INTERNATIONAL DESIGN CONFERENCE – DESIGN 2022
https://doi.org/10.1017/pds.2022.184

SysML 4 Digital Twins – Utilization of System Models for the Design and Operation of Digital Twins

F. Wilking ✉, C. Sauer, B. Schleich and S. Wartzack
Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
✉ wilking@mfk.fau.de

Abstract

The implementation of Digital Twins has become a common task for many industrial companies to ensure a sufficient digitization of their products and maintain competitiveness. This results in the question of how to compensate additional effort caused by designing Digital Twins. With this paper, an approach for this compensation is presented by creating Digital Twin behaviour through utilizing SysML diagrams and directly derivate usable code from them for a further implementation. This offers a part solution of lowering the threshold for using MBSE and increasing its benefits.

Keywords: *model-based systems engineering (MBSE), cyber-physical systems, systems engineering (SE), digital design, complex systems*

1. Introduction

Enabling an enhancement of product benefit and adding further value to products is accompanied by an increase of complexity (Sheard et al., 2015) through interdependency and adaptability, which is shown in the ongoing digitization of products, e. g. Digital Twins. This complexity is not only reflected in a higher number of elements included in a system but mainly in a higher degree of interdependencies between elements and subsystems. Hence, this confronts many enterprises, formerly focused on non-digital products, with a variety of different challenges to grasp the increasing complexity of their products and ensure a competitiveness within their market. The approach of Model Based Systems Engineering (MBSE) attained popularity among various industry sectors within the last years as an approach to enable the application Systems Engineering. While Systems Engineering was initially created to be applied within industry sectors shaped by tremendous complexity, e. g. aerospace, these new industry sectors are faced with a situation in which the approach needs to be heavily tailored and adapted towards their specific constraints. In addition, it must be considered that an MBSE approach is not limited to change processes within the enterprise of an OEM but also can be enlarged towards design efforts of system and component suppliers. While it is already very difficult to measure key performance indicators or even an ROI for Systems Engineering (Eigner et al., 2015; Honour, 2013). Especially, these other enterprises along the supply chain, affected by the introduction of MBSE, might be in a situation in which they have not yet reached a degree of system complexity that justifies the integration of a whole MBSE approach. However, the effort for MBSE activities is imposed on them by their customers, e. g. through a lower negotiation power. This is described as extrinsic and intrinsic motivations for the application of MBSE (Wilking et al., 2020). An intrinsic motivation would imply the aim to grasp complexity and ensure improved systems through its application. An extrinsic motivation forces the enterprise to apply MBSE due to imposition by a customer or through legal constraints. These enterprises along the value chain, which have an extrinsic motivation for MBSE, are often left out of scope in existing approaches. But especially they are confronted with the question on how to compensate the integration of MBSE or how to benefit from it. Limited resources for small and medium-sized enterprises aggravate this situation. Therefore, it is crucial to find solutions for compensating MBSE efforts or increase the overall benefit of its application, especially in new industry sectors. A common situation for suppliers involves a demand of their customers to deliver additional system models, e.g. through System Modelling Language (SysML) diagrams. These diagrams support the interdisciplinary communication between stakeholders and document the structure and behaviour of a system. Suppliers need to develop competencies in creating such models but rarely benefit from it, as their subsystem or component complexity does not justify the additional effort. Another challenge is the increasing digitization of products. Many conventional and mechanical-oriented enterprises are facing a situation in which interdisciplinary competencies are required. Hence, this led to a further increase of required effort for maintaining the competitiveness. The creation of Digital Twins, a recent development within the digitization of products (Gartner, 2018), has a natural link towards the integration and usage of MBSE by using models. This technology of Digital Twins seems to be promising by helping conventional enterprises taking the next step to become system suppliers. This design of Digital Twins confronts them with several challenges regarding design approaches and handling additional required efforts to implement Digital Twins. Therefore, within this paper, the central research question is to identify solutions for the compensation of effort through the utilization of SysML for the creation of Digital Twins. The potential of machine-recognizable SysML-Models are analysed and evaluated within the context of the Digital Twin Design. The concept is shown in a demonstrator software linking SysML diagrams with a preliminary Digital Twin concept to discuss whether a compensation is given.

2. Related Work

2.1. Digital Twins & MBSE for Digital Twin Design

The concept of Digital Twins originates from NASA's Apollo Program. Two identical space vehicles were used to mirror similar conditions during flights and enable more precise interpretation of data and thus recommendations (Rosen et al., 2015). However, this concept was developed further and took advantages of modern developments, such as the Internet of Things (IoT) to enable data streaming and models that are more sophisticated. Nevertheless, the discussion about a precise definition of Digital Twins is still not yet finished. In 2020 the German Scientific Society of Product Development (WiGeP), in which multiple German university professors present a common understanding of Digital Twins, proposed a generic definition (Stark et al., 2020). This definition includes several important aspects, e. g. the necessity of an existing physical counterpart (Schleich et al., 2019; Schleich et al., 2017; Hribernik et al., 2013; Schroeder et al., 2016), bidirectional orientations (Schleich et al., 2017) and real time data capabilities (Lee et al., 2013; Tao et al., 2019), and is therefore feasible as a basic definition. A derived definition is visualized in Figure 1 and described in the definition below.

In this definition a Digital Twin is described as the connection of an existing, physical object with a virtual counterpart. This counterpart contains several models to describe the physical object. Generally, these models were created during the product development, e. g. through system models, simulation models, mathematical models, etc. They are adapted within the development phase of the life cycle and form the virtual prototypes of the product. Depending on various constraints, e. g. calculation speed, models must be adapted towards the specific concept of the Digital Twin. With these models and real time as well as historic product data, received by sensors within the physical product, the Digital Twin analyses the current state and can make predictions about future behaviour. This analysis enables the creation of direct recommendations towards the user, e. g. through service notification, or directly towards the physical product, e. g. through optimization. This control circle enables the Digital Twin to adapt and enhance its own models by constantly increasing the maturity through current data. The optimal usage of the concept is achieved by the connection of multiple Digital Twins, an Aggregated Digital Twin. This connection allows the holistic view on the system. (Stark et al., 2020)

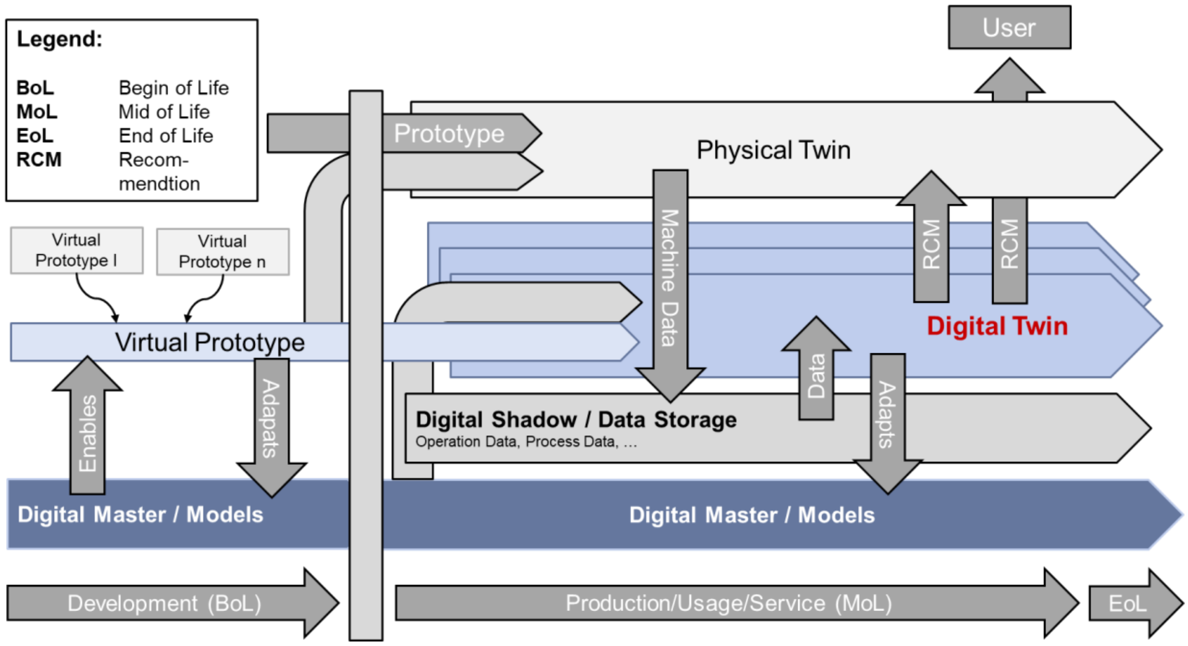

Figure 1. Derived Definition of Digital Twins

With this definition, a further classification of Digital Twins is reasonable to ensure the cover of several different use cases for it (Wilking et al., 2021). While some Digital Twins are only used for delivering detailed information, others are supporting the user's decision-making. Therefore, three different classes of Digital Twins are defined: Informational Digital Twin (IDT), Supporting Digital Twin (SDT) and Autonomous Digital Twin (ADT). They further vary in model quality, real time capability and their integration depth as component, subsystem or system shown in Figure 2.

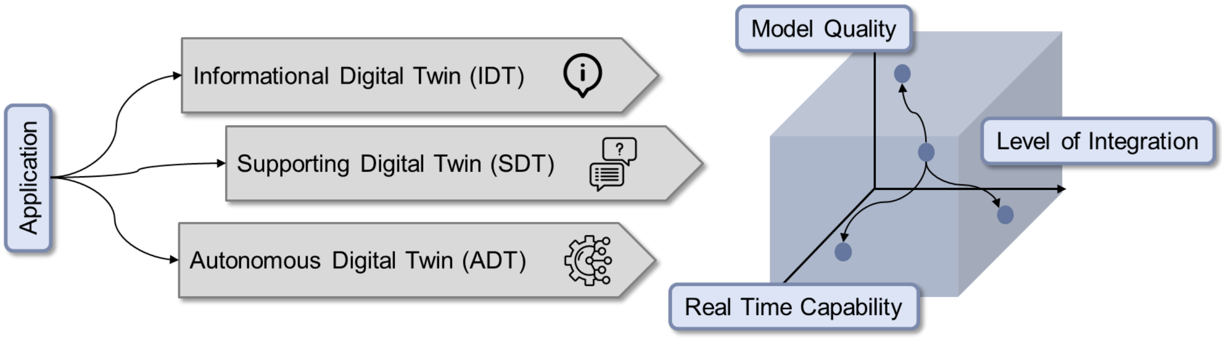

Figure 2. Classification and Dimensions of Digital Twins acc. to Wilking et al., 2021

The model quality (or fidelity) ensures that the degree of details within the models is enabling the designated use case of the system, e.g. a high model quality can lead to a high computing time for simulations. If the results are needed immediately this could lead to conflicts. Furthermore, Digital Twins can be using real time data or are solely depending on stored data. In addition, the level of integration is taking an important role for defining interdependencies between several system components or even different Digital Twins.

2.2. SysML Usage & Utilization for Digital Twins

Systems models are used as images of systems focused on specific profiles and perspectives to ensure an efficient view on the system (Weilkiens, 2014). SysML, as a modelling language, enables a standardized expression of these system models. Hence, it ensures the view on a system from different

https://doi.org/10.1017/pds.2022.184 Published online by Cambridge University Press

perspectives and enhances an interdisciplinary and cross-stakeholder communication. This communication scenario defines the main use cases for these diagrams and therefore improves the comprehension of complex systems through the describing character of its notation. In fact, SysML, as an adaption of the software describing Unified Modelling Language (UML), can also be converted into an exchange data format, such as XML Metadata Interchange (XMI) or through specific tool interfaces that are integrated into the modelling software (Handley et al., 2021). This allows another use case for SysML diagrams, which the authors define as "technical utilization" of SysML models. For this utilization, the SysML diagrams are further used in a machine-recognizable context. However, this utilization of SysML diagrams is still discussed in research and first approaches are shown, e. g. in automated simulations (Kapos et al., 2014), validation of system design (Karban et al., 2016), hybrid modelling approaches (Schumacher and Inkermann, 2021) or linking Virtual Reality environments (Mahboob et al., 2018) to it. In context of the Digital Twin Design, SysML has taken a significant role to describe a system and grasp its complexity, e. g. through graphical visualisation of its architecture and behaviour and preliminary simulations (Makarov et al., 2019; Wang et al., 2021). Indeed, models are a vital component of digital twins during their operation phase as shown in Figure 1. However, further utilization of SysML directly for the design and operation of Digital Twins has not yet been identified. Only first attempts of model automation for Digital Twins were made outside of the SysML, e. g. using the Automation Markup Language by Bao et al. (2019)

3. Utilization of SysML for Designing and Operating Digital Twins

3.1. Capabilities & Potentials

The different classes of Digital Twins require different approaches for their design and implementation. But the capability of current SysML diagrams is limited and must be further discussed. Exporting SysML diagrams by using XMI enables the export of the diagram structure as well as information within the diagram. This ensures that a diagram can be exported from its visual form, used for communication, towards a machine-recognizable data format. However, these capabilities are limited, as XMI just reflects the diagram by its hierarchical structure. This requires further interpretation of the XMI-file but ensures the universal utilization of all SysML-diagrams. Nevertheless, SysML offers the great potential of utilizing its multi-perspective view on the system towards an exported structure which can be further used in either giving structured information about a system through a diagram, set requirements or describing a behaviour, e. g. through an auto-generated JavaScript-Code which can be directly used within other code.

3.2. Requirements for utilizing SysML Models

During the design of the demonstrator several requirements for the machine-recognizable modelling of SysML diagrams were revealed. Being mainly used for improving interdisciplinary and cross-stakeholder communication, SysML diagrams do not require a completely consistent syntax, as the viewer might correctly interpret minor deviations. Humans might be aware of tacit information within elements or processes within the behavioural structure. However, to ensure a machine-recognizable model this consistency of the syntax is crucial. In general, an approach for utilizing SysML models therefore requires the correct implementation of such diagrams. The information as well as the information flow must be complete and consistent throughout the whole model or the diagrams. In fact, depending on the current modelling approaches, this could lead to higher efforts for creating such models and modellers must be aware of each connection, its direction and impact within a machine-readable context and not only for communication purposes. While establishing the required precision and accuracy within one enterprise will already be a challenge, an enlargement towards a supply chain with multiple system (model) suppliers will lead to significant management efforts to prevent several iterations. In addition, as a utilization approach for SysML diagrams is highly goal oriented, the definition of such a goal is vital for a correct implementation. It must be precisely defined, what these models are used for and how to set up an adequate model architecture. This leads to three main requirements for modelling and utilizing SysML diagrams within this context:

- Consistency: Models must be created without any space for human interpretation, as it is often the case with SysML diagrams. Connections and interdependencies must be modelled using the specific tools to enable a clear XMI interpretation or code clean code-generation.
- Common Modelling Strategy: To lower later iterations, a common model strategy must be established to ensure that models are created in the same manner throughout the enterprises as well as the supply chain.
- Goal Orientation: Diagrams for communication and diagrams for machine-recognition must be separated. While one can help to understand or specify the other, their goals are highly different. A diagram goal must be set initially, and the two purposes should not be combined in a single diagram.

3.3. Effort Compensation through Utilization of SysML Models

With the mentioned requirements for utilizing SysML models, a quantification of possible effort compensation is difficult. Indeed, in theory, a compensation is possible and a further use of the models can be enabled throughout the life cycle. This requires that the benefit or compensation exceeds the additional effort that must be invested. Especially in scenarios where system models are used anyway, it seems likely that a minor additional effort can lead to a significantly higher benefit of these as the initial modelling effort is already done and an adaption towards a format that can be utilized and is mainly using the already existing models. The vital components of Digital Twins are models. Hence, the connection with an approach like MBSE therefore seems reasonable and modelling the structure and behaviour of this intertwined physical and digital twin pair is the next logical step to grasp its complexity. Therefore, the hypothesis is postulated that SysML models can be further utilized to enable and improve the design of Digital Twins as well as being used during the usage phase of such a twin as a key model for information acquisition and behaviour definition, e. g. giving adequate supportive actions. For the design phase, they ensure an overview of different behaviours and how they are connected as well as the trigger mechanisms to achieve this state. For Digital Twins system models support the overview of its complex structure. These systems might consist of thousand different behaviours. If modelled correctly this behaviour can directly be transferred into a code framework, saving time and enabling a direct verification. Especially in context of complex system behaviour, this allows a sufficient overview of different states and prevents misunderstandings, which could not be achieved by simply viewing its code. This leads to two benefits by combining system modelling with the further utilization of the model. First, the overview and complexity grasping aspect leads to a higher quality of the products by preventing unnecessary iterations and secondly through directly transforming the models into usable components of the later Digital Twin.

4. Integration Example for SysML Utilization in a Digital Twin

4.1. Goal and Scenario

The goal of the demonstrator is to proof whether the SysML model can be utilized as an information database, e. g. for material properties, and for a flow of actions, so a central program of the Digital Twin does not need to be filled with several conditions and if-else-statements, being unclear and confusing in complex scenarios. The main challenge was, to check whether the consistent and accessible visualization of a SysML-diagram can be transformed into a machine-recognizable form to become an element of the Digital Twin and enable its operation as modelled. The required additional effort was observed and analysed whether the utilization of SysML-diagrams leads to a compensation of effort. For the scenario a two staged gear drive was used, which was equipped with sensors. The scenario was chosen to check, whether a transformation of communication models towards machine-recognizable models can be achieved and whether these models are suitable for operating such a Digital Twin. However, the demonstrator reflects a common approach of many enterprises, as they equip existing products with Digital Twin functionalities in the form of smart retrofits. The classification of the Digital Twin was limited to a Supporting Digital Twin, which gives advice based on the received sensor data and simulation results. This SDT has the goal to

https://doi.org/10.1017/pds.2022.184 Published online by Cambridge University Press

identify critical states of the gear by using the sensor data and feed the models with them. Based on the results a specific recommendation is given to the operator, e. g. to turn off the system. As no actuators or bidirectional connections to the gear are given, no autonomous actions are taken place in the scenario, but it is limited to guidance. The SysML diagrams were initially created to describe the structure and behaviour for the twin to support the communication during its design. To see how much effort must be invested to utilize them towards building a machine-recognizable flow of actions and structure, it was analysed how much they need to be adapted to ensure this data basement and whether this additional effort can be viewed as a reasonable compensation.

4.2. Conceptual Architecture and Digital Twin Design

For the proof of concept, the gear was equipped with sensors, measuring vibrations, temperature, and forces. The sensors are connected with a Raspberry Pi 4 and are streamed to a Computer via Microsoft Azure and saved into a csv-file. Component data, requirements and behaviour of the gear drive were modelled within a SysML model, using requirement diagrams, block-definition, internal blocks, and state machines. A central program was designed that reads the real time data of the gear drive and uses the exported JavaScript from the state machine diagram, as well as system information taken from XML-exports of the SysML model. Based on the predefined state machines the tool is able to interpret the data and take adequate actions, e.g. resuming back to idle, running FEM simulations, and send out recommendations to the user. A simplified overview is shown in Figure 3.

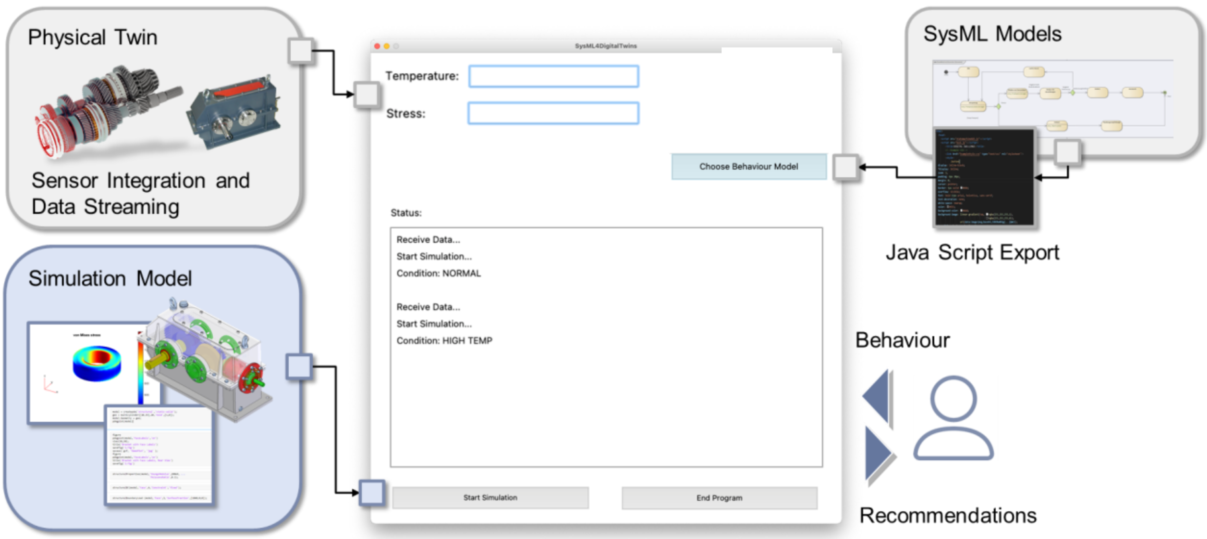

Figure 3. Overview of the created Demonstrator for a Supporting Digital Twin of a Gearbox

The behaviour was modelled within SysML using executable state machines and the JavaScript export functionality. Modelling these conditions and statements within SysML enabled a clear overview of the system behaviour and ensured that no conditions were excluded as well as the interdependencies accurately considered. The usage of SysML diagrams was an essential part during the design of the Digital Twin as they were used for discussions and communication between involved persons. The sequence of the Digital Twin system is represented in Figure 4. It shows that the system is set into a loop in which sensor data, read by the sensors and the Raspberry Pi, is streamed, using Microsoft Azure and Power BI towards a central user interface and program. In a next step the modelled system behaviour as well as information stored within the system model is used to check anomalies, based on conditions modelled within behavioural diagrams of the system model. If such is detected the Matlab FEM simulation is used to analyse the further impact. The given results are read by the central program as a Dashboard and, depending on the modelled behaviour, are transferred into specific actions, e. g. breaking the loop and turning off the system. These specific actions are shown as guidance on the UI for the operator and do not lead to autonomous actions initiated by the program.

https://doi.org/10.1017/pds.2022.184 Published online by Cambridge University Press

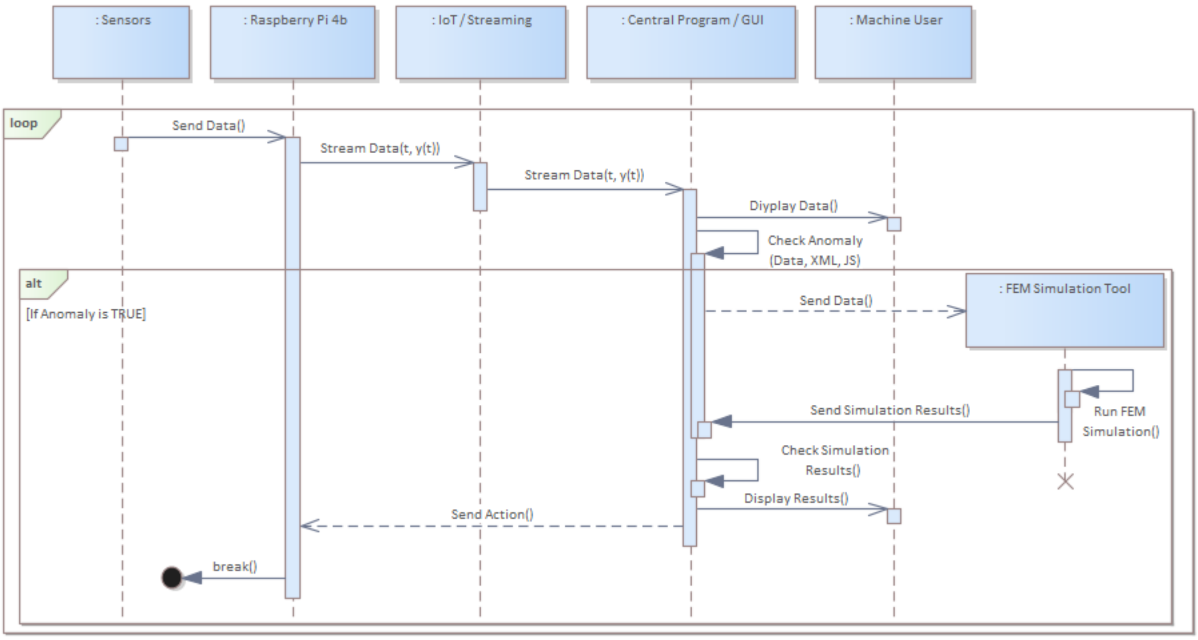

Figure 4. Sequence Diagram of the Digital Twin Behaviour for the Gearbox

The basic value structure of the gear was modelled in block diagrams to create a readable XMI-structure, e. g. the used material and properties of elements. Further usage of this data is planned in next generations of the demonstrator when a transformation from an SDT towards an ADT is implemented. It is possible to then involve even production data into the decision making by individualizing the physical product through the data integration in SysML diagrams. The creation of the whole system model was conducted to represent a realistic scenario for modelling systems, i. e. using diagrams for communicative purposes as well as creating machine-readable diagrams. The internal IoT Infrastructure was not modelled within SysML, future benefits of these models are discussed in chapter 5. Specific interpretations of the perceived data were left out to focus on the utilization of the system models and show the general feasibility of the concept. Therefore, neither realistic vibration data for a specific use case was used, nor the interpretation of the data is representing state of art in gear research.

4.3. Behaviour & Impact Analysis

In a first iteration, the behaviour and impact analysis was kept simple to analyse whether the system model can be utilized for this purpose. A simple activity flow of the system behaviour was created to discuss different possibilities of received data and how the system should react to them. The activity flow gives an overview of the principal sequence of activities and therefore describes the general behaviour of the Digital Twin system. This designed behaviour of the Digital Twin, shown in the activity flow, was transferred into different specific states defining the specific guards and signals to ensure identifiable behaviour within the later exported code. During the concept creation of the Digital Twin system, SysML prove itself as a helpful way to model the behaviour and grasp the variety of different behaviours, which becomes even more important for complex systems. The states and pseudo states of the state machine diagram are given in Figure 5. Turning the system on creates a loop within the operation mode to constantly check the perceived sensor data, received by the Raspberry and the sensors in the gearbox. If the vibration y(t) stays under the critical level the loop continues. In case the vibration surpasses the critical vibration, the system turns into the Error Mode 1 state giving the information to the user and stopping the system by displaying a recommendation on the dashboard. If the vibration comes close to the critical one, the FEM simulation state is activated, triggering a simulation. Depending on the results, the loop is continued or the second Error Mode state is activated, leading to a stop of the system and ending the state machine. The transitions of the states were connected with specific conditions and later transformed to an executable version of the state machine.

https://doi.org/10.1017/pds.2022.184 Published online by Cambridge University Press

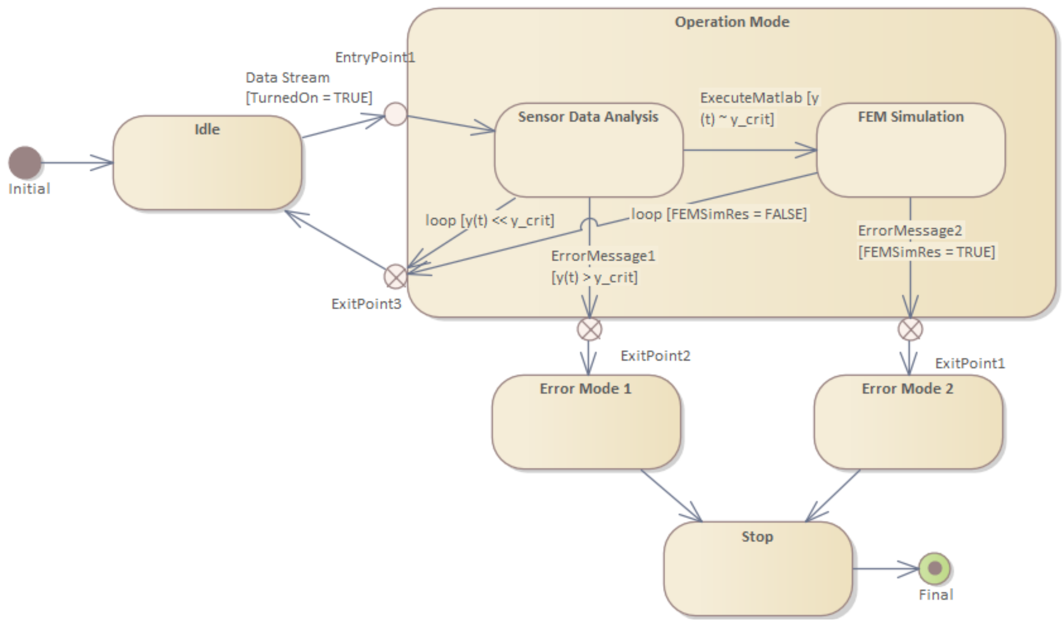

Figure 5. State Machine Diagram of the Digital Twin Concept

With an executable state machine diagram this generic behaviour was exported into JavaScript-code (Figure 6), which is further used as the defining code structure for the later system behaviour. This code can easily be implemented into the UI and its backlog to achieve that, given a specific state, e. g. critical sensor data, leads to the modelled actions planned within the design of the Digital Twin. The Dashboard as central program of the Digital Twin demonstrator used this code to interpret incoming data from the Raspberry PI and controlled the actions of the system as well as the displayed recommendations for the user.

```
function Block1(manager, instanceName) {

    /* Begin - EA generated code for StateMachine */

    this.base    =   new    StateMachineContext(this,    manager,
instanceName, "Block1");
    //Initialize Region Variables
```

Figure 6. Beginning of the exported State Machine Code

4.4. Effort Compensation for the Scenario

Using SysML for defining a Digital Twin concept can be reasonable to improve the overall communication. However, the effort of modelling these diagrams can be compensated with directly utilizing the behaviour diagrams to establish a behaviour structure for other software. The here used JavaScript file, which is very simplified, includes around 2000 lines of code. The creation of this state machine and deviating executable code takes a few minutes, being very accessible about every state and its conditions. Hence, a "part effort compensation" is given by saving additional coding and further using the created models instead of simply using them for communicational benefits or documentation. Especially the preliminary usage of the diagrams to discuss the design of the demonstrator helped significantly, even for such a simple system, to implement the planned behaviour into the component design. However, this compensation comes with a price of modelling significantly more detailed and with a high awareness of using tool features, which might slow down the modelling process.

5. Further Discussion & Assessment

The presented simplified demonstrator for utilizing SysML diagrams in later lifecycle stages of a Digital Twin has shown that the technical possibility is given to use the diagrams as basis for software that

https://doi.org/10.1017/pds.2022.184 Published online by Cambridge University Press

enables the operation of Digital Twins. However, it must be evaluated whether the achieved effort compensation justifies a detailed modelling in SysML. While SysML diagrams are defined by a specific standard, human interpretation allows to a specific degree the understanding of non-consistent diagrams or models. This is not the case for machine-readable diagrams. In fact, while the models were created quickly, the design of the central program, bringing together the streamed data and the defined behaviour, has shown that a predominantly effort had to be invested into the alignment of these. While internal development teams might be able to agree on a specific standard for modelling, enlarging this on the supply chain contains tremendous challenges for agreeing on a consistent approach of modelling similar machine-readable diagrams to prevent further effort for their alignment and enabling the interchangeability of different diagrams into the tool. Especially the usage of a central state-machine must be viewed critically. Using a single diagram allows a comfortable adaption towards this diagram. In fact, a system behaviour is defined by a variety of different diagrams, which need to be considered for designing a Digital Twin. Using SysML diagrams however has achieved a detailed awareness of the system behaviour and interdependencies. The usage of the diagrams during the design phase helped to create a common understanding of the gear and its Digital Twin. Especially during the further utilization of these diagrams, it became clear that a common understanding of the system was also based on a common interpretation but not in any case given by a completely consistent model focusing the satisfaction of the mentioned requirements. Nevertheless, also non-technical problems might evolve by using this approach along the supply chain. Many details about a system behaviour, e. g. given in FEM-simulations and the derivation of information about the remaining lifetime, are a significant element of enterprise's knowhow. Passing this information on to enable the Digital Twin concept for the customer could bear the risk of security gaps. Additional possibilities could have been identified along the modelling process, by combining SysML with other modelling capabilities such as modelling the IoT infrastructure. This gives a clear overview of the whole Digital Twin concept and existing interdependencies between its elements or subsystems. While some modelling tools offer specific export functions, e.g. state machines to JavaScript, for many diagrams the only export solution is using XML. Especially through other models while initially being implemented to ensure model exchange between different software tools, a further utilization is possible but implies other challenges, stated by Handley et al. (2021).

6. Summary & Outlook

The paper has shown that in principle the utilization of SysML diagrams for defining a behavioural structure of a Digital Twin is possible. The concept for this utilization used given possibilities, e.g. the JavaScript export and existing simulation tool capabilities. However, the effort compensation for complex systems can be given by combining the lucidity of a SysML diagram with auto-generated code. Combining these two benefits could be a feasible approach for future Digital Twin software that improves the accessibility for technically enabling Digital Twins. The presented approach proves that combining these two benefits within a software tool is possible and especially for complex systems an effort compensation is given by autogenerated code that saves time and reflects the current state of the system model. With this approach, a step towards the overall MBSE goal of a single source of truth is accomplished. Nevertheless, the stated challenges must be faced to ensure a feasible integration of the approach into the actual usage of system models. In general, "SysML4DigitalTwins" is a promising concept that must be developed further to lower the obstacles of Digital Twin design. Utilizing SysML could be further used as a main approach of handling Digital Twin software by easily creating a behaviour through system models instead of coding it. It must be further investigated how several different diagrams can be considered by the software and how a transformation towards an Autonomous Digital Twin can be made by using the diagrams to control actuators within the system and ensure a consistent control loop.

References

Bao, J., Guo, D., Li, J. and Zhang, J. (2019), "The modelling and operations for the digital twin in the context of manufacturing", Enterprise Information Systems, Vol. 13 No. 4, pp. 534–556. DOI: 10.1080/17517575.2018.1526324.

Eigner, M., Huwig, C. and Dickopf, T. (2015), “Cost-Benefit Analysis in Model-Based Systems Engineering: State of the Art and Future Potentials”, in– Proceedings of the 20th International Conference on Engineering Design (ICED15), pp. 227–236.

Gartner (2018), “Hype Cycle for Emerging Technologies”, https://www.gartner.com/smarterwithgartner/5-trends-emerge-in-gartner-hype-cycle-for-emerging-technologies-2018/ (accessed 21 July 2020).

Handley, H.A.H., Khallouli, W., Huang, J., Edmonson, W. and Kibret, N. (2021), “Maintaining the Consistency of SysML Model Exports to XML Metadata Interchange (XMI)”, in 2021 IEEE International Systems Conference (SysCon), Vancouver, BC, Canada, IEEE, pp. 1–8. DOI: 10.1109/SysCon48628.2021.9447105.

Honour, E.C. (2013), “Systems Engineering Return on Investment”, Dissertation, Defence and Systems Institute, University of South Australia, 2013.

Hribernik, K., Wuest, T. and Thoben, K.-D. (2013), “Towards Product Avatars Representing Middle-of-Life Information for Improving Design, Development and Manufacturing Processes”, in Kovács, G.L. and Kochan, D. (Eds.), IFIP TC 5 International Conference, NEW PROLAMAT 2013, Dresden, Germany20, pp. 85–96.

Kapos, G.-D., Dalakas, V., Nikolaidou, M. and Anagnostopoulos, D. (2014), “An integrated framework for automated simulation of SysML models using DEVS”, SIMULATION, Vol. 90 No. 6, pp. 717–744. DOI: 10.1177/0037549714533842.

Karban, R., Dekens, F.G., Herzig, S., Elaasar, M. and Jankevičius, N. (2016), “Creating system engineering products with executable models in a model-based engineering environment”, in Angeli, G.Z. and Dierickx, P. (Eds.), Modeling, Systems Engineering, and Project Management for Astronomy VI, 2016, Edinburgh, United Kingdom, SPIE, 99110B. DOI: 10.1117/12.2232785.

Lee, J., Lapira, E., Bagheri, B. and Kao, H. (2013), “Recent advances and trends in predictive manufacturing systems in big data environment”, Manufacturing Letters, (1). 38–41. DOI: 10.1016/j.mfglet.2013.09.005.

Mahboob, A., Husung, S., Weber, C., Liebal, A. and Krömker, H. (2018), “SysML Behaviour Models for Description of Virtual Reality Environments for Early Evaluation of a Product”, in Proceedings of the DESIGN 2018 15th International Design Conference, pp. 2903–2912. DOI: 10.21278/idc.2018.0382.

Makarov, V.V., Frolov, Y., Parshina, I.S. and Ushakova, M.V. (2019), “The Design Concept of Digital Twin”, in Cvirkun, A.D. (Ed.), Proceedings of 2019 Twelfth International Conference MLSD: Russia, pp. 1–4. DOI: 10.1109/MLSD.2019.8911091.

Rosen, R., Wichert, G. von, Lo, G. and Bettenhausen, K. d. (2015), “About The Importance of Autonomy and Digital Twins for the Future of Manufacturing”, IFAC-PapersOnLine, Vol. 48 No. 3, pp. 567–572. DOI: 10.1016/j.ifacol.2015.06.141.

Schleich, B., Anwer, N., Mathieu, L. and Wartzack, S. (2017), “Shaping the digital twin for design and production engineering”, CIRP Annals, Vol. 66 No. 1, pp. 141–144. DOI: 10.1016/j.cirp.2017.04.040.

Schleich, B., Dittrich, M.-A., Clausmeyer, T., Damgrave, R., Erkoyuncu, J.A., Haefner, B., Lange, J. de, Plakhotnik, D., Scheidel, W. and Wuest, T. (2019), “Shifting value stream patterns along the product lifecycle with digital twins”, Procedia CIRP, Vol. 86, pp. 3–11. DOI: 10.1016/j.procir.2020.01.049.

Schroeder, G.N., Steinmetz, C., Pereira, C.E. and Espindola, D.B. (2016), “Digital Twin Data Modeling with AutomationML and a Communication Methodology for Data Exchange”, IFAC-PapersOnLine, Vol. 49 No. 30, pp. 12–17. DOI: 10.1016/j.ifacol.2016.11.115.

Schumacher, T. and Inkermann, D. (2021), “Heterogene Modellierung - Verknüpfung und Integration von Systemmodellen der SysML mit CAD-Modellen”, in: Proceedings of the 32nd Symposium Design for X, The Design Society. DOI: 10.35199/dfx2021.20.

Sheard, S., Cook, S., Honour, E., Hybertson, D., Krupa, J., McEver, J., McKinney, D., Ondrus, P.: Ryan, A., Scheurer, R., Singer, J., Sparber, J. and White, B. (2015), A Complexity Primer for Systems Engineers.

Stark, R., Anderl, R., Thoben, K.-D. and Wartzack, S. (2020), “WiGeP-Positionspapier: „Digitaler Zwilling“”, ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb, Vol. 115 No. special, pp. 47–50. DOI: 10.3139/104.112311.

Tao, F., Sui, F., Liu, A., Qi, Q., Zhang, M., Song, B., Guo, Z., Lu, S.C.-Y. and Nee, A.Y.C. (2019), “Digital twin-driven product design framework”, International Journal of Production Research, Vol. 57 No. 12, pp. 3935–3953. DOI: 10.1080/00207543.2018.1443229.

Wang, Y., Steinbach, T., Klein, J. and Anderl, R. (2021), “Integration of model based system engineering into the digital twin concept”, Procedia CIRP, Vol. 100, pp. 19–24. DOI: 10.1016/j.procir.2021.05.003.

Weilkiens, T. (2014), Systems Engineering mit SysML/UML: Anforderungen, analyse, architektur, Germany.

Wilking, F., Schleich, B. and Wartzack, S. (2020), “MBSE along the Value Chain – An Approach for the Compensation of additional Effort”, in 2020 IEEE 15th International Conference of System of Systems Engineering (SoSE), Budapest, Hungary, IEEE, pp. 61–66. DOI: 10.1109/SoSE50414.2020.9130497.

Wilking, F., Schleich, B. and Wartzack, S. (2021), “DIGITAL TWINS - DEFINITIONS, CLASSES AND BUSINESS SCENARIOS FOR DIFFERENT INDUSTRY SECTORS”, Proceedings of the Design Society, Vol. 1, pp. 1293–1302. DOI: 10.1017/pds.2021.129.