VR-SysML: SysML Model Visualization and Immersion in Virtual Reality

Roy Oberhauser[0000-0002-7606-8226]

Computer Science Dept.

Aalen University

Aalen, Germany

e-mail: roy.oberhauser@hs-aalen.de

***Abstract* - As systems grow in complexity, the interdisciplinary nature of systems engineering makes the visualization and comprehension of the underlying system models challenging for the various stakeholders. This, in turn, can affect validation and realization correctness. Furthermore, stakeholder collaboration is often hindered due to the lack of a common medium to access and convey these models, which are often partitioned across multiple 2D diagrams. This paper contributes VR-SysML, a solution concept for visualizing and interacting with SysML models in Virtual Reality (VR). Our prototype realization shows its feasibility, and our evaluation results based on a case study shows its support for the various SysML diagram types in VR, cross-diagram element recognition via our backplane followers concept, and depicting further related (SysML and non-SysML) models side-by-side in VR.**

Keywords - Systems Modeling Language (SysML); virtual reality; systems modeling; systems engineering.

I. Introduction

Systems engineering (SysE) is an interdisciplinary collaborative engineering field dealing with the design, integration, and management of complex system solutions over their lifecycle. The field faces a continuous challenge of growing system complexity, an increasing share of functionality shifted to software, system resource constraints, while coping with compressed development timeframes and project budget and resource constraints. Furthermore, the interdisciplinary nature of SysE means that diverse stakeholder types and groups with their specialty competencies and concerns are involved and who may not be readily acquainted with the model types and modeling languages involved. Any models may be digitally isolated or practically inaccessible to all stakeholder types, "hidden" within "cryptic" modeling tools that certain modeling specialists may understand. Due to the interdisciplinary nature of SysE, the inaccessibility and lack of model comprehension can hamper collaboration and affect overall system validity and correctness with regard to requirements.

While SysE can involve various models including physical, mechanical, electrical, thermodynamic, and electronic, the focus of this paper is on the Systems Modeling Language (SysML®) [1]. SysML is a dialect of the Unified Modeling Language (UML®) and defined as a UML 2 Profile. Views and their associated diagrams can help reduce cognitive overload, yet their divided nature also risks overlooking a relation or element and comprehending the overall model. Ideally, a model should be whole and complete to the appropriate degree for the reality it is depicting and simplifying. Yet the modeling languages and associated tooling typically assumes a 2D display and portrays portions of models sliced onto 2D diagrams. Although 3D models can be portrayed on 2D displays, they lack an immersion quality.

VR is a mediated visual environment which is created and then experienced as telepresence by the perceiver. VR provides an unlimited immersive space for visualizing and analyzing a growing and complex set of system models and their interrelationships simultaneously in a 3D spatial structure viewable from different perspectives. Lacking a proper 3D system modeling notation, in the interim we propose retaining the well-known SysML notation and interconnecting 2D SysML diagrams in VR, which can suffice for depicting the relations between elements across diagrams and assist with navigating and validating complex models. As system models grow in complexity and reflect the deeper integration and portrayal of their system reality and environment, an immersive digital environment provides an additional visualization capability to comprehend the “big picture” model for structurally and hierarchically complex system models via interconnected diagrams and associated digital elements.

As to our prior work, VR-UML [2] provides VR-based visualization of UML diagrams. VR-BPMN [3] visualizes BPMN-based business process models in VR, while VR-EA [4] visualizes Enterprise Architecture (EA) ArchiMate® models in VR. This paper contributes VR-SysML, a solution concept for visualizing and interacting with SysML results in VR. Our prototype realization shows its feasibility, and a case-based evaluation provides insights into its capabilities.

The remainder of this paper is structured as follows: Section 2 discusses related work. In Section 3, the solution concept is described. Section 4 provides details about the realization. The evaluation is described in Section 5 and is followed by a conclusion.

II. Related Work

As to visualization approaches with SysML, Nigischer and Gerhard [5] proposed a lightweight 3D visualization for SysML models in Product Data Management. They describe an approach and concept, but no prototype is shown. Barosan et al. [6] describes a 3D SysML digital-twin-in-loop virtual simulation environment of a distribution center for truck driving test scenarios integrating IBM Rhapsody with Unity3D; VR and immersion are not considered. Mahboob et al. [7] describe a model-based approach to generate VR object collision simulation scenes from SysML behavior models.

Besides our own VR-UML [2], VR features are not yet commonplace in UML tools: Ozkaya [8] analyzed 58 different UML tools without any mention of VR, and Ozkaya and Erata [9] surveyed 109 practitioners to determine their UML preferences without any mention of VR.

In contrast, VR-SysML provides a VR-centric visualization and immersive experience with SysML models, providing automatic layout of views as stacked 3D hyperplanes, visualizing the reality of inter-view relations and recurrence of elements, and enabling interactive modeling in VR. Hypermodeling enables SysML, UML, and other models to be simultaneously visualized in the same virtual space, supporting cross-model analysis across various diagram types and stakeholder concerns.

III. Solution Concept

Our solution concept is based on VR. In support of our view that an immersive VR experience can be beneficial for model analysis, Müller et al. [10] compared VR vs. 2D for a software analysis task, finding that VR does not significantly decrease comprehension and analysis time nor significantly improve correctness (although fewer errors were made). While interaction time was less efficient, VR improved the user experience, was more motivating, less demanding, more inventive/innovative, and more clearly structured.

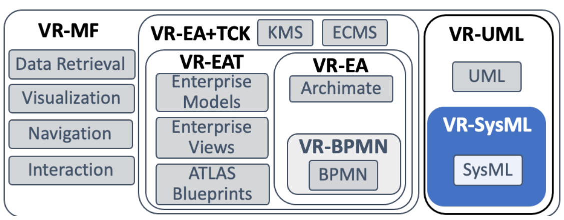

Figure 1. Conceptual map of our various VR solution concepts.

SysML is a general-purpose architecture modeling language for systems and systems-of-systems, supporting their specification, analysis, design, verification, and validation. Out of UML 2's diagrams, it reuses seven (modifying four of these) while adding two additional ones. Thus, for VR-SysML (Figure 1) we chose to extend our VR-UML [2] solution concept, which is based on our generalized VR Modeling Framework (VR-MF) (detailed in [4]). VR-MF provides a VR-based domain-independent hypermodeling framework addressing four aspects requiring special attention when modeling in VR: visualization, navigation, interaction, and data retrieval. Our other VR modeling solutions include VR-BPMN [3], VR-EA [4], and VR-EAT, which integrates the EA tool Atlas to provide dynamically-generated EA diagrams in VR. VR-EA+TCK adds additional capabilities, integrating enterprise Tool, Content, and Knowledge such as a Knowledge Management Systems (KMS) and/or Enterprise Content Management Systems (ECMS). While SysML is popular for embedded and model-based systems, it is also applicable to domains such as EA.

A. *Visualization in VR*

Our concept attempts to leverage the best of 2D and VR: to support diagram comprehension, we chose not to diverge significantly from the SysML notation. Yet placing 2D SysML images like flat screens in front of users would provide little added value in the 3D VR space. A plane is used to intuitively represent a diagram. Stacked hyperplanes are used to support viewing multiple diagrams at once, while permitting a user to readily have an overview of the number and types of diagrams. Furthermore, hyperplanes serve a grouping function and allow us to utilize the concept of a common transparent or invisible backplane to indicate common elements across diagrams via multi-colored inter-diagram followers. Versus side-by-side, stacked diagrams are a scalable approach for larger projects since the distance to the VR camera is shorter. Multiple stacks can be used to group diagrams or delineate heterogeneous models. Diagrams of interest can still be viewed side-by-side by moving them from the stack via an anchor sphere affordance on a diagram corner, which is also used to hide or collapse diagrams to reduce visual clutter. To distinguish SysML elements types, 2D icon images can be placed on generic (e.g., block) model elements, in order to reduce the effort of modeling each SysML element type as a separate 3D form for VR.

B. *Navigation in VR*

One navigation challenge arising from the immersion VR offers is supporting intuitive spatial navigation while reducing potential VR sickness symptoms. Thus, we incorporate two navigation modes in our solution concept: the default uses gliding controls for fly-through VR, while teleporting instantly places the camera at a selected position. Although potentially disconcerting, it may reduce the likelihood of VR sickness induced by fly-through for those prone to it.

C. *Interaction in VR*

As VR interaction has not yet become standardized, in our concept user-element interaction is supported primarily through VR controllers and a *Virtual Tablet*. The Virtual Tablet provides detailed element information with context-specific Create, Retrieve, Update, Delete (CRUD) capabilities including a *virtual keyboard* for text entry via laser pointer key selection. The aforementioned corner *anchor sphere* affordance supports moving / hiding / displaying diagrams. Inter-diagram element *followers* can be displayed, hidden, or selected (emphasized).

IV. Realization

The logical architecture for our VR-SysML prototype realization is shown in Figure 2.

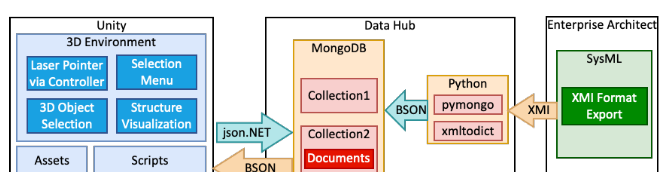

Figure 2. VR-SysML logical architecture.

SysML models are imported in XMI format to our Data Hub that is implemented in Python. Xmitodict is used to convert the XMI to a key-value dictionary and the built-in json package is used for JSON conversion. Pymongo is used to store the JSON (as BSON) in the NoSQL document database MongoDB. The scripts in the Unity environment utilize json.NET. SysML XMI files produced from SparxSystems Enterprise Architect were used. Our prototype currently does not consider the Allocation Table (relationship matrices).

V. Evaluation

We base the evaluation of our solution concept on design science method and principles [11], in particular, a viable artifact, problem relevance, and design evaluation (utility, quality, efficacy). A case study is used with an emphasis on SysML diagram type support, how these are visualized in VR, and additional capabilities in VR. A sample SysML project with all 9 SysML diagram types is used to compare the visualization in Enterprise Architect to that in VR-SysML, grouped as requirement, behavior, or structure diagram types.

As shown in Figure 3, the various diagrams of the SysML model are mapped to stacked hyperplanes that provide an anchor affordance (black sphere) with which to expand, collapse, or move a diagram. Planes and elements have a shallow 3D depth with labeled edges to support recognition from different viewing angles. The colors of the planes can be configured to help with differentiation or grouping. Furthermore, our backplane concept creates followers that allow one to quickly find the same element across different diagrams in the same model, to readily see in which diagrams that element participates, or to determine that the element is only shown on one diagram (it not having a follower). The colored followers can be selected (made bold) and the other followers can be hidden if desired to reduce visual clutter for larger models.

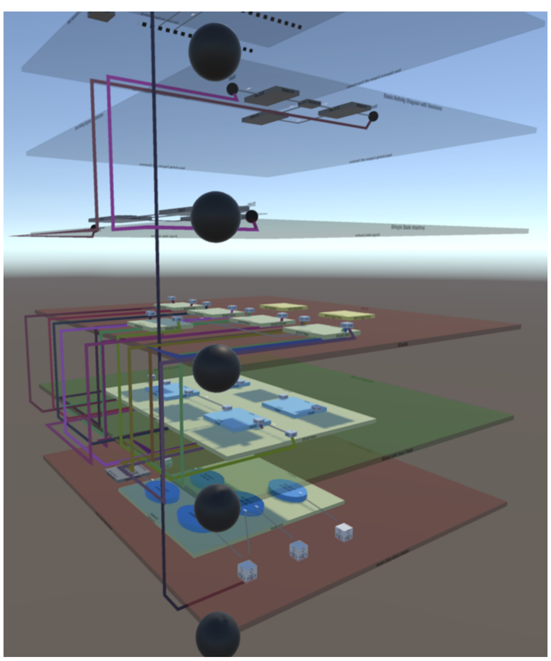

Figure 3. Backplane with inter-diagram followers.

A. SysML Requirement Diagram

SysML extends UML with an additional diagram type, the Requirement diagram. It can be used to specify functional and non-functional requirements for the model. An example viewed in EA and in VR is shown in Figure 4. In VR, elements are labeled on edges to support reading from different angles. The VR Tablet can provide more details or interaction capabilities for a selected element, and while support for modeling capabilities is shown on the interface, these are currently placeholders and have not yet been fully implemented in the prototype (create, modify, delete, export).

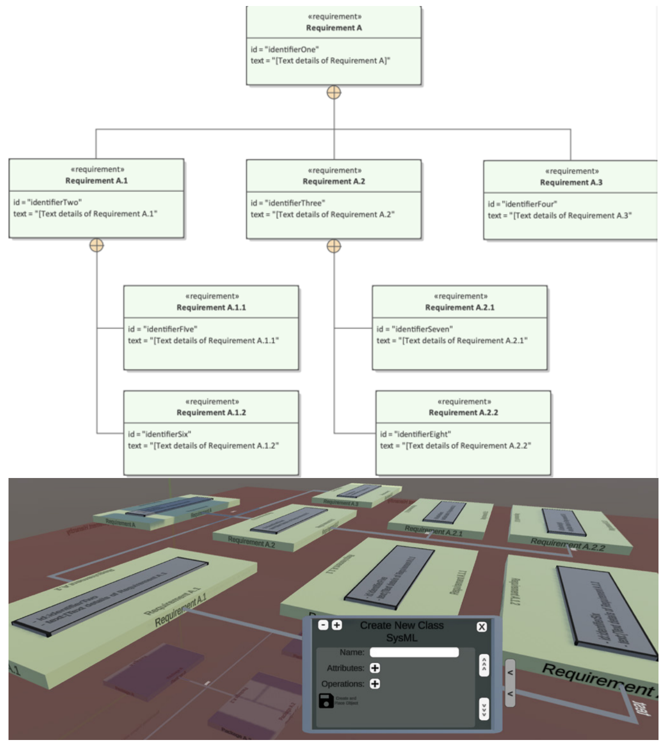

Figure 4. Requirement Diagram in EA (top) and in VR (bottom).

B. SysML Use Case Diagram

As a behavior diagram, SysML includes the Use Case Diagram from UML as shown in Figure 5. In order to more readily recognize and differentiate the diagram type, an oval shape was used for the use cases. However, the actors utilize our generic cube concept with notation symbols placed on the various sides. This provides a flexible mechanism for quickly supporting various notation element types and tailoring or extending model element types using any icons or images.

C. SysML Activity Diagram

Another dynamic behavior diagram type that can be used to specify dynamic system behaviors, such as control flow and object flows, is the Activity diagram in SysML (see Figure 6). It is slightly modified from that in UML, adding additional semantics for Continuous Flow and Probability.

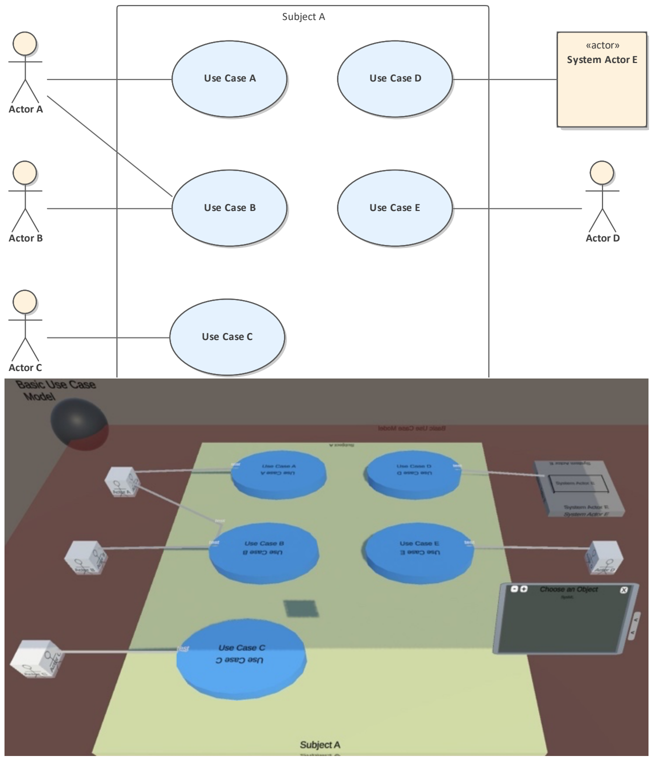

Figure 5. Use Case Diagram in EA (top) and in VR (bottom).

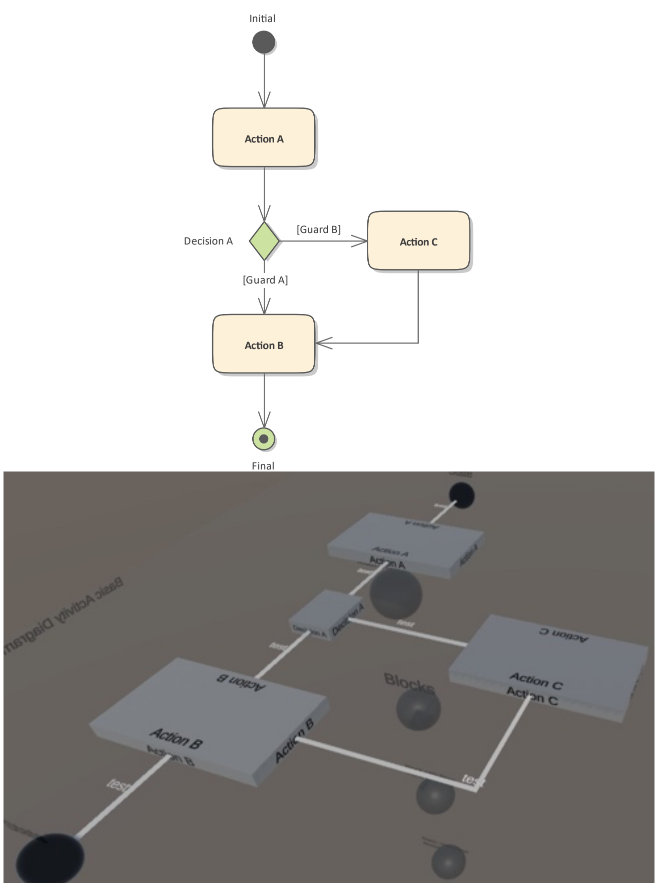

Figure 6. Activity Diagram in EA (top) and in VR (bottom).

D. SysML Sequence Diagram

Sequence diagrams (unmodified from UML) provide a further dynamic behavior diagram, showing interactions via message sequences (see Figure 7).

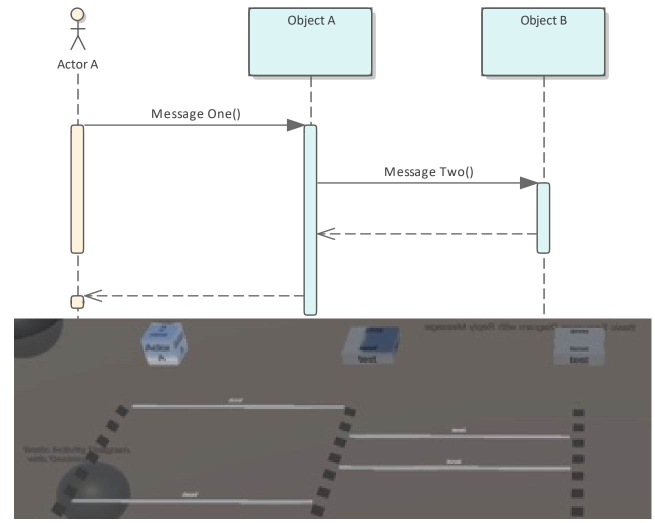

Figure 7. Sequence Diagram in EA (top) and in VR (bottom).

E. SysML State Machine Diagram

State machine diagrams (unmodified from UML) are a dynamic behavior diagram showing states transitions that occur in response to events (see Figure 8).

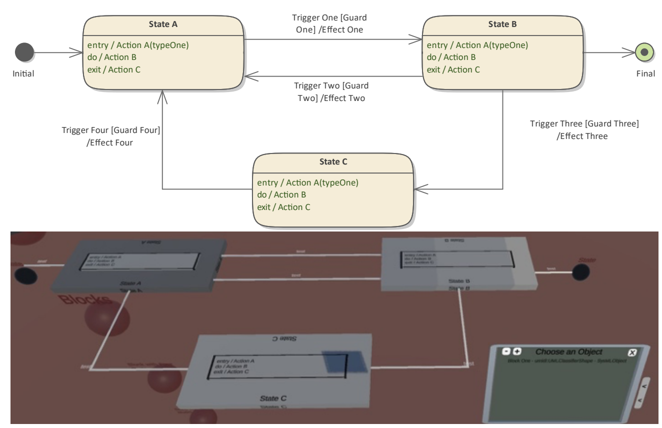

Figure 8. State Machine Diagram in EA (top) and in VR (bottom).

F. SysML Block Definition Diagram (BDD)

A BDD (see Figure 9) is a static structural diagram, analogous to the UML Class diagram type with certain modifications, and shows system components, their contents (as properties, behaviors, constraints), interfaces, and relationships. It can be used for describing the system structure as a hierarchy of relations between systems and subsystems typically consisting of “black-box” blocks. As a possible specialization, it can be useful to explicitly model constraints separately, referred to as Constraint Block diagrams (see Figure 10), which can be referenced by Parametric diagrams.

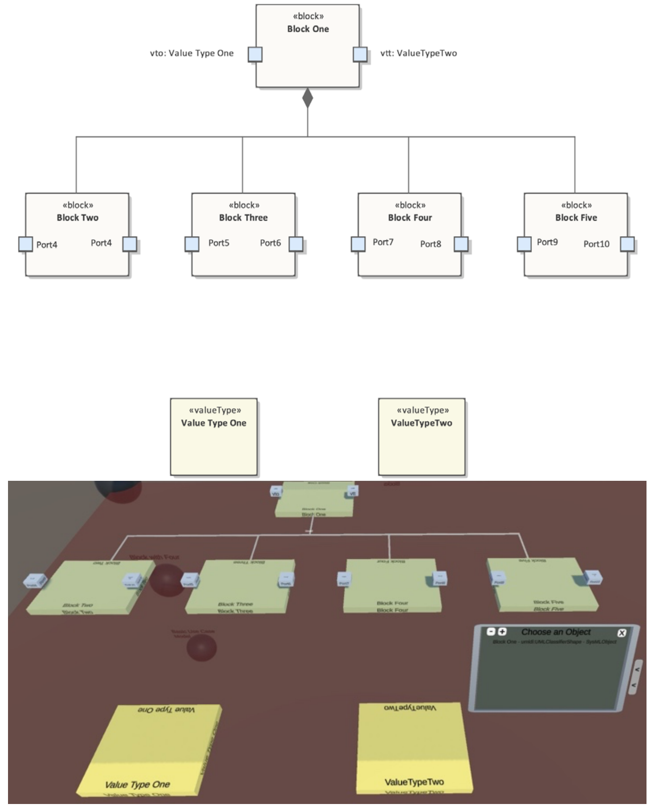

Figure 9. BDD in EA (top) and in VR (bottom).

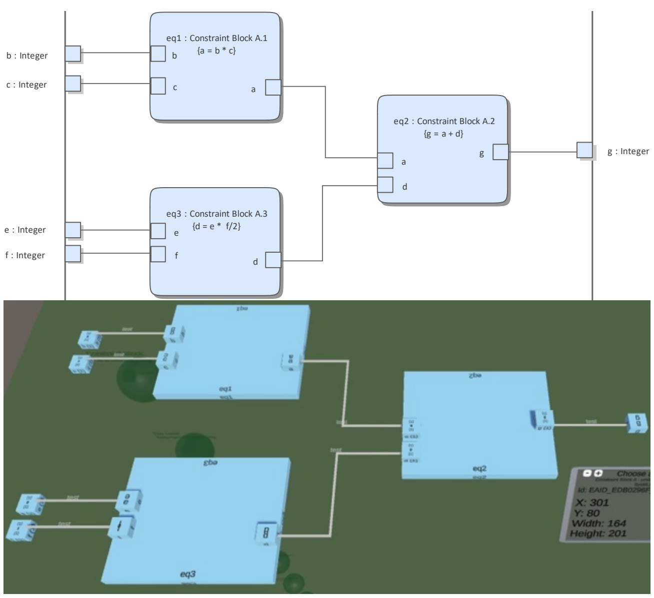

Figure 10. Constraint Block Diagram in EA (top) and in VR (bottom).

G. SysML Internal Block Diagram (IBD)

An IBD is a static structural diagram that depicts the internal (encapsulated) composition (structural contents) of a Block in a BDD, i.e., a "white-box" view. This includes properties, parts, interfaces, connectors, and ports, and can be used to depict the flow of inputs and outputs between them.

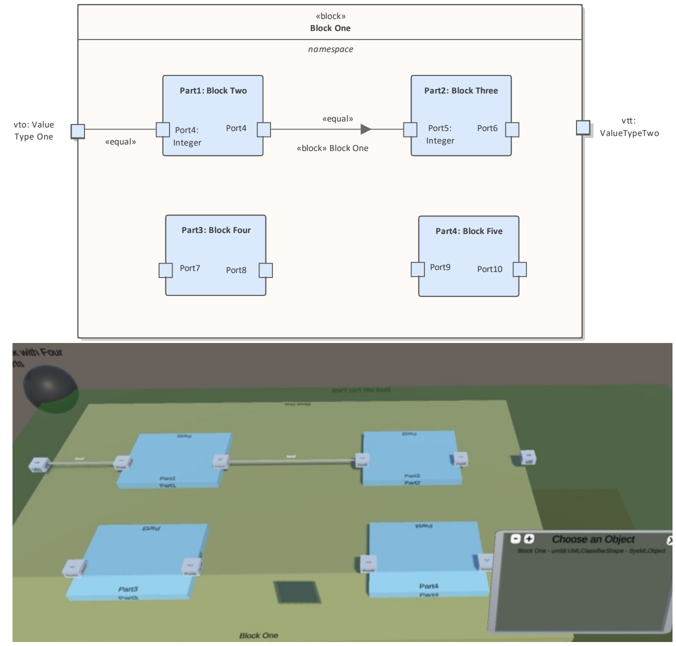

Figure 11. IBD in EA (top) and in VR (bottom).

H. SysML Parametric Diagram

A static structural diagram type, Parametric diagrams (see Figure 12) are a specialization of IBD to model equations with parameters and can be used to enforce mathematical rules or constraints defined via Constraint Blocks.

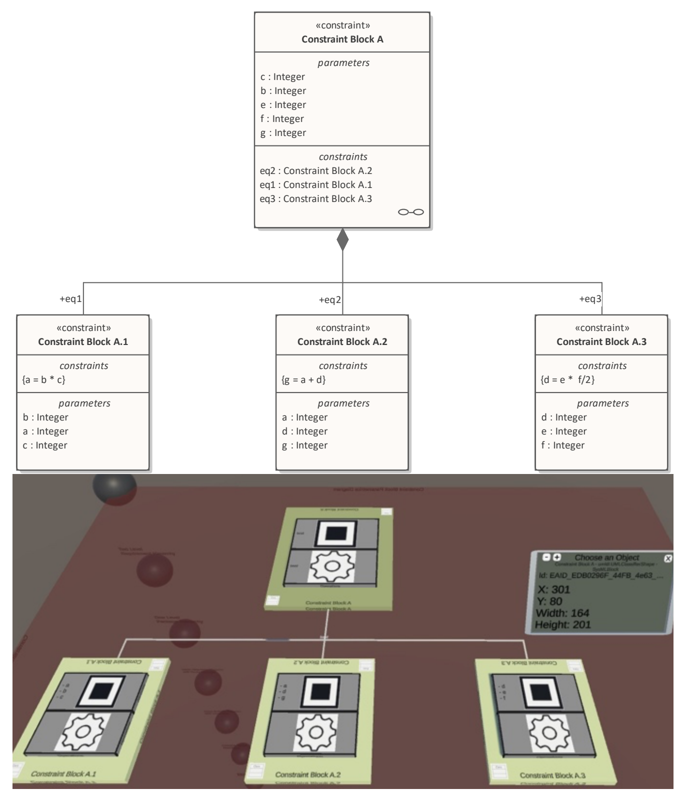

Figure 12. Constraint Parametric Diagram in EA (top) and in VR (bottom).

I. *SysML Package Diagram*

A SysML Package diagram (see Figure 13) is further static structural diagram based on the equivalent UML type (with minor modifications). Packages provide a general-purpose mechanism for grouping model elements and diagrams, and the diagram can be used to show their contents and the relationship between them.

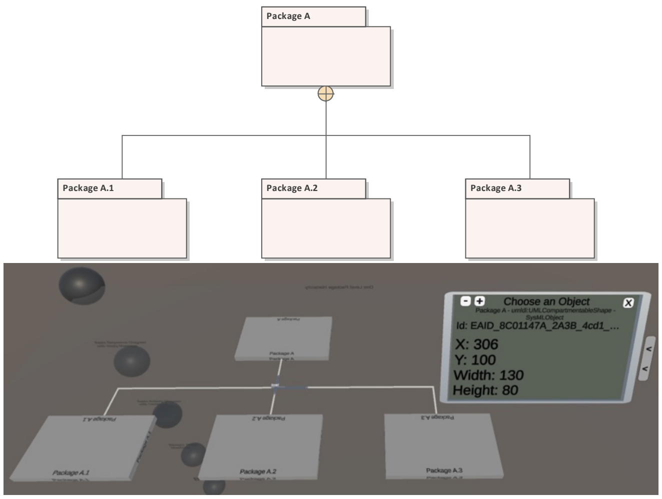

Figure 13. Package Diagram in EA (top) and in VR (bottom).

J. *Multi- and Heterogeneous Model Depiction in VR*

VR's unlimited virtual space provides the potential to view, compare, and analyze multiple SysML (left and center models in Figure 14) or heterogeneous models side-by-side (exemplified with an ArchiMate enterprise architecture model on the right in Figure 14). For SysE, this immersive approach also has the potential to support interdisciplinary collaboration between specialization experts for complex systems.

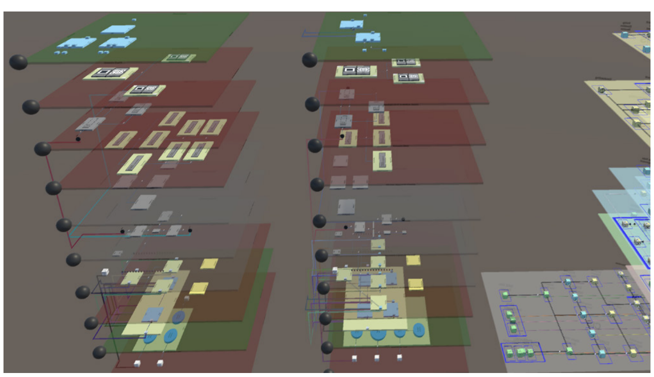

Figure 14. Multiple and heterogeneous side-by-side models in VR.

VI. Conclusion

VR-SysML contributes an immersive SysML model experience for visually depicting and navigating SysML diagrams of models in VR. The solution concept was described and a VR prototype demonstrated its feasibility using a case study. Based on our VR hyperplane principle, SysML diagrams are enhanced with 3D depth, color, and automatically-generated inter-diagram element followers based on our back-plane concept. Interaction is supported via a virtual tablet and keyboard. The unlimited space in VR facilitates the depiction and visual navigation of large models, while relations within and between elements, diagrams, and models can be analyzed. Furthermore, in VR additional related (SysML or non-SysML) models can be visualized and analyzed simultaneously and benefit complex systems-of-systems architectures or collaboration. The sensory immersion of VR can support task focus during model comprehension and increase modeling enjoyment, while limiting the visual distractions that typical 2D display surroundings incur. Future work includes support for modeling directly in VR, integrating further SysML tooling and simulation capabilities, supporting model verification and validation, and conducting a comprehensive empirical study.

Acknowledgment

The authors would like to thank Shadrach Arulrajah and Marie Bähre for their assistance with the design, implementation, evaluation, and diagrams.

References

[1] OMG, "OMG Systems Modeling Language Version 1.6", Object Management Group, 2019.

[2] R. Oberhauser, R., "VR-UML: The unified modeling language in virtual reality – an immersive modeling experience," International Symposium on Business Modeling and Software Design (BMSD 2021), Springer, Cham, 2021, pp. 40-58.

[3] R. Oberhauser, C. Pogolski, and A. Matic, "VR-BPMN: Visualizing BPMN models in Virtual Reality," In: Shishkov, B. (ed.) BMSD 2018. LNBIP, vol. 319, Springer, Cham, 2018, pp. 83–97. https://doi.org/10.1007/978-3-319-94214-8_6

[4] R. Oberhauser and C. Pogolski, "VR-EA: Virtual Reality Visualization of Enterprise Architecture Models with ArchiMate and BPMN," In: Shishkov, B. (ed.) BMSD 2019. LNBIP, vol. 356, Springer, Cham, 2019, pp. 170–187.

[5] C. Nigischer and D. Gerhard, "Lightweight visualization of SysML models in PDM systems," DS 87-3 Proc.eedings of the 21st International Conf. on Engineering Design (ICED 17) Vol 3: Product, Services and Systems Design, 2017, pp. 61-70.

[6] I. Barosan, A. Basmenj, S. Chouhan, D. Manrique, "Development of a Virtual Simulation Environment and a Digital Twin of an Autonomous Driving Truck for a Distribution Center," Software Architecture (ECSA 2020). CCIS, vol 1269, Springer, Cham, 2020, pp. 542–557.

[7] A. Mahboob, S. Husung, C. Weber, A. Liebal, and H. Krömker, "SYSML behaviour models for description of Virtual Reality environments for early evaluation of a product," In DS 92: Proc. 15th Int'l Design Conf. (DESIGN), 2018, pp. 2903-2912.

[8] M. Ozkaya, "Are the UML modelling tools powerful enough for practitioners? A literature review," IET Software, vol. 13, 2019, pp. 338-354. https://doi.org/10.1049/iet-sen.2018.5409

[9] M. Ozkaya and F. Erata, "A survey on the practical use of UML for different software architecture viewpoints," Information and Software Technology, Vol. 121, 106275, 2020.

[10] R. Müller, P. Kovacs, J. Schilbach, and D. Zeckzer, "How to master challenges in experimental evaluation of 2D versus 3D software visualizations," In: 2014 IEEE VIS International Workshop on 3Dvis (3Dvis), IEEE, 2014, pp. 33-36

[11] A.R. Hevner, S.T. March, J. Park, and S. Ram, "Design science in information systems research," MIS Quarterly, 28(1), 2004, pp. 75-105