



SYSML BEHAVIOUR MODELS FOR DESCRIPTION OF VIRTUAL REALITY ENVIRONMENTS FOR EARLY EVALUATION OF A PRODUCT

A. Mahboob, S. Husung, C. Weber, A. Liebal and H. Krömker

Abstract

Virtual Reality (VR) can help designers to realise later life phase situations of a product in presence of a specific actor (or actors) in a specific environment by building product use-cases in VR, thus facilitating an early evaluation of a design. In this paper, we present a model-based approach that uses SysML behaviour models to describe and control VR environments. The behaviour modelling process with SysML and the simulation results obtained using these behaviour models are discussed. An overview over simulation processes is provided by constructing an example product use-case in VR.

Keywords: design to x (DtX), systems engineering (SE), virtual engineering (VE), virtual reality (VR), systems modeling language (SysML)

1. Introduction

During the product development process, it is necessary to consider later life phases of the product and its interaction with neighbouring systems (e.g. actor(s) and environment). Today, for these considerations designers build different use-cases of the product based on (mostly textual) requirements arising from different stakeholders. One goal here is to gain the understanding about the product and its behaviour in its later life phases in order to develop a solution that can meet the relevant needs of all the stakeholders. The products are no longer considered as an isolated system; instead, the context consisting of life phase specific environments and life phase specific actor(s) are also considered during the design phase. According to the guideline VDI 2221 (1987) the general life phases of a product are product planning, product development, production planning, production, distribution, use, service and disposal. There are different actor(s) and environments associated with these life phases; e.g. in the use phase the actor can be a user (or multiple users) inside a usage environment. A successful product in its use phase should be able to fulfil the needs of the user in a particular usage environment. Similarly, the product in each of its life phases has a different context (= actor(s) and environment) that should be considered at the design stage. A relevant concept in this regard is the work of Mahboob et al. (2017a) that discusses the context of a product and the construction of different use-cases of a product for different life phases as shown in Figure 1. In Figure 1, the product is considered as the system under development, E represents all (functional, interactional, disturbing, ...) inputs to the system and A are the outputs produced by the system. Each life phase of a product can have different requirements. Building the use-cases can help greatly to verify the fulfilment of these requirements. Figure 1 is based on the relational properties introduced by Andreasen and Mortensen (1997) which are further extended by Weber (2007). The verification of the different use-cases can be realised with virtual or physical methods. The virtual verification of these use-cases can be performed much earlier than the first physical

prototype and can help the designer to gain an early understanding of product behaviour in its later life phases.

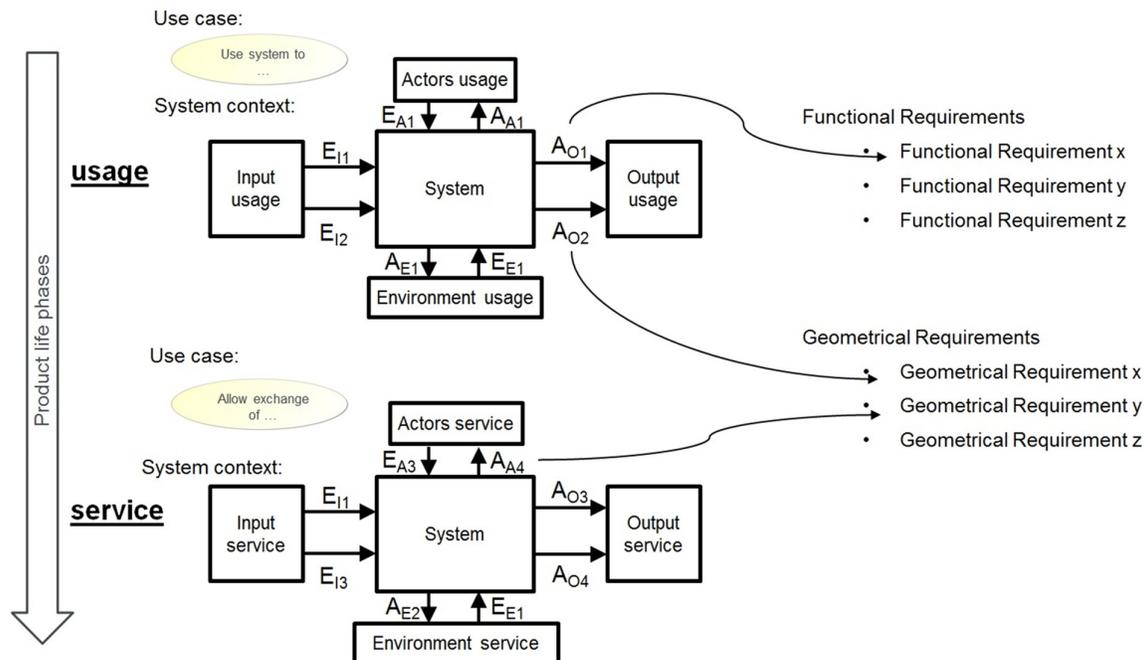


Figure 1. Product context in its different life phases (Mahboob et al., 2017a)

Virtual Reality (VR) technology can be of great use at this point as it can directly show (visually and/or acoustically and/or tactile) a product in its later life phases along with its context. Product's later life situations and product's interaction with actor(s) and environment can be evaluated inside VR by means of product use-case scenarios. However, the preparation of VR models is a challenging task as it is currently based on time-consuming programming of the whole VR-scene in advance. This limits the application of VR in the industry. Mahboob et al. (2017b) identifies the following three reasons for this limit:

- The time and the effort needed for the preparation of virtual models is very high,
- these models are also limited in terms of their reuse and/or reuse of parts of them and
- once constructed, very few modifications are possible, therefore a new situation can force a complete new preparation.

We are focusing on a new approach that is based on dividing the complete VR-scene into three sub-models i.e. product, actor(s) and environment as also described in Mahboob et al. (2017a,b). Also, in this approach, the product designer is the observer of the VR scene, instead of playing him/herself the role of an actor. This approach helps to overcome the above mentioned limitations of VR application. However, this approach puts extra requirements on the sub-models (for actor(s), product and environment) i.e. they must be described as independent from each other as possible, they must be able to interact with each other, this interaction should be described and they should be described with enough degree of generality to make them reusable.

The authors consider the Model Based Systems Engineering (MBSE) approach most suitable for the description of such sub-models. For the standardisation of the sub-models MBSE with Systems Modeling Language (SysML) as the language of behavioural description for the sub-models is used. Furthermore, the VR system itself mainly supports the visual representation of geometrical objects and does not directly process the capability to incorporate the physical properties of objects. A real time product evaluation can only be achieved through the incorporation of physics simulation along with the visual representation of geometrical objects inside a virtual environment. Therefore, we also focus in this paper on the integration of physics simulation in the VR environment. This is done by integrating a

physics simulator with the behaviour modelling in SysML. We use an open source version (education version) of (V-REP) virtual robot experimentation platform V-REP (2017) that offers multiple physics engines to simulate the physical properties of the objects.

In the next section, we will briefly introduce the related work done in the field of VR, the modelling approaches and the current methods in MBSE. Later, we will describe the overall simulation process and our behaviour modelling approach for modelling the sub-models will also be presented.

2. Related work

There are various methods and digital models available that facilitate the product development process. These available models mainly focus on the product and its behaviour. As already discussed in the introduction, the life phase specific context (actor(s) and environment) also need to be considered during the product development process. However, the influence of context is only implicitly contained in present models. VR can support the product development process by realising a later life phase situation of a product along with its context in form of a VR-scene. The current methods and tools available for product development in VR do not fully facilitate the product designer in fulfilment of his/her tasks. These limitations in the software solutions are studied by Stark et al. (2009, 2010) who discuss the discrepancies in currently available software solutions. Metag et al. (2008) focus on a user centred approach using VR to reduce the uncertainties in the product development process. The work of Weber and Husung (2011) discusses the process of virtual product development and proposes a new classification for the development of virtual prototypes. An example in this work also studies the dynamic and acoustic behaviour of a product. A work from Abidi et al. (2016) is very relevant to this paper, as it focusses on the simulation of production lines in VR and, like the study described here, uses a SysML behaviour model as the base of the simulation. However, the SysML behaviour model is not directly linked with the VR simulation, instead, it is first converted into a simulator model (in ARENA) and then the simulation is controlled using the simulator model. Albers et al. (2015) discuss the use of MBSE in product development processes with focus on application of SysML. This work studies the potential of SysML to support the construction kit (components of specific functionality and built that can be combined to form a complex system) development process. Gausemeier et al. (2013) suggest the use of the systems engineering approach, as it can handle the complexity and supports the efficient development of the multidisciplinary products of today. Recently, two cooperation projects between industry and academia have focussed on SysML as a unified language for implementing MBSE: The FAS4M project (Moeser et al., 2015; Moeser et al., 2016; FAS4M, 2017) worked to support the modelling and development of mechatronic systems and has developed an interface between a SysML editor and CAD tools. The project mecPro2, on the other side, has focussed on the Product Lifecycle Management (PLM) and has developed a corporation between MBSE and PLM. The project suggested that this combination can handle the complexity of the system under development (Eigner et al., 2014; Meissner et al., 2014; Eigner et al., 2017; MecPro2, 2017). Follmer et al. (2010) emphasize the use of SysML for the organisation of requirements, structure and behaviour of a product. Silhavy et al. (2011) present an example model for a product in SysML and focus on the behaviour diagrams in SysML. There are other works that try to integrate model-based approaches with VR to achieve a simulation in VR: Chevaillier et al. (2012) use a UML model which is then transformed in order to integrate simulations in VR. Abidi et al. (2015) achieve this by doing the integration of VR and SysML. Furthermore, there are more methods and studies (Schamai et al., 2009; Sindico et al., 2011; Batarseh and McGinnis, 2012; Foures et al., 2012; Gross and Mukherjee, 2015; Bank et al., 2016) that either transform the SysML models to another executable language/tool or try to integrate SysML models into existing executable tools to achieve the simulation. However, there is still a need for a method that can pave the way for controlling simulations in VR directly through SysML models.

For using SysML to represent the sub-models relevant in this study (product, actor(s), environment) the following research questions have to be answered:

- How should the behaviour of the sub-models be described in SysML?
- How can the interaction between the models be modelled?
- How can these behaviour models in SysML be used to facilitate a real-time simulation inside VR?
- How can the SysML behaviour models be coupled with an external physics engine to incorporate the physical properties of objects into the VR-scene?

3. Concept and information flow

In order to achieve clarity for the reader, it is important to first provide an overview of the complete simulation process and the components/tools involved in it before describing the modelling process in detail. Therefore, this section will provide a bird-eye view of the VR simulation achieved using the behaviour models in SysML. First of all, the geometrical models of the environment and the product are transferred to construct a scene inside V-REP. The result is an exact copy of the VR-scene, but in a form on which the physics calculation can be performed using built-in libraries of V-REP. V-REP also offers an open Application Programming Interface (API) in multiple languages to control the objects in a V-REP scene during simulation. We used this API to build a control code that can communicate with the SysML behaviour model.

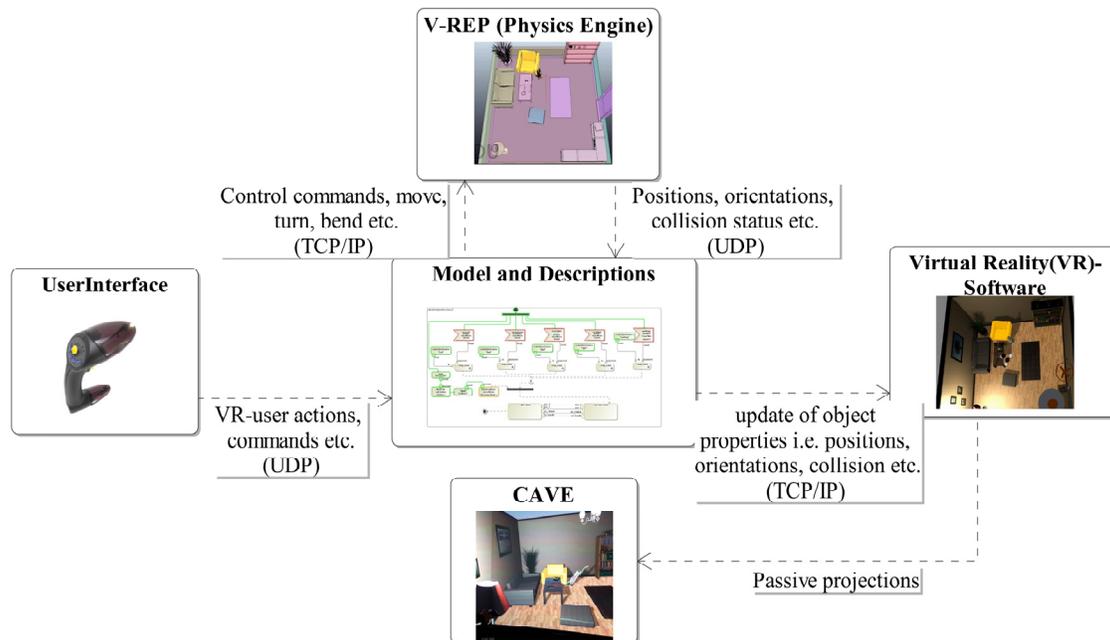


Figure 2. Exchange of information between different tools during simulation

We are using a commercial modelling tool to make the SysML models executable. This execution is based mainly on token flow simulation. The SysML behaviour description at the start of the simulation cycle reads the data from the user interface (interaction device) model. The data that comes as a string, is split up and restructured to extract the action performed by the VR user, e.g. changing positions and orientations, button values, etc. Based on the received data the behaviour description sends a command to V-REP to realise the user actions. The result of this command can change the vacuum cleaner model inside the simulator either by a linear movement or a rotation or a change of the angle of its handle. This induces a change in the position or the orientation of the vacuum cleaner model. This change can now result in a collision between the vacuum cleaner and an environment object. Therefore, the behaviour description in SysML reads the status of the objects from the simulator model continuously and if there is a change in the properties of any object, it is immediately communicated to the VR rendering software.

The collision of two objects inside V-REP is normally indicated by a change of the colour of the colliding objects; this can be read over the API function to see which two objects are colliding. When we send some commands to change object properties (position, orientation, colour, etc.) to the VR software, there should be an interface to understand the incoming command. The VR software used for our simulation has a built-in plugin that can translate or change the orientation of objects by understanding certain commands. However, it was not possible to change the colour of objects in VR using incoming commands. Therefore, we have extended the built-in plugin to include the command

interpretation and implementation for a colour change also, so that a collision can be visualised in VR. This cycle keeps on repeating and the VR scene is updated at a rate of approx. 20 frames/second. In this way, a real time simulation is obtained inside VR using the behaviour description of SysML models. Along with these calculations the VR scene is rendered on a 3 sided CAVE type VR system that provides an immersive VR environment.

Table 1 sums up the commands used for the communication between different tools as already explained in this section.

Table 1. Commands used for communication

Scope	Command
SysML → V-REP	Move forward (movement speed can be given as parameter)
SysML → V-REP	Move backward(movement speed can be given as parameter)
SysML → V-REP	Turn right (turning speed can be given as parameter)
SysML → V-REP	Turn left (turning speed can be given as parameter)
SysML → VR	Translate an object (object name, X, Y,Z as parameters)
SysML → VR	Rotate an Object (object name, degrees of rotation and rotation axis as parameters)
SysML → VR	Select an Object (object name as parameter) This command will select the given object in the tree and this selection will be indicated by a colour change inside CAVE type VR

4. Modelling approach

Last section provided the overview of the overall simulation process and this section will focus on the modelling of the sub-models. The behaviour of the complete VR-scene consists of the behaviours of sub-models and the interaction between them. Therefore, behaviour descriptions of our sub-models must also contain interfaces to communicate with each other. In this section, the process of behaviour modelling will be discussed in detail.

First, we discuss the modelling of interactions between sub-models. Basically interactions can be represented by means of a structural definition in SysML and can be seen in Figure 3 as an internal block definition diagram (IBD). The sub-models communicate/exchange information by means of standard interface points in SysML, i.e. ports. In SysML it is also possible to draw direct dependencies, i.e. linking an element of a sub-model to the other sub-model directly. We have avoided these direct interdependencies between sub-models and instead modelled the interaction over the ports. The reason for this is the reusability perspective of the sub-models.

We have a vacuum cleaner as an example product, a living room as the environment and, instead of a full-blown actor model, in our first studies we have used a “User Interface” model, i.e. a human user operating/controlling the product model in the given environment. An actor model (human avatar in VR) is still under development. Therefore, the actions performed during the simulation so far are driven by the user of the VR system, who interacts with the user interface. The behaviour model of the product can communicate with the user interface and the environment behaviour model over the ports. According to the SysML specification, interactions can be modelled either by defining an “Interaction Block” or by “Call Behaviour Actions”. We modelled the behaviour of both user interface and environment models using “call behaviour action” that can be called directly over the port. The product model continuously calls the behaviour of the user interface to check if the VR user has performed an action/set a command. After that the product model interprets the information received from the user interface model and updates the product behaviour model accordingly. Along with this update, an execution command is sent to V-REP to realise the actions of the user and subsequent behaviour changes in the product model. Once the product model is updated, the product model communicates with the environment model to see if there is a collision taking place inside the simulator model as a result of the last action.

It is important for the reader to focus on the ports labelled as UserAction, UserMovements, call_behav und collision, as they will be used to explain the behaviour models. The behaviour models that we used to achieve the VR simulation consist of multiple activity diagrams, state machine diagrams, opaque

behaviours, etc. (“Opaque behaviour” is a behaviour element in SysML that describes a particular view of a system in a specific language in a modelling tool.) It is not possible to discuss all the diagrams in this paper. Therefore, the authors have decided to discuss one diagram from each sub-model’s behaviour that can provide a brief overview over the basic mechanisms used.

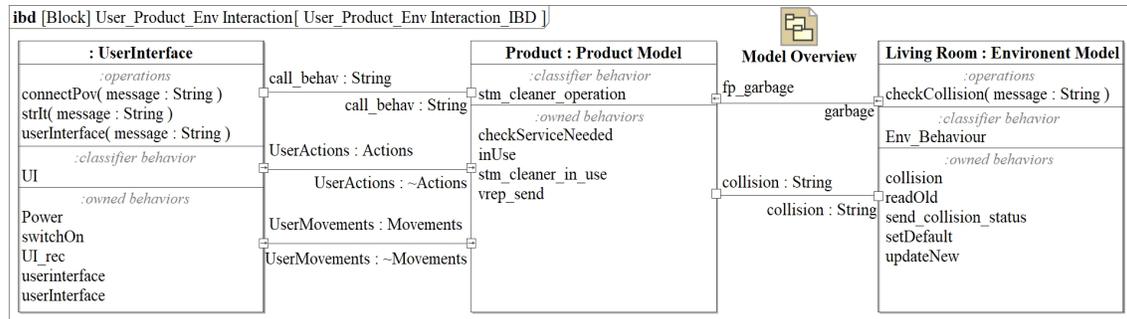


Figure 3. Interaction between sub-models

In Figure 4 the behaviour of the product model is shown. As the product model depends on the actions performed by the VR user, it is modelled using “Accept Event Action” that waits for turnLeft, moveForward, moveBackward, turnRight or nothing signals coming from user interface behaviour model over the port “UserMovements”. As soon as the activity starts accept event actions start to wait for a signal. The product model calls the “UserInterface” behaviour over port “call_behav” using the “userInterface” action. Once the execution of this call behaviour is completed the product model receives one of the signals and sends the appropriate command to V-REP indicated here as “vrep_send”. After that the product model calls the “checkCollision” behaviour of the environment model using the “collision” port. Finally, the product model checks the changes of the product model in V-REP (indicated as rec_data_Vrep) and updates the VR simulation by sending the suitable commands (indicated as VR_Send).

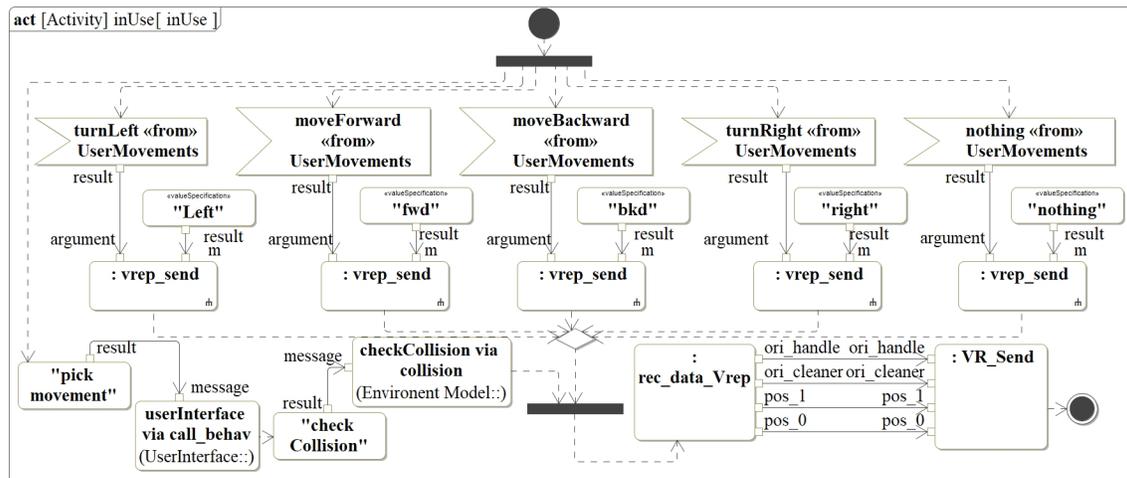


Figure 4. Product model behaviour

In Figure 5 the behaviour of the user interface model is shown. This behaviour is called by the product model to check the user actions in VR. The VR interaction device’s data (position, orientation, buttons status, etc.) is normally available over an internet based UDP protocol. The received data is checked by the user interface behaviour model which then decides which signal will be sent to the product model along with the value “y” that can contain the position or orientation of the interaction device.

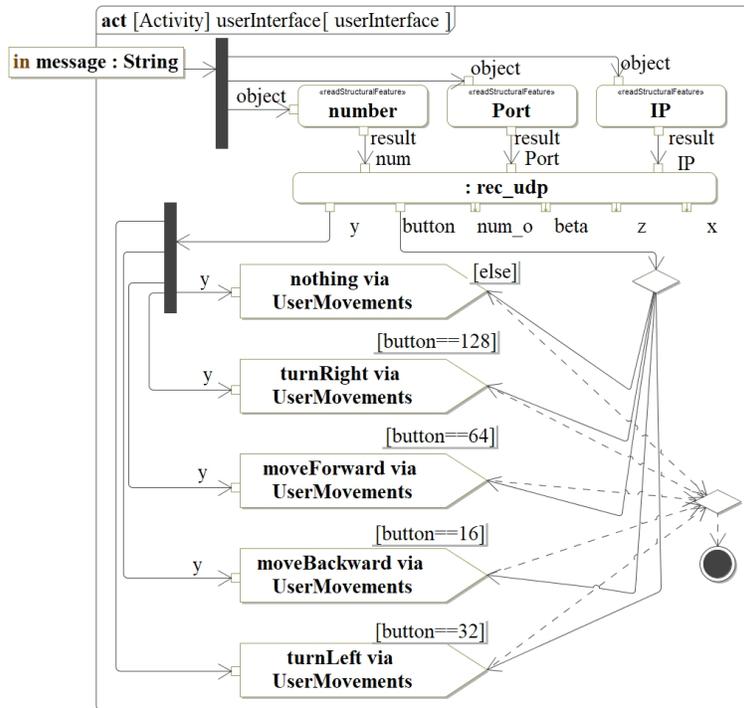


Figure 5. User interface behaviour

Figure 6 shows the behaviour called by the product model from the environment model. When the execution of this diagram starts, the “rec_Simu” receives the status of all the objects in the scene from V-REP.

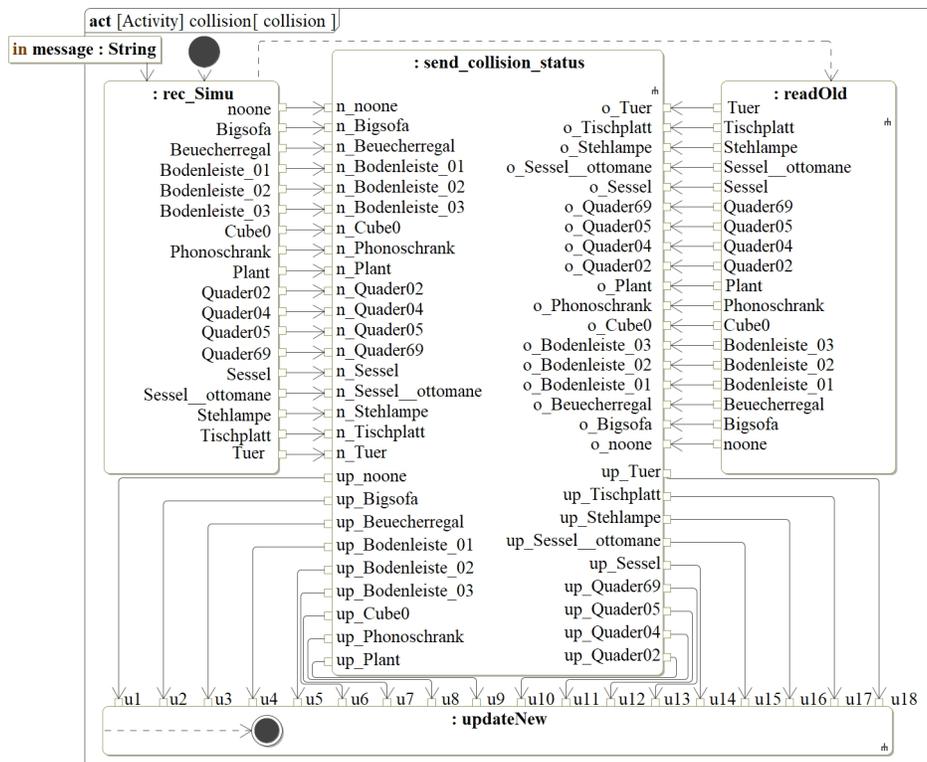


Figure 6. Environment model behaviour

The received data contains the information about collisions of the objects and it is communicated to “send_collision_status” that compares it with the status of the objects from last execution cycle shown as “readOld” on Figure 6. New and old statuses of objects are compared and the changes in the object states (collision/no collision) are communicated to VR system to update the simulation. Finally, the behaviour description of the environment model replaces all the old statuses with the new ones (indicated as “updateNew”), so that they can be used in the next execution cycle. The action “rec_Simu” is modelled as opaque behaviour using jython script (Jython, 2017) and all the rest of the actions shown in Figure 6 contains multiple activity diagrams to complete their behaviour.

5. Discussion

The method presented in this paper shows up a new application of SysML behaviour models, in particular to derive a simulator model and to control a simulation in VR. The actual simulation obtained in a CAVE type VR is real-time as far as its execution capability is concerned. However, there are some limitations in the software solutions available for the implementation of the method. Even though the parameter updates (commands to the VR system) are sent several times in a second, the VR rendering software updates the scene at a much slower rate depending on the rendering speed. This means that the rendering speed is limiting the speed of the simulation and the simulation can be slower as the number of objects and textures in the VR scene will increase. From SysML behaviour model to the VR scene including physical properties of objects is still a complicated simulation process with many data transformation in between different tools. Furthermore, the reverse information flow from VR to the SysML behaviour models can prove difficult (depending on the tools involved.)

On the other hand, the modelling tools have improved considerably in the last years; we have come from just modelling to the execution of models as almost all commercial modelling tools support model execution. We are moving further to integrate modelling tools with existing industrial solutions of product development, e.g. CAD, PLM, requirement management, etc. SysML is developed to be a unified modelling language to apply MBSE methods to all fields and it contains multiple possibilities to describe a certain aspect of some behaviour. This leaves SysML to be abstract and there is a need of more concrete definitions and application guidelines that elaborate the application area for different SysML elements. Today, in industry the SysML models are made alongside the design to perform different analysis on design as discussed by Kleiner und Husung (2016). Therefore, SysML model driven VR simulations can greatly facilitate product evaluation inside VR by overcoming the current limitations in the field of VR and facilitating an efficient preparation of product use-cases in VR.

The modelling approach presented in the last section discussed in detail the methodology for individual behaviour description of the sub-models in SysML, as well as the modelling of the interaction in between the sub-models using SysML-ports. Section 3 provided a complete overview of a simulation in VR that is achieved using the behaviour descriptions in SysML. This section also brought to light the coupling of SysML models with external tools (i.e. physics engine and VR system). The resulting real time simulation achieved in VR also validated the use of SysML models to achieve a real time simulation. Thus, the modelling approach presented in this paper along with the results of simulation achieved validate the research questions that were set up in Section 2.

6. Conclusion and outlook

This paper introduced a new method to control the VR environment by simulations that are based on SysML behaviour descriptions. The aim is to build product use-cases with physically correct behaviour inside VR. The behaviour modelling process in SysML is explained in detail and the results of first VR simulations using SysML behaviour models are also discussed. The results established the approach of dividing the complete VR scene into sub-models and by describing these sub-models in SysML, it is possible to drive a real-time simulation in VR. In the end the SysML modelling tool was linked with a physics engine as well as with the VR rendering software. The integration of SysML and physics engine brings physical behaviour into the VR objects – very important for VR applications in product development. The simulation achieved through division of the complete VR scene into sub-models (in our case: product, actor(s) and environment) and their behaviour descriptions in SysML has clear advantages over the conventional way of programming the whole VR scene in one go. The major

benefits are that the sub-models are reusable, that small changes no longer necessitate a completely new preparation of the VR scene so that a faster way to build new use-cases of a product in VR is opened. In the future we will focus on the completion of actor model, so that a human avatar can be used and the product designer can observe the interaction of product, actor and environment. We will also prepare more sub-models to test flexible recombinations of sub-models in order to make new product use-cases.

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".

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>
- Albers, A., Scherer, H., Bursac, N. and Rachenkova, G. (2015), "Model Based Systems Engineering in Construction Kit Development – Two Case Studies", *Procedia CIRP*, Vol. 36, pp. 129-134. <https://doi.org/10.1016/j.procir.2015.01.044>
- Andreasen, M.M. and Mortensen, N.H. (1997), "Basic thinking patterns and working methods for multiple DFX", *Proceedings of the 8th Symposium 'Fertigungsgerechtes Konstruieren', Schnittach, Germany, October 16-17, 1997*, pp. 7-12.
- Bank, D., Blumrich, F., Kress, P. and Stöferle, C. (2016), "A systems engineering approach for a dynamic co-simulation of a SysML tool and Matlab", *Proceedings of the 2016 Annual IEEE Systems Conference (SysCon 2016), Orlando, Florida, April 18-21, 2016*, IEEE, pp. 1-6. <https://doi.org/10.1109/SYSCON.2016.7490534>
- Batarseh, O. and McGinnis, L.F. (2012), "System modeling in sysml and system analysis in arena", *Proceedings of the Winter Simulation Conference (WSC '12), Berlin, Germany, December 9-12, 2012*, IEEE. <https://doi.org/10.1109/WSC.2012.6465139>
- Chevaillier, P., Trinh, T.H., Barange, M., De Loor, P., Devillers, F. et al. (2012), "Semantic modeling of virtual environments using mascaret", *Proceedings of the 5th Workshop on Software Engineering and Architectures for Realtime Interactive Systems (SEARIS 2012), Orange County, California, March 5, 2012*, IEEE. <https://doi.org/10.1109/SEARIS.2012.6231174>
- Eigner, M., Dickopf, T. and Apostolov, H. (2017), "System Lifecycle Management - An Approach for Developing Cybernetic 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 Cybernetic 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 17.11.2017).
- 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.
- Foures, D., Albert, V., Pascal, J.-C. and Nketsa, A. (2012), "Automation of SysML activity diagram simulation with model-driven engineering approach", *Proceedings of the 2012 Symposium on Theory of Modeling and Simulation - DEVS Integrative M&S Symposium (TMS/DEVS '12), Orlando, Florida, March 26-30, 2012*, Society for Computer Simulation International, San Diego.
- Gausemeier, J., Dumitrescu, R., Steffen, D., Czaja, A., Wiederkehr, O. and Tschirner, C. (2013), *Systems Engineering in der industriellen Praxis*, Heinz Nixdorf Institut, Paderborn.
- Gross, J. and Mukherjee, R. (2015), "Integrating Multibody Simulations With SysML", *Proceedings of the ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Vol. 6, Boston, Massachusetts, August 2-5, 2015*, ASME, pp. V006T10A038. <https://doi.org/10.1115/DETC2015-48095>

- Jython (2017), *Jython: Python for the Java Platform*. [online] Available at: <http://www.jython.org/index.html> (accessed 17.11.2017).
- Kleiner, S. and Husung, S. (2016), “Model Based Systems Engineering: Prinzipien, Anwendung, Beispiele, Erfahrung und Nutzen aus Praxis-sicht”, *Tag des Systems Engineering, Herzogenaurach, Germany, October 25-27, 2016*, Carl Hanser Verlag, München, pp. 13-22. <https://doi.org/10.3139/9783446451414.002>
- Mahboob, A., Liebal, A., Husung, S., Weber, C. and Krömker, H. (2017b), “A method for efficient and task oriented configuration of virtual reality (VR) models for the analysis of technical systems”, *Proceedings of the 59th Ilmenau Scientific Colloquium (IWK), Ilmenau, Germany, September 11-15, 2017*, Universitätsbibliothek Ilmenau ilmedia, Ilmenau.
- Mahboob, A., Weber, C., Husung, S., Liebal, A. and Krömker, H. (2017a), “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.
- 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.
- Moerer, G., Grundel, M., Weikiens, T., Kümpel, 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>
- Moerer, 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>
- Schamai, W., Fritzon, P., Paredis, C. and Pop, A. (2009), “Towards unified system modeling and simulation with ModelicaML: modeling of executable behavior using graphical notations”, *Proceedings of the 7th International Modelica Conference, Como, Italy, September 20-22, 2009*, Linköping University Electronic Press, pp. 612-621. <https://doi.org/10.3384/ecp09430081>
- 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.
- Sindico, A., Di Natale, M. and Panci, G. (2011), “Integrating SysML with Simulink using Open-source Model Transformations”, *Proceedings of the 1st International Conference on Simulation and Modeling Methodologies, Technologies and Applications (SIMULTECH 2011), Noordwijkerhout, The Netherlands, July 29-21, 2011*, SciTePress, pp. 45-56. <https://doi.org/10.5220/0003593600450056>
- 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
- V-REP (2017), *Virtual robot experimentation platform*. [online] Coppelia Robotics, Zürich, Switzerland. Available at: <http://www.coppeliarobotics.com/> (accessed 17.11.2017).
- Weber, C. (2007), “Looking at 'DFX' and 'Product Maturity' from the Perspective of a New Approach to Modelling Product and Product Development Processes”, In: Krause, F.-L. (Ed.), *The Future of Product Development*, Springer, Berlin, Heidelberg, pp. 85-104. https://doi.org/10.1007/978-3-540-69820-3_11
- 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.

Atif Mahboob, Research assistant
 Technische Universität Ilmenau, Department of Mechanical Engineering
 Max-Planck-Ring 12 (Haus F), D-98693 Ilmenau, Germany
 Email: atif.mahboob@tu-ilmenau.de