

THE REUSE OF SYSML BEHAVIOUR MODELS FOR CREATING PRODUCT USE CASES IN VIRTUAL REALITY

Mahboob, Atif (1); Husung, Stephan (2); Weber, Christian (1); Liebal, Andreas (1); Krömker, Heidi (1)

1: Technische Universität Ilmenau; 2: sem engineering methods AG

ABSTRACT

An early evaluation of a product along with the consideration of life phase specific actor(s) and environment(s) can help greatly to gain an understanding of the product's behaviour and interactions. Virtual Reality (VR) can help designers to examine later life situations of a product by means of use case scenarios. However, preparing a VR-scene is still a time-consuming and cumbersome task. A model based approach that uses behaviour models of SysML to describe a VR-scene can reduce the preparation efforts. Such an approach is helpful if it allows the reuse of already described VR-scenes or their contents. This paper talks about the reusability of SysML behaviour models that constitute a VR-scene. This reusability can only be achieved by the generic definition of model interfaces. Therefore, a new modelling approach is presented to facilitate the reuse of SysML behaviour models to form different use cases of a product in VR. This approach also talks about the interface definitions and the management of variants of SysML models. The presented approach is elaborated by an example model that contains variants and uses instances to build different use cases.

Keywords: Product modelling / models, Systems Engineering (SE), Virtual Engineering (VE), SysML/MBSE, Virtual Reality (VR)

Contact:

Mahboob, Atif
Technische Universität Ilmenau
Engineering Design Group
Germany
atif.mahboob@tu-ilmenau.de

Cite this article: Mahboob, A., Husung, S., Weber, C., Liebal, A., Krömker, H. (2019) 'The Reuse of SysML Behaviour Models for Creating Product Use Cases in Virtual Reality', in *Proceedings of the 22nd International Conference on Engineering Design (ICED19)*, Delft, The Netherlands, 5-8 August 2019. DOI:10.1017/dsi.2019.208

1 INTRODUCTION

An important goal during the product development process is to find a solution that satisfies the product requirements from all its life phases. These requirements contain the needs of the stakeholders of the product and the product interaction with its neighbouring systems (e.g. actor(s) and environment). The general life phases of a product are product planning, product development, production planning, production, distribution, use, service and disposal (VDI2221). In each of these life phases, the product finds itself in a different environment and with different actor(s). The multi-disciplinary and complex product cannot be considered isolated; instead, the context of the product is also relevant. A life phase specific actor (or maybe several actors) together with an environment define the context of the product. In order to gain a better understanding of the product's behaviour and its interactions in its later life phases, it is important to consider the contexts of the product already at the development phase. Product designers identify different use cases of the stakeholders and translate them to technical requirements of the product. The goal of building these use cases is to better understand the product behaviour in its later life phases and to verify the fulfilment of the stakeholders' needs.

Following abbreviations and terminologies will be used throughout the rest of the paper:

- MBSE - Model Based Systems Engineering
- SysML - Systems Modeling Language
- PAE - Product, actor and environment
- Context: Life phase specific actor(s) and environment that have an interaction with the product
- bdd - SysML Block Definition Diagram, ibd - SysML Internal Block Definition Diagram
- HLSA - High Level Solution Architecture - The core model for each solution architecture for a given problem (Aleksandraviciene and Morkevicius (2018))
- VR - Virtual Reality
- CAVE - CAVE automatic virtual reality
- HMD - Head Mounted Display

With regard to these use cases, Virtual Reality (VR) technology is very useful and relevant as it can help to model and show later life situations of a product along with its context. Despite the great potential of VR, its use in industry is mainly limited to the visual evaluation of a product and less focus is put on capturing its behaviour in VR. Although the geometrical representations of a product can be prepared by importing CAD geometries directly into VR, the preparation of behavioural representations is mainly based on cumbersome programming of the complete VR-scene in advance. The main obstacles for the application of VR in industry are the long time and high effort needed for the preparations of virtual models, their very limited reusability and the limited modification possibilities. Therefore, in this paper, the focus lies on a new model based approach based on the division of the complete VR-scene into three sub-models PAE, i.e. product, actor and environment. Such a division of the complete VR-scene can allow recombination/ reuse of these models to form different use cases of a product as shown in figure 1. In the presented approach, VR is only used for visualisation and the behaviour description along with physical calculations is performed/executed outside VR.

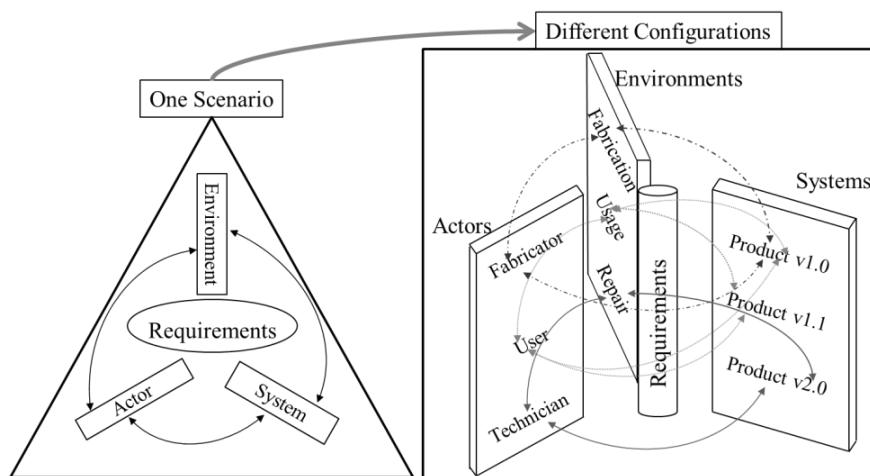


Figure 1: Use case building in a model based approach for VR (Mahboob et al. (2017))

The goal of using a model based approach is to reduce the VR-scene preparation effort and achieve reusability of behaviour models. The different VR-systems (e.g. CAVE, HMD, Powerwall, etc.) usually consist of different hardware and software components that are not compatible with each other. The conventional programming-based solutions do not work in different VR systems. The main reason is that different VR-software used in different VR-systems use different programming languages. That, in turn, makes the VR preparation highly specific to the VR systems at hand. It would also be of great advantage if behaviour models developed once would work in different VR-systems (e.g. CAVE, HMD, Powerwall etc.). Although the PAE division of the VR-scene helps in efficient use case building, it requires a generic description of the models and their interaction with each other. In order to address the abovementioned challenges, the Systems Modeling Language (SysML) is used as the description language. SysML can describe the static and dynamic behaviour of sub-models and their interaction; however, it cannot facilitate the physical calculation (e.g. collision detection) on the geometric objects in a scene. Therefore, a dedicated physics calculation software has to be used to perform needed physical calculations. As the SysML models are developed nowadays alongside design to perform different analyses during the design process in industry (Husung *et al.* (2018)), the connection of SysML and VR can greatly facilitate the use of VR in industry by reducing the additional VR-scene preparation effort. The generic flow of information during a VR simulation that is based on SysML behaviour models can be seen in figure 2.

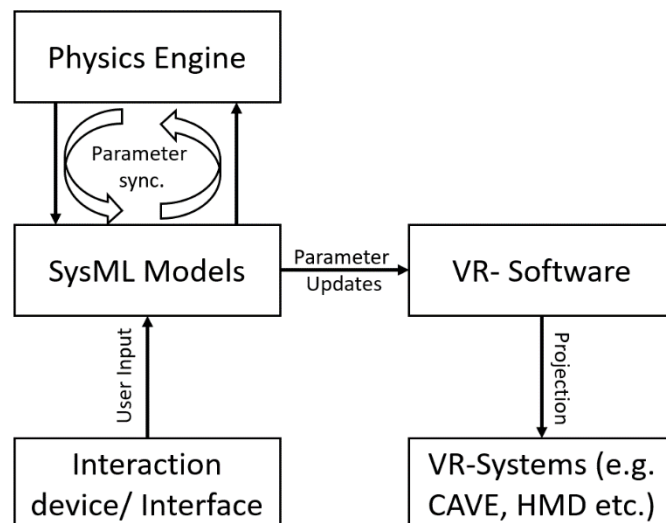


Figure 2: Flow of information between different tools during VR simulation (derived from Mahboob *et al.* (2018a, 2018b))

SysML models contain the behaviour description of the sub-models PAE, handle the user input and communicate with the physics engine. *Physics Engine* contains the exact replica of the VR-scene geometric contents and performs physical calculations on them. *VR-Software* contains the VR-scene consisting of geometric objects and projects them onto the *VR-System*. The SysML models at the core of such a simulation process interpret the user input, control the physics engine and update the contents of the VR-software. Mahboob *et al.* (2018a, 2018b) have already established that the SysML behaviour models can be used to achieve simulation in different VR-systems by using them in CAVE and HMD respectively. In order to analyse different life phase situations of a product, it should be possible to substitute or reuse the sub-models that are already defined in SysML. Therefore, our focus in this paper lies on developing an approach for the reuse of SysML behaviour models of PAE, i.e. *SysML-Models* in figure 2. This paper will address the following research questions:

1. How can the interactions between the sub-models be modelled in such a way that the substitution/reuse of the sub-models is enhanced?
2. How can the already described behaviour models in SysML be reused with minimum effort to construct new use cases of the product in VR?

2 RELATED WORK

The product development process makes use of different digital models and methods to perform different analyses. However, the product and its behaviour stay at the core of these methods and less

The *User_Product_Env Interaction* block owns the three sub-models and can exchange information over the SysML ports (named *interaction*, *product*, *environment* in the figure). These ports are used to call the behaviour of the sub-models simultaneously over the ports so that a parallel execution can be achieved.

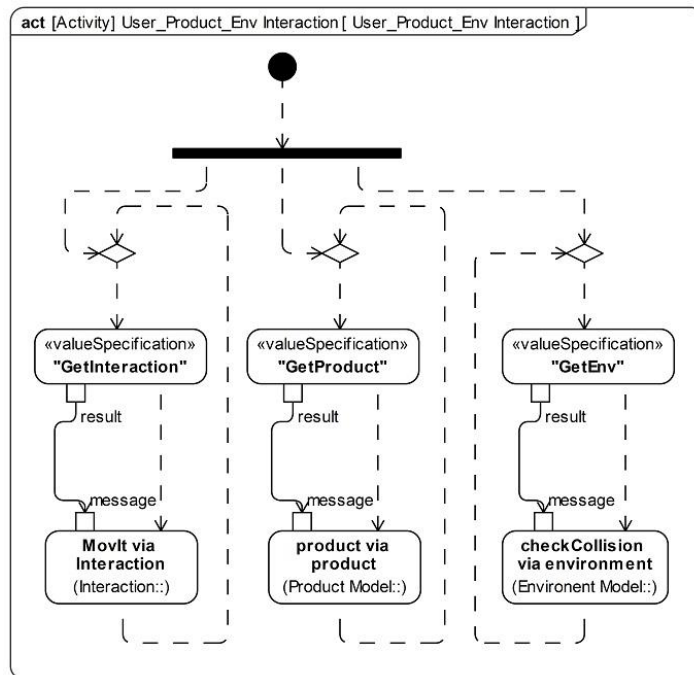


Figure 4: Parallel execution of sub-models (activity diagram)

Figure 4 shows the activity diagram used for executing the sub-models in parallel. The block *User_Product_Env Interaction* calls the behaviour (*MovIt*) of the *Interaction* model over the port *Interaction*, the behaviour (*product*) of the product model over the port *product* and the behaviour of the environment model (*checkCollision*) over the port *environment*. Once the execution of the activity diagram according to figure 4 starts, the sub-models work in parallel and communicate the information between each other over the ports defined in figure 3. This approach manually initialises the behaviour of all sub-models. Although in this manner, the sub-models can be executed in parallel, the replacement of any of the three sub-models will require remodelling. For instance, in order to replace one sub-model from such a parallel execution, it will be necessary to stop the current execution, perform remodelling for a new use case and then start a new execution. This might be an easy task for a relatively simple SysML model; however, it can require additional modelling effort for every use case and can be prone to error, if the overall model is relatively detailed and complex. Therefore, the new approach uses the “*SysML Instances and the concept of inheritance*”. This enables an efficient and relatively easy construction of use cases in VR by avoiding the additional modelling effort needed for each use case. In order to keep the new modelling approach easy to understand, the complete modelling of sub-models will not be discussed in this paper; instead, the necessary details about the structure and behaviour of these models will be discussed.

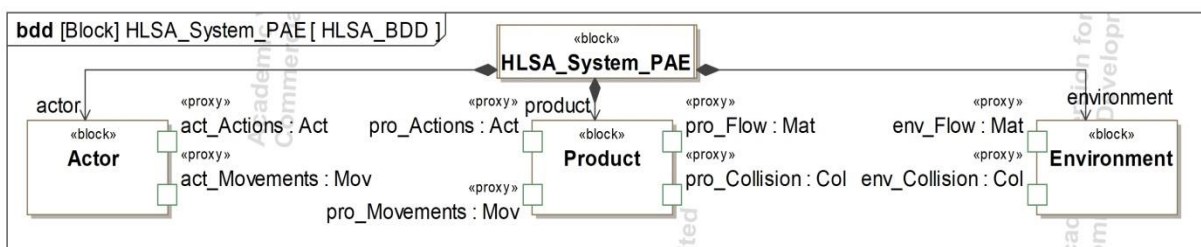


Figure 5: System architecture

Figure 5 shows the *HLSA* of the *HLSA_System_PAE* that contains three sub-models and defines their interaction points (definitions of ports and interfaces). The connections of the ports of PAE are also

specified inside HLSA. This HLSA possesses the generic structure of the complete SysML model for all possible use cases. The goal is to choose the version of each of PAE for any given use case or replace one of the available versions with another one (e.g. replacing version 1 of the product in current use case with version 2 in the next use case). In order to attain this functionality, the *Product* block in the HSLA model should possess all ports and interfaces that any of its given versions can require. Therefore, it is important at this point to consider all possible interfaces between PAE for all their versions and to define them already in HLSA.

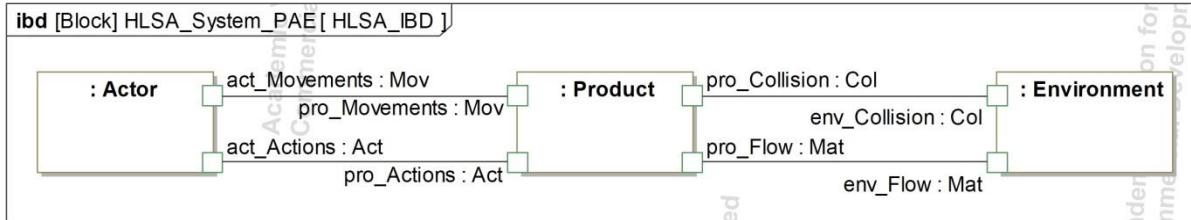


Figure 6: Internal Block Definition Diagram (ibd) of HLSA_System_PAE

Figure 6 shows the *ibd* for *HLSA_System_PAE* block. It can be observed that there is no information exchange between the *HLSA_System_PAE* and the sub-models, unlike the existing approach (see figure 3). Therefore, the individual behaviour of the sub-models can no longer be initialised by the block that contains them. One last thing to define in HLSA is to model the signals and flows that will be used in different versions of PAE. This HLSA should be reused as the reference model while modelling any version of sub-models. The behaviour description of the sub-models should use the interfaces and the signals from the HLSA so that the consistency can be obtained throughout all versions. This can be understood by taking a look at two different versions of the actor model, i.e. *InteractionDeviceV1* and *DomesticUserV1* as shown in figure 7. The blocks *InteractionDeviceV1* and *DomesticUserV1* are linked with the *Actor* block from the HSLA model using the generalisation relationship (denoted by a hollow triangular arrow and a solid line). Generalisation relation means that the two new versions of actor model inherit the constraints modelled in the *Actor* block, e.g. the interfaces, signals and definitions of ports. The sign “^” on the ports of new blocks indicates that the port is not owned by the block; instead, it is inherited from the *Actor* block. The use of generalisation relationships helps to make the interfaces on all the versions of the actor model consistent with that of the HLSA model. The authors suggest making a separate SysML model for each version of a sub-model, as it will facilitate the reuse of these versions in any other model also. Similarly, two versions of the product, i.e. *VacuumCleanerV1*, *VacuumCleanerV2*, and also two versions for the environment model, i.e. *LivingRoomV1*, *FabricationEnvironmentV2*, are used.

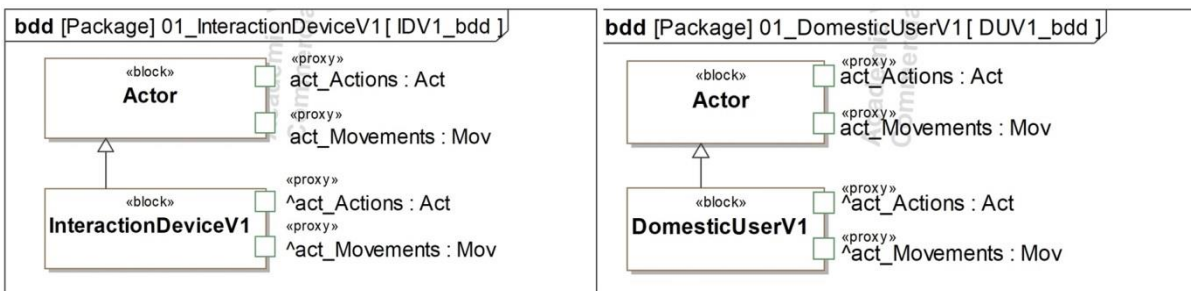


Figure 7: bdd for InteractionDeviceV1 and DomesticUserV1

In order to explain the modelling approach, one use case is built by using instances in SysML and later used to achieve simulation in VR. This use case contains a vacuum cleaner as product, living room as environment and an interaction device as a substitute for actor model. Each of these models' behaviour is described using SysML behaviour models. The interaction device model handles the input from the VR system about the actions performed by the VR-user and communicates it to the product model using SysML ports. The product model receives these actions and decides upon needed implementations of these actions. Furthermore, it communicates with the physics engine to performs physical calculations on the vacuum cleaner geometric model and receives the feedback. Finally, it communicates the changes to the VR-software so that the parameters of the virtual vacuum cleaner can be updated. The living room model keeps a check on the geometric objects in the environment,

communicates with physics engine constantly and it sends the information regarding objects' parameters or collision to the VR-software. Similar to these models, the *DomesticUserV1*, *VaccumCleanerV2* and *FabricationEnviornmentV1* can be modelled and the overall structure of the complete HLSA system can be seen in figure 8.

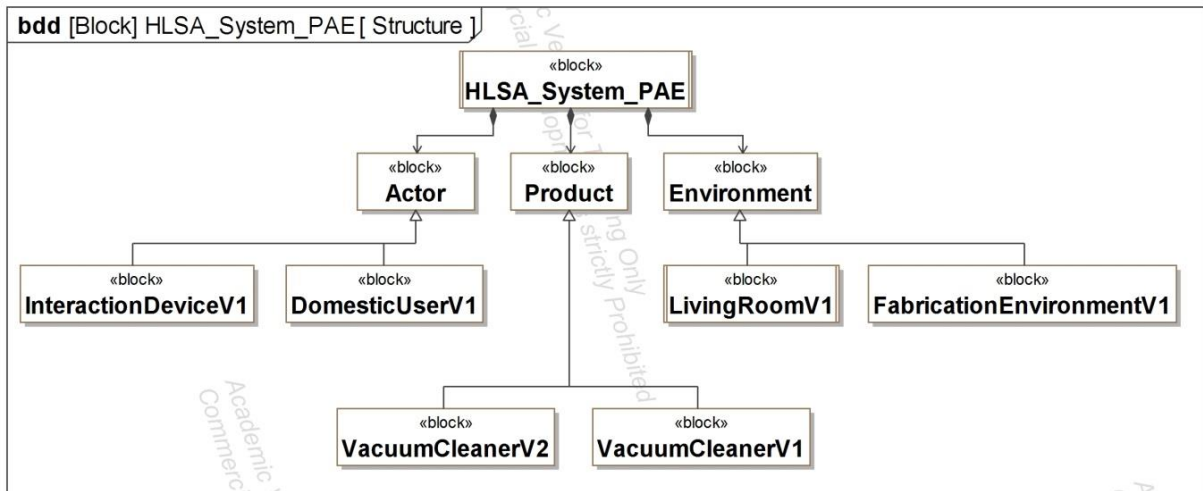


Figure 8: The overall structure and components of HLSA Systems

Now we have six different SysML models saved separately in a repository containing two versions from each of PAE. These models can now be loaded in the HSLA model. The model containment tree looks like in figure 9 and shows that the current model uses shared sub-models and it is indicated by a “hand” sign. As the relationship (generalisation) of different versions with HLSA is already defined in each version, it becomes clear to the modelling tool that each of PAE has two versions. It is now possible to create different instances of the *HLSA_System_PAE* by using the automatic instantiation functionality of SysML.

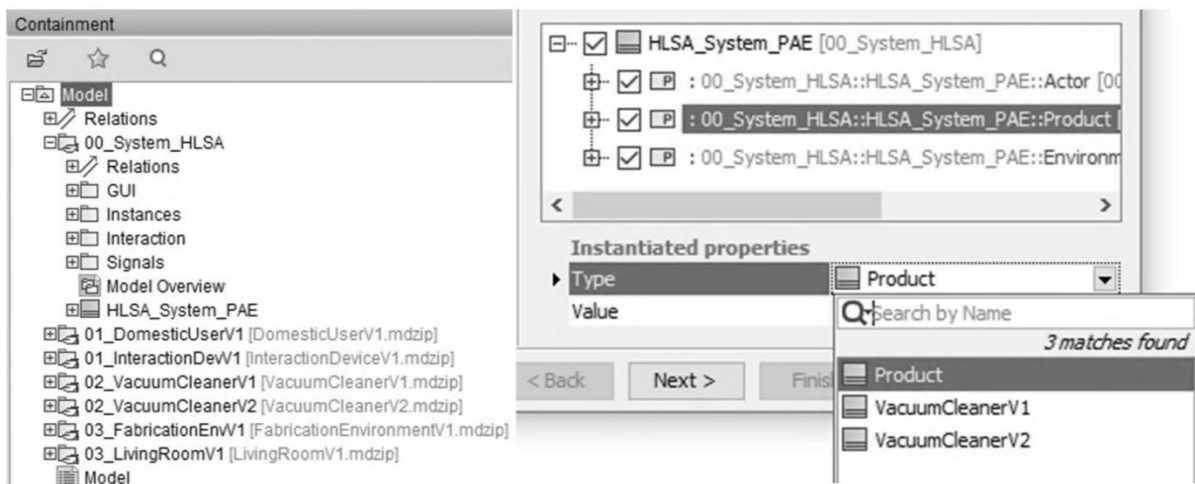


Figure 9: Model containment tree (left), automatic instantiation dialog box (right) in Cameo Systems Modeler of NoMagic

Figure 9 shows the automatic instantiation dialog box from the modelling tool. This dialog box lets the user choose the appropriate versions of PAE for a particular use case. Let us take an example use case consisting of VacuumCleanerV1 inside LivingRoomV1 and being controlled by InteractionDeviceV1. As a result, an instance of the *System_PAE* for this use case is created, shown in figure 10 (left).

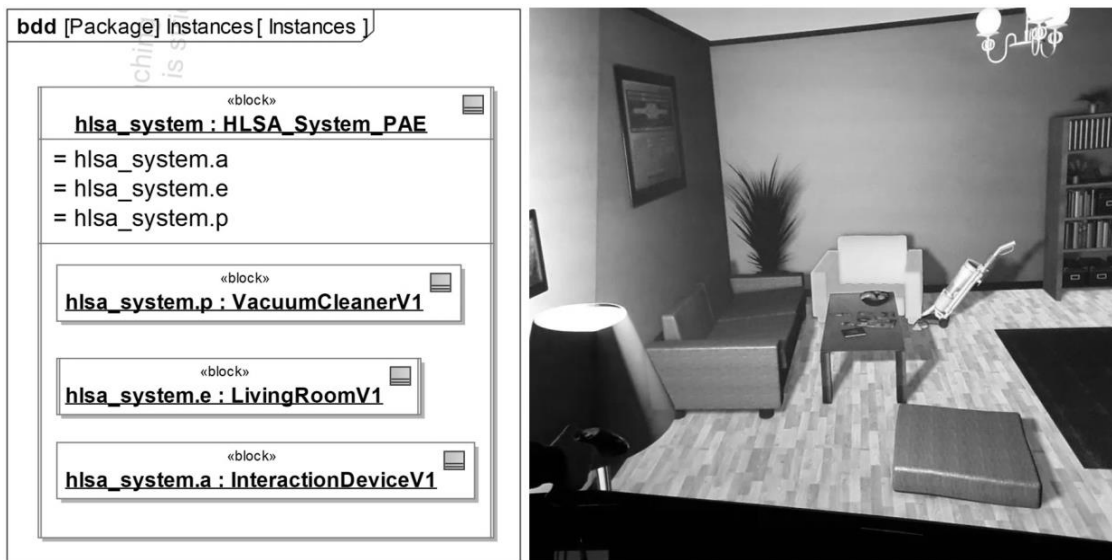


Figure 10: Instance of System_PAE (left), simulation in CAVE-VR (right)

This instance shows that originally *HLSA_System_PAE* block had three parts and also shows the version of each part used to make this instance. If this block is executed, it automatically initiates the classifier behaviour of each of sub-models in parallel. As a result, a real-time product use case is built and simulated in VR as shown in figure 10 (right). Similarly, more instances of the *HLSA_System_PAE* block can be created for building use cases for different combinations of PAE.

The presented modelling approach for the reuse of parts of a system and its sub-models can be summed up in the form of generic guidelines as follows:

- Create an HLSA model containing the main components/parts of the system along with interface definitions, signals and flows
- Each version of a component or sub-model should be linked with the respective block in HLSA using generalisation relationship
- Any changes in the interfaces should be performed in the HLSA and the variants should use the constraints defined in the HLSA
- The behaviour models of the different variants of sub-models should not possess any direct dependency with any other model or its other versions

4 DISCUSSION

Revisiting the research questions defined in the introduction section, the modelling approach presented in this section helps in defining reusable interfaces in a consistent way. It also facilitates the reuse of behaviour models by enabling an efficient and easy configuration of new use cases of a product for use in VR. Defining an HLSA helps not only to keep consistency in the modelling, but also facilitates the generic definition and reuse of the interfaces. The use of a generalisation relationship in SysML helps to create different instances of PAE and indeed different use case situations of a product along with its context. This is an advantage over the conventional organisation of the sub-models as shown in figures 2 and 3, where the reuse or substitution of a sub-model will require additional modelling effort or even a complete redefinition of interfaces between the sub-models. The presented approach is particularly of advantage when modelling a complex product, where different teams are working on sub-parts of the overall system. The execution of an instance allows running all the sub-models in parallel, thus eliminating the need of managing the behaviour models manually. The use of HLSA also allows flexibility in the modelling process, as additional interfaces can be added later or the existing ones can be modified. These modifications will automatically become available for each version of the sub-model as the HLSA is used as the reference model in the modelling process.

5 CONCLUSION AND OUTLOOK

This paper presents an approach for modelling systems that are split up into interacting sub-models of product, actor(s) and environment (PAE) in SysML which enables the use of SysML-Instances to effi-

ciently construct new use cases of a product. It helps to keep the interfaces of a product with its neighbouring systems (i.e. actor and environment) consistent in all different versions. The approach enables an efficient recombination, reuse and automatic parallel execution of the sub-models (PAE). The possibility of reusing sub-models that describe the behaviour of a VR-scene also helps to reduce the preparation effort needed for VR. Furthermore, the presented approach is not limited to the given use case; instead, it can be used to manage different versions of the sub-parts of a product during the development process.

The implementation of automatic creation and execution of instances based on user feedback from VR can be looked at in the future.

REFERENCES

- Abidi, M.-A., Chevaillier, P., Lyonnet, B., Kechiche, M., Baert, P., and Toscano, R. (2015), “How to Create a New Generation of Industrial Processes Simulation by Coupling Simulation Tools with VR Platforms”, *Proceedings of the 28th International Conference on Computer Applications in Industry and Engineering (CAINE-2015)*, San Diego, California, USA, October 12-14, 2015, ISCA, Winona, Minnesota, pp. 59–64.
- Abidi, M.-A., Lyonnet, B., Chevaillier, P., and Toscano, R. (2016), “Contribution of Virtual Reality for Lines Production's Simulation in a Lean Manufacturing Environment”, *International Journal of Computer Theory and Engineering*, Vol. 8 No. 3, pp. 182–189. <https://doi.org/10.7763/IJCTE.2016.V8.1041>
- Aleksandraviciene, A. and Morkevicius, A. (2018): “Solution domain”, *MagicGrid Book of Knowledge- A practical guide to System Modeling using MagicGrid from No Magic*, pp. 80, Kaunas, Lithuania
- Eigner, M., Dickopf, T., and Apostolov, H. (2017), “System Lifecycle Management - An Approach for Developing Cybertronic Systems in Consideration of Sustainability Aspects”, *Procedia CIRP*, Vol. 61, pp. 128–133. <https://doi.org/10.1016/j.procir.2016.11.210>
- Eigner, M., Muggeo, C., Dickopf, T., and Faißt, K.G. (2014), “An Approach for a Model Based Development Process of Cybertronic Systems”, *Proceedings of the 58th Ilmenau Scientific Colloquium (IWK)*, Ilmenau, Germany, September 8-12, 2014, Universitätsbibliothek Ilmenau ilmedia, Ilmenau.
- FAS4M (2017), Functional Architectures of Systems for Mechanical Engineers – FAS4M. [online] FAS4M. Available at: <http://fasform.de/content/> (accessed 27.11.2018).
- Follmer, M., Hehenberger, P., Punz, S. and Zeman, K. (2010), “Using SysML in the product development process of mechatronic systems”, *Proceedings of DESIGN 2010 / the 11th International Design Conference*, Dubrovnik, Croatia, May 17-20, 2010, pp. 1513–1522
- Gausemeier, J., Dumitrescu, R., Steffen, D., Czaja, A., Wiederkehr, O. and Tschirner, C. (2013), “Systems Engineering in der industriellen Praxis”, *Heinz Nixdorf Institut*, Paderborn.
- Husung, S., Lindemann, G., Korobov, S., Hamester, M., and Kleiner, S. (2018): “Use Case driven Model-based Systems Engineering for industrial applications”, *EMEASEC2018/ TdSE 2018*, Berlin.
- Mahboob, A., Weber, C., Husung, S., Liebal, A., and Krömker, H. (2017), “Model based systems engineering (MBSE) approach for configurable product use case scenarios in virtual environments”, *Proceedings of the 21st International Conference on Engineering Design (ICED '17)*, Vol. 3, Vancouver, Canada, August 21–25, 2017, The Design Society, Glasgow, pp. 281–290.
- Mahboob, A., Husung, S., Weber, C., Liebal, A., and Krömker, H. (2018a), “SYSML behaviour models for description of Virtual Reality environments for early evaluation of a product”, *DS92: Proceedings of the DESIGN 2018 15th International Design Conference (pp. 2903–2912)*. <https://doi.org/10.21278/idc.2018.0382>
- Mahboob, A., Husung, S., Weber, C., Liebal, A., and Krömker, H. (2018b), “An Approach for Building Product Use case Scenarios in Different Virtual Reality Systems”, *ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. V01BT02A047-V01BT02A047)*. American Society of Mechanical Engineers. <https://doi.org/10.1115/detc2018-85223>
- MecPro2 (2017), “Modellbasierter Entwicklungsprozess cybertronischer Produkte und Produktionssysteme – mecPro2”, [online] mecPro2. Available at: <https://www.mecpro.de/> (accessed 17.11.2017).
- Meissner, H., Cadet, M., Stephan, N., and Bohr, C. (2014), “Model-Based Development Process of Cybertronic Products and Production Systems”, *Advanced Materials Research*, Vol. 1018, pp. 539–546. <https://doi.org/10.4028/www.scientific.net/AMR.1018.539>
- Metag, S., Husung, S., Krömker, H., and Weber, C. (2008), “User-centered Design of Virtual Models in Product Development”, *Proceedings of the 53rd Ilmenau Scientific Colloquium (IWK)*, Ilmenau, Germany, September 8-12, 2008, Universitätsbibliothek Ilmenau ilmedia, Ilmenau.
- Moeser, G., Grundel, M., Weilkens, T., Kumpel, S., Kramer, C., and Albers, A. (2016), “Modell-basierter mechanischer Konzeptentwurf: Ergebnisse des FAS4M-Projektes”, *Tag des Systems Engineering, Herzogenaurach*, Germany, October 25-27, 2016, Carl Hanser Verlag, München, pp. 417–428. <https://doi.org/10.3139/9783446451414.040>

- Moeser, G., Kramer, C., Grundel, M., Neubert, M., Kümpel, S., et al. (2015), "Fortschrittsbericht zur modellbasierten Unterstützung der Konstrukteurstätigkeit durch FAS4M", *Tag des Systems Engineering*, Ulm, November 11-13, 2015, Carl Hanser Verlag, München, pp. 69–78. <https://doi.org/10.3139/9783446447288.008>
- Silhavy, R., Silhavy, P., and Prokopova, Z. (2011), "Behavioral modeling in system engineering", *Proceedings of the 13th WSEAS International Conference on Automatic Control, Modelling and Simulation (ACMOS'11)*, Lanzarote, Spain, May 27-29, 2011, WSEAS, pp. 100–105.
- Stark, R., Hayka, H., and Langenberg, D. (2009), "New potentials for virtual product creation by utilizing grid technology", *CIRP Annals*, Vol. 58 No. 1, pp. 143–146. <https://doi.org/10.1016/j.cirp.2009.03.066>
- Stark, R., Krause, F.-L., Kind, C., Rothenburg, U., Müller, P., et al. (2010), "Competing in engineering design – The role of Virtual Product Creation", *CIRP Journal of Manufacturing Science and Technology*, Vol. 3 No. 3, pp. 175–184. <https://doi.org/10.1016/j.cirpj.2010.08.003>
- VDI 2221 (1987), "VDI-Guideline 2221: Systematic Approach to the Design of Technical Systems and Products", VDI, Düsseldorf, 1987.
- Weber, C., and Husung, S. (2011), "Virtualisation of product development/design – seen from design theory and methodology", *Proceedings of the 18th International Conference on Engineering Design (ICED '11)*, Vol. 2, Lyngby/Copenhagen, Denmark, August 15-19, 2011, The Design Society, Glasgow, pp. 226–235.

ACKNOWLEDGEMENTS

The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG) for supporting this work under the funding label KR 3297/3-3 and WE 1989/6-3 within the research project "user- and task-centred virtual model for product development with regard to further life phases, actors and environments".