21ST INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED17
21-25 AUGUST 2017, THE UNIVERSITY OF BRITISH COLUMBIA, VANCOUVER, CANADA

MODEL BASED SYSTEMS ENGINEERING (MBSE) APPROACH FOR CONFIGURABLE PRODUCT USE-CASE SCENARIOS IN VIRTUAL ENVIRONMENTS

Mahboob, Atif (1); Weber, Christian (1); Husung, Stephan (2); Liebal, Andreas (1); Krömker, Heidi (1)
1: Technische Universität Ilmenau, Germany; 2: :em engineering methods AG, Germany

Abstract

Today, the designers have to deal with a great amount of uncertainties in the design process due to the lack of information about the product, its behaviour and the context in later life-phases (phase specific actors and surroundings). Current development primarily focuses on modelling of the product and its behaviour. The consideration of the product context also plays an important role. Virtual product development has the potential to support the designer by giving a blink into the later life-phases and context. It supports the early verification of product requirements and can also help to discover unseen requirements by means of use-case scenarios inside VR. A user and task oriented development method using VR can help to reduce the design uncertainties considerably. Therefore, in the paper a concept for user and task oriented model for product development in VR is presented. This model implements the actor(s) and the surrounding(s) along with the product inside VR using a MBSE approach. The standardised multidisciplinary modelling language the Systems Modelling Language (SysML) will be used for the modelling and an example will be presented as a proof of the presented concept.

Keywords: Systems Engineering (SE), Virtual Engineering (VE), Design to X, Use-case scenarios, Systems Modelling Language (SysML)

Contact:
Atif Mahboob
Technische Universität Ilmenau
Institute for Design and Precision Engineering
Germany
atif.mahboob@tu-ilmenau.de

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

1 INTRODUCTION

It is the task of any product designer to consider all life-phases of the product being designed in advance and to ensure that the final solution complies with the requirements of all phases. General product life-phases are product planning, product development, production planning, production, distribution, use, service and disposal (according to VDI 2221). The problem today is that the products have become more complex, they are usually multi-disciplinary, customer demands are increasing and at the same time development time is decreasing. In order to cope with these challenges a large (and still growing) number of methods and digital support tools are being applied in product design. One area of support methods and tools are Virtual Reality (VR) environments and tools. The advantage of VR technologies is that they can directly show (visually and/or acoustically and/or tactile) the future product and its behaviour in a respective context. However, until today it is extremely time-consuming to establish so-called VR scenarios – the whole scene has to be programmed in advance, changes in either the product or the context require major modifications. Another problem is that in many present VR applications in design the designer himself/herself is expected to interact with the product – so to say as a substitute for the humans (or systems) that will work with or on the product in reality later.

This paper reports on advances made in a research project that follows another approach. The basic idea is:

- A VR scene is split up into the product, the environment and human actor(s).
- All of them have structural and behavioural aspects.
- Environment and actor(s) are specific for certain life-phases (e.g. in the use phase actor = product user/operator, in the service phase actor = service technician).
- By (re-) combination of product, environment and actor(s) it should be a lot easier to construct specific VR scenarios (compared to programming the whole scene individually) and to modify them along the design process (e.g. because of the product model evolving).

In order to make the understanding of this paper easier for the reading, the authors would like here to define the terminologies used in the rest of the paper

- "Context" - a specific combination of environment and actor(s) in a specific product life-phase.
- "Use-case" - specific demand and interaction of environment and actor(s) with the product.
- "VR-user" - the designer.
- "Environment" - the surroundings of a product in different life phases.

The context of the product is different in each of its life-phase: A product comes in contact with actors in different roles and expectations in each life-phase. For example, in the use phase of the product the designer tries to develop a product that will fulfil its function and will be in accordance with additional needs of the user (e.g. efficiency, ergonomics, aesthetics, …). However, in the maintenance phase of the product a technician comes in contact with the product and wants a product with easy diagnosis capabilities, easy access to worn-out or broken components, easy replacement, etc. Therefore, a product in each single life-phase has different use-cases, states, and in all these use-cases a different system context has to be considered.

In this paper, we will present a concept for a model for creating different use-cases of the product inside VR. The validation of the concept is performed using a CAVE-type virtual environment (Section 5). At present the representation is focussed on visualisation, despite the fact that the available VR system could also represent acoustics, see Siegel et al. (2016). In addition, tactile (haptic) properties would be of benefit, but are not included in this project.

2 BASIC MODEL

Figure 1 shows two of the mentioned use-cases: using the product and servicing it, at the same time demonstrating the respective contexts.

- *E* represents all (functional, interactional, disturbing, …) inputs to the system – intended as well as unintended – while *A* represents all outputs produced by the system.
- The context can be understood as a use-case-specific environment along with the respective actor(s), e.g. a fabrication machine operator, an end user or a service technician. The context also contains the environment that is not necessary for the functional behaviour, but has to be considered with regard to disturbances.

The designer has to encounter with a lot of uncertainties during the design process and the main challenge during the product development is to foresee and consider all requirements resulting from the use-cases in the different life-phases and the system contexts.

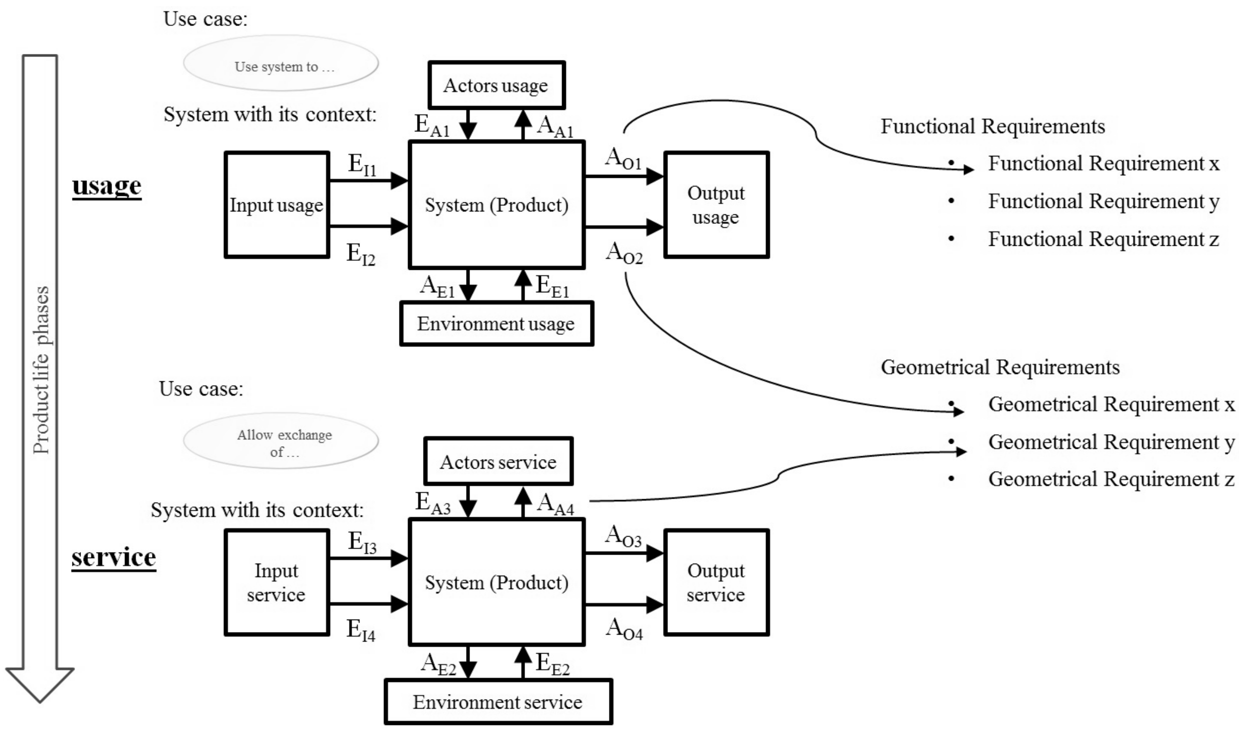

Figure 1. Product context in different product life-phases

In Figure 1, the product is considered as a black box called "system" - a normal way to reduce complexity during the design process. The system has actors who use it or perform service on it. Corresponding to the actors there is a particular environment (maybe several environments), in which the product will be used or serviced. In both use-cases, the product may interact with single or multiple actors in an environment and take some inputs and convert them to outputs after performing certain operations on them. The important thing to understand from both these use-cases is the change in the system requirements. In usage, it is important that the product fulfils its functions and produces the expected/intended outputs. On the other hand, in service the functional requirements may not be important; instead the geometrical properties may be important, e.g. the position and ease of replacement of a defective part. It is well known (e.g. VDI 2221) that in order to make a successful and competitive product for the market, the product designer has to analyse and consider all the later life stages of the product and the requirements arising from them.

Today the designer in general has to imagine all of the relevant use-cases or uses models like in Figure 1 himself/herself. This task requires creative imagination or a lot of experience. The challenge is to identify and understand all relevant requirements correctly. In this research our goal is to support the designer in the early design phases by means of a user and task oriented Virtual Reality (VR) environment. This shall consist of easily configurable scenarios with life-phase specific models along with models of actors and the environments for quick and easy analysis of use-cases in different product life-phases.

In principle, Figure 1 is a more formalised representation of the so-called relational properties introduced by Andreasen and Mortensen (1996), further developed by Weber (2007). The new thing is that in each life-phase the product, the environment and the actor(s) are to be considered and modelled separately in order to be able to re-use and re-combine these constituents easily for new tasks.

Another idea behind the work is that in this way the designer can look at the complete situation (consisting of product, environment and actor(s)) from the outside instead of having to "play" the role of the actor(s) himself/herself while imagining the environment.

The model for each life-phase needs a representation for the product characteristics (or design parameters) and the properties (or behaviour) – designations according to CPM/PDD approach of Weber (2005). The model will be described by means of a standardised multidisciplinary modelling language i.e. Systems Modelling Language (SysML) and a model based approach will be developed for the fast and easy configuration of scenarios inside VR. The 3D representation of a product in its use-case specific environment and the actor(s) along with required behavioural descriptions of product, environment and actor will deliver a real-time product use-case in VR, which can help the designer to spot the strengths and shortcomings of the solution early. An example model will also be presented to establish model based approach.

3 RELATED WORK AND RESEARCH QUESTIONS

Various methods, digital models and tools are propagated to support product development processes. However, they almost entirely focus on modelling the product and its behaviour. As has been shown in the previous section, in addition to these aspects the context consisting of humans and environment in each life-phase of the product plays an important role. Stark et al. (2009, 2010) take a deeper look into this topic and analyse the deficiencies in the currently available software solutions. Metag et al. (2008) analyse the requirements for the product model in VR and focus on the user centred approach to reduce the uncertainties in the development process. Furthermore, this work talks about the advantages of VR as it can help to analyse the design requirements by means of product use-case scenarios inside VR. Weber and Husung (2011) discuss the virtual product development process and propose a new methodology for the validation of product properties through virtual prototypes and provide an example in VR to analyse the dynamic and acoustic behaviour of the product. Abidi et al. (2016) throw light on the contribution of VR for production lines and simulate a system inside VR and discuss the problems of the integration of a simulation tool into the VR system. The work uses a SysML based model for behavioural programing which is converted into a simulator model to achieve the required simulation model inside VR. Albers et al. (2015) talk about the use of MBSE (Model-Based Systems Engineering) in the product development processes and particularly examine the role of SysML in construction kit development. A construction kit has components of specific functionality and built, which can be combined to form a complex system. The work performs the examination on two case studies (early stages in the complete vehicle development and series development of drive-train components) and concludes that SysML has the potential for the construction kit development. Follmer et al. (2010) say that mechatronics design processes are multi-disciplinary and SysML possesses the capability to support engineers by organising the information about requirements, structure and the behaviour of the product under development. The work also mentions the shortcomings of SysML, as it is not directly executable, and also suggests the solution for it i.e. it can be executed by the help of additional software tools. This work also includes a brief discussion of the SysML diagrams by modelling a washing machine. Another work by Silhavy et al. (2011) particularly discusses the behavioural modelling in SysML and explains its behaviour diagrams with the practical example of an audio player. Carrol, (2000) argues about the effectivity of a scenario based design approach and says that scenarios not only help to keep the design process focussed, but also help the designer to evaluate the design in different levels and perspectives.

In conclusion, literature shows that VR has great potential to help the product development process. However, current VR solutions cannot fully support the product development process. A main limitation for VR applications is the effort needed for the preparation of VR scenarios. The related work done in this field suggests the application of SysML as it can help to reduce the needed effort for VR environment considerably. Furthermore, related work also signifies the use of user oriented approaches for an effective VR-supported development process.

The aim of this paper is to present basic concepts in order to answer the following research questions:

- How to create separate isolated models for product, actors and environment including behavioural descriptions?
- How to combine them for specific use-cases in order to build a use-case specific scenario in VR?
- How to achieve a generic (formal) description of a VR model for use in product development?

In order to further investigate and develop the concept of a user and task oriented model-based approach for product development supported by VR the SysML will be used. Therefore, the next section will briefly introduce SysML and the concept of Model Based Systems Engineering (MBSE).

4 MODELLING BASIS: MODEL BASED SYSTEMS ENGINEERING (MBSE) AND SYSML

MBSE is a systematic and holistic development approach by use of seamless system modelling from stakeholder and requirement specification up to system integration, verification and validation. The application of the model driven approach leads to several advantages in contrast to the traditional document oriented way of working. The advantages are for example: vertical and horizontal traceability, re-use of models for product generations and part families, one source for communication between disciplines. The most important society of Systems Engineering is the International Council on Systems Engineering (INCOSE). A common modelling language is SysML (Systems Modelling Language). A big advantage of this language is the division between the model repository and the model views for modelling and analysis. The repository contains the overall model with all elements and connections. In contrast, the views show only context specific elements in order to reduce the modelling and analysis complexity. The views in SysML are realised in terms of diagrams that are divided into structural, behavioural and requirement. The authors consider the detailed explanation of the SysML diagrams out of the scope of this article. However, the diagrams used in the explained example will be briefly discussed later.

5 CONCEPT AND METHOD

As explained in the introduction the motivation behind the concept explained here is to support product designers by enabling him/her to have a look in the later product life-phases, each of them represented by an appropriately modelled use-case. As an example, Figure 2 shows a couple of life-phases of a product (here a vacuum cleaner).

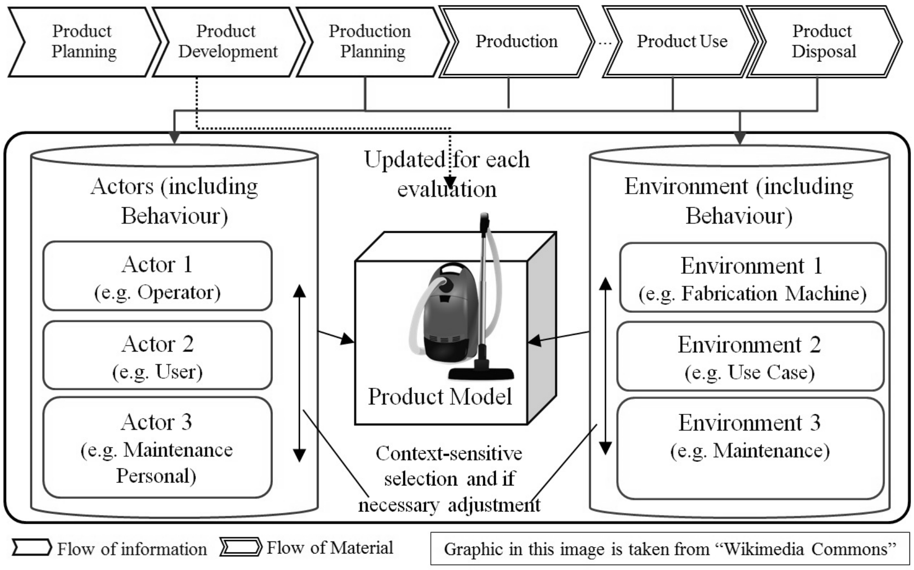

Figure2. Context of a vacuum cleaner (Graphic Reference: Common, (2016))

The product is shown along with its possible life-phases, interacting actors and respective environments. The product itself always bases on the current state of development; therefore, with a new revision (changed characteristics/design parameters) the model has to be updated. A VR-representation of the contexts the product is confronted with in different life-phases can support the designer to have an "outside" look at respective situations, thus helping to recognise and understand missing requirements. VR is a powerful technology that can realise a quick and easy representation (visualisation, auralisation, …) of a product in its possible environments along with the relevant actors. Models in VR are represented via a 3D scene description and interaction channels to manipulate the product in a certain environment with the actor and here the designer can come in as the user of VR viewing at the complete scenario. Different product life-phases can be constructed inside a VR environment by implementing, besides the product itself, the respective environment and the actor(s). All of them have to be equipped with geometrical, but also with behavioural descriptions. While current research and development in the field of VR is mainly focussed on the improvement of visual presentations and increasing the computation speed, our project explores the construction of more complex, i.e. behaviour-containing scenarios and faster preparation of a VR scene by modularisation and the re-use of the environment and actor models for different use-cases.

The geometrical representation is normally available in the form of CAD geometry. Most of the available VR scene rendering software tools provide the possibility for the import of CAD geometrical data by means of different geometrical exchange formats (STEP, VRML, STL etc.), as VR does - unfortunately - usually not work on the CAD geometry but on simplified, secondary geometry such as VRML, JT, etc. For the configuration of the VR scenario characteristic coupling surfaces are needed, see Kirchner et al. (2007).

Currently, there is no well-established method available for building behavioural models in VR. All that can be done is based on manually adding simulations to the corresponding geometrical parts inside the VR rendering software which is cumbersome, time consuming and has to be adapted with every change of the product.

Therefore, a new efficient method for the behavioural VR scene configuration is needed that can make

- The configuration of the scene easier and faster (even for complex systems).
- The models re-usable in different scenes.
- Extended designer interaction with the scene possible.

Furthermore, as discussed further above, a VR scene should comprise of actor, system (product) and environment models and these models should be capable of interacting with each other. This puts extra requirements in the new method i.e.:

- It must be capable of describing the single models separately from each other with appropriate interfaces to connect them with each other.
- It must enable interactions between these models as well as interactions with the user so that an interactive scene can be achieved.

Considering the requirements mentioned above, SysML as the language for behavioural descriptions in VR was chosen. The reasons for this choice are explained by means of the graphical description in Figure 3.

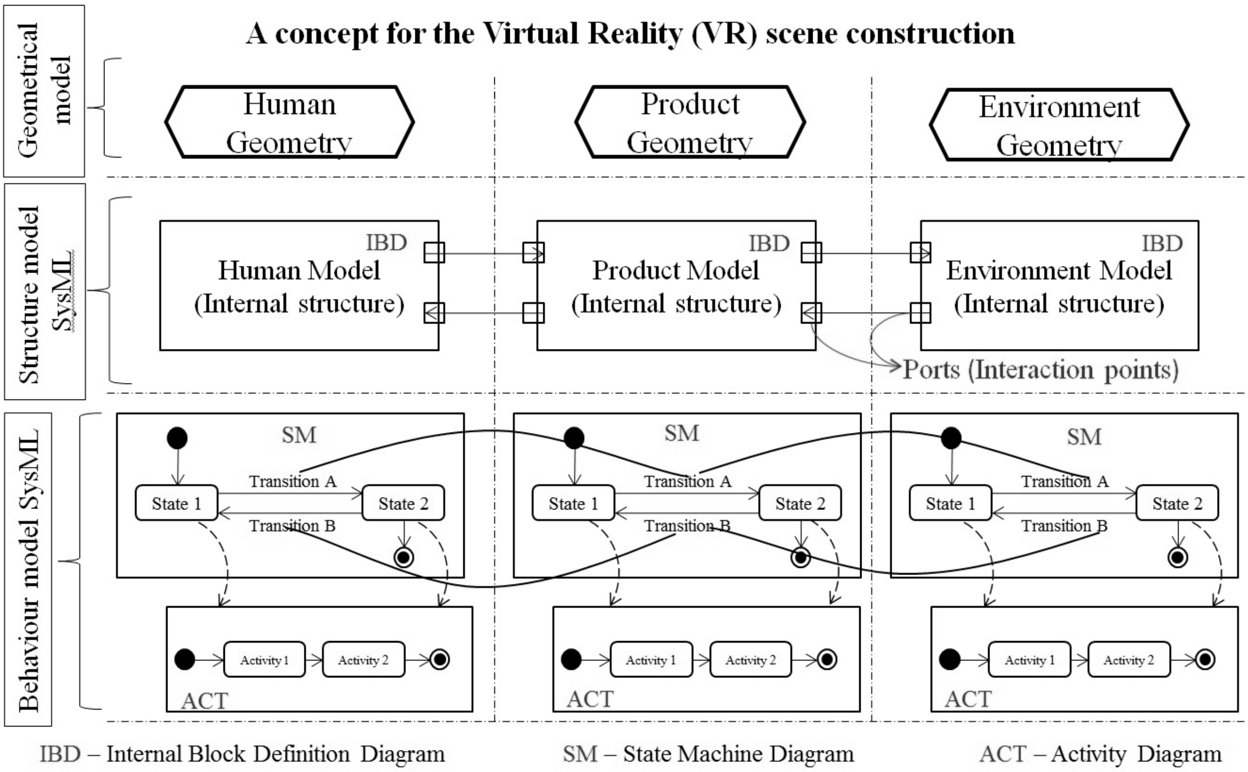

Figure 3. Concept for the Virtual Reality (VR) scene configuration

Figure 3 shows the three models needed to build one use-case for the product. Each model possesses geometry needed for the visualisation in VR. The separate model will be aligned via characteristic coupling surfaces according to the kinematic chain. The second part of each model is the model-based description in the form of SysML. This description is divided in two parts; the structural model and the behavioural model. The structural model consists of the Block Definition Diagram (BDD) and the Internal Block Definition Diagram (IBD). For reasons of space and to keep the focus on the modelling of interactions between different models, BDD is not discussed in this paper. IBD shows the internal structure of the parts and the interaction points (SysML ports) of the model with the outside world. The behavioural model shows a state machine diagram (SM) that represents the different states in which the model can find itself, e.g. "on" (state 1) and "off" (state 2). The change of state requires a transition that can be an internal or external event, a signal or a time event. Every state of the model represented in the state machine diagram can have one or more activities that describe the behaviour of the model in that state. An activity diagram (ACT) actually models the flow of actions or events that should be performed to complete a behavioural feature. For instance, in order to use a vacuum cleaner, it needs to be plugged to the socket, switched on, pulled on the floor, sometimes having collisions with objects in the environment, etc. In this way, an independent SysML model can be constructed that has ports for interconnections with other models when needed. In Figure 3, the models are connected with each other by means of ports. The ports can model the signal flow from one model to the other, but also operation requests, an action performed or a flow of materials. The state machines in the behaviour model can contain the transitions that depend on a transition or behaviour happening in another model and thus enable an interaction between the models. This way SysML can model the actor, product and environment separately and can also model the interaction between them– see the second group of system requirements listed above.

Now the authors would like to discuss the other three requirements about ease of configuration, re-usability and designer interaction with the scene. So far an actor inside the scene, who interacts with the product was considered. If the designer needs to interact with the scene himself/herself without an actor e.g. to check the product usage, it can be performed by taking the model of an interaction device for the VR and using its SysML model instead of the actor model. The models constructed in SysML have high re-usability property, because they can be used in comparable scenes or even for the later generation/different series of the same product.

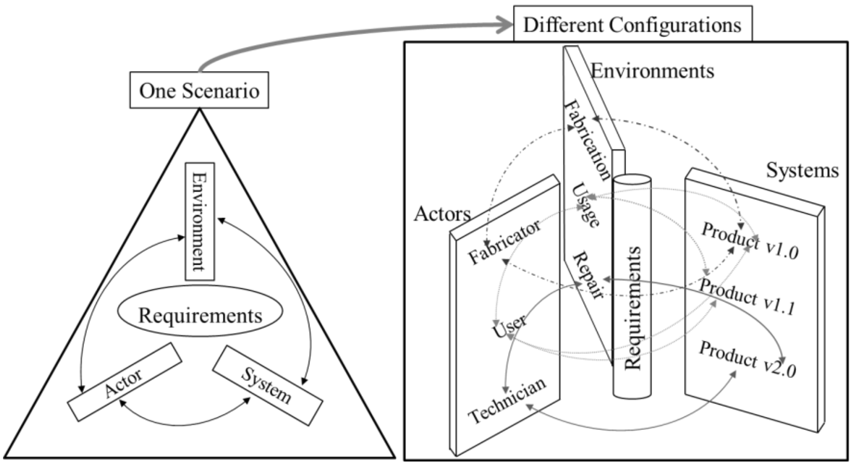

Figure 4. Use-case building in a model based approach for VR

Figure 4 shows a model-based approach for three different configurations represented to create use-cases in VR. Each use-case has requirements associated with it and the use-case in VR can help the designer to verify the fulfilment of the requirements. For example, the designer finds some limitation in the design of *Product v1.0* in the usage scenario in VR that is constructed using the actor that is user and the usage environment. Now, there is revision needed in the design and *Product v1.1* is developed. The new scenario can be constructed by just replacing the product model.

The concept for the flow of information, execution and the scene configuration is demonstrated by Figure 5. The "Database" will contain the models for actors, environments and product versions. In current product development processes in industry, the SysML models are developed alongside the design to perform different analyses on the design Kleiner and Husung (2016), and CAD models of products are constructed before the fabrication/production phase. In the concept explained here, separate actor and environment models are needed in addition; however, their construction is a one-time task, as they are re-usable for evolving versions or the later generations of the product. First the geometrical models of product, environment and the actor will be loaded in the VR software. These geometries are placed inside VR software in the form of hierarchal scene objects. The VR software will now have a scene for a particular scenario and it will be presented to the developer/designer, e.g. using a CAVE-type VR system. The information about the currently loaded models in the scenario will be sent to the "SysML Execution Platform" that will send a request to Database for the relevant SysML model. The SysML model containing the scenario description will also be loaded from the Database. It will monitor for the change in "Scene Object Properties" against any action of the actor, it will change the properties of the objects accordingly. Now the product designer is observing the scenario inside VR; he/she observes the actor acting with/on the product, the product reaction to the actions and the environment behaviour.

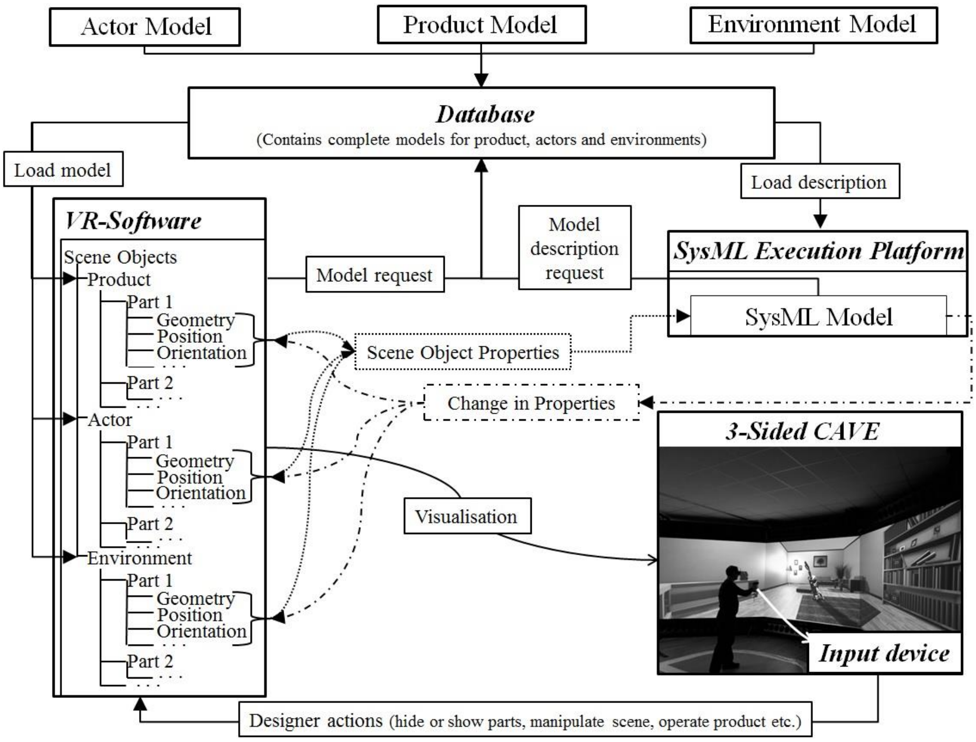

Figure 5. Overview of complete process for VR scene configuration

Manipulations of the VR scene such as a change of the view point, but also changing the product, the environment or the actor can be accomplished by normal VR interaction devices, e.g. a flystick. The designer can also load or suppress certain elements of the product or the environment. In this case, the VR software will show/hide elements or search for the requested model in Database and add it to the scene. At the same time the change in the scene will be communicated to the "SysML Execution Platform" that will also load or suppress the relevant SysML description and will compose the new scenario for visualisation. This mechanism makes the (re-) configuration of a VR scene easier and faster compared to the present situation where, after any change, the complete scenario needs to be reconstructed.

6 EXAMPLE MODEL

In order to demonstrate the application of the concept presented in the last section, in this section a practical example will be shown. Particular focus is on demonstrating the advantages of using SysML; the example itself is quite simple and very common: a domestic vacuum cleaner as the product, a living room as the environment and a user who will perform the cleaning. At first the three models were created with both structural and behavioural descriptions. Later these models were combined with each other to achieve a vacuum cleaner usage scenario inside a living room.

The most important interactions between these models were modelled in IBD, shown in Figure 6. It is very important that each of the models have interfaces (in the form of ports) with the same context in both models that need to be combined. For example, the user can perform some actions (on, off, open cover, etc.) and some movements (turn, move forward/backward, etc.) on the product, so the product model needs to have interfaces to receive these actions and movements. On the other hand, the product can show signals to the user (e.g. lights, indicators, aural signals); therefore, an interface for warnings is required on both models. Each model contains sub-elements that refer to the objects in the VR-software, e.g. a living room has sofa, table, etc. Each part can have a certain behaviour, e.g. in the case of the living room all objects can collide with the product or user during movement.

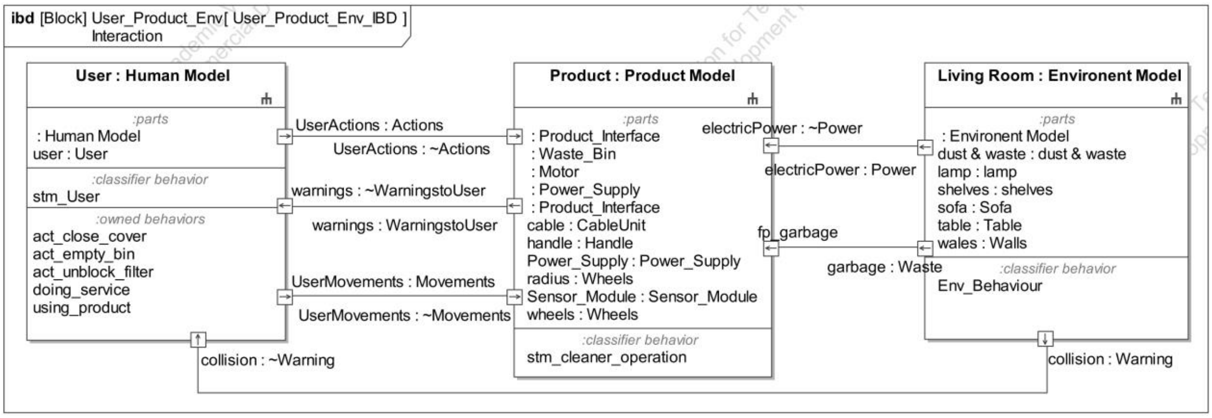

Figure 6. Internal block definition diagram for usage scenario

The behavioural model for the complete scenario can be represented by means of state machine diagrams as shown in Figure 7.

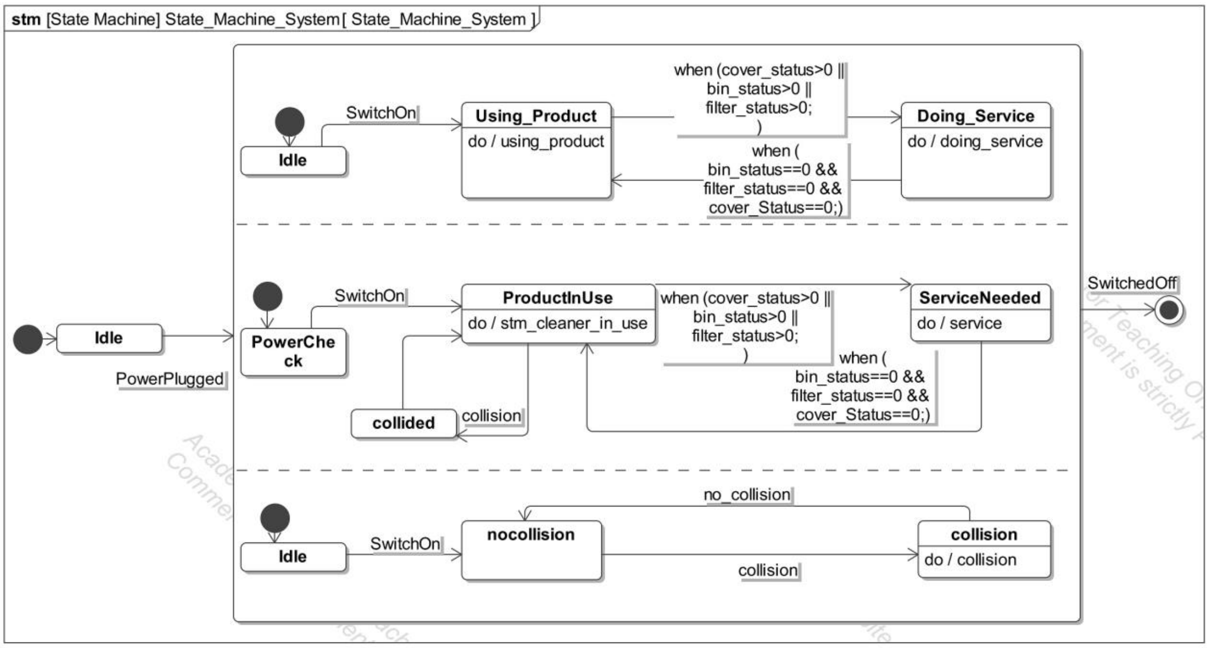

Figure 7. State machine diagram for the overview about the use-case "usage" of a vacuum cleaner

The behaviour of all three models is shown as an individual behaviour inside a "composite state". In the figure, the state having multiple regions and states is a composite state. The objects need certain events or signals to occur for the execution of actions, e.g. plugging the device to the power supply, switching it on, opening the cover, emptying the waste bin, cleaning a blocked filter, etc. On the other hand, the product monitors the status of its internal components and, if necessary, changes its state (e.g. from use to service state if the cover is opened, the dust bin is full or the filter is blocked) and informs the actor by means of relevant warning signals (can be understood as the warning LEDs on the physical vacuum cleaner). The actor can then perform certain actions (closing the cover, emptying the bin, and cleaning the filter) which again cause a transition in the product model state (back to use). Similarly, the environment model can show a collision that will be communicated to the user in the form of a warning while the further movement of the product in the respective direction will be restricted.

7 CONCLUSION AND FUTURE WORK

In this paper, an approach was presented for creating isolated models for systems with its context using SysML and the interaction between the models was modelled inside SysML that results in the combined behavioural description of a use-case for creating a VR application. These SysML models offer interactions, the possibility of reusability and a combined behavioural description for a VR-application. This paper also sheds light on the efficient use of VR technology by introducing the behavioural features for a model along with the visualisation and the possibilities for reducing the effort for VR scene configuration. In the future, the authors will focus on the construction of the actor and environment models for more use-cases and the first simulation will be tested inside the VR environment.

REFERENCES

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*, 8(3), p.182.

Albers, A., Scherer, H., Bursac, N. and Rachenkova, G. (2015), "Model Based Systems Engineering in Construction Kit Development – Two Case Studies", Procedia CIRP, 36, pp.129-134.

Andreasen, M.M., Mortensen. N.H. (1996), "Basic thinking patterns and working methods for multiple DFX". In: *Proceedings of the 7th Symposium "Design for X"*, Universität Erlangen-Nürnberg, 1996/97, pp. 7-12

Carrol, J.M. (2000), "Five reasons for scenario-based design", *Interacting with computers*, 13(1), pp.43-60.

Follmer, M., Hehenberger, P., Punz, S. and Zeman, K., 2010, "Using SysML in the product development process of mechatronic systems", In DS 60: *Proceedings of DESIGN 2010, the 11th International Design Conference*, Dubrovnik, Croatia.

Common, W.(2016), "The vacuum cleaner graphic", (Accessed on 12.12.2016) URL:https://commons.wikimedia.org/wiki/Vacuum_cleaner#/media/File:Blue_vacuum_cleaner.svg

Kirchner, A., Husung, S., Krömker, H. and Metag, S. (2007), "Functional Investigations of Mechanisms using CAD-modeled Constraints", In: Shigeaki Tsunoyama (ed.): *10th International Conference on Humans and Computers*, Düsseldorf/Aizu, 13.-15.12.2007. University of Aizu, Japan, pp. 59-64.

Kleiner, S. and Husung, S. (2016), "Model Based Systems Engineering: Prinzipien, Anwendung, Beispiele, Erfahrung und Nutzen aus Praxissicht", Tag des Systems Engineering, pp. 13-22.

Metag, S., Husung, S., Krömker, H. and Weber, C. (2008), "User-centered Design of Virtual Models in Product Development", Proc. of 53rd IWK, Ilmenau.

Siegel, A., Weber, C., Mahboob, A., Albers, A., Landes, D., & Behrendt, M. (2016), "Virtual Acoustic Model for the Simulation of Passing Vehicle Noise", *In ASME 2016 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. V01BT02A055-V01BT02A055)

Silhavy, R., Silhavy, P. and Prokopova, Z. (2011), "Behavioral modeling in system engineering", In: *13th WSEAS international conference on automatic control, modelling & simulation*, pp. 100-105.

Stark, R., Hayka, H. and Langenberg, D. (2009), "New potentials for virtual product creation by utilizing grid technology", *CIRP Annals-Manufacturing Technology*, 58 (1), pp. 143-146.

Stark, R., Krause, F.-L., Kind, C., Rothenburg, U., Müller, P., Hayka, H. and Stöckert, H. (2010), "Competing in engineering design – The role of Virtual Product Creation", *CIRP Journal of Manufacturing Science and Technology*, 3(3), pp. 175-184.

VDI 2221, (1987), "VDI-Guideline 2221: Systematic Approach to the Design of Technical Systems and Products", VDI, Düsseldorf, 1987.

Weber, C. (2005), "CPM/PDD – an extended theoretical approach to modelling products and product development processes", In: *Proceedings of the 2nd German-Israeli Symposium on Advances in Methods and Systems for Development of Products and Processes*, pp. 159-179.

Weber, C. (2007), "Looking at 'DFX' and 'Product Maturity' from the Perspective of a New Approach to Mod-elling Product and Product Development Processes. In: "The Future of Product Development" (ed. by Krause, F.-L.), pp. 85-104. Springer, Berlin-Heidelberg 2007.

Weber, C. and Husung, S. (2011), "Virtualisation of product development/design – seen from design theory and methodology", In: DS 68-2: *Proceedings of the 18th International Conference on Engineering Design (ICED 11), Impacting Society through Engineering Design*, Vol. 2: Design Theory and Research Methodology, Lyngby/Copenhagen, Denmark, 15.-19.08. 2011.

ACKNOWLEDGEMENT

The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG) for supporting this work under the funding label KR 3297/3-1 and WE 1989/6-1 within the research project "user and task oriented virtual model for product development".