



TOWARDS A DIGITAL TWIN: HOW THE BLOCKCHAIN CAN FOSTER E/E-TRACEABILITY IN CONSIDERATION OF MODEL-BASED SYSTEMS ENGINEERING

Heber, Dominik (1,2); Groll, Marco (1,2)

1: Daimler AG, Germany; 2: University of Twente, The Netherlands

Abstract

Complexity of electric/electronics (E/E) in automobiles increases tremendously due to more connectivity and real-time data processing. Hence, huge volatility of E/E artifacts is the consequence. In order to keep track of all changes made from early systems engineering to late after sales, a permeable traceability in data management systems has to be achieved. Therefore, this elaboration adapts the blockchain technology to achieve traceability of E/E development artifacts from early model-based systems engineering (MBSE) till after sales. By this, MBSE is linked to product data management and the automobile's configuration is known at each instant of time. The Digital Twin, a digital, domain-specific representation of the physical vehicle in one front end tool, makes the complexity still feasible to handle and is empowered by the blockchain. Hence, traceability of E/E artifacts over an automobile's lifecycle including MBSE is fostered in a manageable manner.

Keywords: Multi-, Cross- and Trans-disciplinary processes, Product Lifecycle Management (PLM), Product structuring, Product modelling / models

Contact:

Dominik Heber
Daimler AG
Engineering IT MBC
Germany
Dominik.Heber@daimler.com

Please cite this paper as:

Surnames, Initials: *Title of paper*. In: Proceedings of the 21st International Conference on Engineering Design (ICED17), Vol. 3: Product, Services and Systems Design, Vancouver, Canada, 21.-25.08.2017.

1 INTRODUCTION

The capability to determine and reproduce the current state of the composition of electric/electronic (E/E) devices within automobiles throughout their entire lifecycle is a decisive necessity for various application domains such as error correction in software and feature upgrading for ameliorated customer experience. The increasing complexity in automobiles' E/E architecture, a vast variety of software applications running on over a hundred electronic control units (ECUs), as well as new technologies, for instance flashing over the air, make the issue of reproducibility and determination of the current state of settings inevitable and extremely challenging. Therefore, a persistent foundation of documentation and traceability of development artifacts are crucial. This rises research question 1: How can traceability of E/E components in a car's lifecycle be ameliorated?

Within 15 years the amount of ECUs in a premium upper class limousine nearly decupled (Daimler, n.a.). This is on the one hand due to more and more assistance systems which require a lot of computing power, on the other hand due to human-centered enhancements such as smart phone integration or sophisticated human machine interfaces. Here it can be emphasized that particularly real-time and safety relevant applications drive the necessity for development of communication systems with higher performance (Reif, 2015). Multiple software with high volatility on one ECU further increase complexity. For developers it is essential that this complexity is presented in a manageable manner, e.g. in one front end IT tool and not multiple monolithic tools and data bases, so that documentation still can be handled efficiently what leads to research question 2: How can management of this complexity be facilitated and become more efficient?

Development with focus on systems, i.e. systems engineering, and their precise definition including all simulation models already starting in the requirements engineering phase is state of the art yet (Eigner et al., 2014b). However, because systems engineering's complexity rises rapidly parallel with, or even due to, the above mentioned increasing complexity in an automobile's E/E architecture, systems engineering likewise experiences soaring intricacy. Particularly the traceability throughout the lifecycle in case of changes during the development process or even in the phase of production and after sales is a pivotal aspect to handle this intricacy (Eigner et al., 2014a, 2015, 2016). Hence, research question 3 is: How can early system models be connected traceably to product documentation?

In this work focus will lie on mechatronics, i.e. mechanics, electronics, and software, e.g. of what comprises an ECU. Actors and sensors as well as cyber-physical systems will not be in scope. This work wants to describe a first approach for explaining how traceability within the product data management (PDM) could be facilitated. Furthermore, model-based systems engineering also shall be connected to the PDM. Additionally, a concept of a digital 1:1 domain-specific representation of the composition of a physical product in each point of time of its lifecycle, extending product lifecycle management (PLM), is presented. This concept could be operationalized by transferring the concept of a decentralized, irreversible, and public transaction history to PDM and PLM. Figure 1 depicts the research interests.

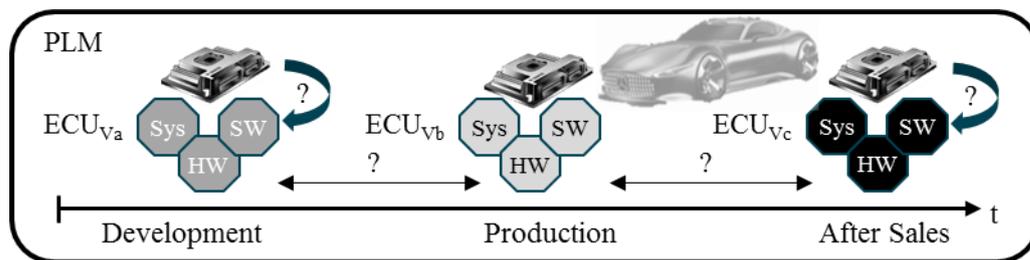


Figure 1. Research interests: How can traceability of Systems (Sys), hardware (HW), and software (SW) along an automobile's lifecycle be fostered and managed?

2 RELATED WORK

2.1 Product data management

Product data management (PDM) is defined as the management of a product model as well as a process model with the target of generating unambiguous and reproducible product configurations (Eigner and Stelzer, 2009). Configuration management is the logical consequence of a permeable product and

process management (Eigner and Stelzer, 2009). Many activities in the context of configuration management aim at the reproducibility and traceability of the current state of construction of a product. Additionally, there is information about which procedure resulted in the recent state of construction (Eigner and Stelzer, 2009). Identification of the current state of construction is given by the so-called effectivity. This effectivity is defined, depending on case of application, by either date or index of change or by a serial number (Eigner and Stelzer, 2009). As shown by Groll and Heber (2016), the permeability in a PDM landscape, particularly in the automobile development and production when focusing on the E/E and mechatronic product development, is crucial (cf. Groll and Heber (2016) for further details). For PDM it is decisive to know the constructability and consistency assertions of products (Küchlin and Sinz, 2000), or associativity, particularly when dealing with mechatronic products which each of themselves already consists of multiple artifacts, e.g. mechanics, electronics, and several software. In some areas of the automobile industry classifications according modules (clustering of product combinations), positions, variants, including left-hand and right-hand steering positions, as well as code rules are applied to certain parts. This is depicted in Figure 2 (Küchlin and Sinz, 2000).

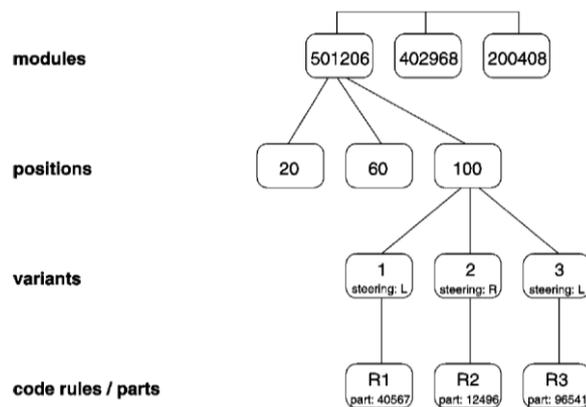


Figure 2. Structure of the part list including structure of model-class-dependent constructability as used by the automobile industry (Küchlin and Sinz, 2000)

Code rules are combined using Boolean algebra for verification of consistency and constructability (Kaiser and Küchlin, 2001). Those code rules have limited space of values and complexity increases tremendously given each new possibility of combination within the purchasing process of an automobile. When considering mechatronic products, then code rules also have to apply to different variants of ECUs to verify the consistency of hardware and software as a stand-alone device and, moreover, in the context of the system the ECU directly is integrated as well as in the context of all other systems in the automobile. Zengler and Küchlin (2013) assess the applicability of different algorithms for existential Boolean quantifier elimination in the configuration of automobiles. For circa 200 variables, depending on the automobile type and approach chosen, between about 600 and 2000 clauses have to be defined to calculate all variants (Zengler and Küchlin, 2013). This shows the extreme complexity which has to be handled throughout the lifecycle. Due to the limited space of values, as stated above, and the extreme complexity, code rules have to be redefined during the lifecycle of an automobile nullifying traceability. Therefore, Objective 1, referring to research question 1, can be stated as following:

Objective 1. There has to prevail traceability of product data management, particularly including all its inherent product configurations, throughout the entire lifecycle of an automobile.

2.2 Product lifecycle management and the concept of the Digital Twin

Historically, product data management (PDM) projects were shaped by the management of technical documents connected to computer aided design (CAD) (Eigner and Stelzer, 2009). Changes in law and product liability created the compulsion of linking those documents to the product root in structure tree of the bill of material and hence application programming interfaces to enterprise resource systems were developed (Eigner and Stelzer, 2009). By means of a permeable configuration management (cf. Section 2.1), PDM evolves to a backbone of a complete product lifecycle management (PLM). There exists a plethora of definitions of PLM. All in common is that generally there exists a broader scope of implementation and integration over all phases of the product lifecycle (Eigner and Stelzer, 2009).

Integration means that there has to prevail a continuous associativity, i.e. linkage, of all different product structures over the different phases of the lifecycle's domains and their corresponding IT systems, starting with requirements management, PDM, manufacturing process management, production planning and steering system, and after sales management (Eigner et al., 2014b). This extends the sole permeability within a PDM focusing on design and engineering phase, as presented by Groll and Heber (2016), by the integration of further lifecycle phases and across a variety of domain-specific tools.

According to Boschert and Rosen (2016), the concept of building an actual, physical "hardware twin" stems from NASA's Apollo program, where an exact replication of a spaceship remained on earth to let the engineers track and test each modification of the spaceship in space, try alternatives, and to use it for training aspects. Yet another familiar instance of a "hardware twin" is the so-called "Iron Bird". In this case it is an Airbus systems integration test bench which is ground-based and used to assess, integrate, and verify physical integrations of aircraft systems such as electrical and hydraulic ones. Via a virtual cockpit the Iron Bird actually can be flown, i.e. virtually flown, computers simulating the necessary environment and hence generating a hardware twin without the purpose of full functionality (Boschert and Rosen, 2016; Airbus Industries, 2016). On the grounds of ever more powerful and prevailing simulation technologies, physical hardware components might be replaced by more accurate, virtual models and therefore extending the Iron Bird towards a digital replication, also known as *Digital Twin*, of the formerly sole physical instance (Boschert and Rosen, 2016). The vision of the Digital Twin is depicted as "a comprehensive physical and functional description of a component, product or system, which includes more or less all information which could be useful in all - the current and subsequent - lifecycle phases" (Boschert and Rosen, 2016, p. 59). However, Boschert and Rosen (2016) focus more on the simulation aspects meaning that the Digital Twin serves as a complete digital model of multiple simulated systems. In the same way, Shafto et al. (2012) from NASA formed the term *Digital Twin* (Boschert and Rosen, 2016), with emphasis on simulation models, accordingly: "A Digital Twin is an integrated multiphysics, multiscale simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its corresponding flying twin" (Shafto et al., 2016, p. 64). Accentuation shall lie on "to mirror the life", i.e. covering the entire lifecycle of its twin sibling from design to disposal.

Thus, the Digital Twin shall incorporate following characteristics in order to assuage the increasing volatility of E/E product documentation (in alignment to Boschert and Rosen, 2016):

1. The Digital Twin has to facilitate interfaces to all relevant IT systems in which the data in scope is stored persistently and combine it to an exact reproduction of the current condition of the composition of ECUs and their immanent characteristics in this way as the real automobile currently is built up.
2. The Digital Twin has to ensure a permeability of documentation of ECUs and their properties throughout the entire lifecycle - from the beginning of design and engineering till the operating phase of the automobile in the after sales. This ensures the traceability of each component or artifact in each variant and each version.
3. The Digital Twin has to be domain specific in order not to become too complex and therefore unhandy. Here, the focus is on E/E and specifically ECUs.

This requirements towards the design of a Digital Twin yield Objective 2, addressing research question 2:

Objective 2. There has to prevail a digital representation of an automobile in each point in time which fosters traceability of artifacts and ensures a permeable documentation. This digital representation has to be domain specific and in one front end IT tool to mitigate complexity and increase manageability for developers.

2.3 Model-based systems engineering

According to Groll and Heber (2016), a decisive impediment in the product development method of Bender (2005) is that a model-based product development is not explicitly addressed. Bender (2005) solely focusses on mechatronic product development. For this purpose, Bender (2005) extended the V-Model delineated in VDI (2004) to a three-layered V-Model, addressing the different domains within mechatronic products separately. In the two upper levels in Figure 3, i.e. vehicle and system level, development occurs jointly. Successively, sub-systems and components are developed domain-specifically (cf. Figure 3). Precisely this separation of development in domain-specific tracks, which are

often divided technically and organizationally, is what makes the development of mechatronics so challenging (Groll and Heber, 2016).

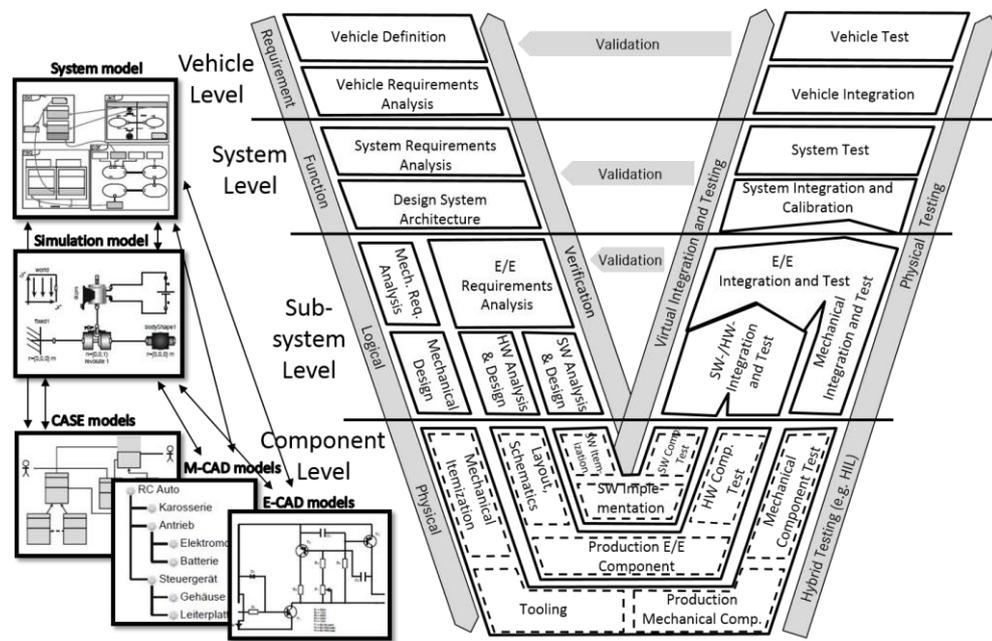


Figure 3. V-Model of mechatronic product development extended for systems development and applied to automobile development (in alignment to Bender (2005), Eigner et al. (2014b), VDI (2004))

If the description within each particular phase of the lifecycle is based upon a formal language (a formal language is an abstract language which focuses on mathematical or physical deployment (Eigner et al., 2014b)), and digital models are built and placed at the disposal by transformation for the next phases, then the IT-process chain is denominated as model-based (Eigner et al., 2014b). Systems engineering (SE) is defined as an inter-disciplinary, document-based approach for the development and implementation of technical and complex systems in order to assure a high-value fulfilment of stakeholders' and customers' requirements throughout the entire lifecycle (Eigner et al., 2014b). Model-based systems engineering (MBSE) makes use of a formal language and designated system models to determine the product within the product development process in each of its phases and also in downstream processes (Eigner et al., 2014b). Figure 3 already extends Bender's (2005) V-Model to that effect that the different development phases for model-based systems engineering are considered: requirement, function, logical, and physical. Testing, i.e. the right arm of the V in Figure 3, is not in scope of this elaboration. On system level, one defines the system model which constitutes of interdisciplinary aspects of requirements, structure, behavior, parameters, context, and test cases associated with one or many systems (Friedenthal et al., 2012; Hybertson, 2009 in Eigner et al., 2014b). The analysis of discipline-specific and multidisciplinary approaches jointly identify aspects of the interdisciplinary system model. Those system models usually are descriptive and non-executable, i.e. cannot be simulated (Eigner et al., 2014b). In the next phase, quantitative aspects are integrated into the interdisciplinary simulation models. Further downstream in the development process, discipline-specific simulation models will be derived from the multidisciplinary simulation models. Those discipline-specific simulation models are models from, for instance, computer-aided software engineering (CASE), mechanical CAD (M-CAD), and electric/electronic CAD (E-CAD). Throughout the entire systems engineering process, granularity of details increases from requirements, via functions and logical system architecture, to the physical system architecture, as depicted in Figure 3 (Eigner et al., 2014b). Eigner et al. (2014a, 2015, 2016) present a draft how model-based systems of different disciplines can be managed throughout the entire lifecycle so that all of the time a virtual image, i.e. the Digital Twin, is available. This approach is called systems lifecycle management (SysLM). SysLM depicts the information management of models in general which extends common PLM by an explicit consideration of upstream and downstream phases of development (Eigner et al., 2016). The PLM backbone shall incorporate the entire system lifecycle management (Eigner et al., 2016). Gilz (2014) further highlights

in his outlook the need for a combination of system models and configuration as well as variant models. In addition, Boschert and Rosen (2016) propose that the Digital Twin is the next wave in simulation technology. This leads to Objective 3 and refers to research question 3:

Objective 3. Model-based systems engineering and its artifacts from the early development phase have to be included in the product data management to foster traceability of system models and their simulations along the lifecycle.

3 CONCEPT OF TRACEABILITY OF ECUS ALONG THE LIFECYCLE

The concept of the Digital Twin extends PLM by the idea that the user, e.g. developer, has an identical replication of an entire product combined in one front end IT tool. Hence, the vision of the Digital Twin of an automobile constitutes of a persistent documentation of the current and successive states of any specifically limited and defined combination of information artifacts in a centralized IT system. Besides, several separated IT systems might contain the data which is decisive for the Digital Twin but a metadata layer links and combines the dispersed data into one single point of truth. In the case of, for example, ECUs, the combination of artifacts would be composed of at least one hardware, i.e. the entire association of all hardware components such as the ECU's case, different kinds of memory, central processing unit, plugs including pins, et cetera. Those hardware components naturally are less volatile with regard to their persistence in an automobile because an ECU usually is only changed in service garage when a defect exists, and hence the imperative to provide high flexibility within a documentation system, where the Digital Twin is located, does not stem mainly from hardware changes. Additionally, several parts of software also are information artifacts within the specifically defined combination which shall be captured by a Digital Twin. Software is subject to many changes due to a very high complexity and hence error proneness. Moreover, customers expect automobiles to be more like their smartphone: adaptable by personal applications, connected to many devices or even other cars and infrastructure, and online in the internet. Also, functions and interfaces are more intertwined and integrated in the automobile and outside of it. Those functions are more and more realized and executed by software (Boschert and Rosen, 2016). This very high volatility and complexity yields needs for traceable documentation of the current composition of these parts in the ecosystems where they operate.

3.1 The blockchain technology and its application possibilities

The concept of the crypto-currency *Bitcoin*, published in a whitepaper by Nakamoto (2008), fosters online payments which are sent directly from one participant to another participant without making the necessity of a third intermediary, e.g. in this case a financial institution. Here, the peer-to-peer nodes create a network which timestamps each transaction by using hashes of each transaction and combining them into a continuous chain of hash-based evidence of work, the so-called *blockchain*, using digital signatures to verify the authenticity of transactions. The Bitcoin network is the most famous application of the blockchain technology (Sixt, 2017). In the blockchain, a block consists of a hash of the previous block, a time-stamp, usually both included into the header, and a payload which is limited in size (see Figure 4). This payload can carry information, such as texts, or references to other data sources in case this referenced data would be too big to include into the blocks. In the case of Bitcoin, the payload is information about financial transactions. The blockchain is a decentralized, public, and irreversible transaction history. The decentralized structure is achieved by the peer-to-peer nodes' network of which each individually holds the entire transaction history, i.e. the entire blockchain. Hence, there is a dispersed and redundant database for which no intermediary or central authority is needed in order to guarantee authenticity of transactions. A consensus-based validation of transactions secures uniqueness of a transaction.

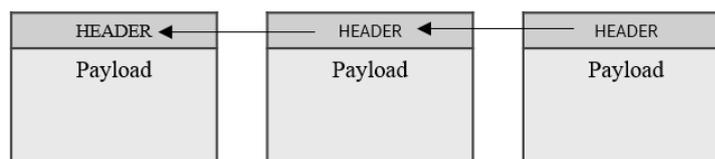


Figure 4. The blockchain: A decentralized, public, and irreversible transaction history. The header includes hash values referencing to the previous block. The payload may contain a limited size of data and linkages to other data sources

The public structure is accomplished by a spread knowledge about each transaction and everyone can access this knowledge which is stored in the blockchain. The irreversible structure is realized by interlinking of hash values and hence data integrity is fostered. Use cases for the application of the blockchain technology usually have to have the following autonomy so that the blockchain technology could be efficient:

- Multiple independent parties are involved;
- Redundant administration and exchange of data;
- Inter-company processes;
- Current process is inefficient, e.g. due to inherent high costs of coordination;
- A centralized system is not feasible, maybe due to trust issues;
- High data transfer rate is not necessary because dispersion of data in the peer-to-peer network still is slow with the blockchain technology.

This low data transfer rate also is one of the hugest disadvantages of the blockchain technology. Moreover, the decentralized structure increases requirements for maintenance and security.

3.2 A methodical approach applying the blockchain technology as means for the promotion of traceability of ECUs along their lifecycle

As depicted in Section 3.1, the blockchain fosters an irreversible transaction history. This irreversibility guarantees that blocks, i.e. transactions or any kind of payload linked via hashes to their predecessor, in the downstream development process cannot be lost or linkages remain unknown. When we consider that in the development process a block is created each time when an information artifact is created or updated and this new block is linked to its predecessor via hash values, this fosters permeable traceability of all information artifacts - from the very first one to the last. Information artifacts, i.e. the payload of blocks, with focus on ECUs can be systems, hardware, software, variants, and consecutive versions of the aforementioned. However, due to limited block size (cf. Section 3.1), the payload of each block cannot be geometric models nor the entire source code but solely metadata and, for instance, a uniform resource identifier (URI) directing to the corresponding data source, team data management, etc. where data is created and/or stored. In case of model-based systems engineering, this approach will be presented subsequently and later will be applied to the use case of the Digital Twin.

3.2.1 Connect model-based systems engineering to PLM with the blockchain technology

When commencing the development process chronologically, one starts with the definition of requirements and then moves on to conceptualize functional, logical, and physical aspects of the product's systems in the sense of MBSE (cf. Section 2.3). In Figure 5 it is delineated how the information artifacts of the MBSE process can be documented persistently and chronologically, following 1 to 3, in a PLM backbone adopting the blockchain technology and hence creating traceability along the lifecycle. This addresses Objective 3. The MBSE process starts with the definition of the system model on the system level in the V-model of development (cf. Figure 3). Due to motivational reasons for the use case in Section 3.2.2, we will assume that a supplier develops the system outside-mirror.

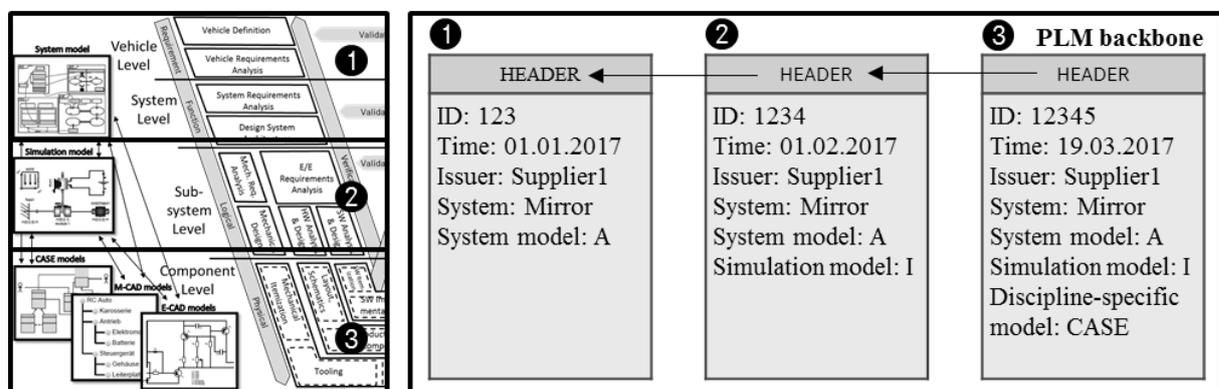


Figure 5. Creating blocks for documentation in the PLM backbone from each model-based systems engineering process step. Block headers refer to their predecessor for the purpose of traceability along the lifecycle

When the interdisciplinary system model was defined in its author IT tool, then the supplier will write the first block of this particular blockchain into the PLM backbone. The block contains, besides the header with a hash value, metadata such as a unique identifier, a time-stamp (sometimes in the header), information about the issuer, the name of the system and a reference to the particular system model which will be stored elsewhere. Afterwards, on the sub-system level those interdisciplinary system models will be enriched with quantitative information and will be simulated. The supplier adds this new metadata not to the previous block but by appending a fresh block to the PLM backbone containing all the previous information but with a recent ID, generating effectivity of this version, and additional data for the simulation model and where it is stored. The same repeats itself when proceeding to the component level where discipline-specific simulation models are built. The irreversibly connected new block additionally contains the reference to the discipline-specific simulation model, here a CASE-model (see Figure 5).

3.2.2 Use case Digital Twin: The PLM blockchain as means of traceability of ECUs' product documentation along their lifecycle

As depicted in Section 2.2, the Digital Twin aims at fostering a permeable traceability of domain specific product documentation of a single entity of a product in one front end IT tool. The blockchain enables this traceability (cf. Section 3.1). Hence, this use case describes an approach how the Digital Twin can be enabled by implementation of the blockchain. In Section 3.1 prerequisites for a blockchain use case are given. Applied here, we speak of a Digital Twin resembled in a PLM blockchain and stored at the car manufacturer. The supplier's and partner's blockchain only hold the relevant information about the parts delivered by the supplier or in scope of the partnership, but not all information of the Digital Twin due to data disclosure. This setup addresses prerequisites such as multiple parties, exchange of data, inter-company processes, high costs of coordination, trust issue, and low data transfer rate sufficient due to mirroring the Digital Twin in each blockchain is not time-critical. Chronologically, we start at the MBSE process in Section 3.2.1. However, this process is abbreviated and summarized in the first block created at the supplier's blockchain including metadata for system "Mirror" (cf. Figure 6). By consensus (cf. Section 3.1), co-operation partner and manufacturer agree upon the validity of the block. Then, the block is mirrored in the partner's blockchain (here combined with the supplier's due to graphical reasons) and the manufacturer's, resembling the Digital Twin in the PLM blockchain. In the PLM blockchain one associates the metadata to backbone IT tools (cf. Section 3.2.1). During development, one solely develops car lines and their different model types (sedan, convertible, etc.) so there is only a Digital Twin for each of those which are stored in the PLM blockchain.

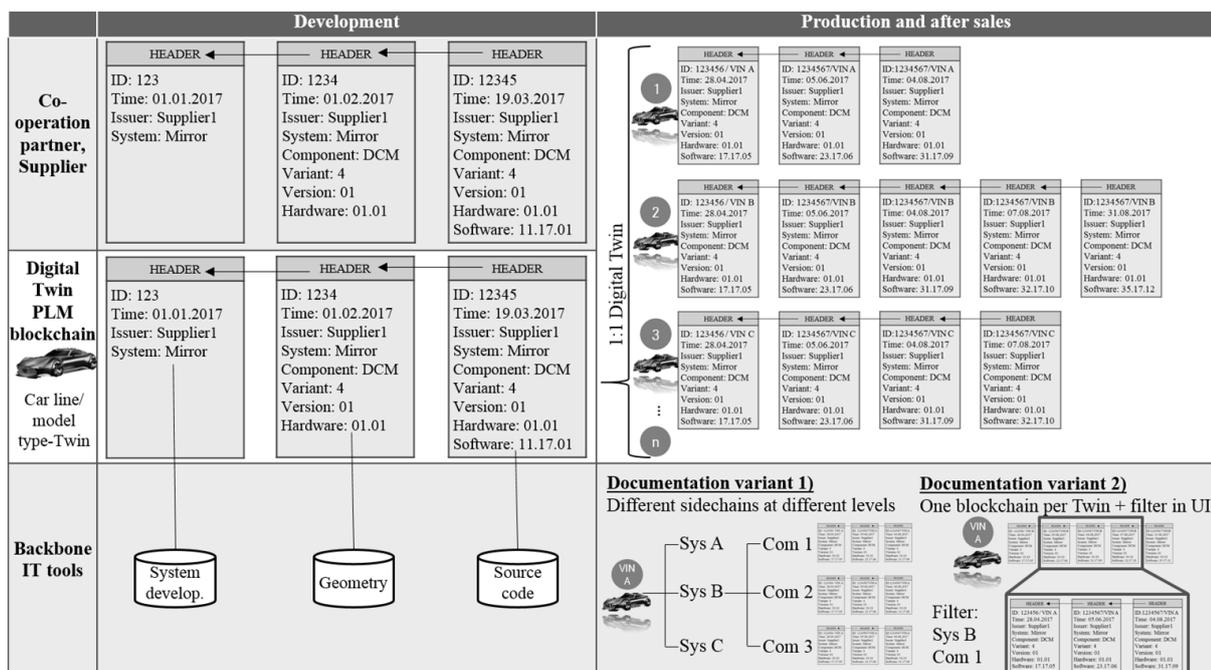


Figure 6. Use Case Digital Twin: How the blockchain fosters traceability of each automobile's configuration from development to after sales

Moving to the component level, the supplier develops component-specific hardware in a defined variant and version, publishes this block in his blockchain, partner and manufacturer agree, mirror the block in their blockchains, and the manufacturer links the hardware's metadata to his data source where e.g. 3D models etc. are stored. The same process holds for software. Each new development artifact is linked in the blockchain to the previous block enabling traceability amongst hardware and software configurations (Objective 1) and traceability to MBSE (Objective 3). When a customer orders her car, this particular car gets a vehicle identification number (VIN) and when production starts, the physical twin is born. From this point in time there is an individual Digital Twin for each physical car, as per definition (cf. Section 2.2). Due to focus on E/E, permeable traceability by means of the blockchain, and representation in one metadata layer (PLM blockchain) bridging monolithic backbone IT tools, Objective 2 is satisfied. Documentation variant 1 in Figure 6 applies different blockchains at e.g. component level and hereby creates traceability by linkages to the respective systems and each car. Those different blockchains pegged to, for instance, an overall car's blockchain, are called sidechains. Documentation variant 2 applies only one blockchain per car. Several thousand development artifacts, i.e. blocks, would not be manageable for a developer. Hence, a filter has to prevail for the system or component of interest in the user interface.

This approach of the blockchain enabling the Digital Twin would not address the issue of a complete systems lifecycle management, as proposed by Eigner et al. (2016) in Section 2.3. Here, the focus is only on a linkage of metadata without functional aspects but to achieve traceability in the sense of a Digital Twin which is not focused on simulations (cf. Section 2.2).

4 CONCLUSION AND OUTLOOK

Due to an increasing share of electric/electronic (E/E) components, such as electronic control units (ECUs), in an automobile, complexity in product documentation becomes very challenging. Therefore, the current state of composition of different variants and versions has to be known all over the lifecycle of an automobile. This is crucial for simulations in the development phase, for assembly in the production phase, and for e.g. flashing over the air in the after sales in order not to induce errors, provoke erratic behavior, or know which system has to be re-simulated in advance. This issues lead to three research objectives motivated by the state of the art literature analysis. Objective 1 aims at providing traceability of product data management (PDM) all over the entire lifecycle of an automobile. Objective 2 concerns the digital representation of each physical automobile's E/E configuration in every instant of time in order to foster traceability of E/E artifacts. This digital representation still has to be manageable and hence shall be domain specific and in one front end tool. Objective 3 states that model-based systems engineering (MBSE) has to be incorporated into product data management to keep track of changes over the lifecycle.

In this elaboration, a concept of traceability of ECUs along their lifecycle is presented and hence an approach how to keep track of an automobile's configuration status is developed. For that purpose, the blockchain technology is adapted. The blockchain is a decentralized, public, and irreversible transaction history or ledger which is efficient for cases in which multiple parties are involved, data is managed redundantly, and there is some sort of trust issue, amongst others. The blockchain consists of blocks linked to their predecessor by hash values and contain a payload. It is depicted how this payload can be metadata of MBSE creating a permeable product lifecycle management (PLM) backbone resembled by interconnected blocks, yielding a PLM blockchain. This solves Objective 3. Furthermore, a use case of the Digital Twin, the digital, domain-specific, front-end IT tool representation of a physical car over its lifecycle, is presented. For this purpose the blockchain is used to describe and permanently connect metadata, such as variants and versions of E/E artifacts. This creates traceability from the first systems engineering phase via production till after sales and hence solves Objective 1. Objective 2 is accomplished by representing the Digital Twin in a separate blockchain containing all relevant metadata and linking to the proper author tools and data sources where information of geometry or source codes lie and therefore facilitate usability and manageability of an entire automobile's information artifacts.

Further research will focus on which level of the product structure the blockchain will be situated, whether there will be only one blockchain per vehicle or with multiple sidechains. Moreover, the level of detail of payload within blocks of the blockchain has to be assessed. Additionally, interaction with suppliers and co-operation partners within the engineering design network (Parraguez and Maier, 2016) and hence distribution of the blockchain will be in scope.

REFERENCES

- Airbus Industries (2016), *Iron Bird*. Airbus Industries. Available at: <http://www.airbus.com/innovation/proven-concepts/in-design/iron-bird/> (accessed 12. Dec. 2016).
- Bender, K. (2005), *Embedded Systems – qualitätsorientierte Entwicklung*, Springer, Heidelberg. DOI 10.1007/b138984
- Boschert, S. and Rosen, R. (2016), "Digital Twin - The Simulation Aspect", In: Hehenberger, P. and Bradley, D., *Mechatronic Futures - Challenges and Solutions for Mechatronic Systems and their Designers*, Springer International Publishing, Switzerland, pp. 59-74. DOI 10.1007/978-3-319-32156-1
- Daimler AG (n.a.), *Internal Documentation*.
- Eigner, M., Apostolov, H., Dickopf, T., Schäfer, P. and Faißt, K.-G. (2014a), "System Lifecycle Management - am Beispiel einer nachhaltigen Produktentwicklung nach Methoden des Model-Based Systems Engineering", *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, Vol. 109, pp. 853-860.
- Eigner, M., Roubanov, D. and Zafirov, R. (2014b), *Modellbasierte virtuelle Produktentwicklung*, Springer Vieweg, Berlin. DOI 10.1007/978-3-662-43816-9
- Eigner, M., Faißt, K.-G., Apostolov, H. and Schäfer, P. (2015), "Kurzer Begriff und Nutzen des System Lifecycle Management - im Kontext von Industrial Internet mit Industrie 4.0 und Internet der Dinge und Dienste", *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, Vol. 110, pp. 475-478.
- Eigner, M., Muggeo, C., Apostolov, H. and Schäfer, P. (2016), "Kern des System Lifecycle Management - im Kontext von Industrial Internet mit Industrie 4.0 und Internet der Dinge und Dienste", *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, Vol. 111, pp. 63-68.
- Eigner, M. and Stelzer, R. (2009), *Product Lifecycle Management - Ein Leitfaden für Product Development und Life Cycle Management*, Springer-Verlag, Berlin Heidelberg. DOI 10.1007/978-3-540-68401-5
- Friedenthal, S., Moore, A., Steiner, R. (2012), *A Practical Guide to SysML. The Systems Modeling Language*. Morgan Kaufmann, Waltham. DOI 10.1016/B978-0-12-385206-9.00018-1
- Gilz, T. (2014), *PLM-Integrated Interdisciplinary System Models in the Conceptual Design Phase Based on Model-Based Systems Engineering*, Dissertation, Technical University of Kaiserslautern, Kaiserslautern.
- Groll, M. W. and Heber, D. (2016), "E/E-Product Data Management in Consideration of Model-Based Systems Engineering", In: Borsato, M., Wognum, N., Peruzzini, M., Stjepandić, J. and Verhagen, W. J. C. (Eds.), *Advances in Transdisciplinary Engineering, Vol. 4: Transdisciplinary Engineering: Crossing Boundaries, Proceedings of the 23rd ISPE Inc. International Conference on Transdisciplinary Engineering October 3–7, 2016*, IOS Press BV, Amsterdam, pp. 289-298. DOI 10.3233/978-1-61499-703-0-289
- Hybertson, D. W. (2009), *Model-Oriented Systems Engineering Science. A Unifying Framework for Traditional and Complex Systems*. CRC Press, Boca Raton.
- Kaiser, A. and Küchlin, W. (2001), "Automotive Product Documentation", In: Monostori L., Váncza J., Ali M. (Eds), *Engineering of Intelligent Systems. International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems 2001. Lecture Notes in Computer Science*, Vol. 2070, pp. 465-475, Springer, Berlin Heidelberg. DOI 10.1007/3-540-45517-5_51
- Küchlin, W. and Sinz, C. (2000), "Proving Consistency Assertions for Automotive Product Data Management", *Journal of Automated Reasoning*, Vol. 24, pp. 145-163. DOI 10.1023/A:1006370506164
- Nakamoto, S. (2008), *Bitcoin: A Peer-to-Peer Electronic Cash System*, bitcoin.org. Available at: <https://bitcoin.org/bitcoin.pdf> (accessed 15. Dec. 2016).
- Parraguez, P. and Maier, A. (2016), "Using Network Science to Support Design Research: From Counting to Connecting", In: Cash, P., Stankovic, T. and Storga, M. (Eds.), *Experimental Design Research*, Springer International Publishing, Switzerland, pp. 153-172. DOI 10.1007/978-3-319-33781-4_9
- Reif, K. (2015), *Automobilelektronik - Eine Einführung für Ingenieure*, Springer Vieweg, Wiesbaden. DOI 10.1007/978-3-658-05048-1
- Shafto, M., Conroy, M., Doyle, R., Glaessgen, E., Kemp, C., LeMoigne, J. and Wang, L. (2012), *Modeling, Simulation, Information, Technology & Processing Roadmap - Technology Area 11*, National Aeronautics and Space Administration, Washington, DC.
- Sixt, E. (2017), *Bitcoins und andere dezentrale Transaktionssysteme - Blockchains als Basis einer Kryptoökonomie*, Springer Gabler, Wiesbaden. DOI 10.1007/978-3-658-02844-2
- VDI (Verein Deutscher Ingenieure) (2004), *VDI-Richtlinie 2206: Design methodology for mechatronic products*. Beuth Verlag, Berlin.
- Zengler, C. and Küchlin, W. (2013), "Boolean Quantifier Elimination for Automotive Configuration - A Case Study", In: Pecheur, C. and Dierkes, M. (Eds.), *Formal Methods for Industrial Critical Systems. FMICS 2013. Lecture Notes in Computer Science*, Vol. 8187, pp. 48-62, Springer, Berlin Heidelberg. DOI 10.1007/978-3-642-41010-9_4