery service levels will be decreased. Furthermore, if the mentioned applications use the information model, it will save computation time and memory space [42].

Nevertheless, the complexity of the car as a product increases more and more. Trends like embedded systems and e-mobility are not yet considered in full within the product structures. New dependencies of technical and electronical components and the compatibility between hardware and software will change the car architecture and therefore influence logistics and thus the DCM process. Thus, this information needs to be integrated into the product representation in the near future.

Even more so, when targeting an integrated product structure, further product characteristics from other departments like sales or productions may need to be taken into account. In consequence, it is believed that a more generalized graph structure instead of the applied tree structure may hold further benefits in terms of complexity reduction. Against this background, generic graph structures shall be analyzed by the authors in the near future.

Acknowledgment

Special thanks to the German “Bundesministerium für Forschung und Bildung” (BMBF) for making this study possible by providing funding to the ILogTec project. The funding of BMBF was granted with Funding-ID 03FH008IA5 and 03FH008IB5.

References

[1] K. Pawlikowski, D. Fruhner, K. Klingebiel et al., “Benefits of an Integrated Hierarchical Data Structure for Automotive Demand and Capacity Management,” in ICCGI 2016: The Eleventh International Multi-Conference on Computing in the Global Information Technology, C. Merkle Westphall, K. Nygard, and E. Ravve, Eds., pp. 20–25, ThinkMind, Barcelona, Spain, 2016.

[2] J. Schuberthan and S. Potrafke, “Die Anforderungen des Kunden…,” in Logistik in der Automobilindustrie: Innovatives Supply Chain Management für wettbewerbsfähige Zulieferstrukturen, F. Gehr and B. Hellingrath, Eds., pp. 8–18, Springer Berlin Heidelberg, 2007.

[3] D. Alford, P. Sackett, and G. Nelder, “Mass customisation — an automotive perspective,” International Journal of Production Economics, vol. 65, no. 1, pp. 99–110, 2000.

[4] E.-H. Krog and K. Statkevich, “Kundenorientierung und Integrationsfunktion der Logistik in der Supply Chain der Automobilindustrie,” in Das Beste der Logistik, H. Baumgarten, Ed., pp. 185–195, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[5] S. Meissner, Logistische Stabilität in der automobilen Variantenfließfertigung, Technische Universität München, Garching b. München, 2009.

[6] A. Trojan, “…und die Auswirkungen auf den 1st-Tier-Lieferanten,” in Logistik in der Automobilindustrie: Innovatives Supply Chain Management für wettbewerbsfähige Zulieferstrukturen, F. Gehr and B. Hellingrath, Eds., Springer Berlin Heidelberg, 2007.

[7] R. T. Yu-Lee, Essentials of capacity management, Wiley, New York, 2002.

[8] E. H. Krog, G. Richartz, R. Kanschat et al., “Kooperatives Bedarfs- und Kapazitätsmanagement der Automobilhersteller und Systemlieferanten,” Logistik-Management : internationale Konzepte, Instrumente, Anwendungen ; Zeitschrift der Kommission Logistik im Verband der Hochschullehrer für Betriebswirtschaft e.V., 2002.

[9] D. Arnold, H. Isermann, A. Kuhn, et al., eds., Handbuch Logistik, Springer, Berlin, 2008.

[10] F. Klug, Logistikmanagement in der Automobilindustrie, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.

[11] J. Gebhardt, H. Detmer, and A. L. Madsen, “Predicting Parts Demand in the Automotive Industry: An Application of Probabilistic Graphical Models,” Proc. Int. Joint Conf. on

Uncertainty in Artificial Intelligence (UAI'03, Acapulco, Mexico), Bayesian Modelling Applications Workshop, 2003.
[12] H. Meyr, "Supply chain planning in the German automotive industry," OR Spectrum, vol. 26, no. 4, 2004.
[13] T. Stäblein, Integrierte Planung des Materialbedarfs bei kundenauftragsorientierter Fertigung von komplexen und variantenreichen Serienprodukten, Shaker, Aachen, 2008.
[14] J. Dörmer, ed., Produktionsprogrammplanung bei variantenreicher Fließproduktion, Springer Fachmedien Wiesbaden, Wiesbaden, 2013.
[15] M. Heitmann, IT-Sicherheit in vertikalen F&E-Kooperationen der Automobilindustrie, Deutscher Universitäts-Verlag, Wiesbaden, 2007.
[16] T. Zernechel, "Gestaltung und Optimierung von Unternehmensnetzwerken — Supply Chain Management in der Automobilindustrie," in Die Automobilindustrie auf dem Weg zur globalen Netzwerkkompetenz, F. J. Garcia Sanz, K. Semmler, and J. Walther, Eds., pp. 367–378, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
[17] L. Herold, Kundenorientierte Prozesssteuerung in der Automobilindustrie: Die Rolle von Logistik und Logistikcontrolling im Prozess „vom Kunden bis zum Kunden", Dt. Univ.-Verl., Wiesbaden, 2005.
[18] A.-W. Scheer, Wirtschaftsinformatik: Referenzmodelle für industrielle Geschäftsprozesse, Springer, Berlin, 1997.
[19] H. Barthel, Modell zur Analyse und Gestaltung des Bestellverhaltens für die variantenreiche Serienproduktion, Jost-Jetter, Heimsheim, 2006.
[20] A. Wagenitz, Modellierungsmethode zur Auftragsabwicklung in der Automobilindustrie, Technische Universität Dortmund, 2007.
[21] M. Holweg and F. K. Pil, The second century: Reconnecting customer and value chain through build-to-order ; moving beyond mass and lean production in the auto industry, MIT Press, Cambridge, Mass, 2004.
[22] M. Toth, A. Wagenitz, and K. Klingebiel, "Dynamic Supply Chain Planning with Logistic Assistance Systems," in ASOR Bulletin: The Australian Society for Operations Research, E. Kozan, Ed., 2010.
[23] K. M. Liebler, Eine prozess- und IT-gestützte Methode für die Produktionsplanung in der Automobilindustrie, Verl. Praxiswissen, Dortmund, 2013.
[24] G. Schuh, Produktkomplexität managen: Strategien; Methoden; Tools, Carl Hanser Fachbuchverlag, s.l., 2014.
[25] X. Deng, G. Huet, S. Tan et al., "Product decomposition using design structure matrix for intellectual property protection in supply chain outsourcing," Computers in Industry, vol. 63, no. 6, pp. 632–641, 2012.
[26] M. Kissel, Mustererkennung in komplexen Produktportfolios, Dissertation, Technische Universität München, 2014.
[27] C. Blees, D. Krause, and H. Meerkamm, Eine Methode zur Entwicklung modularer Produktfamilien, TuTech Verl., Hamburg, 2011.
[28] P. Gu and S. Sosale, "Product modularization for life cycle engineering," Robotics and Computer-Integrated Manufacturing, vol. 15, no. 5, pp. 387–401, 1999.
[29] D. Krause, G. Beckmann, S. Eilmus et al., "Integrated Development of Modular Product Families: A Methods Toolkit," in Advances in Product Family and Product Platform Design: Methods & Applications, T. W. Simpson, J. Jiao, Z. Siddique et al., Eds., pp. 245–269, Springer New York, New York, NY, s.l., 2014.
[30] H. Kesper, Gestaltung von Produktvariantenspektren mittels matrixbasierter Methoden, Verl. Dr. Hut, München, 2012.
[31] G. Schuh and U. Schwenk, Produktkomplexität managen: Strategien - Methoden - Tools, Hanser, München, Wien, 2001.
[32] G. Schuh, Gestaltung und Bewertung von Produktvarianten: Ein Beitrag zur systematischen Planung von Serienprodukten, 1988.
[33] Y. Zeng and P. Gu, "A science-based approach to product design theory Part II: Formulation of design requirements and products," Robotics and Computer-Integrated Manufacturing, vol. 15, no. 4, pp. 341–352, 1999.
[34] K. Fujita, "Product variety optimization under modular architecture," Computer-Aided Design, vol. 34, no. 12, pp. 953–965, 2002.
[35] J. H. Lee, S. H. Kim, and K. Lee, "Integration of evolutional BOMs for design of ship outfitting equipment," Computer-Aided Design, vol. 44, no. 3, pp. 253–273, 2012.
[36] A. Brière-Côté, L. Rivest, and A. Desrochers, "Adaptive generic product structure modelling for design reuse in engineer-to-order products," Computers in Industry, vol. 61, no. 1, pp. 53–65, 2010.
[37] N. Noy and D. L. McGuinness, "Ontology development 101," Knowledge Systems Laboratory, Stanford University, 2001.
[38] V. López-Morales and O. López-Ortega, "A distributed semantic network model for a collaborative intelligent system," Journal of Intelligent Manufacturing, vol. 16, 4-5, pp. 515–525, 2005.
[39] M. Vegetti, H. Leone, and G. Henning, "PRONTO: An ontology for comprehensive and consistent representation of product information," Engineering Applications of Artificial Intelligence, vol. 24, no. 8, pp. 1305–1327, 2011.
[40] S. Ohl, Prognose und Planung variantenreicher Produkte am Beispiel der Automobilindustrie, VDI-Verl., Düsseldorf, 2000.
[41] H. Wagner, Kollaboratives Bedarfs- und Kapazitätsmanagement am Beispiel der Automobilindustrie: Lösungsansatz zur Sicherstellung der Wandlungsfähigkeit, Huss, München, 2006.
[42] A. Wagenitz, K. M. Liebler, and S. Schürrer, "A Holistic Approach to Demand and Capacity Management in the Automotive Industry," in Innovation in product and production: July 31 - August 4, 2011 in Stuttgart, Germany ; conference proceedings ; 21th International Conference on Production Research, ICPR 21, D. Spath and R. Ilg, Eds., Fraunhofer Verlag, Stuttgart, 2011.
[43] K.-U. Meininger, Abstraktionsbasierte Bereitstellung bereichsübergreifender Planungsdaten für die Produktionsplanung bei Serienfertigung variantenreicher Erzeugnisse, Idstein, 1994.
[44] R. L. Goodstein, Boolean Algebra, Dover Publications, Newburyport, 2012.