33rd Annual INCOSE international symposium
hybrid event
Honolulu, HI, USA
July 15 - 20, 2023

An Approach for Linking Heterogenous and Domain-Specific Models to Investigate Cabin System Variants

Mara Fuchs
German Aerospace Center (DLR)
Hein-Saß-Weg 22, 21129 Hamburg, Germany
+49(0)402489641371
mara.fuchs@dlr.de

Yassine Ghanjaoui
German Aerospace Center (DLR)
Hein-Saß-Weg 22, 21129 Hamburg, Germany
+49 (0)40 2489641 384
yassine.ghanjaoui@dlr.de

Jörn Biedermann
German Aerospace Center (DLR)
Hein-Saß-Weg 22, 21129 Hamburg, Germany
+49 (0)40 2489641 376
joern.biedermann@dlr.de

Björn Nagel
German Aerospace Center (DLR)
Hein-Saß-Weg 22, 21129 Hamburg, Germany
+49 (0)40 2489641 302
bjoern.nagel@dlr.de

Abstract. This paper presents an approach to link heterogeneous and domain-specific models. The background of this research is the complete investigation and comparison of cabin system variants, where many different aspects have to be represented. These include functional requirements, safety regulations, and geometric properties (e.g. installation space). However, these cannot always be validated or represented with just one model, as different levels of detail are required. Therefore, different discipline models have to be created, which in turn increases the complexity as a whole. Furthermore, the system to be represented by the models, such as the aircraft cabin, is already complex in itself. The many dependencies among each other and subsystems make it difficult to integrate new variants or technologies (e.g. liquid hydrogen) into the existing system architecture. The approach presented here therefore shows how the data and models of the different disciplines can interact with each other in order to be able to investigate variants holistically. This is demonstrated using the design of hatrack variants for a commercial aircraft.

Introduction

The customer's desire for ever more individualized products continues to grow. A high customizing level does not always lead to higher satisfaction or performance, so that mostly product variants (e.g. seats of the same vehicle may differ in terms of some of their functions) are developed to keep quality high and costs low (Hedge, et al. 2005). One area in which customizing is very high is the development of aircraft cabins. The reasons for this are the shorter service life of around seven years compared to the aircraft itself and the possibility for airlines to implement their personal branding. Since the customer, in this case the passenger, interacts directly with the cabin, the strongest impression can be made there: for example, through functionally enhanced seats, dynamic lighting or entertainment systems (Ackert 2013). In order to implement the individual wishes of the airlines, the aircraft manufacturers usually offer a standard specification, which can then be customized by the airlines with pre-selected options from a pool (Ackert 2013). However, these are not always sufficient, so the airline wants more flexibility and easier reconfiguration. Nevertheless, the integration of the cabin has an influence on other systems in the aircraft, such as electrics or structure. Thus, the goal is to link functional requirements with physical design domains (Du, et al. 2006). Moreover, certain modifications must be made at an early stage and comply with industrial standards in terms of on-time and on-quality (Richter, Walther 2017).

In order to fully map, evaluate, and subsequently benchmark product variants, a flexible system architecture is needed in which different disciplines can be considered. Moreover, different model types are required depending on the degree of fidelity (Figure 1). For example, the Systems Modeling Language (SysML) is suitable for system architecture specifications and for requirements traceability. However, if geometric design and placement of system components are required, a SysML model alone is not sufficient because it cannot depict 3D modeling. Another model for the geometrical configuration of the components is necessary, e.g. Blender, Matlab, CATIA. If further aspects have to be checked by simulations, e.g. FEA (Hesse et al. 2021), in turn models with higher accuracy and other tool environments (e.g. HyperMesh, Patran) are required. In addition, there are also requirements that cannot be quantified and modeled accordingly. These are, for example, the visual appearance or the feeling of comfort. But these aspects must also be considered and evaluated in order to produce a product that satisfies all stakeholders' needs. One possibility is the early connection and presentation of the variants in a virtual environment, in which the user can experience the product in scale 1:1. Therefore, interoperability between the models must be ensured and properties as well as model objects must be traceable and available, to enable a faster and more holistic design of the systems considering all stakeholder needs. Depending on the system analysis, a different abstraction level (top-down) is required, which can be freely selected in each model (SysML for requirements and architecture, Matlab for geometry, virtual reality (VR) high resolution for design).

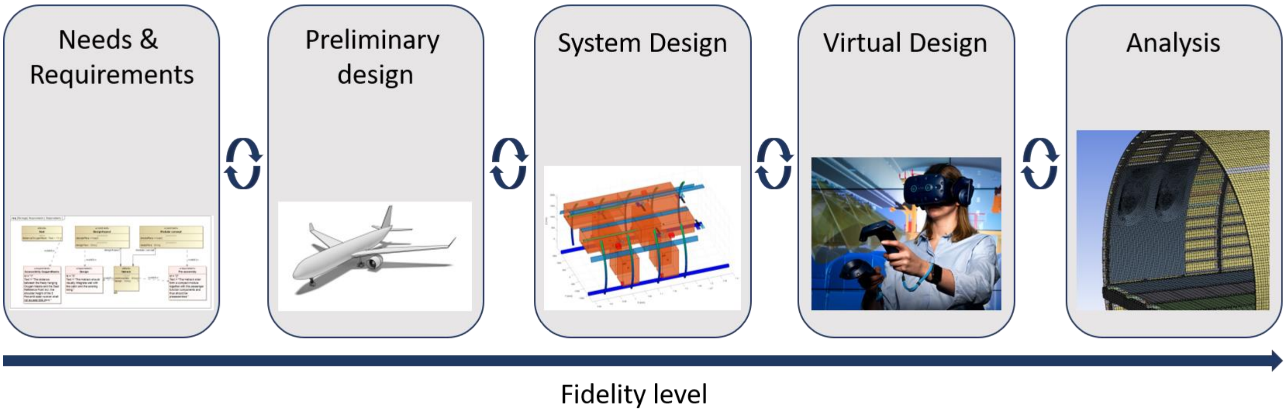

Figure 1. Different fidelity levels of models during the conceptual design process of the aircraft cabin.

All in all, digital continuity between all heterogeneous domains is needed to bring all disciplines involved in the process into an exchange at an early stage and to interlink them. In this way, when changes are made to the system as a result of a wide range of variants or additional requests in the context of customizing, the effects can be identified early on, impacts calculated, and evaluations performed. This research paper describes how the data between the domain-specific models are exchanged and linked, enabling a holistic investigation of variants in the aircraft cabin. In the next section a literature review is presented giving an insight into different types of models and how they support variant modeling. Finally, the approach and methodology that were developed are demonstrated using the design of overhead stowage compartment (OHSC) variants for a commercial aircraft as an example.

Literature review

In the literature a number of approaches and methods already exist for either modeling variants within a model or coupling models of different domains with each other. In addition, SysML v2 will include variant modeling in its specification by introducing new elements to support variant management, indicating the importance and necessity of variant modeling. For modeling variants within a model, there is, for example, the VAMOS method defined by Tim Weilkiens for modeling variants with SysML. Here, the SysML language is extended by a concept for variant modeling through additional stereotypes such as variant points or variation (Weilkiens 2016). One application of the VAMOS method is the systematic development of a federated database infrastructure and management system by Melzer et al. (Melzer et al. 2022). Melzer et al. used VAMOS to model three different database types so that project teams can develop their own database independently using the same infrastructure. Another example of modeling variants within SysML has been given by Forlingieri and Weilkiens (Forlingieri and Weilkiens 2022) in the area of Model-based Product Line Engineering from an industry perspective. Two variant modeling approaches were compared, whereby using VAMOS requires more effort in the modeling of variation but provides a better management and containment of the variability within the model (Forlingieri and Weilkiens 2022).

Frischen et al. used a rule-based configurable bill of material to show how variant management can benefit from a consistent data basis. The complex bill of materials serves as a basis for establishing control mechanisms for variant management and for mapping the status of business decisions at an early stage (Frischen et al. 2019). In addition, the concept of the 150% model has become established in product development. Here, several variants of a product are modeled in advance and then a variant is selected on the basis of different requirements. Thus, all model elements of existing variants of a system are contained in one system model (Menninger et al. 2022). This promises consistency within the model and the reduction of development artifacts (Menninger et al. 2022). However, geometric dependencies are already predefined, so that the system components match for each component without an additional investigation which limits the examination of new variants.

Some approaches have already been developed for linking different domains and thus enabling the modeling of variants. For example, Müller et al. (Müller et al. 2020) have developed an automated approach to link functional models with CAD (computer-aided design) models and to generate alternative 3D solutions depending on the function using an assembly algorithm. Schumacher and Inkermann in turn linked model elements between the language SysML and CAD. Information is exchanged via a standardized data model based on STEP AP 233[1] and XMI[2] (Schumacher and Inkermann 2021). Without the coupling of domain-specific models, the continuity and consistency of information is affected and thus domain-specific interactions are only weakly represented. Mahboob (Mahboob 2021) showed the coupling of virtual reality with Matlab Simulink and SysML to generate real-time product simulations in VR and demonstrated this using a vacuum cleaner. Here, the evaluation is not only conducted by the visualization, but also includes the behavior of the product. All in all, several approaches exist in the SysML as well as in other domains to describe variants and to connect two model domains. Nevertheless, further degrees of fidelity for a holistic view and integration of variants are missing, implying the necessity to link several domains together.

According to the literature review, it can be noted that the coupling of different domains is necessary to enable a holistic investigation of a system. The shift from individual models that have to be linked manually to an automatically interconnected model architecture is shown in Figure 2. Thereby, it is

[1] STEP-based data exchange standard consistent with standards in CAD, structural and engineering analysis.
[2] XML Metadata Interchange is a standard for exchanging metadata information via XML.

important to transfer the data between the different models without loss and to maintain the consistency of the parameters. In addition, the level of detail increases with the modeling (see Fig.1). Thus, the SysML is suitable for modeling requirements and stakeholder wishes as well as for the functional and logical layout of the system components. However, geometric investigations such as installation space integration or optical evaluation of the overall composition must be carried out in other model environments. Furthermore, these model approaches must enable the modeling of variants. Therefore, in this paper an approach is shown in which system variants can be modeled and then configured, evaluated, and compared with each other by coupling them with other domains according to multiple criteria. This gives the opportunity to customize products and adapt them to specific stakeholder wishes as well provides the flexibility to react quickly to new markets and technological developments. This raises the need for an approach of a flexible system architecture to generate, evaluate, and benchmark variants, in order to identify the most beneficial.

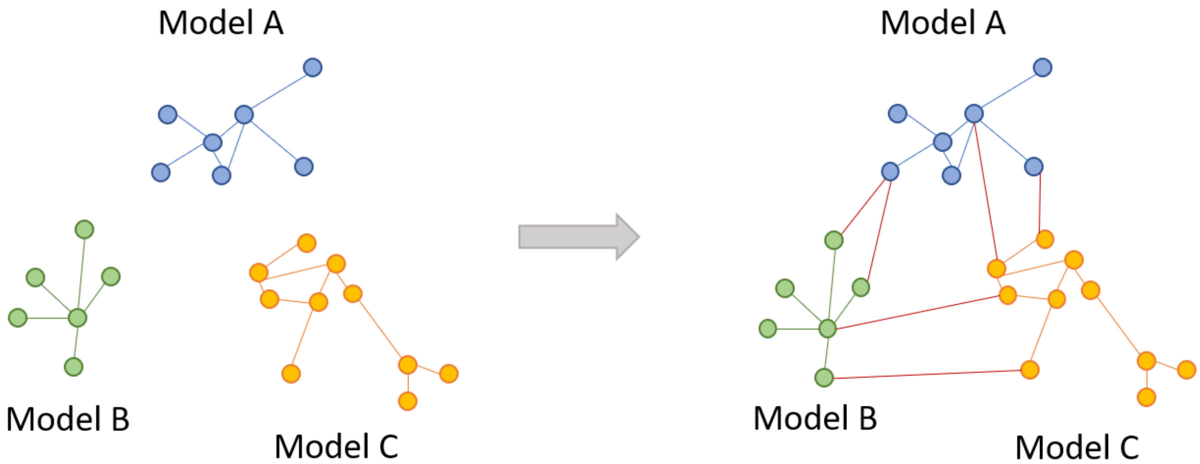

Figure 2. Graphical representation of linking models from different domains for modeling variants.

Model Setup and Methodology to Link Models

The existing wide tool landscape and the predominant use of commercial tools in the industry remain challenges. One reason for this are the many disciplines that require different levels of detail depending on the development stage. Among the tools used in aircraft development, two development environments like Cameo Systems Modeler and Matlab/Simulink are established in a way of an industry standard and are widely used. The Cameo Systems Modeler is used for requirements management, but also offers other advantages such as modeling of functionalities as well as tracing and visualization of all aspects of a system within the tool. Therefore, it is mainly used in the beginning of the development stage. Matlab, on the other hand, has established itself as a numerical computing platform for analyzing data and developing algorithms. It is used to work out details and perform analyses, especially in a later design stage. Another development environment is Unity. This enables the construction of a virtual environment in which cabins can be tested with the aid of virtual reality. In recent years, VR technology has become increasingly important in the aircraft design process. Examples include the integration of virtual reality to the conceptional and functional design process of cabin systems shown by Fuchs et al. [Fuchs, Ghanjaoui et al. 2022] and the evaluation of regional aircraft cabin interiors [Crescenzio et al. 2022] and business jet aircraft cabin interiors [Crescenzio et al. 2019] shown by Crescenzio et al. Connecting Unity to the cabin design process thus enables the early integration of designers into the process as well as providing a platform for exchange with their concept studies. In addition, virtual reality enables the testing of innovative cabin concepts by probands and the integration of humans into the design process, hence its frequently appeal at the end of the development process. An (early) networking of all disciplines in the conceptional design enables an exchange between the disciplines through a digital thread and consequently improves development times and product quality.

As a result of the digital link, changes and their effects can be tracked and made visible more quickly. This is particularly useful in variant modeling, where variants can be designed and then compared with each other according to different criteria. The applicability of the developed methodology is demonstrated with the tools described above.

In the following, the methodology for holistically designing, investigating and subsequently evaluating variants is presented. For this purpose, a SysML model is coupled with a Matlab model and the Unity development environment model. Figure 3 shows the graphical representation of the data connection between the three domain models. First, the system architecture and the requirements are modelled in SysML. Here, the MBSE environment Cameo Systems Modeler (Version 2021) is used. Initial parameters, such as the number of passengers, are read from an XML data file. This file was created as part of the Overall Aircraft Design (OAD) and is used to store and exchange information, such as the top-level requirements for the aircraft, in distributed environments. This allows the cabins to be configured according to conceptual aircraft configurations and to respond to changes in the aircraft structure. Subsequently, the SysML objects for the cabin are instantiated according to the initial parameters. Following, the objects are transferred to Matlab (Version R2022b). Here they are placed geometrically and further property values are filled in the objects (e.g. position in x, y, z). Then, in an intermediate step, the simple geometry models are exchanged with high-fidelity 3D geometry models. For the high-resolution modeling of the cabin, the open-source 3D computer graphics software Blender (Version 3.3.1) is used. The final cabin will be automatically transferred to the virtual reality environment Unity (Version 2018.4.27f1). There, the design can be interactively explored and evaluated. Throughout the entire process, the objects in each domain are uniquely assignable via the ID. This ID is used as a recognition feature between all domains, whereby each object and its links are clearly identifiable and allocable to ensure consistency. This enables generated values from Matlab and Unity to be uniquely assigned to the objects in Cameo and exchanged without loss. These values are then used for the final verification of the requirements.

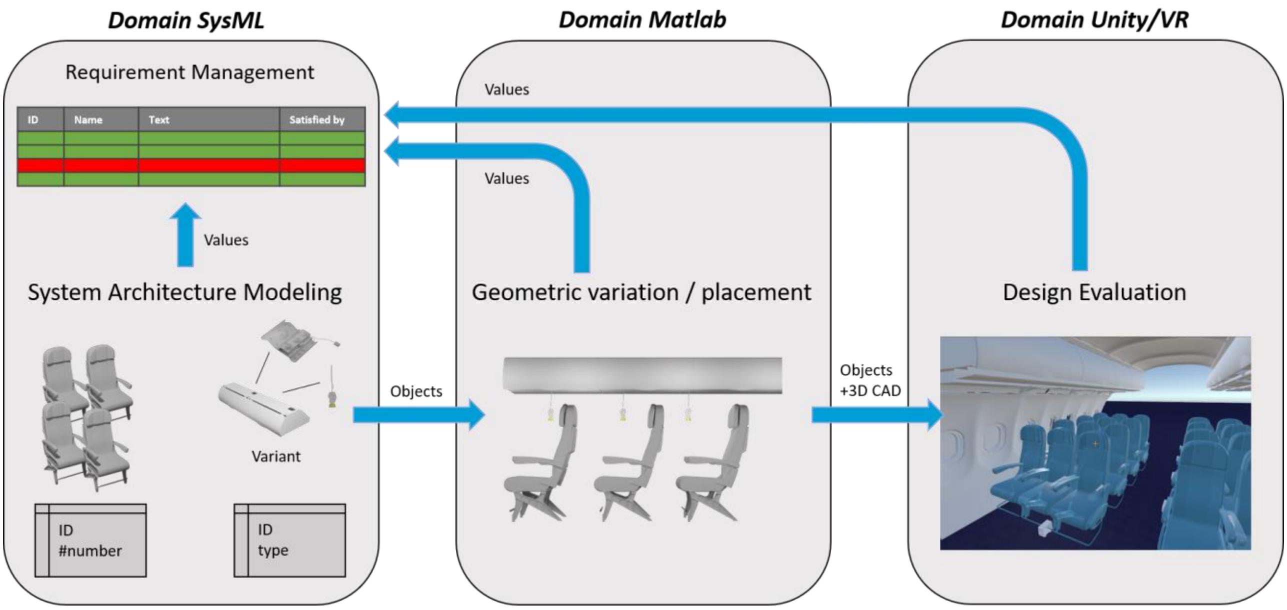

Figure 3. Graphical representation of the three domain models and the data connection between them.

The following subchapters describes the SysML model and the data transfer in more detail. A more detailed description of the methodology setup in Matlab can be found in [Fuchs, Ghanjaoui et al. 2022] and the virtual reality environment is explained in [Fuchs, Beckert et al. 2022].

System Architecture and Requirement Model in SysML

The SysML model is used to trace and validate the requirements and to create the system architecture. Based on the requirements for the cabin concept under development, such as which hatrack variant will be studied and how many passengers will be transported, the associated system architecture is created. Depending on the object type, different value properties are filled with information. As an example, based on the number of passengers to be transported, the number of seats will be instantiated. Depending on the hatrack variant, different architectures are created in addition to the varying geometric properties such as construction dimensions. The regular baggage compartment requires the passenger service channel to attach the passenger service functions (PSF). This includes the oxygen masks, the individual air ventilation and the passenger service unit. In a recent version, however, this is no longer needed. Due to a compact modular design, the passenger service functions can be attached directly to the baggage compartment. Therefore, depending on the variant selection, the corresponding subsystem components are also instantiated. Finally, this type of objects will be filled with a unique ID to ensure consistency. The block definition diagram with the cabin architecture and the described relationships between them is shown in Figure 4. In this study, three different hatrack variants are considered; the regular, large, and extra-large variant. The challenge in customizing the cabin for the hatrack variants is the different system architectures. The subcomponents vary geometrically and in their composition. For example, the number of passenger service systems associated with each overhead bin depends on the seat layout. In business class only one PSF system can be integrated per overhead bin due to the large seat pitch, while in economy class up to three PSF systems belong to one overhead bin. Furthermore, the integration of each variant into different aircraft types such as A320 or A350 needs to be examined to find the optimal hatrack variant that fits most aircraft configurations. All these challenges can be addressed with the developed approach shown in this paper.

For the evaluation of the individual cabin concepts with the different hatrack variants, the system requirements are reviewed. Figure 5 shows the requirements diagram for the cabin. In this example, three different requirements are examined. The first is a safety regulation from the CS-25[3], which states that the seated passenger must be able to reach the oxygen masks easily in case of an emergency. This was further strengthened by taking the grip range of the 5th percentile Asian woman[4]. This ensures that every passenger is able to reach the oxygen masks in case of emergency. The value for checking this is supplied by the seat itself. During the geometric placement of the seat in Matlab, the distance between the freely hanging oxygen mask and the reference shoulder point of the seated passenger is measured and returned into the SysML model. The second requirement is the verification of a modular construction of the hatrack. All passenger service function parts should be located within the dimensions of the baggage compartment. This is also checked in Matlab during the geometric design and transferred as a property value named *preAssembly* to the hatrack objects into the SysML model. The third requirement examines the design aspect. The aim is to ensure that the 3D construction of the selected hatrack is well integrated into the cabin. For this purpose, the user can experience the cabin in a 1:1 scale in the virtual environment and then transfer his or her feedback on perception and clashes as a property value named *design* back to the SysML model.

[3] Certification Specification for Large Aeroplanes released by EASA.
[4] DIN CEN ISO/TR 7250-2 (DIN SPEC 91279) – August 2013.

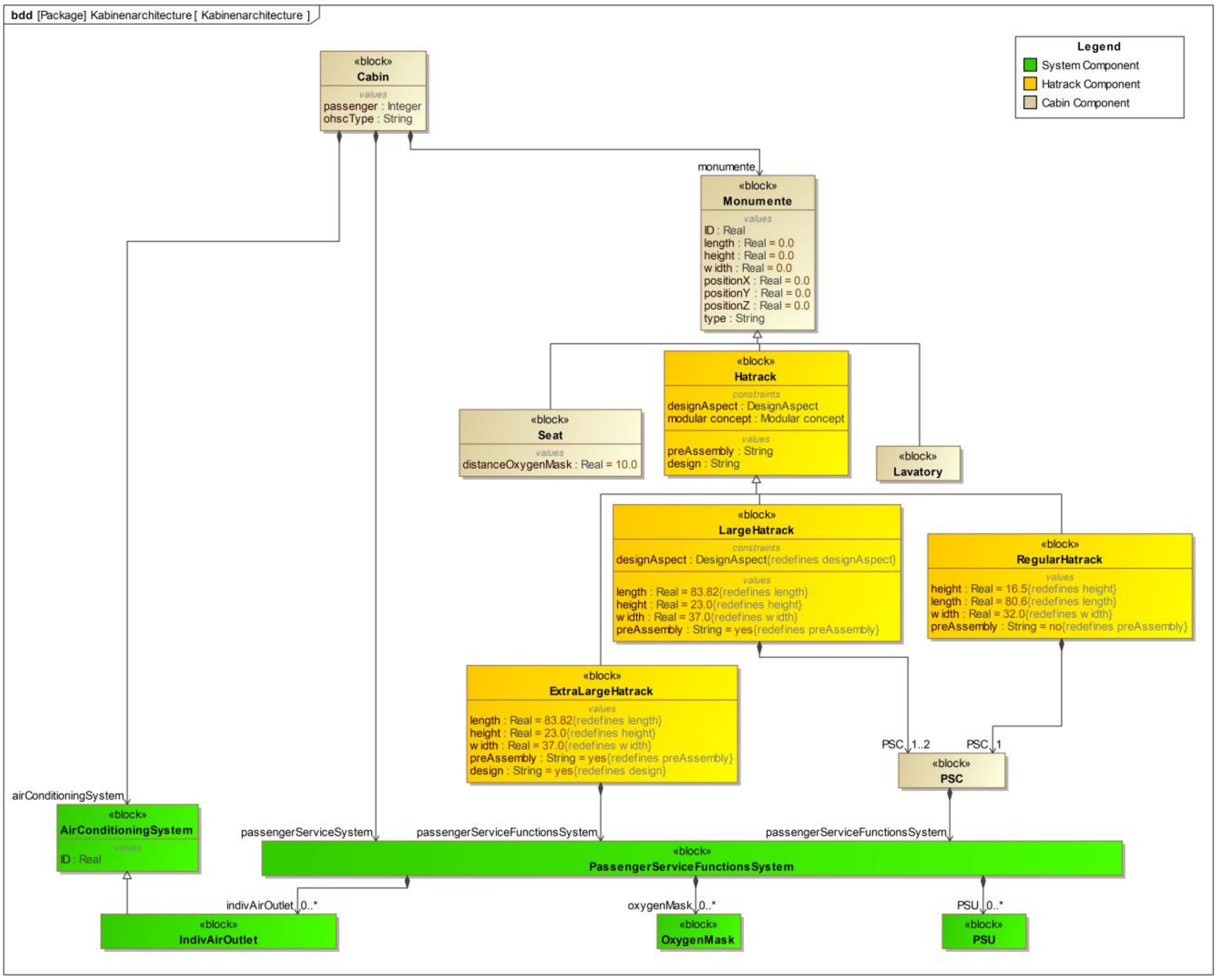

Figure 4. Cabin system architecture using the block definition diagram.

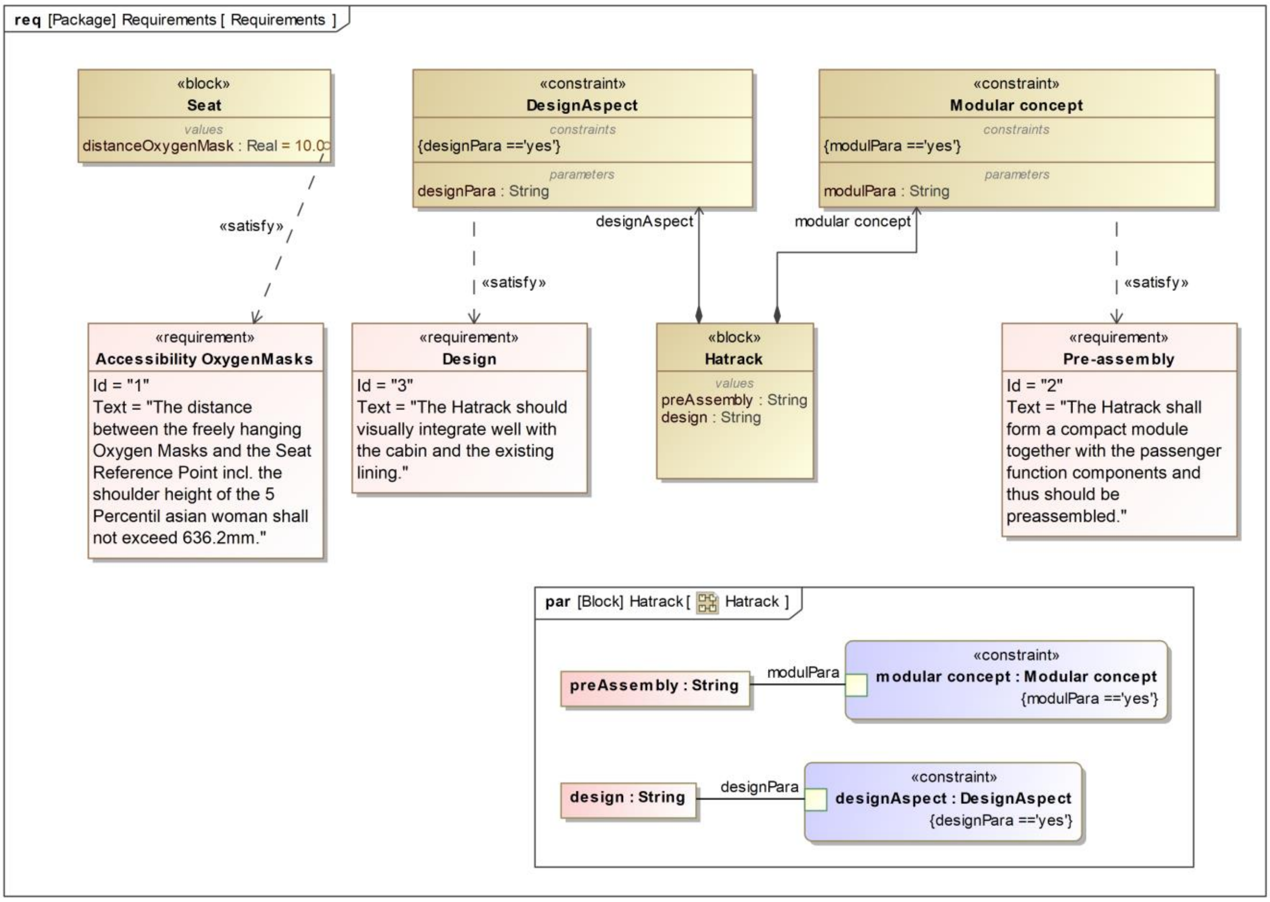

Figure 5. Requirement diagram for the cabin.

Connecting Data between domain-specific Models

The data transfer between the individual models is partially automated. The process starts with the system architecture and the instantiation of the SysML-objects in Cameo. Figure 6 shows the corresponding activity diagram, which is executed in the context of an Airbus 320 family. The element *opaque behavior* is used for instantiating the objects as well as for importing and exporting data. The opaque behavior offers the possibility to use different programming languages or external tool environments to interact with. First, initial parameters delivered by the overall aircraft design are read from an XML file and all cabin components are instantiated with the opaque behavior :createLOPA, according to the LOPA (Layout of Passenger Accommodations). The instantiated SysML-objects are stored in Cameo in part properties. Next, Cameo reads itself and searches for all existing objects (:detectObjects) in order to pass them to Matlab. This is done with the opaque behavior :exportMatlab. Hereby, the type and the ID of the object are identified. Subsequently, the corresponding object classes are called in Matlab and the same objects are created in Matlab's workspace. For example, if a seat object is created in Cameo, an object of the type seat is also created in Matlab. The class definition in SysML and Matlab are the same. Finally, further values are passed (length, width, height). The next step is the geometric placement, design and evaluation of the cabin objects in Matlab. As soon as this process step is completed, Cameo imports the newly generated data with the opaque behavior: importMatlabUnity. First, all objects in Matlab are read and their newly filled property values are passed back to the already instantiated objects in Cameo. As an example, Figure 7 shows the mapping of the property value *preAssembly* between Matlab and Cameo. For the verification of a compact modular construction of the hatrack and its subcomponents, a three-dimensional assembly space check is executed in Matlab. This checks whether all subcomponents of the system are within the construction dimensions. If this is the case, as in Figure 7, the parameter *yes* is passed. In Cameo, the value *preAssembly* of the corresponding object now appears with the value *yes* and is highlighted in green. This is the case because the constraint and thus the requirement for a compact modular construction is satisfied. The entire checkup is done for every object of type hatrack.

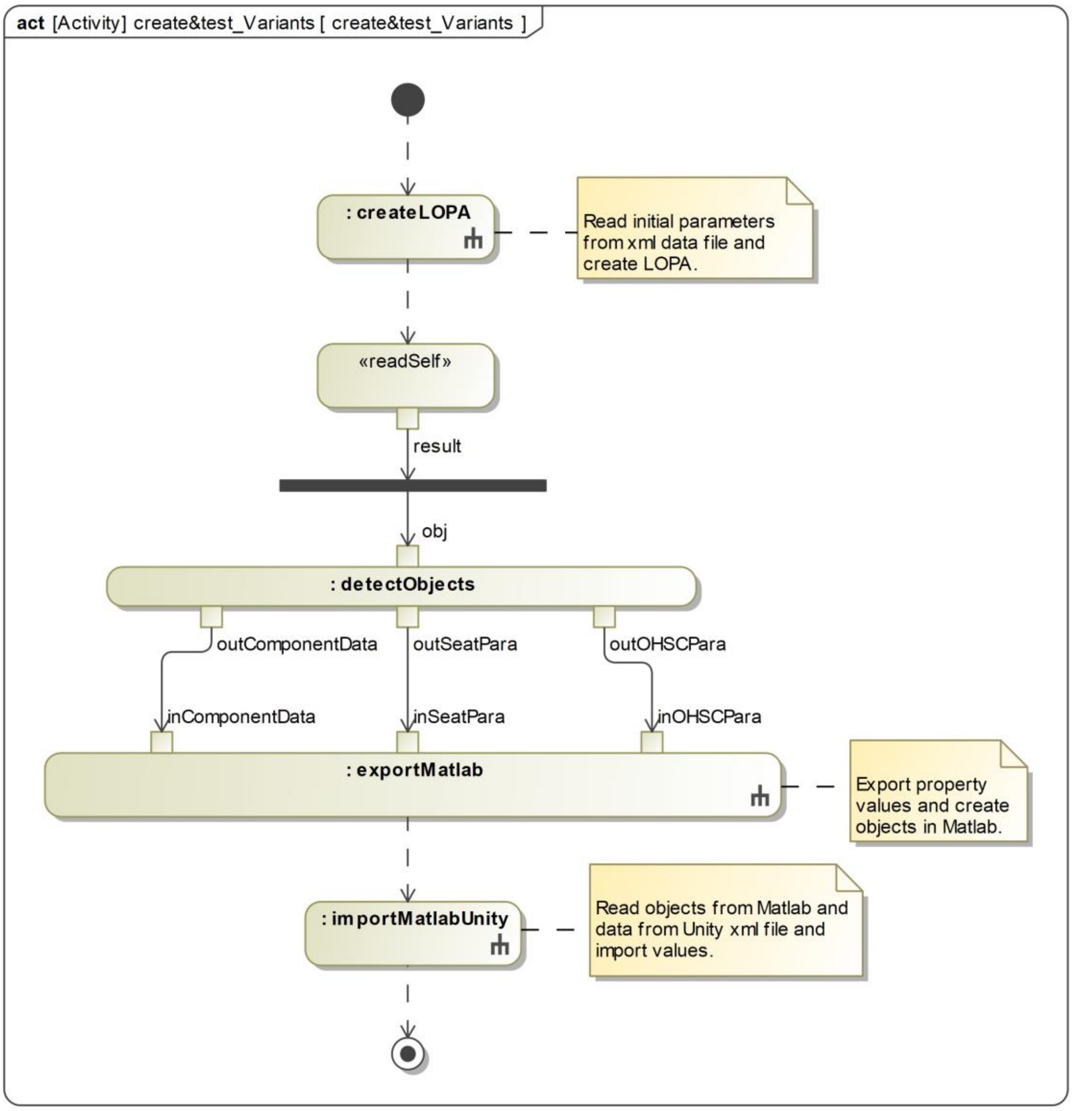

Figure 6. Activity diagram showing the sequence of the functions import initial parameters, instantiate objects, export of objects to Matlab and import of Matlab data.

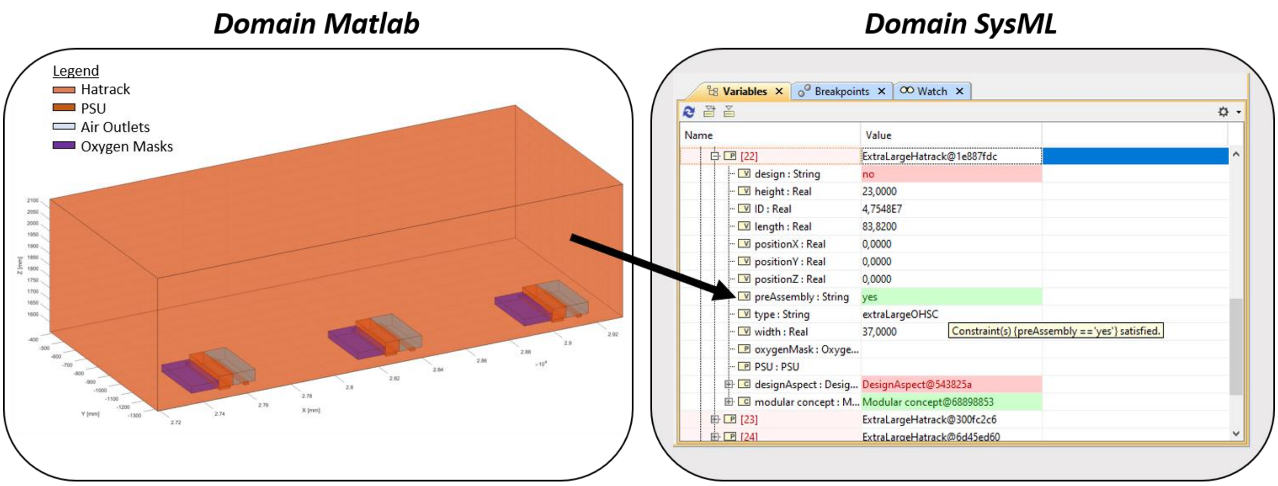

Figure 7. Modul check for the hatrack and value assignment between Matlab and SysML.

Second, the design check performed in the Unity domain is visualized in Figure 8. Here, the user can interact with a virtual panel using an HTC Vive controller. When the user clicks on the hatrack with the controller, the panel A opens to start the design review. When the user selects the *Design Review* Button, a second panel pops up (panel B). There, the design can be marked as rejected or approved. In the background, the corresponding string values are written to an xml file so that they can be re-imported into the SysML domain. At last, the value *design* appears in Cameo with the value *yes* and is highlighted in green.

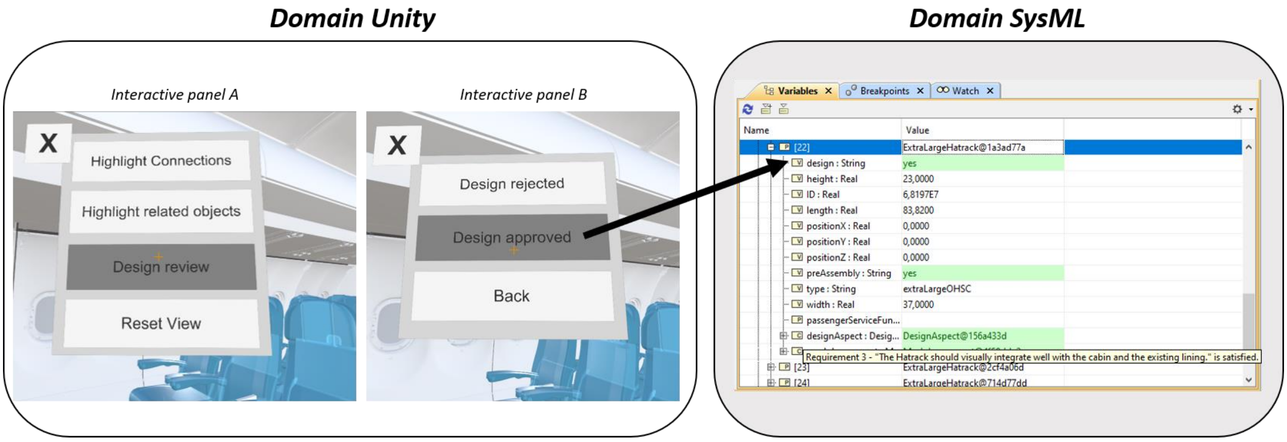

Figure 8. Design check for the hatrack and value assignment between Unity and SysML.

At this point, the objects in Cameo also have all the property values as in Matlab and in Unity. Finally, the overall check of the requirements can be performed.

Use Case Demonstration: Overhead Stowage Compartment Variants

To demonstrate the method, three different overhead stowage compartment variants are examined. Figure 9 shows the three variants in a 3D view and a front view. The first variant is the regular OHSC and state of the art. The second variant is the large OHSC. Compared to the first variant, this one offers more storage space and the possibility to distribute the passenger service functions over two service channels. In addition, it offers a modular design. The third one (extra-large OHSC) offers the largest storage space of all three variants, is also modular and has no service channel. The passenger service functions are directly attached to the OHSC.

Figure 9. Graphical representation of the three overhead stowage compartment variants.

An A321 aircraft with a LOPA from Qatar Airways is selected as a base reference for the investigation of the three hatrack variants. This layout provides 4 rows of seats in business class and 25 rows of seats in economy class, carrying 166 passengers. Following the process described in the previous subsection, the three variants are designed and the objects are instantiated. Subsequently, the requirements are verified by further investigations in Matlab and the virtual environment. An exemplary representation of the cabin integration into the aircraft is shown in Figure 10 of the 3D geometry model with the extra-large OHSC generated in Blender.

Figure 10. 3D model of the aircraft cabin concept with the extra-large hatrack.

The visual representation of the results of all three variants in the virtual environment is shown in Figure 11. The first image shows the installation of the regular OHSC, the second one the large OHSC and the third one the extra-large OHSC. Based on the image sections, the different installation scenarios of the passenger service functions can be distinguished clearly as well as their position. When looking at the regular hatrack, it is noticeable that the dimensions of the hatrack were not considered as a construction limitation in order to accommodate a modular design. The limiting factors are the dimensions of the individual modules of the passenger service function system and the passenger supply channel. The wider distribution of the individual elements in the other variants makes a modular construction possible.

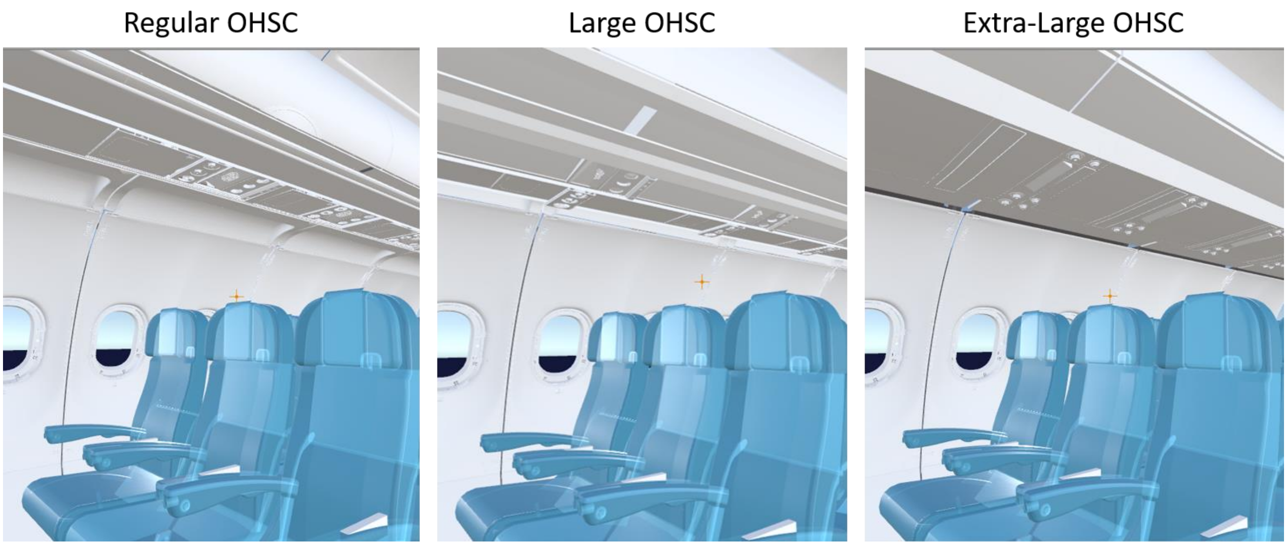

Figure 11. Visualization of the hatrack configuration in VR for all variants.

Figure 12 shows the results of the variant analysis in Cameo using the cabin component instance table, exemplified by the variant with the large OHSC. Here, the parameters that are needed to check the requirements are listed. With the help of the table, it can be seen which object fulfils which requirement in each case. As soon as a requirement is not fulfilled, the responsible parameter is colored red and can be distinguished clearly. If a requirement is fulfilled, the line is not colored. It can be seen here that for the large OHSC, luggage tray no. 22 fulfils both the requirements for modular construction and design. Luggage racks 23 and 24, for example, do not meet the requirements for modularity. Seat no. 1 meets the strict requirement for oxygen mask accessibility, while seat no. 2 slightly misses the maximum oxygen mask distance of 636mm.

#	Name	ID : Real	preAssembly : String	design : String	distanceOxygenMask : Real
23	a321_Qatar_LOPA.ohsc[22]	4801756	yes	yes	
24	a321_Qatar_LOPA.ohsc[23]	7.6359E7	no	yes	
25	a321_Qatar_LOPA.ohsc[24]	3.5051E7	no	yes	
26	a321_Qatar_LOPA.ohsc[25]	4.5988E7	yes	yes	
27	a321_Qatar_LOPA.ohsc[26]	4853557	yes	yes	
28	a321_Qatar_LOPA.ohsc[27]	2.933E7	no	yes	
29	a321_Qatar_LOPA.ohsc[28]	7.4535E7	no	yes	
30	a321_Qatar_LOPA.seat[1]	802845			272.7157
31	a321_Qatar_LOPA.seat[2]	503300			654.9674
32	a321_Qatar_LOPA.seat[3]	176904			654.9674
33	a321_Qatar_LOPA.seat[4]	269346			272.7157
34	a321_Qatar_LOPA.seat[5]	102222			272.7157
35	a321_Qatar_LOPA.seat[6]	527022			654.9674

Figure 12. Cabin component instance table with verification of requirements for the large hatrack.

A summary of the evaluations of all three variants is listed in Table 1. A total of 28 hatracks can be placed in the cabin layout. In the regular version, only 4 luggage compartments meet the requirements for modularity. The reason for this is the larger seat pitch in business class and thus a lower utilization of the passenger service functions to be installed. In addition, the distribution of the PSFs on only one PSC and the larger design of the individual components causes the greatest distance between the oxygen mask and the passenger compared to the other variants. The large variant distributes the PSFs over two PSCs, so that 18 hatracks could be pre-assembled. However, more seats fail to meet the strict requirement for accessibility of the oxygen masks than in the regular variant. The reason for this is a slight shift of the first passenger service channel towards the window. This is compensated for by the fact that distributing the functions over two channels allows the oxygen masks to be placed closer to the seat. In the extra-large version, all hatracks can be pre-assembled, so that the requirements for a modular design are met. Due to the wider distribution of the individual passenger service functions, more seats meet the requirements for the accessibility of the oxygen masks. Only 26 seats do not meet the requirements. The reason for this is the consideration of the modular construction, whereby the components to be installed must be offset by a few centimeters in order not to violate the construction limits of the hatrack. Thus, in the extra-large variant, the remaining 26 seats deviate by only 6% from the target value, compared to regular (19%) and large (16%) variant. In conclusion, the extra-large variant offers the best advantages in terms of modularity, accessibility of the oxygen masks, design and storage space.

Table 1: Variant requirement analysis results

Variant	Number of modular hatracks	Number of oxygen masks req. pass	Number of oxygen masks req. failed	Mean distance oxygen mask if req. failed
Regular	4	94	72	754 mm (+19%)

Large	18	60	106	739 mm (+16%)
Extra-Large	28	140	26	693 mm (+9%)

Discussion

In the following, two aspects of the developed methodology will be discussed. On the one hand, the execution times to perform a variant modeling. On the other hand, the adaptability of the method and its suitability for other systems.

Execution time

For the execution of the property verification and architecture modeling, the time of the individual processing steps or model executions was measured. The results are shown in Table 2. The times listed were recorded for the third use case of the hatracks (extra-large hatrack variant for Qatar A321-LOPA). The architecture modeling and creation of cabin component instances based on the imported parameters of the xml file takes 7.3s. This time also includes the instantiation of objects in Matlab. Following, the further use of the objects for geometric placement and evaluation takes 22.3s in Matlab. Next, the automated generation of a high-fidelity 3D geometry model for an entire cabin and fuselage structure takes 1108s (using an Dell Alienware M15 R7 laptop). The subsequent model and data import to the virtual reality platform requires 321s. Here, the user or stakeholder can interact with the model and check the design unlimited in time. Finally, 5s are needed to transfer the objects and their properties back to Cameo for requirements management and evaluation. The overall process for an investigation of one variant requires 1464s (24.4min). Since most of the time was spent modeling the 3D geometry, this step has the greatest impact on the overall process. The reason for this are the long import times, since a 3D model has to be imported from a library into the scene and placed for each cabin object. In this example, the highest resolution was used. Consequently, the total process time can even be reduced if less detailed models are used. All in all, the execution of the variant check is sufficiently fast and brings a considerable advantage compared to the current customizing process with several on-site meetings, data exchange via e-mail and manual post-processing of the 3D models.

Table 2: Execution Times for each model domain

Model domain	Execution time	Tasks
SysML/ Cameo Systems Modeler	7.3 s	▪ Architecture modeling ▪ Object generation
Matlab	22.3 s	▪ Geometrical placement of objects ▪ Evaluation
3D model generation / Blender	1108.0 s	▪ 3D geometry modeling
VR visualization/ Unity	321.0 s	▪ Data transfer ▪ 3D model import
SysML / Cameo Systems Modeler	5.0 s	▪ Requirement Management and value import

Model generation and expansion

The demonstrated methodology is suitable for the shown use case with the architecture modeling and investigation of different hatrack variants. Due to the same object structure of the cabin components in the SysML environment and in the Matlab environment, as well as the clear assignment of the individual instantiated objects through unique IDs, all three sub-models can interact well with each other beyond the tool environment borders. The data exchange works without loss and the assignment is explicit. Moreover, the exchange of data is also possible in other domains and the data can be reused. Thus, the 3D geometry models generated in the process as well as the cabin component properties can be used for further analyses, e.g. requirement-based creation of FEM model variants on the level of detail inclusive overall evaluation, or investigations study, e.g. cabin assembly process planning (Markusheska et al. 2022). As a result of the coupling, it is possible to achieve different degrees of fidelity. Furthermore, the methodology is also suitable for other systems and can be adapted to them. However, it should be noted that when investigating larger systems or adding further system aspects in SysML, the modeling size increases and the model could quickly become complex and difficult to navigate. One factor is reading the data and preparing it to pass the values to Matlab. Currently, this process is very fragmented and requires many steps to pass the data in large array structures. As a result, the larger the model grows, the more time-consuming the programming of this part of the process becomes. Since the method shown here is a proof of concept, optimizations of the software were not part of it, but could be conceivable in a next step. In future, the individual systems in the cabin could either be modeled in partial models instead of one big model and each connected separately to the Matlab environment. Here, the results (objects) would be coupled and related to each other. Due to the object-oriented programming structure in Matlab and the generic modeling in Unity (VR environment), these can be reused universally. Only the SysML model has to adapt to the use case and provide the corresponding subcomponents. Another option would be to link and share data using the Open Services for Lifecycle Collaboration (OSLC) integration of Cameo. OSLC is an open standard designed to facilitate the collaboration of software development tools. This allows linking artifacts in different tools without additional plugins or seeing information without switching between the different tools. Hyperlinks are used to map the model elements in SysML to the model elements provided by an OSLC provider. However, there has been few studies on this to date in the context mentioned here and it therefore needs to be tested in more detail in the future.

Conclusion and Future Work

In this paper, an approach for linking heterogenous and domain specific models to investigate variants in the aircraft cabin has been demonstrated. In aircraft design, each discipline has its own models with different degrees of fidelity and abstraction, already optimized for its particular research problem. Yet each discipline uses its own tool set, which must be linked together by a digital thread. A method for coupling and linking data from various tool domains was introduced in this research. All in all, three domains were linked with each other for a conceptual design of an aircraft cabin and its systems. First, SysML models (tool environment Cameo Systems Modeler) were used for modeling the architecture of the aircraft cabin and requirements management. Second, the geometrical placement of the cabin objects was done with a knowledge model in Matlab. Third, the data from Matlab was used to automatically generate a 3D high fidelity geometry model for further investigation in a virtual environment (Unity). During the design process of the cabin, the three domains exchange data with each other and transfer all values back to the model in Cameo. Finally, all requirements can be checked.

The workflow has been exemplified using three different overhead stowage compartments. Each compartment comes with different advantages and concepts for integrating the passenger service functions. All three variants were conceptual designed using the developed approach and compared to each other. It could be shown that the extra-large hatrack, besides the advantage of more storage space, also enables better accessibility of the oxygen masks for the passenger and provides a modular design. The latter results in potential savings in time and money due to the ability to pre-assemble and reduced installation time in the final assembly line. The demonstrated approach of an automatically interconnected model architecture enables digital continuity between different domains while maintaining data consistency and providing for each disciplinary analyses the needed fidelity level.

In a next step, variant modeling in the cabin will be further expanded, especially since variant modeling concepts shall be integrated in the future with the introduction of SysML 2.0. The inclusion with further disciplines like acoustic analysis and assembly process planning is planned for future work. In addition, further approaches such as OSLC for linking partial models within SysML as well as to external tools are investigated in order to keep the complexity of large system groups low. In this paper, variant modeling was shown in the context of customizing the design of hatracks in the cabin, where other variant forms such as behavior modeling will be considered in future work. Overall, the coupling of heterogenous tools is essential for a holistic conceptual design and evaluation of a system. The approach shown allows variants to be created quickly, a wide range of requirements to be checked and concepts to be compared with each other in order to be able to make quick predictions in the area of customizing in the cabin.

References

Ackert, S. 2013, ‘Commercial Aspects of Aircraft Customization’, News Paper Article, Aircraft Monitor.

De Crescenzio, F.; Bagassi, S.; Asfaux, S. and Lawson, N. 2019, ‘Human centred design and evalu ation of cabin interiors for business jet aircraft in virtual reality’, *International Journal on Interactive Design and Manufacturing* 13, pp. 761–772. https://doi.org/10.1007/s12008-019-00565-8

De Crescenzio, F.; Bagassi, S. and Starita, F. 2021, ‘Preliminary user centred evaluation of regional aircraft cabin interiors in virtual reality‘, *Sci Rep* 11, 9662. https://doi.org/10.1038/s41598-021-89098-3

Du, X.; Jiao, J. and Tseng, M. M. 2006, ‘Understanding customer satisfaction in product customiza- tion’, *Int J Adv Manuf Technol* 31 (3-4), pp. 396-406. doi: 10.1007/s00170-005-0177-8.

Forlingieri, M. and Weilkiens, T. 2022, ‘Two Variant Modeling Methods for MBPLE at Airbus‘, INCOSE International Symposium, 32: 1097-1113. doi: https://doi.org/10.1002/iis2.12984.

Frischen, C.; Marbach, A.; Tichla, F. and Mantwill, F. 2019, ‘Durchgängige Variantensteuerung mit Hilfe der regelbasierten Komplexstückliste‘, 30. DFX-Symposium 2019 Proceedings. doi: 10.35199/dfx2019.2.

Fuchs, M.; Beckert, F.; Biedermann, J. and Nagel, B. 2022, ‘A collaborative knowledge-based Method for the interactive development of cabin systems in virtual reality‘, *Computers in Industry* 136, 103590, ISSN 0166-3615. doi: 10.1016/j.compind.2021.103590.

Fuchs, M.; Ghanjaoui, Y.; Abulawi, J.; Biedermann, J. and Nagel, B. 2022, ‘Enhancement of the virtual design platform for modeling a functional system architecture of complex cabin systems’, *CEAS Aeronaut J* 13, 1101–1117. doi: https://doi.org/10.1007/s13272-022-00608-z.

Hedge, V. G.; Kekre, S.; Rajiv, S. and Tadikamalla, P. R. 2005, ‘Customization: Impact on Product and Process Performance’, *Production and Operations Management* 14(4), pp. 388-399. doi: 10.1111/j.1937-5956.2005.tb00228.x.

Hesse, C.; Walther, J.-N.; Allebrodt, P. and Wandel, M. 2021, ‘Integration of multi-physics analysis into the cabin design process using virtual reality’, AIAA AVIATION Forum and Exposition, 02.-06. Aug. 2021, Washington, USA. doi: 10.2514/6.2021-2776.

Mahboob, A. 2021, ‘Modelling and use of SysML behaviour models for achieving dynamic use cases of technical products in different VR-systems’, PhD thesis, Technische Universität Ilmenau (Ilmenau, Germany).

Markusheska, N.; Srinivasan, V.; Walther, J.-N.; Gindorf, A.; Biedermann, J.; Meller, F. and Nagel, B. 2022, ’Implementing a system architecture model for automated aircraft cabin assembly processes’, *CEAS Aeronaut J* 13, 689–703. doi: https://doi.org/10.1007/s13272-022-00582-6.

Melzer, S.; Peukert, H.; Wang, H. and Thiemann, S. 2022, ‘Model-based Development of a Federated Database Infrastructure to support the Usability of Cross-Domain Information Systems‘, IEEE International Systems Conference (SysCon), 2022, pp. 1-8. doi: 10.1109/SysCon53536.2022.9773811.

Menninger, B.; Wiechel, D.; Rackow, S.; Höpfner, G.; Oleff, C.; Berroth, J.; Gräßler, I. and Jacobs, G. 2022, ‘Modellierung und Analyse funktionaler Varianz komplexer technischer Systeme’, 33rd Symposium Design for X – DFX 2022. doi: 10.35199/dfx2022.15.

Müller, J.; Panarotto, M. and Isaksson, O. 2020, ‘Function Model Based Generation of CAD Model Variants‘, CAD’20 Proceedings, Barcelona, Spain, pp. 204-208, ISSN 2769-8440.

doi: 10.14733/cadconfP.2020.204-208.
Richter, K. and Walther, J. (ed.) 2017, *Supply Chain Integration Challenges in Commercial Aerospace*, Springer International Publishing, ISBN 978-3-319-46154-0.
Schumacher, T. and Inkermann, D. 2021, 'Heterogeneous models to Support Interdisciplinary Engineering – Mapping Model Elements of SysML and CAD', Procedia CIRP, vol 109, pp. 653-658.
doi: 10.1016/j.procir.2022.05.309.
Weilkiens, T. 2016, *Variant Modeling with SysML*, MBSE4U Booklet Series, ISBN 978-3-9817 875-4-2.

Biography

Mara Fuchs. Mara's career as an Aircraft Systems Engineer started with her graduation from the Hamburg University of Technology in 2019. She started as a research associate and PhD student at the DLR Institute of System Architectures in Aeronautics in Hamburg, Germany. She investigates the coupling of heterogenous domain models for customizing and variant modeling of cabin systems. Moreover, her research focuses on the integration of new technologies into the aircraft (e.g. hydrogen system) and its impact on existing system architectures.

Yassine Ghanjaoui. Yassine received his M.Sc. in Aeronautical Engineering with a focus on cabin systems and systems engineering from the Hamburg University of Applied Sciences in 2021. He is currently conducting research on the application of MBSE in aircraft production planning at the DLR Institute of System Architectures in Aeronautics in Hamburg, Germany. His focus is on developing methods for creating Digital Threads to link aircraft design with semantic production models, as well as automated simulation and validation of assembly processes.

Jörn Biedermann. Jörn leads the virtual cabin mockup and industrialization groups at the DLR Institute of System Architectures in Aeronautics in Hamburg. He obtained his PhD in Mechanical Engineering in the field of cabin noise and vibration in 2016. In his ten years of research at DLR he coordinates and involved with his groups in several cabin and manufacturing digitalization projects to accelerate the development of new cabin configurations.

Björn Nagel. Björn is leading the DLR Institute of System Architectures in Aeronautics in Hamburg as founding director for the last seven years. He has been working at DLR for 20 years, starting as a research engineer in the field of composite structures and adaptive systems after graduating from the Technical University Braunschweig in Aerospace, Aeronautical and Astronautical Engineering in 2003.