Multi-Valued Decision Diagrams for the Verification of Consistency in Automotive Product Data

Rüdiger Berndt, Peter Bazan, Kai-Steffen Hielscher, Reinhard German
Computer Networks and Communication Systems
Friedrich-Alexander University of Erlangen-Nürnberg, Germany
{berndt, peter.bazan, hielscher, german}@informatik.uni-erlangen.de

Martin Lukasiewycz
TUM CREATE
Singapore
martin.lukasiewycz@tum-create.edu.sg

***Abstract*—Highly customizable products and mass customization — as increasing trends of the last years — are mainly responsible for an immense growth of complexity within the digital representations of knowledge of car manufacturers. We developed a method to detect and analyze inconsistencies by employing a Multi-Valued Decision Diagram (MDD) which is used to encode the set of all valid product configurations. On this basis, we stated a number of rules of consistency that are checked by a set-based verification scheme.**

***Index Terms*—multi-valued decision diagram; product configuration; knowledge representation; digital manufacturing; formal verification**

I. Introduction

The development of new markets along with an increasing demand for customizable products, results in a continuously growing complexity within the business processes and information flows of car manufacturers.

The configuration data contains information on the product model range, the respective derivatives and all available equipment features and serves as basis for the communication between customers and enterprise. It also incorporates the knowledge about valid and invalid product configurations.

Any information about parts, sub-assemblies and their mutual structural relationships is contained in the Bill Of Materials (BOM) — it constitutes a common basis for the planning of logistic and production processes.

The final assembly plan incorporates the structure of the final assembly line; every work station contains a set of manufacturing steps which include information on handling gear, parts, work methods, time requirements and control information regarding the Human-Machine-Interfaces.

Any of the above information is developed during different stages within the product development cycle and by various business units. In order to detect and analyze inconsistencies within the information and their interplay, we develop a verification method that is based upon the set of valid product configurations which is represented by a Multi-Valued Decision Diagram (MDD).

II. Related Work

In [1], SAT-checking — based upon a set of binary constraints — is employed to ensure consistent automotive product configuration data. The article provides a formal model of the involved business processes and the configuration data as well as a modification of the underlying SAT-solver.

In model checking the introduction of SAT-solvers made it possible to solve large problems where Binary Decision Diagrams (BDDs) would exhaust the available memory [2].

A method to solve the problem of interactive configuration is presented in [3]. In order to avoid time-consuming search effort in the configuration process, a BDD representing the space of solutions, i.e. valid configurations, is precompiled.

In [4], an overview of AND/OR Multi-Valued Decision Diagrams (MDDs) as data structure for constraint networks is given.

Approximate constraint compilation to keep the structure of MDDs small is introduced by [5]. The size of decision diagrams can be influenced by the variable ordering — i.e. the arrangement of layers within the diagram. This topic is covered in many scientific articles. Unfortunately, finding an optimal ordering for BDDs is known to be NP-complete [6].

Compiling real-world automotive configuration problems using MDDs has been proposed in [7]. A new algorithm combining dynamic variable ordering techniques along with adaptive constraint compilation sequence is introduced. In order to detect slow moving items within the bill of materials, a verification scheme — based on the set of valid configurations and an intersection operation — is established. However, in this article we will focus on the formalization of new automotive product data and an extension of the rules of consistency.

III. Product Data

This section introduces the formal model of the product data. An overview of the relevant data entities can be found in Fig. 1.

A. *Product Configuration Data*

Each configuration, i.e. customer order, is encoded by a set of distinct attributes. A partition on the set of attributes is induced because subsets of attributes encode similar features — e.g. the color can either be blue, red or green.

Let the finite, non-empty set of distinct symbols $A := \{a_1, \dots, a_n\}$ be referred to as the set of *attributes*. Moreover, let a partition $F := \{F_1, \dots, F_m\}$ of A be given. F is referred to as the set of *features*. For a given A and F we define

$$W := F_1 \times \cdots \times F_m \tag{1}$$

to be the set of *configurations*. Let $w \in W$ be given, the mapping function $S : F_1 \times \cdots \times F_m \to A$ yields a set containing all attributes of w.

Constraints are used to restrict the combination of attributes like, for example, a diesel engine that cannot be combined with automatic gear. A *constraint* is an m-tuple, $c := (P_1, \dots, P_m),\ P_i \subseteq F_i$, which induces a set

$$W(c) := P_1 \times \cdots \times P_m. \tag{2}$$

The conjunction of constraints is defined by a *composite constraint*, $\hat{c} = \{c_1, \dots, c_l\}$. Each composite constraint induces a set $W(\hat{c}) := \bigcup_{1 \le i \le l} W(c_i)$.

As from now on we denote the set of *restrictive constraints* within the configuration data by $R := \{c_1, \dots, c_k\}$. Let W and R be given, the set

$$\overline{W}(R) := W \setminus \bigcup_{1 \le i \le k} W(c_i) \tag{3}$$

can be derived, which henceforth will be referred to as the set of *valid configurations*.

B. Technical Information

Information on parts, which are necessary during the production process, is organized in a so-called *Bill Of Materials* (BOM). Apart from the knowledge about the mutual structural relationships — is-part-of relations — each part is linked to a constraint which encodes the set of product configurations where the part is mandatory for the production process. We denote the BOM by $B := \{b_1, \dots, b_l\}$.

Let a configuration w be given — a part b_i is determined to be necessary, iff

$$w \in W(c_{b_i}), \tag{4}$$

where c_{b_i} is the constraint of the according part.

Subsets of parts like, for example, all available variants of steering wheels, are summarized within a *part category*. Each part category is connected to a composite constraint which specifies the set of product configurations where one of the enclosed parts will be assembled — e.g. every customer order with automatic gear, will be equipped with a variant of the according gear selectors. We denote the set of part categories by $P := \{p_1, \dots, p_o\}$. Let a configuration w be given, a part category $p_i \in P$ is said to be mandatory, iff

$$w \in W(\hat{c}_{p_i}), \tag{5}$$

where $\hat{c}_{p_i}$ is the composite constraint of the part category.

In order to determine the parts which are enclosed by a part category, we state the following function.

$$B(p_j) = \{b_{j_1}, \dots, b_{j_n}\}. \tag{6}$$

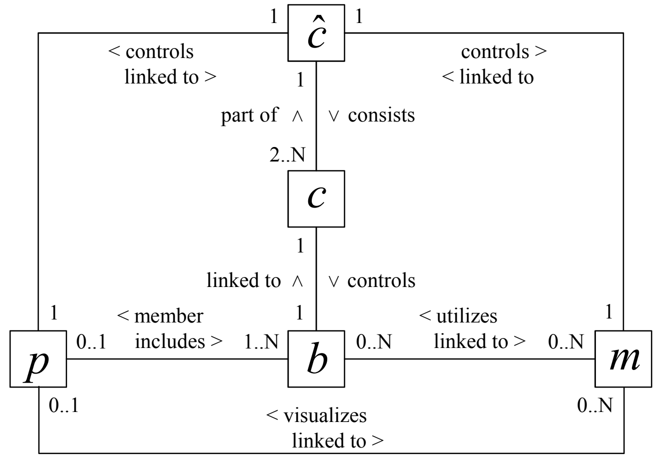

Fig. 1. Entity-relationship model of automotive product data

C. Final Assembly Plan

The third component of the product data bears the structural information of the final assembly line. It incorporates a set of *manufacturing steps* whose application is controlled by composite constraints (cf. part category). Henceforth, we denote the set of manufacturing steps by $M := \{m_1, \dots, m_p\}$. Let a configuration w be given, a manufacturing step $m_i \in M$ is said to be performed, iff

$$w \in W(\hat{c}_{m_i}), \tag{7}$$

where $\hat{c}_{m_i}$ is the composite constraint of the manufacturing step. All manufacturing steps that are linked to a part category are yielded by the following mapping function

$$M(p_j) = \{m_{j_1}, \dots, m_{j_n}\}. \tag{8}$$

Finally, we provide a mapping function which yields all parts that are referenced by a manufacturing step

$$B(m_j) = \{b_{j_1}, \dots, b_{j_n}\}. \tag{9}$$

IV. Rules of Consistency

A. Consistency of Product Configuration Data

Whenever constraints mutually exclude each other, the set of valid product configurations will be empty. Thus the set of restrictive constraints is said to be inconsistent, iff

$$\overline{W}(R) = \emptyset. \tag{10}$$

In order to guarantee that all attributes of the configuration data are admissible, the following rule is introduced.

$$\forall a_i \in A : \exists w \in \overline{W}(R), \quad \text{such that} \quad a_i \in S(w). \tag{11}$$

It ensures that each $a_i \in A$ is contained in *at least one* valid configuration $w \in \overline{W}(R)$.

B. Slow-moving Items

A part will never be used during the production process when there is no valid configuration where it is determined to be mandatory. Therefore, a part is said to be a *slow-moving item*, if it infringes the following proposition

$$\forall b_i \in B : W(c_{b_i}) \cap \overline{W}(R) \neq \emptyset. \tag{12}$$

C. Part List Completeness

The admissibility of each part category is guaranteed by

$$\forall p_j \in P : W(\hat{c}_{p_j}) \cap \overline{W}(R) \neq \emptyset. \tag{13}$$

Each valid product configuration that is contained in $W(\hat{c}_{p_j})$ will be equipped with *exactly* one part of the corresponding part category.

First, we ensure that there is no valid configuration which won't be equipped with any of the enclosed parts.

$$\forall p_j \in P : \left[W(\hat{c}_{p_j}) \setminus \bigcup_{1 \leq i \leq n} W(c_{b_{j_i}}) \right] \cap \overline{W}(R) = \emptyset \tag{14}$$

Next, the enclosed parts' constraints need to be pairwise disjoint. The following equation needs to hold for all l, k, where $1 \leq l, k \leq n$ and $l \neq k$

$$\left[W(c_{b_{j_l}}) \cap W(c_{b_{j_k}}) \right] \cap \overline{W}(R) = \emptyset \tag{15}$$

D. Manufacturing Step Validity

A manufacturing step that is never applied is detected by the following equation.

$$\forall m_j \in M : W(\hat{c}_{m_j}) \cap \overline{W}(R) \neq \emptyset \tag{16}$$

Whenever parts are affected by a high variance — e.g. gear selectors — the operator needs to be supported during the assembly process. Each manufacturing step, therefore, may contain information to guide the assembly process by being connected to a part category.

For all part categories $p_j \in P$ and the according manufacturing steps $m_{j_i} \in M(p_j)$, the following equation needs to hold.

$$W(\hat{c}_{p_j}) \cap \overline{W}(R) = \bigcup_{1 \leq i \leq n} W(\hat{c}_{m_{j_i}}) \cap \overline{W}(R) \tag{17}$$

Finally, we ensure consistency with respect to the parts and their connection to manufacturing steps. The following equation needs to be satisfied for all $m_j \in M$ and for all $b_{j_i} \in B(m_j)$.

$$W(\hat{c}_{m_j}) \cap \overline{W}(R) = \bigcup_{1 \leq i \leq n} W(c_{b_{j_i}}) \cap \overline{W}(R). \tag{18}$$

V. MDD-based Verification

Let A, F and R be given as shown in Table I and Table II. Constraint c_2, for example, states that all configurations which contain $E1$ and $M4$ or $E2$ and $M4$ are not valid — this means that customers are not able to choose a combination of either 2.0 l TDI and 4-wheel or 2.5 l Otto and 4-wheel.

TABLE I
FEATURES AND ATTRIBUTES

Feature	Attribute	Denotation
F_E (Engine)	E1	2.0 l TDI
	E2	2.5 l Otto
	E3	Hybrid
F_S (Steering Arrangement)	SL	Left
	SR	Right
F_M (Modes of Drive)	MF	Front
	MR	Rear
	M4	4-wheel
F_P (Power Steering)	PE	Electronic
	PV	Vacuum
	PN	No

TABLE II
RESTRICTIVE CONSTRAINTS

R	c_1	$(\{E3\}, \{SR\}, F_M, F_P)$
	c_2	$(\{E1, E2\}, S_L, \{M4\}, F_P)$
	c_3	$(F_E, F_S, \{MF, M4\}, \{PN\})$

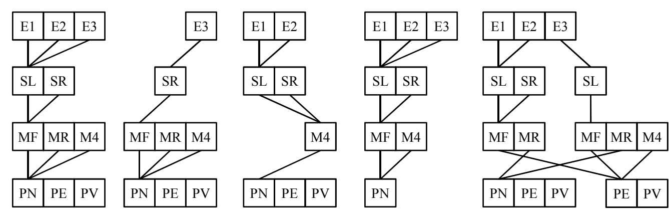

Fig. 2. Representation of W, c_1, c_2, c_3 and $\overline{W}(R)$.

A. Representation of Valid Configurations

First, an MDD representation of W is constructed — each of its layers represents one feature $F_i \in F$. Any edge connects an attribute of the upper layer with a node of the lower layer and hence encodes all combinations of this attribute and the attributes of the respective node. Therefore, each path within the structure — from the top layer to the bottom layer — yields one element of W (Fig. 2).

Next, each constraint $c_i \in R$ is encoded in a similar way, where the layers represent the sets $P_1, \ldots, P_m$. Based upon the MDD representations of W and $c_1, \ldots, c_k$ we now gradually apply the minus operation to calculate $\overline{W}(R)$ where the variable ordering of the diagrams need to coincide. During the successive subtraction of $c_i \in R$, the constraints are verified with respect to self-consistency by checking whether the resulting structure is empty after each subtraction step (10).

B. Verification

In general, all verification criteria are based upon the implementation of a *set-intersection*, a *set-minus* and a *set-union* operation, respectively. A precondition regarding their application is that the ordering of variables of the MDD operands coincide. Besides that, the MDD which represents $\overline{W}(R)$ is mandatory for all verification schemes.

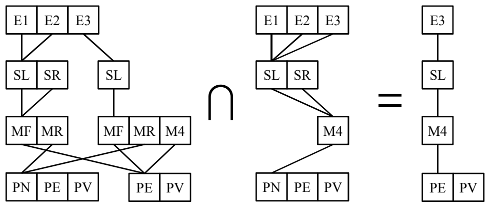

Fig. 3. Admissibility of $M4$.

TABLE III
RESULTS

	Dimensions						
	$\|A\|$	$\|F\|$	$\|R\|$	$\|B\|$	$\|P\|$	$\|M\|$	size [MB]
P1	943	174	4679	19023	775	2730	3.79
P2	1240	181	4413	21947	695	1508	48.32

For example to ensure the admissibility of each attribute (11), first the set $\overline{W}(R)$ is compiled — if the resulting set is empty, the restrictive constraints are inconsistent (10). Next, for each attribute a constraint is constructed that omits all attributes of the respective feature except for the selected attribute; any other feature is fully occupied. After that, each of the constraints is intersected with $\overline{W}(R)$. If the result is empty, the respective attribute is not admissible within the set of valid configurations (see Fig. 3).

For the verification of the part list completeness (14), the composite constraint of each part category is encoded by using a union operator — usually a composite constraint consists of about 4 simple constraints, thus the resulting structure is small. Next, the constraints of each part $b_{j_i} \in B(p_j)$ are gradually subtracted from the set $W(\hat{p}_j)$. Finally, the remaining set of product configurations is intersected with $\overline{W}(R)$.

The verification of the BOM, regarding slow-moving items (12), uses the intersection operation — thus, the constraint of each part needs to be represented as decision diagram. The intersection operation is carried out and the resulting set bears all valid configurations, where the part is necessary for the manufacturing process. If it is empty, the part is said to be a slow-moving item.

VI. Results

First, we introduce the product data with respect to 2 different models in Table III. The columns provide information about dimensions with respect to the corresponding product data and the memory consumption of the decision diagram, $\overline{W}(R)$. All computations have been carried out by using a 2.8 GHz Intel Core 2 Quad Q9550 processor with 8 GB of RAM.

The compilation time of $\overline{W}(R)$ is 13.11 (P1) and 47.83 (P2) seconds. On behalf of all rules of consistency, we have chosen to provide the timing requirements of equation (12) because it needs the most computational effort — 21947 intersection operations for P2; the time for the verification of the parts is 3.29 (P1) and 40.81 (P2) seconds.

VII. Conclusion

In this article we introduced the formal model of the product data of our application domain, i.e. car manufacturing — moreover, we stated a set of rules to ensure consistency. Due to the structural properties within the automotive product data, we motivated the usage of Multi-Valued Decision Diagrams to efficiently encode sets of product configurations. Based on the rules of consistency and the MDD approach, we introduced the verification scheme which implements a number of basic set operations, like intersection or union. Finally, we provided the dimensions of the product data and the computational effort of the verification. Apart from verification, precompiled configuration spaces are used in other applications like, for example, interactive design or property checking; whenever a certain query or operation yields subsets of configurations, like in equation (14), the MDD approach is well suited.

The timing requirements for the compilation of $\overline{W}(R)$ heavily depends on its memory consumption; the more memory is consumed by the MDD, the more time is needed for the construction and the verification. But finding an optimal variable ordering to minimize its size is NP-complete. Therefore, structural knowledge of the problem domain can be explored to compute a good apriori variable ordering [8] [9]. Besides, the development of dynamic reordering methods which combine the knowledge of the size along with the structural information of the constraint network might lead to further improvements.

References

[1] C. Sinz, A. Kaiser, and W. Küchlin, "Formal methods for the validation of automotive product configuration data," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 17, p. 2003, 2003.

[2] E. Clarke, A. Biere, R. Raimi, and Y. Zhu, "Bounded Model Checking using Satisfiability Solving," *Formal Methods in System Design*, vol. 19, no. 1, pp. 7–34, 2001.

[3] T. Hadzic, S. Subbarayan, R. M. Jensen, H. R. Andersen, J. Møller, and H. Hulgaard, "Fast backtrack-free product configuration using a precompiled solution space representation," in *PETO Conference, DTU-TRYK*, 2004, pp. 131–138.

[4] R. Mateescu and R. Dechter, "And/or multi-valued decision diagrams for constraint networks," in *Proceedings of the Thirteenth International Conference on Principles and Practice of Constraint Programming*. Springer, 2007, pp. 498–513.

[5] T. Hadzic, J. N. Hooker, B. O'sullivan, and P. Tiedemann, "Approximate compilation of constraints into multivalued decision diagrams," in *Principles and Practice of Constraint Programming*, 2008, pp. 448–462.

[6] B. Bollig and I. Wegener, "Improving the Variable Ordering of OBDDs Is NP-Complete," *IEEE Trans. Comput.*, vol. 45, pp. 993–1002, September 1996. [Online]. Available: http://dx.doi.org/10.1109/12.537122

[7] R. Berndt, P. Bazan, and K.-S. Hielscher, "Mdd-based verification of car manufacturing data," in *Proceedings of the 3rd international conference on Computational Intelligence, Modelling and Simulation*, 2011.

[8] N. Narodytska and T. Walsh, "Constraint and variable ordering heuristics for compiling configuration problems," in *Proceedings of the 20th international joint conference on Artifical intelligence*, 2007.

[9] R. I. Siminiceanu and G. Ciardo, "New metrics for static variable ordering in decision diagrams," in *TACAS*, ser. LNCS, vol. 3920:90104. Springer, 2006, pp. 328–342.