33RD CONGRESS
OF THE INTERNATIONAL COUNCIL
OF THE AERONAUTICAL SCIENCES
STOCKHOLM, SWEDEN, 4-9 SEPTEMBER, 2022

A HOLISTIC AIRCRAFT CABIN METAMODEL AS AN APPROACH TOWARDS AN INTERCONNECTED DIGITISED CABIN LIFECYCLE

Markus C. Berschik[1], Marvin Blecken[2], Hiteshkumar Kumawat [2], Jan-Erik Rath[3], Dieter Krause[1], Ralf God[2], Thorsten Schüppstuhl[3]

[1]Hamburg University of Technology (TUHH), Institute of Product Development and Mechanical Engineering Design (PKT), Hamburg, Germany
[2]Hamburg University of Technology (TUHH), Institute of Aircraft Cabin Systems (FKS), Hamburg, Germany
[3]Hamburg University of Technology (TUHH), Institute of Aircraft Production Technology (IFPT), Hamburg, Germany

Abstract

Advancing digitalisation, growing passenger needs and constantly changing requirements are urging aircraft manufacturers to offer new and digital services as well as highly individualised cabin layouts. Thus, increased co-development and co-customisation along the whole lifecycle are necessary. To address the associated challenges of data consistency and availability, we propose a holistic metamodel for combining all relevant product information in one central model. Its application is shown in an example connecting product development and assembly process planning. Consistent usage of the metamodel can improve the interconnection along the whole product lifecycle and drive the coordination of all stakeholders.

Keywords: Metamodelling, Aircraft Cabin, Product Lifecycle, Digitalisation, Customisation.

1. Introduction

Due to increasing competition in the transportation sector, airlines are forced to offer their customers competitive prices, higher reliability and greater comfort. To set themselves apart from the competition, the aircraft cabin of the future will be an even more customised product for the airline and the passenger than it is today. New digital services for passengers, such as controlling seat functions or ordering drinks via their smartphone, will contribute to passenger comfort and enable the generation of additional revenue for airlines [1]. At the same time, requirements for safety, security, reliability and user-friendliness, especially also for crew members, must always be met. For product development and production, this trend of individualised cabins and the demand for shorter lead times is a growing challenge, added onto the existing characteristics of mostly manual production and assembly with high tolerances and high variety. Product development cycles are becoming shorter, and the number of product variants and production ramp-ups is increasing [2],[3].

To cope with the increase in costs, time and effort associated with a greater range of product variants and product functions, not only the product itself but also product development and production need to be improved. Up to now, product development, production system development and production are not well connected. Each of the stakeholders and departments uses different software tools, data formats and means of communication. Working with outdated data and repetition of the same work in different departments becomes more likely. Errors are discovered only at later stages resulting in costly re-engineering or rework [4]. Thus, as in other industries, co-development and co-customisation of aircraft cabins and their production systems need to be enabled.

The most important requirements for effective interdisciplinary co-development and cooperation are a common system understanding and a single data source. New, consistently model-driven approaches offer the opportunity to simultaneously optimise product functionality, design and production [4],[5],[6]. Thus, within the research project *VERDIKA* (Connected Digital Cabin) a common cabin metamodel is developed by the Hamburg University of Technology (TUHH). The overall objective of the project is to significantly advance networking and digitisation throughout the entire product lifecycle of the aircraft cabin from design, through production and installation, to usage and operation. Models for the creation, distribution, transfer and use of data and information enable a detailed overview of these complex relationships during the value creation process.

Existing approaches on (meta-)modelling to cover and support the whole product lifecycle as well as related work concerning a networked aircraft cabin, development strategies and tools will be briefly described in the next section. The envisioned holistic cabin metamodel and its modelled realisation are presented in section 3. An application on a small example and its added value for product development and assembly process planning are outlined in section 4 and an outlook on further applications is given in section 5. The paper closes with a summary and an outlook on future steps and developments in section 6.

2. State of the Art

An approach to manage product development, production and product usage, as well as corresponding data along its whole lifecycle, is Product Lifecycle Management (PLM). The concept of PLM to date is realised by proprietary software platforms integrating different discipline-specific development tools, most of the same vendor, to build development workflows and manage product data. However, today's PLM tools mostly support the lifecycle only up to the end of production and are inflexible and not customisable for different applications. They mostly use structural information in form of a bill of material as a reference for the used elements, which leads to missing links to other domains, especially with the increasing development of mechatronic products. With emerging technologies and concepts such as the Internet of Things and model-driven engineering, the technical implementation of the PLM concept might be expanded to cover the actual whole product lifecycle in the form of a Digital Twin and increase the linkage also to behaviour information [6].

In some phases of the product lifecycle, modelling is already commonly used. Especially in the development phase, sophisticated approaches for model-driven development exist but often lack the connection to later life phases [7],[8]. This applies in particular to the individual departments with specialised modelling tools such as CAD systems, FE solvers or offline robot simulation tools. However, on a higher level, system models aim at describing the whole system (e.g. a product) with its requirements, architecture, functions and behaviours. A model is thereby defined with the three characteristics of representation, reduction and pragmatism concerning the modelled object [9]. An Ontology is a special descriptive model, which uses partial description and under-specification as a key feature of abstraction. It provides all relevant modelling concepts of terminology and interrelation and uses a context to differentiate the view onto the model [10].

In the context of Software Engineering, software architectures are modelled and created based on the Unified Modelling Language (UML), which is defined by the Object Management Group (OMG). The domain of Systems Engineering also takes advantage of modelling and a shift from the traditional document-based to a model-based approach can be seen throughout the industry. The central idea of Model-Based Systems Engineering (MBSE) encompasses a system model that represents the system specification and serves as a single source of truth by using the Systems Modelling Language (SysML). The system-level model thereby does not replace discipline-specific models but aims at connecting the different disciplines and integrating the data used and generated by discipline-specific tools [5].

In doing so, common methods and processes for developing complex systems such as aircraft must be followed by the whole development team to enable consistency, traceability and clear data flow. Commonly used development guidelines include CMMI, SPICE, ISO61508 as well as the INCOSE process guidebook, describing processes and activities in general systems engineering [11]. In aviation, guidelines for the processes are provided by the SAE ARP-4754 [12]. Herein, the development process is generally described based on the V-Model and tailored considering the specific needs of the aviation industry. To enable the handling of the development of complex aircraft systems, the design phase of the V-Model consists of different development levels. These include but are not limited to the aircraft level, system level and item level. At each development level, information for the development, e.g. functions, is refined [12],[13].

While the SAE ARP-4754 is not specifically linked to model-based systems engineering, the RFLP (requirements, functional, logical, physical) development framework aims at covering the modelling steps in any model-based project. It was established to bridge the gap between requirements engineering and physical system development typically present in the system analysis and system development phases of mechatronic products, e.g. the descending branch of the V-Model. While requirements answer the question of why a product is developed, functional models shall define what is to be developed in accordance with what the system does. The logical structure states how and by which logical components the functions will be fulfilled and the physical definition represents the virtual product with its physical product data [14],[15],[16].

The framework is used to enable clear traceability from the requirements to the operational, functional and component layers of the system, enabling impact analysis and system behaviour simulation integrating and linking functions with logical components and their physical data. It was applied in the automotive [16] as well as the aircraft assembly planning [15] sector and applies to all hierarchical layers of the product. Kleiner et al. [14] implemented an RFLP modelling approach in Dassault Systèmes CATIA and ENOVIA V6 as an integrated commercial software platform enabling the linkage of requirements, functional, logical and physical model elements including CAD data. In many industries, however, multiple heterogeneous software tools are used along the development process, preventing the implementation of the RFLP framework [14],[15],[16],[17].

A generic metamodel can benefit the system understanding and improve the development processes even in a heterogeneous environment. From a hierarchical point of view, the metamodel is an abstraction layer above the model. Thus, a metamodel defines the elements and the syntax of the modelling language. For this definition, the concept of abstraction is commonly used. It is described in the Meta Object Facility (MOF), which is a standardised meta-metamodel published by the Object Management Group (OMG) in 1997. The MOF describes a top-down syntax for modelling languages such as Unified Modeling Language (UML), the Common Warehouse Meta-Model (CWM) or the Software Process Engineering Meta-Model (SPEM) [18].

With the concept of abstraction, different hierarchical levels of metamodelling are possible. Especially in low abstraction levels, different metamodels depend on the different fields and specific application cases. A domain metamodel defines the concepts - classes of elements - relevant from the application point of view, along with their relationships [19]. An example is the metamodel for cyber-physical systems by Scheuermann, based on the models of embedded systems, sensor networks and smart objects. An excerpt of it is shown in Figure 1, defining a cyber-physical system with a smart object as a leaf node. Each smart environment can contain persons that are part of or use a smart object. The metamodel allows developers to add new abstractions and new stereotypes for example wearables, smartphones etc. [20].

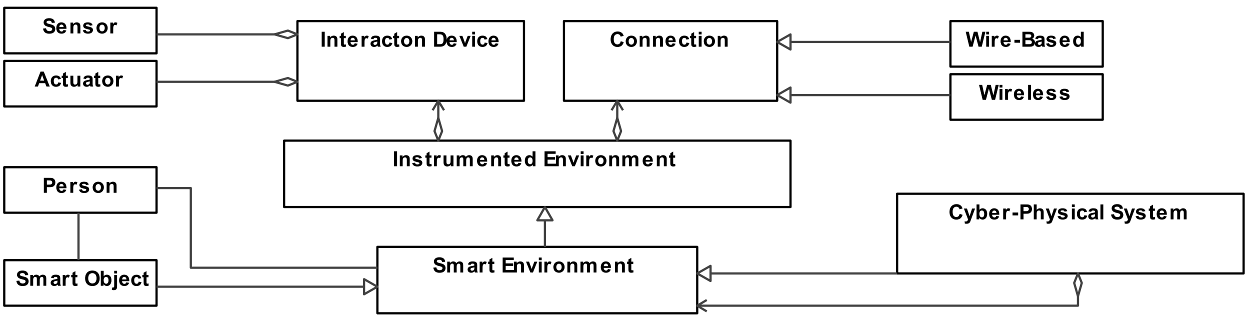

Figure 1 - Metamodel of cyber-physical system [20]

However, in the area of aircraft systems development and usage, no common holistic metamodel for the modelling of an aircraft cabin along its entire lifecycle exists.

3. Holistic Cabin Metamodel

With the increasing connectivity, product diversity and cost pressure, for effective interdisciplinary co-development and co-operation, the further connection of the product lifecycle is inevitable [4],[5],

[6]. Thus, in the research project VERDIKA (connected, digital cabin), an aircraft cabin metamodel is developed, defining generic elements and their relationships to be used in a cabin model. It represents the abstract syntax and concepts to create a holistic cabin model with a modelling language such as SysML, supporting the whole cabin life cycle from development until usage and MRO. As depicted in Figure 2, the metamodel links the domains of different product life phases so that the information and data flows can be modelled holistically. In addition, the cabin is represented as a product itself and modelled with the consideration of service- and communication platforms and its cyber-security.

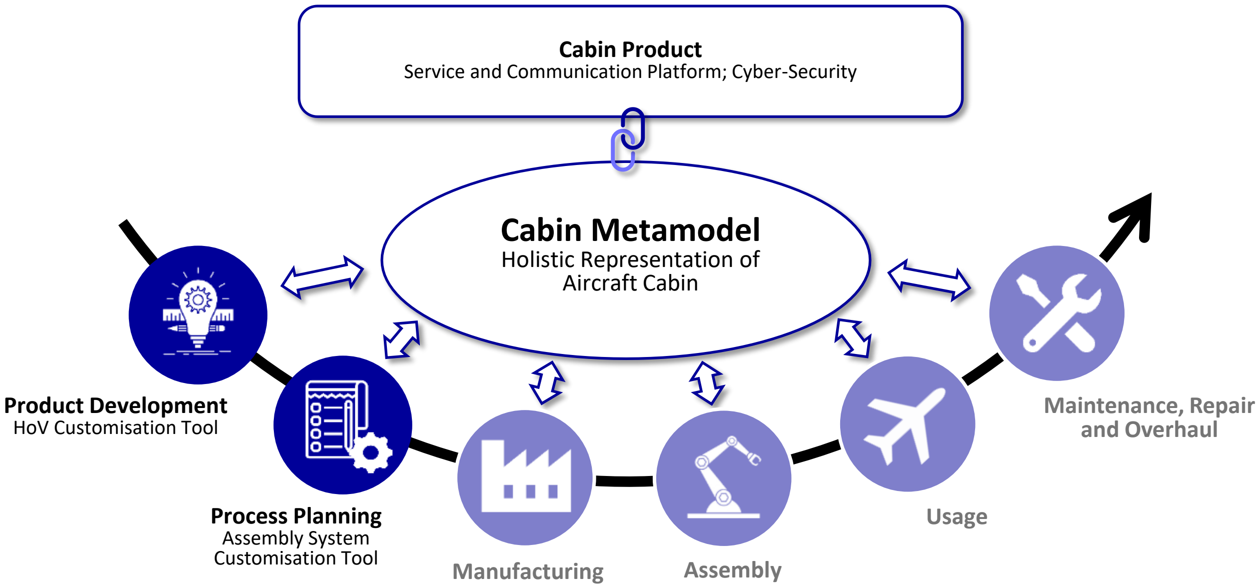

Figure 2 - Overview of the envisaged cabin metamodel concerning the lifecycle phases

The selection of life phases of the cabin shown in Figure 2 represents a simplification of the cabin life cycle. In the current state of the project, only the first two steps, product development and process planning are represented in detail. In addition, different services such as the communication platform are also considered for the cabin including its cyber security. Other life phases can easily be added in the future due to the top-down approach used for metamodelling. If further data and information need to be modelled, new specific datatypes and facets generically defined in the metamodel can simply be added in lower hierarchical levels without affecting the consistency of the already existing data. The structure of the abstract metamodel is depicted in Figure 3.

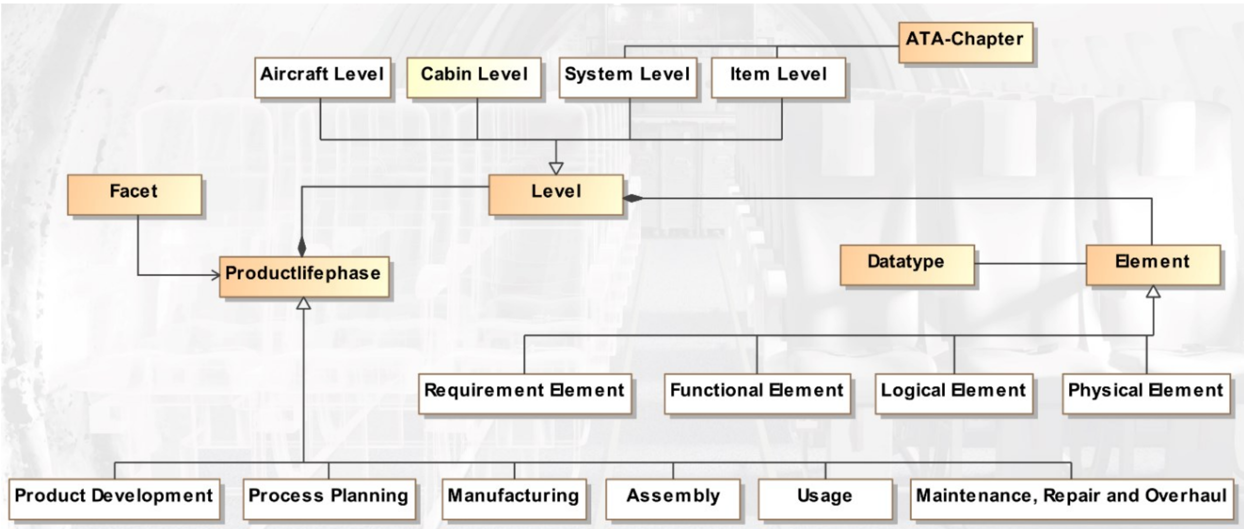

Figure 3 - Description of the cabin metamodel

The envisioned metamodel of the cabin builds around its different product lifecycle phases. As shown before, the missing interconnections of the life phases are a key driver for inconsistent data and other inaccuracies throughout the development process and beyond. Therefore, they are used in the metamodel as one of the focused aspects. Furthermore, in the description of the metamodel, different facets of considerations can be added for every product life phase. These facets augment the general description of the model and hence provide room for extension. Safety and reliability are examples of facets, which are relevant for the certification of critical systems like aircraft systems. Because of connecting previously isolated systems in the aircraft cabin, security is also considered a facet that must be addressed in the development.

For aircraft development processes, the ARP provides levels for the hierarchical classification of the whole product of an aircraft. The typically used levels are the aircraft level, system level and item level. For the cabin metamodel, these levels are augmented with the cabin level, to provide the space for all relevant modelling regarding the cabin. In this representation, the aircraft level is used for completeness and to provide the system boundary of the cabin. On the system and item levels, the ATA-Chapters are used for the clustering of all technical equipment of the aircraft into corresponding groups and subgroups, being of great importance for certification of any civil aircraft and thus included in the metamodel.

To not only provide a hierarchical differentiation for different aspects of modelling but also a differentiation between different model elements, the RFLP modelling approach is used. It provides clusters and differs elements into four groups relating to requirements, functional, logical and physical aspects. This clustering is especially important in the phase of product development but is also expanded towards all other life phases.

The cabin metamodel shown in Figure 3 builds the sound foundation for defining elements needed to enable modelling along the whole aircraft cabin lifecycle. The element *datatype* serves as an origin for defining different aspects of the product life phase. Different kinds of data are represented by different datatypes, which can be classified as elements on a specific level of a product life phase. The definition of the generic datatype is shown in Figure 4.

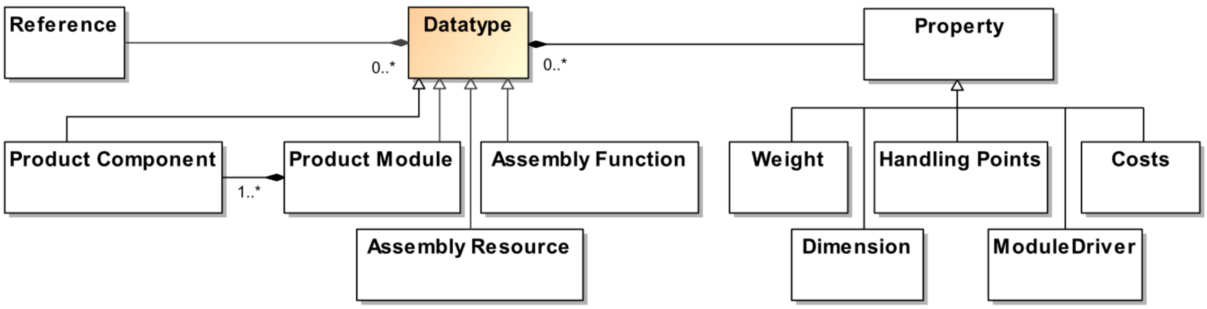

Figure 4 - Metamodel of a datatype and inherited datatype elements

The element *datatype* consists of zero to many *properties* as well as zero to many *references* represented by the composition, respectively. A datatype can further be specified by defining a unique identifier, a name and a description which enables the detailed characterisation of datatypes and inherited elements but is intentionally neglected in this simplified representation. Generic properties that might be used by any datatype can be *weight*, *dimensions*, *handling points*, *module drivers* and *costs*, among others.

One important example of a datatype is the *product module*. It is derived by inheriting the properties of a datatype using the generalization relationship. The datatype product module can be defined by assigning an ID, a name and a description, accordingly. It is specified using different module properties. In this extract *weight*, *interface*, *module driver*, *dimension*, *variance* and *costs* can be mentioned. With this selection of properties, it is clear to see, that properties provide not only physical information like weight or dimension but also product strategical information like module driver, which are used for different optimisation and development aspects e.g. Life Phase-Modularisation [21]. Properties can also be refined which is not depicted here. An example would be the costs-property, wherein it can be distinguished between complexity costs, assembly costs, manufacturing costs and more. As CAD and corresponding models are widely used in product development, linking CAD models with a product module is necessary to take full advantage of the product model. In the case of the product module datatype, the reference element is therefore used to enable referencing CAD models in the cabin model.

Similarly, the assembly resource as another datatype example uses the reference element to reference robot or PLC programs, CAD data or instruction manuals for tools. Different assembly resources such as robots, workers, tools, measuring and test equipment all have different properties such as *handling capacity*, *dimensions* and *costs*. Due to the generic definition and expandability of the generic datatype properties, relevant properties can be added to and inherited from the metamodel. However, all assembly resources have at least one *skill* which is relevant for an assembly task. The skill is a logical modelling element which is strongly related to the datatype *assembly function* as a functional modelling element. Assembly function instances are divided into different subgroups as defined in [22], namely *join*, *handle*, *test*, *adjust* and *auxiliary operations*. Examples for assembly functions out of the handling-subgroup are *store*, *hold* or *move*, which can be further split into subfunctions such as *turn* and *position*. The position function datatype uses properties such as the *location* for the definition of starting point and endpoint of the movement, *speed* or *time* as well as *accuracy*. The assembly functions and the resources that implement them are modelled at a generic level as part of the co-design of the product and assembly system, using the metamodel.

4. Application of the Metamodel on the Example of Assembly Process Planning

The metamodel enables consistent modelling and interlinkage of aspects across all aircraft cabin lifecycle phases. Since assembly is a major cost factor, particularly in the aircraft cabin, and previous errors often only become noticeable here [22], a focus in the *VERDIKA* project is on optimising and automating assembly process planning. A decisive prerequisite for efficient assembly system development and process planning is the co-design of product modules and assembly system resources. When developing variable modules for a modular product system, the associated requirements for the assembly system must be developed at the same time and linked to the corresponding product module. This linkage is depicted in a developed ontology shown in Figure 5, specifying that each module is mounted in the aircraft cabin by realizing different assembly functions.

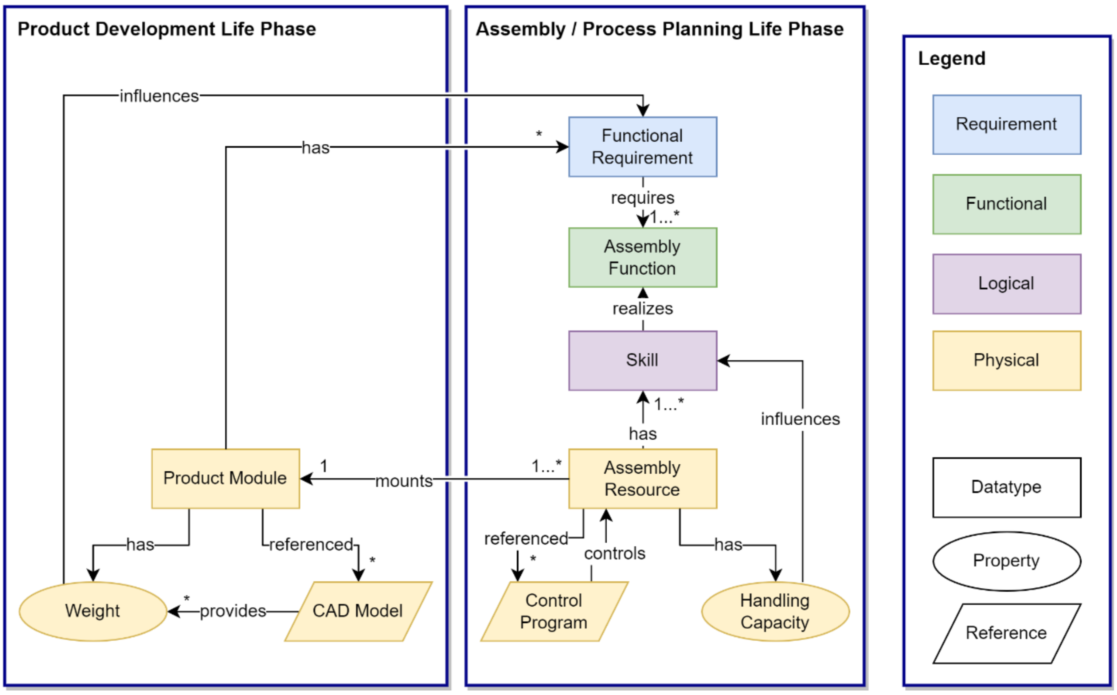

Figure 5 - Assembly process planning ontology

The ontology is built upon the metamodel by connecting the product development and process planning life phases at the cabin level, defining relations between the different datatypes described above representing requirement, functional, logical and physical elements. Multiple ontologies can coexist simultaneously, which makes it possible to describe all relevant processes and life phases according to the syntax of the metamodel. Thereby, a holistic description of the cabin can be achieved. The definition of product modules as interchangeable parts of a customer-specific cabin configuration leads to an obvious linkage to the assembly process planning life phase through functional requirements concerning assembly.

In the example of a hatrack, assembly requirements would be to move and position it at the correct place in the cabin, hold it in place and join it to the periphery by inserting bolts. These requirements specify certain assembly functions, in this example *move*, *position*, *hold* and *insert*. Each function can be realized by at least one assembly resource through its skills as logical elements. The necessary work instructions or control programs of an assembly resource can be linked to the instantiated resources in the assembly system model via references, similar to CAD files in the product development life phase. The generated programs can be simulatively validated and, in particular, the assembly time can be determined, which can be stored as a property in the model and reflected to the customer or the product configuration process. The properties of a product module, which may be also derived from referenced CAD models, have a direct influence on the assembly requirements and thus on the choice of an assembly resource with its properties. For example, positioning a hatrack with the property *weight* of 10 kg cannot be realized by an assembly resource with the property *handling capacity* of 5 kg.

Matching product modules requiring assembly functions with assembly resources providing skills is an elemental part of assembly system configuration. This capability-based allocation of skills of existing possible resources and the required assembly functions can be captured in an assembly sequence model. As the central element of the assembly system configuration, the assembly flow chart links the product modules with their assembly function and the executing and used resources. An excerpt of an example flow chart is depicted in Figure 6.

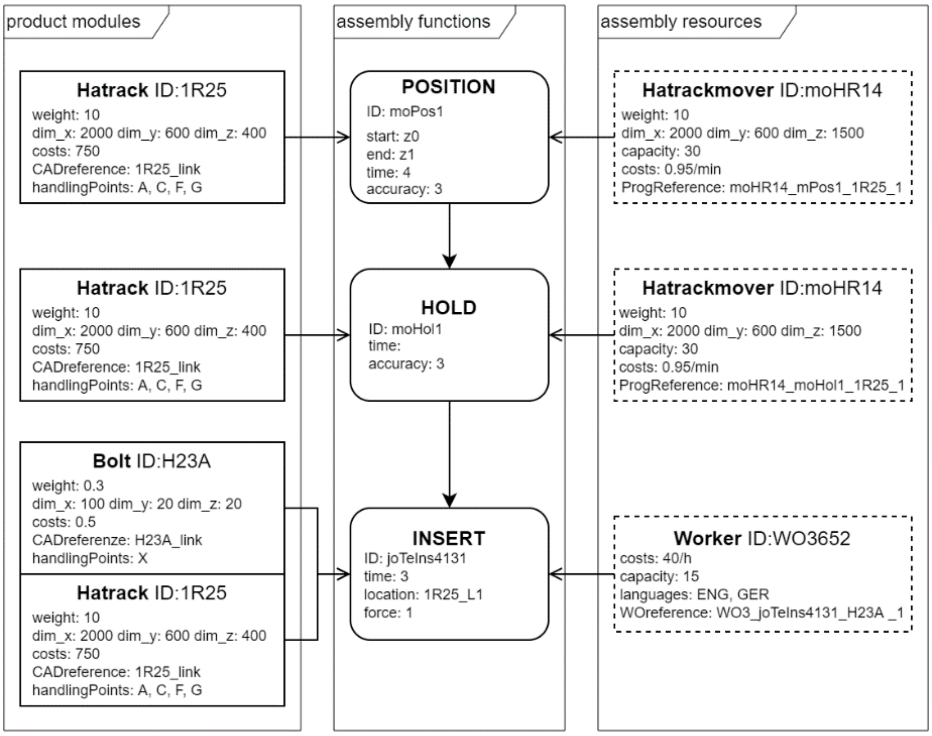

Figure 6 - Excerpt of an example assembly sequence

In the centre are the assembly functions that need to be realised in a sequence from top to bottom. Visible on the left side, the product modules being mounted are linked to the corresponding assembly functions. The assembly functions are realized by assembly resources, visible on the right side, which is connected to the functions via corresponding skills. The example starts with the positioning of a hatrack module at the correct height in the cabin by an automated handling device, called *Hatrackmover* in this case. After positioning, the handling device is required to hold the hatrack in position until further notice. While the hatrack is held in place, an assembly worker shall insert a bolt into the hatrack at a certain position. Apart from the connections of product modules, assembly functions and assembly resources, the example shows the usage of relevant properties of the multiple used datatypes. These include for example a weight specification of the product modules which can be matched with the handling capacity of possible assembly resources. Other properties such as the time allocated for the assembly functions and the costs of assembly resources, specified as money per time, can be used to calculate assembly costs which might influence the choice of product modules in the product configuration process. Furthermore, references to CAD data as well as to control programs or human-readable work orders are stored in the elements of product modules and assembly resources, respectively. Further properties specifying the assembly functions can be directly used for automated configuration of the programs and work instructions.

5. Outlook on Further Applications of the Metamodel

Supporting an economical realisation of a new modular cabin platform and services as well as the individual cabin hardware configurations, a digital end-to-end customisation process is enabled by the common metamodel. A cabin product can be modelled from a product development perspective based on the holistic metamodel to integrate features such as product family and variance considerations and to reduce Head of Version (HoV) costs. After implementing the metamodel, different customisation approaches such as a Digital Offer Catalogue (DOC) can be developed based on it. With the DOC, airlines can configure their cabin HoV and the further production system customisation process is initiated. For each new unique HoV configuration, the corresponding product model needs to be analysed to determine necessary changes to the assembly system. Based upon a model library with digital representations of the available assembly system resources and their skills, these logical and parametric changes will be implemented in the instantiated assembly system model. Due to the links and references enabled by the metamodel, a configuration can not only take place on the top-level logical model, but also on a specific robot or PLC program. These can then be used for detailed simulations generating information about feasibility and assembly times, allowing improved design decisions once fed back to the customer and the product engineers.

With the resulting configured cabin and assembly system models, there are digital representations of the planned specific cabin and assembly system configurations at the time of assembly. With regards to the concept of Digital Twins, these are ideal datasets that can be used as so-called Digital Masters and, thus, be the base for the instantiation of Digital Twins of said specific aircraft and its assembly system [23]. With its inherent life-phase overarching approach, the cabin metamodel lays the ground to manage and track a diversity of data (like product models, process models, programs) and provide the different information to all concerned stakeholders of an aircraft cabin lifecycle. This information is valuable for the ongoing maintenance and especially retrofit processes, as the planning of cabin-conversions currently faces uncertainties regarding the actual state of the aircraft [24]. The exchange of historical and real-time data via a clearly defined format enables other developers and stakeholders to keep up to date with the product's current state. With the growing formalized number of datasets, the Digital Twin, enabled by the cabin metamodel, can then also be used to apply services like data analytics and machine learning algorithms which can provide information about historical events and states of various aircraft cabin products for example for predictive maintenance.

The development of new services is explicitly supported by the metamodel definition in addition to classical product development and process planning. In the development phase, requirements from all stakeholders can be acquired, for example, to develop multiple use cases for a future cabin management system enabling new services for airlines, passengers and crew. To implement the various services, a network cabin platform architecture model will help to connect all the data sources and sinks of an aircraft cabin. Through the facets provided in the metamodel, particular focus can be placed on the safety and security of, among other things, the critical interface to the cabin management system. A cyber-physical smart system can be developed by integrating sensors and actuators to record and control processes in the physical world. It connects them to people via information technology solutions and human-machine interfaces (HMI), for example, passenger mood adaptive lighting. The used data models can be excessed to view the development progress, historical events, real-time data operations and predictions for concerned departments. Additionally, business analytics tools built on machine learning technologies can help to deliver real-time viewing into business Key Performance Indicators (KPIs) and improve IT and business collaboration. Linking all the life cycle phases in the context of security in the metamodel also facilitates assessing security risks which arise in the assembly or maintenance. Hereby, the security analysis comprises not only the product but also risks that occur in other life phases of the product and different stakeholders that are included, e.g. original equipment manufacturers (OEMs).

6. Summary

Against the backdrop of growing competition in the aviation sector and the resulting trends towards more individualised aircraft cabins, new smart services and automated production, the project *VERDIKA* lays the foundation for greater networking in the cabin itself, as well as in the product lifecycle phases from product development through production to the MRO business. In this paper, a holistic metamodel for a digitally connected, networked aircraft cabin is proposed which interconnects stakeholders in various life cycle phases. Every product life phase can be described with different datatypes representing requirement, functional, logical and physical elements on different system levels. This enables coherent modelling of all aspects in a multidimensional holistic model. Special focus is set on the development and implementation of smart services while strongly considering security requirements, as well as on the interconnection of product development and assembly process planning. The common cabin metamodel with the underlying ontologies describing product and assembly process planning, which is the basis for an efficient co-design. The creation of links between model elements of different domains and life phases is the most important benefit of the metamodel, enabling traceability and verification and preventing miscommunication and errors. Furthermore, the metamodel provides the possibility of establishing a common data source, which is an important factor when data changes rapidly as in the early stages of co-design and co-customization including domain-specific decisions. With all its benefits this holistic approach can improve the foundation for the implementation of software platforms for data management or PLM systems within the whole product lifecycle. It can also be used as a digital master for upcoming Digital Twin approaches.

Building upon the metamodel, model-based and data-driven technologies offer various methods, tools and implementation options that make the development and use of complex systems more manageable. The general approach proposed in this paper is thereby not limited to the aircraft cabin, but might be expanded to cover the whole aircraft or other complex systems. The generic formulation of life phases, hierarchy levels, facets, datatypes and element types provide the possibility of generic application as well as easy extension.

Acknowledgements

The presented research is funded by the Federal Ministry for Economic Affairs and Climate Action (BMWK) as part of the sixth Federal Aeronautical Research Program (LUFO VI-1).

Contact Author Email Address

mailto: markus.berschik@tuhh.de

Copyright Statement

References

[1] AerQ GmbH. AERQ Takes Digital Transformation of Aircraft Cabins to a New Level. Online available *https://www.aerq.com/news/aerq-takes-digital-transformation-of-aircraft-cabins-to-a-new-level*, accessed on 22.02.2022.

[2] Sinnwell C. Methode zur Produktionssystemkonzipierung auf Basis früher Produktinformationen: ein Beitrag zur Integration von Produktionssystemplanung und Produktentwicklung unter Einsatz des MBSE. *Produktionstechnische Berichte aus dem FBK*, Technische Universität Kaiserslautern, Dissertation, 2020.

[3] Schüppstuhl T, Schlosser C. Anwendungsfeld Flugzeugbau. *Handbuch Industrie 4.0. Geschäftsmodelle, Prozesse, Technik.* Hanser, pp 635–651, 2017. https://doi.org/10.3139/9783446449893.023.
[4] Brusa E, Calà A, Ferretto D. *Systems Engineering and Its Application to Industrial Product Development*, 1st edition, Springer International Publishing, 2018. https://doi.org/10.1007/978-3-319-71837-8.
[5] Grösser S N, Reyes-Lecuona A, Granholm G. *Dynamics of Long-Life Assets*. 1st edition. Springer International Publishing, 2017. https://doi.org/10.1007/978-3-319-45438-2.
[6] Muggeo C; Pfenning M. Die Rolle von MBSE und PLM im Industrial Internet. *Tag des Systems Engineering (TdSE)*. Hanser, S. 279–287, 2016. https://doi.org/10.3139/9783446447288.028.
[7] Berschik M C, Laukotka F, Krause D, Ansatz eines holistischen Datenstamms zur modellbasierten, domänenübergreifenden Nutzung in der Produktentstehung, *Proceedings of the 32th Symposium Design for X (DFX2021)*, Tutzing, Germany, 2021.
[8] Seiler F M, et. al. MBSE zur Unterstützung der Produktentwicklung von modularen Produktarchitekturen. *Produktentwicklung und Konstruktionstechnik*. Vol. 10, Krause D, Springer, pp 111–134, 2020.
[9] Stachowiak H. *Allgemeine Modelltheorie*. Springer, 1973.
[10] Aßmann U, Zschaler S and Wagner G. Ontologies, Meta-models,and the Model-Driven Paradigm. *Ontologies for Software Engineering and Software Technology*. Springer, pp 249–273, 2006.
[11] Alt O. *Modellbasierte Systementwicklung mit SysML*. Carl Hanser, 2012. https://doi.org/10.3139/9783446431270.
[12] SAE. Guidelines for Development of Civil Aircraft and Systems. *SAE-ARP-4754A*, 2010.
[13] Hintze H, God R. Airworthiness Security: Zuverlässigkeit ist nicht mehr genug!, *Ingenieurspiegel*, 2020.
[14] Baughey K. Functional and Logical Structures: A Systems Engineering Approach. *SAE Technical Paper Series*. SAE 2011 World Congress & Exhibition, Warrendale, United States, 2011. https://doi.org/10.4271/2011-01-0517.
[15] Kleiner S, Kramer, C. Model Based Design with Systems Engineering Based on RFLP Using V6. *Smart Product Engineering*. Springer, pp 93–102, 2013. https://doi.org/10.1007/978-3-642-30817-8_10.
[16] Li T, Lockett H and Lawson C. Using requirement-functional-logical-physical models to support early assembly process planning for complex aircraft systems integration. *Journal of Manufacturing Systems*, Vol. 54, pp 242–257, 2020. https://doi.org/10.1016/j.jmsy.2020.01.001.
[17] Fourgeau E, et. al. System Engineering Workbench for Multi-views Systems Methodology with 3DEXPERIENCE Platform. The Aircraft RADAR Use Case. *Complex Systems Design & Management Asia*, Vol. 426, pp 269–270, 2016. https://doi.org/10.1007/978-3-319-29643-2_21.
[18] Object Management Group, Meta Object Facility (2.5.1), Online available *https://www.omg.org/spec/MOF/2.5.1/#document-metadata*, accessed on 14.02.2022.
[19] Estublier J, Ionita A D and Vega G. A domain composition approach. *Proceedings of the International Conference on Software Engineering Research and Practice*, Las Vegas, USA, Vol 1, pp 389-395, 2005.
[20] Scheuermann C S. A Metamodel for Cyber-Physical Systems. PhD thesis. Munich Universtity of Technology, 2017.
[21] Krause D, Gebhardt N. *Methodische Entwicklung modularer Produktfamilien*. 1st edition, Springer, 2018. https://doi.org/10.1007/978-3-662-53040-5.
[22] Lotter B, Wiendahl H-P. *Montage in der industriellen Produktion*. Springer, 2006. https://doi.org/10.1007/3-540-36669-5.
[23] Laukotka F; Hanna M and Krause D. Digital Twins of Product Families in Aviation based on an MBSE-assisted approach. *Procedia CIRP*, Vol. 100, pp 684-689, 2021. https://doi.org/10.1016/j.procir.2021.05.144.
[24] Moenck K H W, et. al. Towards an Intelligent Digital Cabin Twin to Support an Aircraft's Retrofit and Base Maintenance, *SAE Technical Paper 2022-01-0046*, 2022. https://doi.org/10.4271/2022-01-0046.