Managing Complex Data for Electrical/Electronic Components: Challenges and Requirements

Julian Tiedeken[1], Joachim Herbst[2], and Manfred Reichert[1]

[1]Institute of Databases and Information Systems, Ulm University, Germany
{julian.tiedeken, manfred.reichert}@uni-ulm.de
[2]Daimler AG, Germany
joachim.j.herbst@daimler.com

Abstract: In the automotive domain, innovation is driven by the introduction and continuous improvement of electrical and electronic (E/E) components (e.g. sensors, actuators, and electronic control units). This trend is accompanied by increasing complexity of interdependencies between these E/E components. In addition, external impact factors (e.g. changes of regulations) demand for more sophisticated E/E product data management (E/E-PDM). Since E/E product data is scattered over distributed heterogeneous IT systems, application-spanning use cases (e.g. consistency of artifacts, plausibility of logical connections between electronic control units) are difficult to realize. Consequently, the partial integration of the corresponding application data models becomes necessary. Evolution of application data models is common in the context of E/E-PDM, but not considered by existing application integration approaches. Furthermore, no methodology for realizing application integration models exists. This paper elaborates the challenges to be tackled when integrating E/E product data from different applications. It further presents properties of the IT landscape involved in E/E-PDM and reveals occurring problems. Finally, requirements for E/E-PDM are discussed.

1 Introduction

A modern car consists of up to 70 *electronic control units* (ECUs) [MHHR06]. New functionality, product innovation, and customer requirements continuously increase the complexity as well as the number of ECUs in the car. Due to country-, market-, and customer-specific requirements, in addition, a large number of variants of electrical/electronic (E/E) product data exists [HBR08]. In this context, simultaneous engineering, manufacturer-supplier-relationships, and development processes show complex interdependencies (cf. Fig. 1). In particular, E/E product data refers to digital development artifacts, e.g. requirements, circuit diagrams, wiring diagrams, bootloader, flashware, and software. The latter are created and managed by autonomous information systems satisfying different requirements. Legal requirements (e.g., product liability (ISO 26262 [fS11]) and envi-

ronmental regulations (RoHS[1], WEEE[2], EuP[3]) demand for traceability and verification of development decisions. Hence, any change of E/E product data must be documented. For this purpose, E/E product data management (E/E-PDM) systems maintain configurations that bundle interrelated E/E products occuring in all development phases. For example, basic information include the developer or supplier of an E/E product (e.g., ECU, sensor, or actuator), change requestor, and change reason.

Each E/E product has a lifecycle that comprises the steps required to create and distribute it, e.g. planning, development, production, and after sales. In particular, product development constitutes an integral part of the product lifecycle. In this context, emerging market trends (i.e. green-mobility, e-mobility), increasing product liability, environmental regulations, tighter integration of suppliers, and shorter development cycles demand for high data quality as well as integrated E/E product development processes. To cope with these challenges, an IT landscape must adequately cope with product changes. In current practice, a multitude of heterogeneous applications are used for accomplishing different development tasks. In turn, this results in numerous artifacts that partially overlap. In particular, ensuring data consistency and data quality (e.g. accuracy, actuality) constitutes a challenging task in such an environment. Especially, use cases requiring data from different sources are difficult to support, due to missing mappings between interrelated artifacts. To realize these use cases, a partial integration of heterogeneous and distributed data models from different applications is required. Usually, these data models apply different concepts for supporting versioning, variability, and aggregation of E/E product data. Furthermore, semantic dependencies between artifacts from different application data models are undocumented. Another challenge in this context is the independent evolution of the data models maintained in different autonomous applications.

Usually, an IT landscape fostering E/E product development has evolved over years and is based on engineering expertise. To create a common integration model in such an environment, a methodology is needed that explicitly considers existing application data models and technologies. Furthermore, standards fostering the exchange and interoperability of product data (e.g. MSR[4], AUTOSAR[5] [AUT11b, AUT11a], ISO 10303) must be supported by the methodology as well.

This paper elaborates problems that occur when partially integrating heterogeneous applications and their data models in the context of E/E-PDM. Based on this, we identify fundamental requirements an IT support for E/E data management must meet in such a setting.

The remainder of this paper is organized as follows: Section 2 discusses problems and requirements for effective E/E-PDM. Section 3 discusses related work along the identified requirements. The paper concludes with a summary in Section 4.

[1] Restriction of Hazardous Substances Directive
[2] Waste Electrical and Electronic Equipment Directive
[3] Energy-using Products Directive
[4] Manufacturer Supplier Relationship
[5] AUTomotive Open System ARchitecture

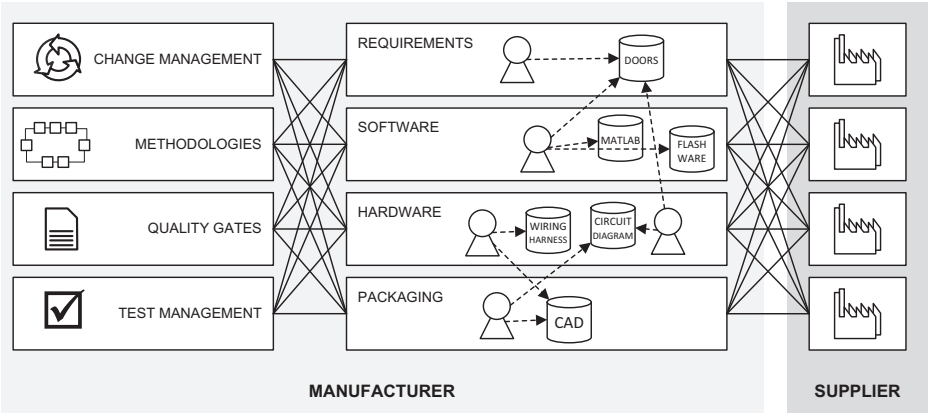

Figure 1: Complexity of E/E product data.

2 Findings, Challenges, Requirements

We analyzed applications involved in E/E development processes and considered several use cases enabling application-spanning consistency checks of E/E product data (e.g. plausibility of logical connections between electronic control units and networks). This section summarizes results of these analyses. It further identifies fundamental challenges emerging in the context of E/E development processes and their IT support. Finally, requirements for holistic and effective E/E-PDM are elicited.

2.1 Challenge 1: Common ontology for data integration

Findings. A main characteristic of E/E development processes is imposed by manufacturer-supplier relationships: Systems and components are developed in close collaboration with suppliers. Systems and sub-systems describe parts of a car characterized by the functions provided. Usually, systems aggregate features perceived by the customer (e.g. heat, ventilation, or air conditioning). Furthermore, they are technically realized based on interacting components (i.e., ECUs, actuators, and sensors). Due to the networked enterprises in a supply chain, for different development stages, deadlines become necessary to guarantee product delivery times. To cope with such a scenario, engineers execute development tasks concurrently (simultaneous engineering) using different applications (cf. Fig. 2). In addition, these applications provide different concepts for enabling versioning, variability, and aggregation of product data. Typically, engineers use numerous applications, hosted and maintained by different departments, to create and maintain E/E product data (e.g. applications for software function modeling, requirements engineering, and computer-aided design). These applications were designed and developed

independently from each other, and their data models reflect long-term experiences of engineers. Furthermore, there are important use cases not covered by the data models of these applications at all. For example, a car usually has ECUs to control functions of installed seats (e.g. position, heat and ventilation). If these ECUs do not differ in their functions, they are, represented by a single model object. To enable these use cases, a number of workarounds are applied, which constitute tacit knowledge of the engineers and are not documented at all. Other use cases, which utilize artifacts from different heterogeneous data sources (e.g. checking plausibility of references between ECU pins and logical signals) are performed manually by responsible actors. The latter collect and aggregate development artifacts from different departments. In particular, this constitutes a cumbersome and time-consuming task during which some of the relevant artifacts may have to be replaced by new versions.

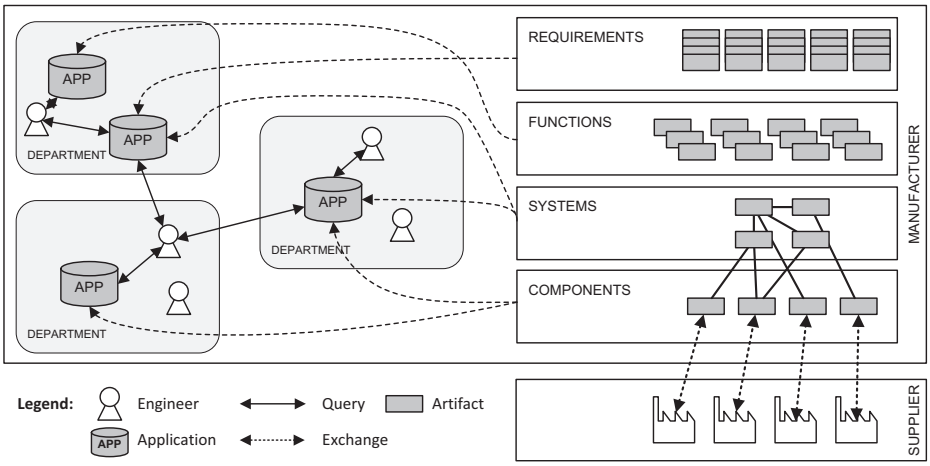

Figure 2: Logical structure of E/E product data.

Challenges. Usually, development tasks are separated into system- and component-specific parts. Both terms are ambiguously defined and used in application data models. For example, while a component in a wiring diagram refers to a variant of an E/E component at a specific geometric position, in the context of requirements documentation, the same term describes all variants of an E/E product in general.

Requirements. For use cases requiring read access to E/E data from different application, a common integration ontology is needed (*Requirement 1*). Such an ontology constitutes the core of any application involved in E/E development processes (cf. Fig. 3). To integrate E/E product data in a consistent and transparent manner, bidirectional mappings between related concepts are needed (*Requirement 2*). As an advantage of such mappings, semantic inconsistencies between application data models can be eliminated. In particular, through mappings changes are made transparent and reproducible. Moreover, integration of development artifacts can be accomplished semi-automatically and, thus, contribute to save

time and reduce errors (as introduced in the context of manual integration). To prevent ambiguities, any common integration ontology must allow modeling artifacts semantics. In addition, application-specific concepts for versioning, variability, and aggregation of E/E product data, as well as documentation methodologies must be considered in the context of a common integration ontology (*Requirement 3*).

2.2 Challenge 2: Consistency of E/E product data

Findings. E/E product data are created, maintained, and used in all stages of the automotive lifecycle (development, production, after sales). Usually, different stakeholders (e.g. E/E architects, system and component responsible, engineers) participate in these lifecycle phases using different applications [MHHR06]. In turn, this results in redundant artifacts and models (e.g. wiring harness, logical connection). In this context, many of the artifacts created in a particular lifecycle phase are required in subsequent development phases. We denote such interdependent applications as tool chains. Generally, development artifacts are reused along these chains to reduce development time. For example, if a new car model series is developed, artifacts of the previous model series serve as basis for product improvements and new ideas. To coordinate development steps, E/E systems and components are developed using the Vee Model [FMC05], which defines necessary steps and responsibilities of all involved parties for the distributed development and testing of artifacts. To control and test required functionality of an entire system, integration tests for the used components become necessary. Hence, quality gates constitute integral parts of any development process [Coo90]. In particular, they describe fixed development stages with predefined quality requirements to be fulfilled at the specified point in time. Generally, development processes involve different stakeholders, departments, and applications. Although global process overviews exist to some degree (e.g. Fig. 3), these have not realized based on workflow technology [RW12]. Changes of an E/E product are common and may impact other E/E products. For example, if the software interface of an ECU changes, logically connected ECUs must be identified and analyzed. For this purpose, global change management processes exist that coordinate and document change requests.

Challenges. Although development is performed simultaneously, artifacts from heterogeneous data sources must be continuously integrated at specific points during development (e.g. design review or prototype analysis). In this context, complex transformations between heterogeneous data models must be defined and maintained. In particular, explicit mappings between artifacts from different data models become necessary. Currently, these mappings are defined manually. Hence, communication between different stakeholders and development departments become necessary, which is error prone and time consuming. Overall, consistency of related artifacts from heterogeneous, distributed data models is a costly, manual task. Another problem is the lack of any technical processes implementation in E/E development, turning the monitoring of E/E product data changes a challenging and costly task; i.e., requests by different stakeholders and departments must be handled manually.

Requirements. The tight integration of OEM and suppliers requires data of high qual-

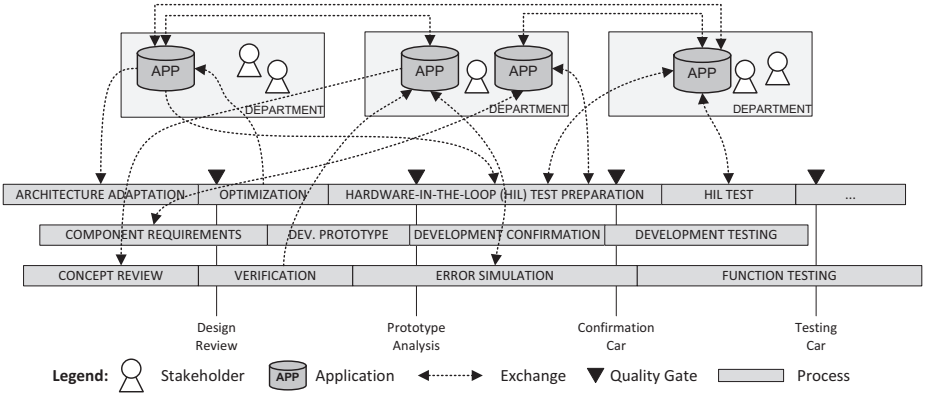

Figure 3: Overview on E/E development processes.

ity as well as standardized processes. For example, during the development of an ECU, functional models and communication information (software ports, frames, and signals) are concurrently created and maintained in different applications. Further, ECU development processes are separated into manufacturer- and vendor-specific parts; e.g vendors realize software for ECUs based on interfaces provided by the manufacturer. Hence, the consistency of artifacts must be guaranteed. Note that any error of a signal, frame, or bit might lead to a malfunctioned product. Hence, consistency management allowing for the identification of inconsistent E/E product data is crucial (*Requirement 4*).

2.3 Challenge 3: Data Model Changes

Findings. In general, changes of application data models are performed autonomously based on department-specific schedules. Although these schedules are distributed across the different E/E development departments, the coordination of data model changes remains a challenging task.

Challenges. Changes of application data models might affect other application data models which are difficult to identify due to undocumented interdependencies between artifacts from different data models. If a change is realized without notifying affected applications, the operation of existing tool chains might be impaired. Hence, before a change of an application data model can be realized, additional data models of other applications must be analysed.

Requirements. To handle data model changes in a more systematic manner, the dependencies between artifacts from different data models must be explicitly modeled and maintained (*Requirement 5*). Usually, data models comprise numerous artifacts. Hence, the manual comparison of all artifact combinations across different data models would be too costly. To reduce this complexity, matching algorithms based on well defined criteria

(e.g. name, data type), are needed. Additionally, data model interdependencies should be analyzed to evaluate the impact of data model changes in advance of their execution (*Requirement 6*). To enable flexible change management, transformations between application data models should be derivable from the dependencies maintained (*Requirement 7*). Hence, adaptations to data model changes can be semi-automated, which contributes to reduce development costs as well as errors.

2.4 Challenge 4: Methodology for documenting E/E product data

Findings. In the automotive domain, AUTOSAR is the de facto standard for developing embedded systems. In this context, the definition of standardized protocols for software development allows exchanging software running on ECUs from different vendors. In addition to the software architecture, methodologies for describing and defining reusable and scalable software components constitute an integral part of AUTOSAR. Besides standardizing ECU software, processes for developing safety related E/E products (e.g. airbag, electronic stability control) are an important component of contemporary E/E-PDM: First, *Failure Mode and Effects Analysis* (FMEA) [Sta03] is used for failure prevention and security management. In particular, FMEA focuses on reducing change costs of artifacts, i.e., on reducing late changes and hence reducing implementation costs. Second, ISO 26262 as emerging standard for functional safety in the automotive domain, needs to be considered. It defines a security lifecycle covering all aspects of development processes. To classify different levels of security requirements, so-called *Automotive Safety Integrity Levels* (ASILs) must be assigned to safety-relevant E/E products. Besides these standards, existing applications define methodologies for repeating tasks and product data documentation.

Challenges. Even though AUTOSAR's integrated methodologies support soft- and hardware, other aspects like wiring harness and connectors of E/E product data are not taken into account. Additionally, tacit knowledge makes the integration of data models a difficult task to accomplish, since semantic inconsistencies might remain unresolved. Finally, an explicit monitoring and control of existing documentation methodologies and guidelines are missing. As a consequence, ambiguities in E/E product data occur.

Requirements. A comprehensive methodology for documenting E/E product data is needed, taking existing standards and methodologies (e.g. AUTOSAR, ISO26262, Automotive SPICE [MHDZ12]) into account (*Requirement 8*). To create a common integration ontology allowing for the integration of existing application data models, a modeling methodology is needed. Note that applications creating and maintaining E/E product data have resulted from long-term experiences. Hence, a modeling methodology must support the creation of a common integration ontology from scratch (top-down) as well as from existing data models (bottom-up) (*Requirement 9*).

Table 1 summarizes the discussed requirements, and associates them along the common integration ontology in Figure 4.

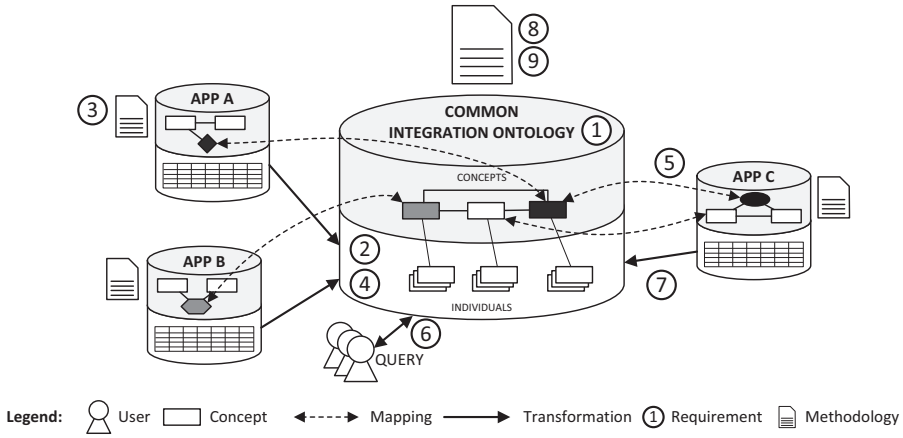

Figure 4: Common integration ontology.

3 Related Work

A common meta-model for tool integration based on EAST-ADL2 and HRC is presented in [Bau10]. This approach focuses on embedded systems engineering and derives concepts for a common meta-model from HRC and EAST-ADL2 meta-models. Although the authors take concepts, like component, part, and port into account, other ones (e.g. requirement, system) are missing. [CCWC09] uses an ontology for integrating distributed product knowledge and proposes a product knowledge service model realized through web services. In addition, a meta-knowledge schema for knowledge modeling is described. Finally, similarity measures for labels and relationships are introduced. The presented approach focus on the integration of heterogeneous, distributed knowledge. Integrating underlying application data schemes is not considered.

[BGN+04] is a model-based software (re-)engineering approach. It integrates tools at the meta-model level and proposes two design patterns for data integration. Further, consistency rules and integration constraints are provided. For this, simple graph grammar rules are defined restricting a meta-model to enable interoperability. Note that this allows for consistency checking as well. Finally, a triple graph grammar is used to model semantic relationships between different meta models. Other approaches using triple graph grammar are presented in [KS06]. ToolNet [FK03] focuses on tool integration and data consistency. For this purpose, consistency relations between reference objects are modeled manually. The approach focuses on requirements specifications, function models, and geometric product data. As a major drawback, changes of application data models are not considered. [NK04] compares the evolution of database schemes with the one of ontologies evolution. In particular, it highlights the differences of both research areas and gives insights into challenges for ontology evolution. These challenges for ontology evolution constitute a starting point for future research on change management of the

No.	Description
1	Common integration ontology.
2	Bidirectional mappings between artifacts.
3	Transparency of data model semantics and documentation methodologies.
4	Consistency management for E/E product data.
5	Maintenance of data model interdependencies.
6	Support of impact analysis.
7	Transformations between application data models must be derivable from the maintained dependencies.
8	Comprehensive documentation methodology, taking current standards into account.
9	Modeling methodology for the common integration ontology.

Table 1: Requirements for E/E product data management.

required common integration ontology. In summary, there exists no holistic approach covering the aforementioned challenges for E/E product data management in an integrated and comprehensive manner. Although many approaches focus on tool integration, change management of partly integrated application data models is not properly considered. In addition, methodologies for E/E product data management are not covered.

4 Summary and Outlook

This paper has identified challenges emerging in the context of E/E-PDM. Based on a comprehensive analysis of applications involved in E/E development processes as well as characteristic use cases for application-spanning consistency checks, we have elaborated requirements for effective E/E-PDM. To enable application-spanning uses cases (e.g. consistency), a common integration ontology is required allowing for the integration of relevant concepts from existing application data models. In this context, ensuring the consistency of artifacts is a challenging task. Another challenge concerns the evolution of application data models. Note that such changes of application data models are common and hence must be supported. A possible approach is to explicitly document data model interdependencies. The latter are crucial for automatically deviating data model transformations. Finally, a documentation methodology is needed, that takes existing applications data models, technologies as well as standards into account.

In future work we will provide detailed insights into our integration framework and methodology. Finally, we will focus on changes of application data models in context of E/E-PDM.

Acknowledgements. This work has been conducted within the PROCEED[6] project funded by Daimler.

[6]**PRO**active **C**onsistency for **EE** product **D**ata management

References

[AUT11a] AUTOSAR. AUTOSAR Methodology, Technical Report Version 1.2.2, Release 3.2, Rev 0001, 2011.

[AUT11b] AUTOSAR. Technical Overview, Technical Report Version 2.2.2, Release 3.2, Rev 0001, 2011.

[Bau10] A. Baumgart. A common meta-model for the interoperation of tools with heterogeneous data models. In *In 3rd Workshop on Model-Driven Tool and Process Integration (MDTPI 2010)*, Paris, France, June 2010.

[BGN+04] S. Burmester, H. Giese, J. Niere, M. Tichy, J.P. Wadsack, R. Wagner, L. Wendehals, and A. Zündorf. Tool integration at the meta-model level: the Fujaba approach. *Int J Softw Tools Technol Transfer*, 6:203–218, 2004.

[CCWC09] Y.M. Chen, Y.J. Chen, C.C. Wen, and H.C. Chu. Ontology-Based Knowledge Integration for Distributed Product Knowledge Service. In *Proc. World Congress on Engineering and Computer Science*, 2009.

[Coo90] R.G. Cooper. Stage-gate systems: a new tool for managing new products. *Business Horizons*, 33(3):44–54, 1990.

[FK03] R. Freude and A. Königs. Tool integration with consistency relations and their visualization. In *Proc. Workshop on Tool-Integration in System Development (TIS 2003)*, pages 6–10, 2003.

[FMC05] K. Forsberg, H. Mooz, and H. Cotterman. *Visualizing project management: Models and frameworks for mastering complex systems*. Wiley, 2005.

[fS11] International Organization for Standardization. ISO/FDIS ISO 26262-2:2011: Road vehicles – Functional Safety – Part 2: Management of functional safety, 2011.

[HBR08] A. Hallerbach, T. Bauer, and M. Reichert. Managing Process Variants in the Process Lifecycle. In *10th Int'l Conf. on Enterprise Information Systems (ICEIS'08)*, pages 154–161, June 2008.

[KS06] A. Königs and A. Schürr. Tool integration with triple graph grammars-a survey. *Electronic Notes in Theoretical Computer Science*, 148(1):113–150, 2006.

[MHDZ12] M. Mueller, K. Hoermann, L. Dittmann, and J. Zimmer. *Automotive SPICE in Practice: Surviving Implementation and Assessment*. Rocky Nook, 2012.

[MHHR06] D. Müller, J. Herbst, M. Hammori, and M. Reichert. IT Support for Release Management Processes in the Automotive Industry. In *4th Int'l Conf. on Business Process Management (BPM'06)*, number 4102 in LNCS, pages 368–377. Springer, September 2006.

[NK04] N.F. Noy and M. Klein. Ontology evolution: Not the same as schema evolution. *Knowledge and information systems*, 6(4):428–440, 2004.

[RW12] M. Reichert and B. Weber. *Enabling Flexibility in Process-Aware Information Systems: Challenges, Methods, Technologies*. Springer, Berlin-Heidelberg, 2012.

[Sta03] D.H. Stamatis. *Failure mode and effect analysis: FMEA from theory to execution*. ASQ Press, 2003.