INTERNATIONAL DESIGN CONFERENCE - DESIGN 2020
https://doi.org/10.1017/dsd.2020.41

METHOD FOR 3D-ENVIRONMENT DRIVEN DOMAIN KNOWLEDGE ELICITATION AND SYSTEM MODEL GENERATION

S. Japs ✉, L. Kaiser and A. Kharatyan

Fraunhofer IEM, Germany

✉ sergej.japs@iem.fraunhofer.de

Abstract

The development of cyber-physical systems requires close cooperation between stakeholders from different disciplines. Model-based systems engineering support this by the design of a system model. Non-identified domain knowledge by the stakeholders is a challenge when creating the system model. The CONSENS 3D-Modeling Method supports the domain-independent elicitation of domain knowledge using a 3D environment and enables the derivation of a SysML system model. We applyed the method by implementing a prototype, called 3D Engineer, to an application example from the automotive industry.

Keywords: 3D modelling, virtual engineering (VE), model-based engineering, cyber-physical systems, systems modeling language

1. Introduction

The development of intelligent technical systems, such as autonomous vehicles, is characterised by close cooperation between mechanical engineering, electrical engineering and software engineering. The interdisciplinary and complexity of these systems leads to an increasing challenge for an effective and efficient development. Model-based Systems Engineering (MBSE) addresses these challenges by methods for the development of a common system model involving different stakeholders.

The precise and early elicitation of the domain knowledge of the stakeholders constitutes a challenge for the development of the system model. Missing domain knowledge within the system model is a risk for the discipline-specific development regarding development costs and development time.

There are several approaches (Coulin and Zowghi, 2005; Asger and Yousuf, 2015) that address the elicitation of domain knowledge, such as the requirement elicitation techniques interview or brainstorming. However, these techniques are aimed for textual elicitation of requirements and not for the creation of a system model. In particular, interviews lead to ambiguous requirements (Alessio et al., 2016). Conversely, MBSE approaches such as CONSENS (Gausemeier et al., 2014) provide a method for developing a system model, but do not help to elicitate the domain knowledge of stakeholders.

The theory of situational cognition postulates that the elicitation of associated knowledge is more successful when people are placed in a situational context (Brown et al., 1989). For example, if one stands on the side of the road and watches the traffic, more safety-critical situations come to mind than if one were standing in front of a whiteboard. There are already a number of approaches that support the elicitation of domain knowledge using 3D environments (Florides et al., 2015; Bhimani and Spolentini, 2017; Bhimani, 2017; Brown et al., 2015; Brown et al., 2017). 3D environments allow e.g. the playful interaction with 3D

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press

objects inside the 3D environment as well as the change of the perspective. However, these approaches can only be applied domain-specific. Another problem with these approaches is that after the knowledge has been elicitated, it is necessary to transfer it manually to an MBSE software tool. The approach (Brown et al., 2015) allows additionally deriving models, but is domain-specific.

In this paper we introduce the CONSENS 3D-Modeling Method, which is domain-independent and, using 3D environments, improves the elicitation of domain knowledge.

In order to reduce the modeling effort, the CONSENS 3D Modeling Method enables the automatic derivation of a system model. For this we extend the CONSENS method for use in 3D environments. The CONSENS method has been applied in numerous industrial projects in the product conception phase, e.g. to develop smart home products (UNITY Innovation Alliance, 2020) or as a general method for designing mechatronic products (Smart Mechatronics, 2020). We use SysML (OMG, 2015) as modeling language, since SysML is the de-facto modeling language for systems engineering (Dori, 2016).

We illustrate our approach with the application example Platooning. Platooning is a method for driving a group of vehicles autonomously together. Here we illustrate a safety hazard by the occurrence of an obstacle in front of the platoon. Furthermore we illustrate a safety critical security threat by the manipulation of sensor data to manipulate the platoon behavior via a fake obstacle.

We structure this work as follows: First, in Section 2, the necessary foundations for our approach will be established. For this purpose the MBSE method CONSENS is presented, on which our approach is based. Furthermore, a classification of related work and a delimitation will be performed. In section 3, we introduce the CONSENS 3D Modeling Method, which allows the application of the existing CONSENS method for the use in a 3D environment and ensures an automatic derivation of a SysML system model. Section 4 presents the application of the method using the platooning example. For this we present a general software architecture for a tool implementation. Furthermore, we show the application of our method by means of a prototypical implementation by our tool 3D Engineer. Finally, a discussion, summary and an outlook are presented.

2. Fundamentals and related work

In this section we present the fundamentals of the CONSENS 3D-Modeling Method presented in section 3 and delimit it from relevant existing approaches.

2.1. CONSENS method and SysML4CONSENS-profile

As the fundamental approach we use the CONSENS method (Gausemeier et al., 2014). CONSENS is a Model-based Systems Engineering (MBSE) approach and designed for the interdisciplinary design of complex intelligent technical systems. The method defines, centrally for an MBSE approach, based on a modeling language the design of a system model in the concept phase. The CONSENS method requires the design of different partial models for the different aspects of a technical system. The Environment Model views the system as a black box in the context of its environment. In this model, interactions between environmental elements and the system are modelled. Application Scenarios describe a situation-specific view of the described system. The partial model Requirements comprises a structured collection of requirements for the system. The partial model Functions allows a hierarchical breakdown of the functionality of the system. In the System Structure, the components of the system and the relationships between the components are modelled. In the partial model Behavior, the system behavior is modeled by activities, states or sequences. The partial model Shape, addresses the early definition of the shape of the system. By means of cross-relationships, the different partial models can be connected with each other. Relationships within these models are distinguished according to the following relationship types: information, energy, substance and if unclear logical relationship. While CONSENS defines its own modeling language, we use the modeling language SysML (OMG, 2015). This defines a requirements diagram and several structure and behavior diagrams. In order to be able to use the CONSENS method in combination with SysML, we use a corresponding SysML profile (Dumitrescu et al., 2013). Profiles allow the modelling language to be extended by adding further stereotypes. E.g. the stereotype “information” (see Figure 4), allows the relationship “Manipulated data” between the environmental element “hacker” and the system “car” to be categorized as an information relationship.

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press

2.2. Classification of related work and delimitation

We have examined a number of approaches that support system design and allow to derive a system model in a modeling language. We exclude approaches like (Atukorala and Chang, 2018), which are very formal and can only be understood by discipline experts, or approaches that do not use an established modelling language. For this purpose, we consider the requirement elicitation and documentation as an early form of system design. For this we have created a classification (see Figure 1). We distinguish between approaches which support domain-dependent and independent system design. Furthermore, we distinguish between approaches which additionally allow a derivation of a system model or not. We illustrate the classification in Figure 1, exemplarily on the basis of approaches that compete with our approach or are relevant to it. The Driving Simulator based method (Florides et al., 2015) supports the elicitation of requirements for the automotive industry by means of a simulation-based 3D environment. The approach does not deal with the derivation of requirements or a system model. The Business Process Modeling Method (Brown et al., 2015) supports the identification of business processes through a 3D environment in the form of an office complex. Here the determined business processes are automatically created in the form of BPMNs. Both approaches are not independently applicable to domains.

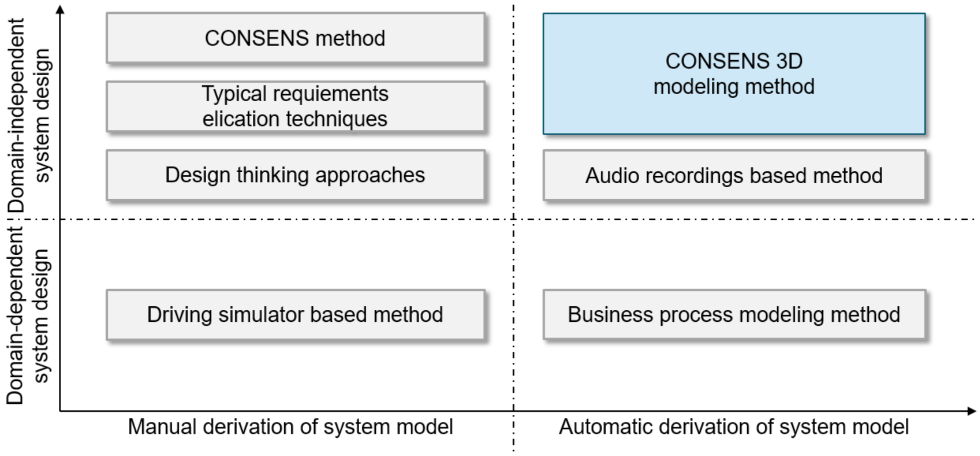
Domain-independent system design
Domain-dependent system design
CONSENS method
Typical requiements elication techniques
Design thinking approaches
CONSENS 3D modeling method
Audio recordings based method
Driving simulator based method
Business process modeling method
Manual derivation of system model
Automatic derivation of system model

Figure 1. Classification of MBSE approaches

An example for the domain independent system design is the CONSENS method (Gausemeier et al., 2014) (cf. Section 2.1). The method helps to create the system model by means of different partial models. The system aspect shape can be linked with other Partial models by cross-relationships. However, this aspect is not methodically addressed in order to support the design of the individual partial models. In particular, CONSENS offers no automatic derivation of a system model. Techniques for the elicitation of requirements such as interviews or document analysis can be used independently of domains. However, these do not support the automatic derivation of requirements of a system model. Design thinking refers to the cognitive, strategic and practical processes by which design concepts are developed. According to (Brenner and Uebernickel, 2016), design thinking contains phases which are run iteratively to develop specific and stepwise more detailed prototypes. (Tomita et al., 2017; Lewrick et al., 2018; Tekaat et al., 2019) are showing the feasibility and potentials of combining design thinking with systems engineering. The creation of prototypes in design thinking approaches harmonizes with approaches that use 3D objects as system representatives. However, we did not find any approach that allows the automatic generation of a system model. The Audio Recordings based method (Abad et al., 2018) is domain independent. By analyzing audio recordings through e.g. interviews, requirements can be automatically recorded in text form and be categorized. No support in the elicitation of these requirements is considered in this method.

Based on our literature analysis, we did not find approaches that simultaneously improve the elicitation of domain knowledge, can be applied domain-independently and allow the automatic derivation of a system model to reduce the manual modeling effort. We fill this research gap with the CONSENS 3D-Modeling Method (cf. Section 3).

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press

3. CONSENS 3D-Modeling Method

In this section we present the CONSENS 3D-Modeling Method. By using 3D environments, this method enables the playful elicitation of requirements and design of an informal system model. Furthermore, the method allows the automatic derivation of a SysML system model.

The method generally consists of four process steps (see Figure 2). The first step consists of the definition of the general use case, here platooning. This is followed by the application of the CONSENS 3D analysis phase (see Section 3.1) and the application of the CONSENS 3D synthesis phase (see Section 3.2). The existing CONSENS analysis and synthesis phase was extended for use in 3D environments. The last step is the automatic derivation of a SysML system model based on the steps performed in the analysis and synthesis phase. For this purpose, a meta model is used which defines the mapping between text, elements of the 3D environment and SysML elements (see Section 3.1).

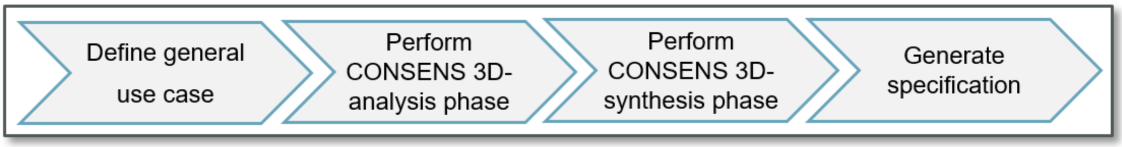
Define general use case
Perform CONSENS 3D-analysis phase
Perform CONSENS 3D-synthesis phase
Generate specification

Figure 2. Method overview

3.1. Meta model

The meta model shown in Figure 3 is fundamental for the CONSENS 3D analysis and synthesis phase and the derivation of SysML system models. This defines the mapping between text elements, elements of the 3D environment and SysML elements. Stereotypes allow the specific distinction between the respective elements. Furthermore, by noting the position sequence, it is possible to capture the motions of 3D objects.

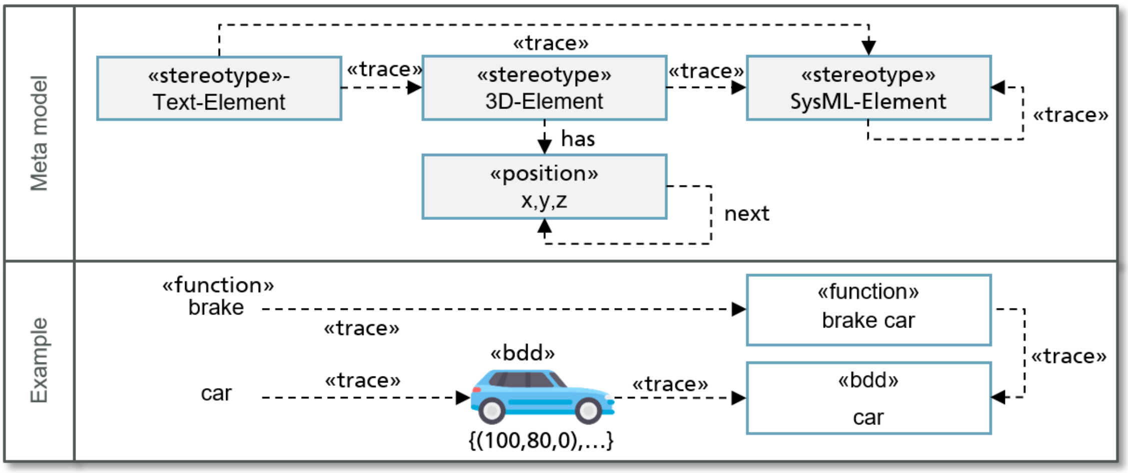

Figure 3. Meta model

Figure 3 shows the application of the meta model to the function “brake car”. The text element of the function can be allocated via the trace-relationship directly to the SysML element “brake car” with the stereotype “function”. In addition, “car” can be assigned to a 3D and SysML element with the stereotype “bdd”. A SysML Block Definition Diagram (bdd) represents system elements as black-boxes. Vehicle movements are captured using the position sequence. Since there is a relationship between the function “brake car” and a specific “car”, this is annotated accordingly.

3.2. CONSENS 3D-analysis phase

For the application of the CONSENS method using a 3D environment, an extension or adaptation of the method is necessary. In this section we introduce the 3D analysis phase, which is based on the existing CONSENS analysis phase. The 3D analysis phase consists of three sub-processes (see Figure 4).

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press

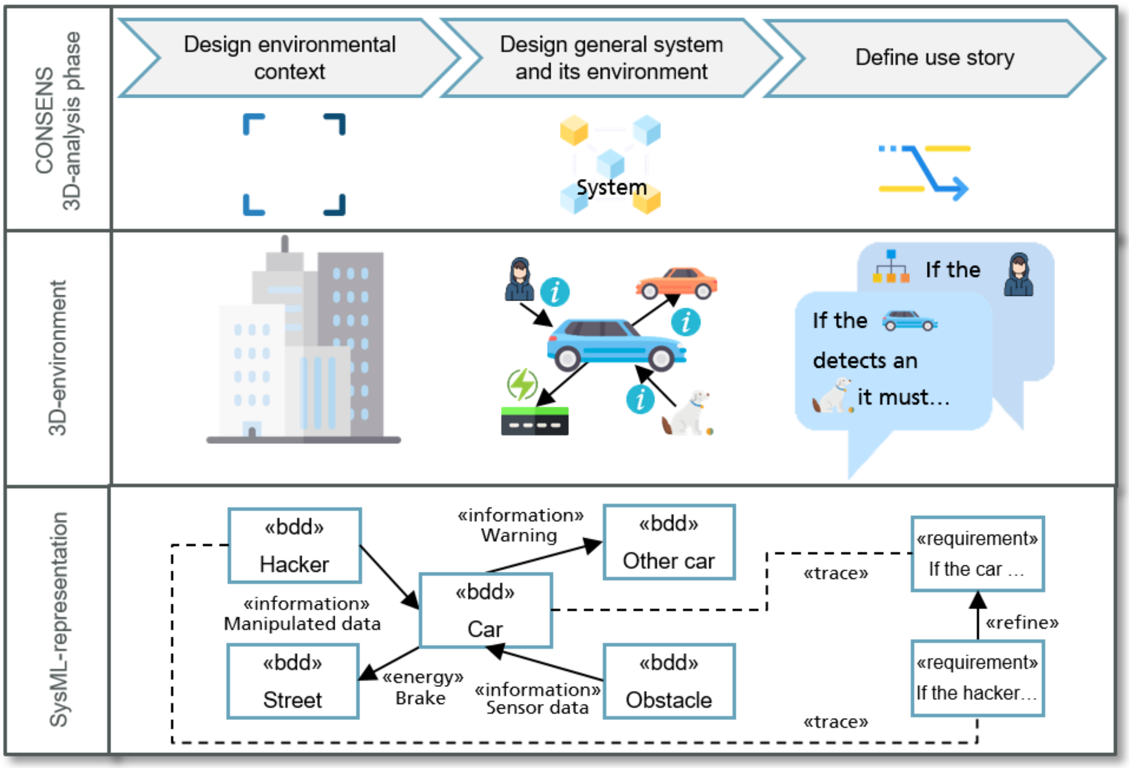

Figure 4. CONSENS 3D-analysis phase overview

The first sub-process, design of an environmental context, is new and precedes the previous analysis phase. For the general use case, here platooning, an environmental context in the 3D environment has to be designed. The environmental context provides the foundation for situational cognition in the 3D environment. For platooning we choose a major city as the environmental context.

The second sub-process uses the environmental context in order to design the general system and its direct environment, supported by the use of 3D objects. In this sub-process a boundary is defined between the system under design and its direct environment. In addition, relationships shall be established between the system and the environmental elements. We define the individual vehicle in the platoon as the system. The situational cognition is reinforced by the environmental context and enables the derivation of environmental elements for the system out of the environmental context.

Examples would be other vehicles in the platoon, the roadway and obstacles that could be located in the major city. The detection of obstacles corresponds to a CONSENS information relationship between obstacle and vehicle. The braking of the vehicle on the road represents an energy relationship. In addition, the environmental element hacker can be derived for the environmental context of major cities. The hacker could be in a house or café close to the roadway in order to manipulate vehicle sensor data or the communication between vehicles in the platoon.

The third sub-process deals with the definition of user stories. This sub-process corresponds with the creation of an application scenario in CONSENS. User stories are short requirements formulated in common language. In the context of the 3D analysis phase, user stories are used to describe what is to be modelled and what has already been modelled in the 3D environment. User stories are linked with environmental elements and the general system from the 3D environment. They can also be linked to each other. The modeled relationships between the 3D objects and the modeled behavior (see section 3.3) provide potential for the automatic generation of user stories.

An example is the following user story: If the vehicle detects a sudden appearing obstacle, it must perform an emergency braking. In order to increase precision and avoid ambiguities, a connection is established between the specific vehicle and the obstacle with the corresponding 3D elements. In contrast to the existing CONSENS analysis phase, the use of user stories represents a simplification and serves as a basis for the formulation of requirements.

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press

Based on the process steps of the 3D analysis phase, SysML models can be derived from Section 3.1 using the meta model. Figure 4 shows a CONSENS environment model for the Platooning application example in the form of a SysML-BDD. The user stories from the 3D environment are presented in the form of SysML requirements. Cross-referencing of user stories with each other is also done for SysML requirements. Since user stories are linked with 3D elements, a link is set between the SysML requirements and the SysML-BDD. This is used to formulate precise requirements.

3.3. CONSENS 3D-synthesis phase

In this section we introduce the 3D synthesis phase, which is based on the existing CONSENS synthesis phase. The 3D synthesis phase consists of three sub processes (see Figure 5).

The first sub process deals with the design of system functions. System functions are initially independent of solutions and can be implemented in different ways. They are defined in CONSENS by a subject + verb combination. The extension consists of linking subjects of a function with 3D elements. For example, for the function “Detect obstacle” a link from “obstacle” to the representing 3D element is created. If there is a hierarchical relationship between functions, this is also captured. The function “Detect obstacle”, for example, requires the function “Brake car”.

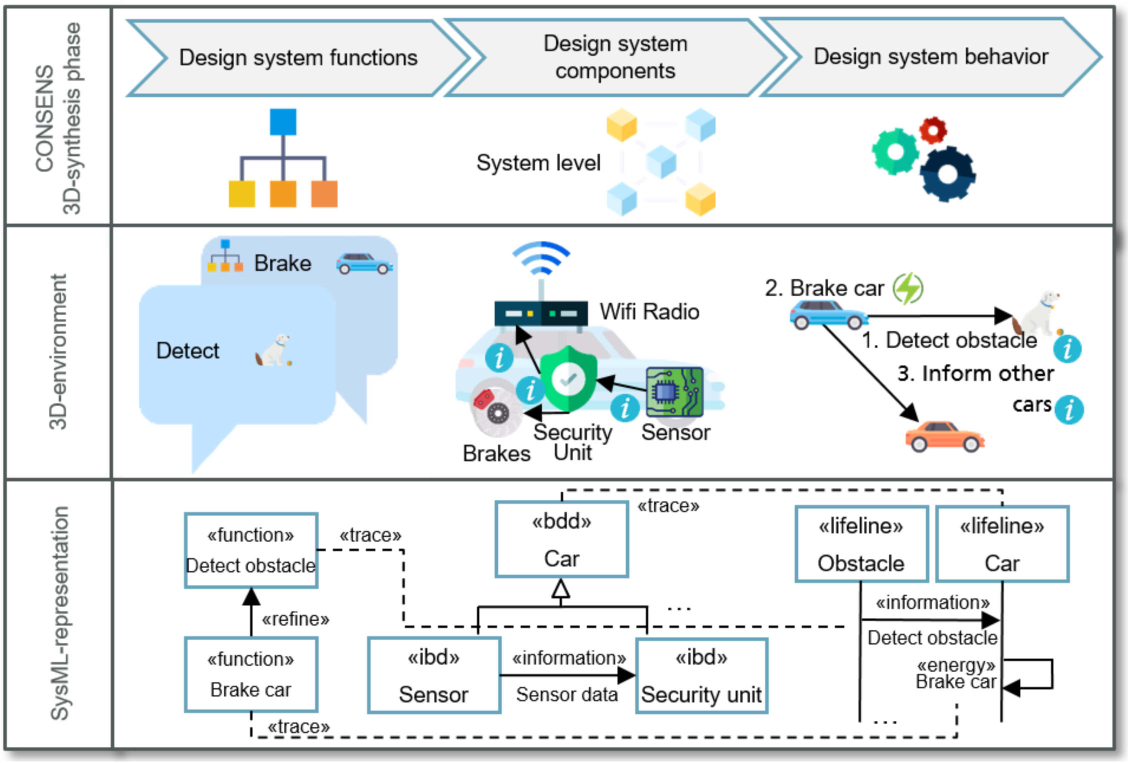

Figure 5. CONSENS 3D-synthesis phase overview

The second sub-process deals with the design of system components and their relationships to each other. Using 3D objects, system components can be placed relative to each other in the general system and can be connected to each other. The modelled artefacts environment model, user stories and functions serve as a basis. In the context of the application example, a sensor component is required for the detection of obstacles. A security unit has to analyze the sensor data in case of external attacks. There is an information relationship between the sensor component and the security unit.

The third process step deals with the design of the system behavior using the 3D environment. The system behavior can be modelled on two different levels. On the one hand on level of the general system and its environment and on level of the system components. The artifacts that have already been created in previous process steps serve as a basis. An important aid is the positioning of 3D elements in the 3D environment. Analogous to the procedure in SysML, modeling can be condition-

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press

based, activities-based and interaction-based, depending on the behavior focus. Figure 5 shows an interaction-based sequence within the 3D environment. Because an interaction represents a relationship between two 3D elements, it can be differentiated by a relationship type. If an obstacle is detected, the vehicle should then brake and finally warn other vehicles. Based on the process steps of the 3D synthesis phase, SysML models can be derived from Section 3.1 using the meta model. Figure 5 shows SysML models for functions, the system architecture, and a sequence diagram for system behavior. If a SysML model element occurs within another SysML model element, it is linked to other model elements to avoid ambiguities. For example, the function “Brake car” appears in the sequence diagram as an interaction message. Here, “car” refers to the vehicle that first recognized the obstacle.

4. Application of the method and prototypical implementation of 3D Engineer

In this section we introduce the application of the CONSENS 3D-Modeling Method. First, a general software architecture for tool support is discussed, then a prototypical implementation of our tool - 3D Engineer (Japs, 2020) - will be presented.

Figure 6 shows the general architecture of 3D Engineer. 3D Engineer requires a 3D model library. For this we use the Unity Asset Store (Unity, 2019a), which contains more than 31000 3D object packages and is continuously extended. Most object packages contain several 3D objects up to over 100. 3D Engineer extends a 3D engine by the process steps of the CONSENS 3D analysis and synthesis phase. The 3D engine we use is Unity (Unity, 2019b), one of the most commonly used development environments for 3D games. Part of 3D Engineer is the generation of SysML models. A Model-based Systems Engineering Environment is required for further processing of the SysML models. For this we used the CAMEO Systems Modeler (CMS) (NoMagic, 2019). For the model import we developed our own CMS plugin.

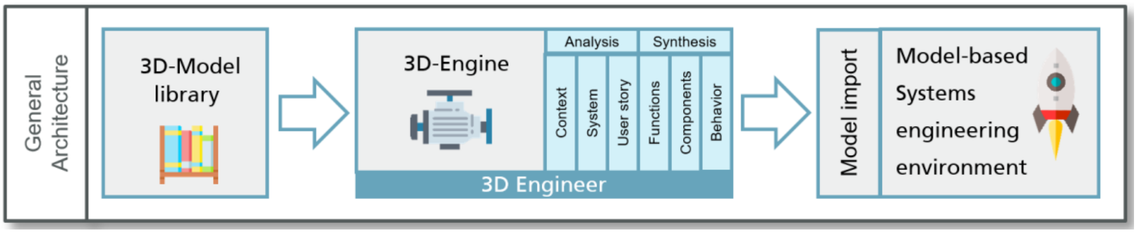

Figure 6. Software architecture overview

Figure 7 shows the prototype implementation of 3D Engineer. We used the CONSENS 3D method for the general Use Case Platooning using 3D Engineer. We created an environment model in 3D Engineer and automatically generated a SysML-BDD (see Figure 8). Furthermore, we modelled two interaction sequences in 3D Engineer (see Figure 7) and automatically generated two sequence diagrams (see Figure 9).

The application of the method is described in detail below (see Figure 7). During the analysis phase, an environmental context was created for the general use case. For the environmental context “Major City”, we used Windridge City from the Unity Asset Store, which is often used in the scientific community. We used 3D objects from the Asset Store to model the general system and the environment.

The general system is the front vehicle. As environmental elements we use a woman as a hacker, a box as an obstacle and a rear vehicle as another vehicle in the platoon. Relationships are established between the general system and the environmental elements and differentiated according to relationship types. We have indicated a possible automatic text recognition by speech input by the microphone symbol.

In play mode, requirements are determined in a playful way by touch gestures by placing 3D objects, changing perspectives and moving objects. The Specification Mode is used to intuitively design the system model. In order to keep up the modeling flow, dummy objects are used, here the box. These boxes can potentially be identified by a label.

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press

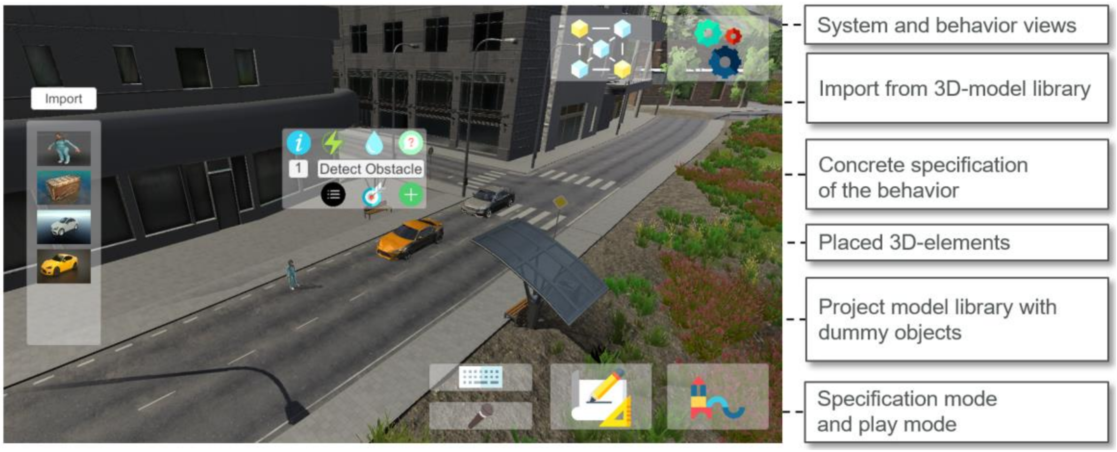
Import
Detect Obstacle
System and behavior views
Import from 3D-model library
Concrete specification of the behavior
Placed 3D-elements
Project model library with dummy objects
Specification mode and play mode

Figure 7. Implementation of 3D Engineer

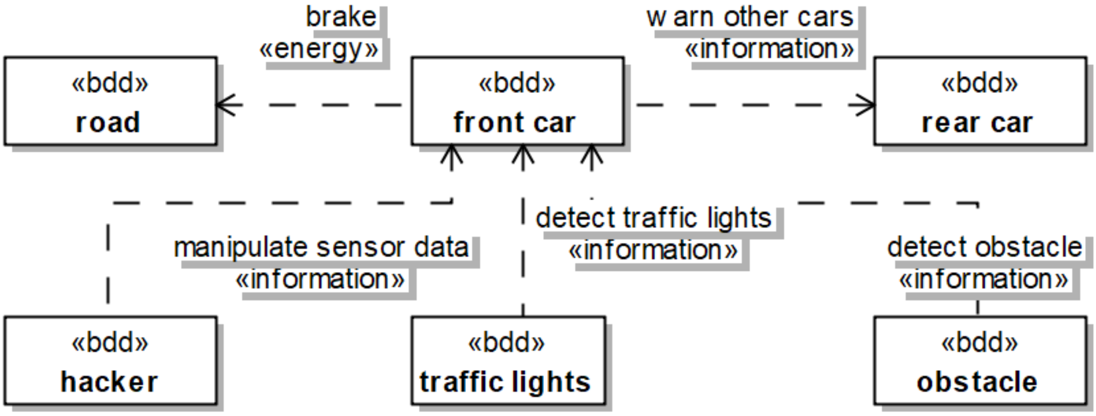
«bdd» road
brake «energy»
«bdd» front car
w arn other cars «information»
«bdd» rear car
manipulate sensor data «information»
detect traffic lights «information»
detect obstacle «information»
«bdd» hacker
«bdd» traffic lights
«bdd» obstacle

Figure 8. Generated system environment model

During the synthesis phase, system components and their relationships are designed. For this purpose, the implementation of the general system and its environment could be reused. For the design of the system behavior, we focused on the realization of the interaction behavior (see Figure 7). For this purpose, a sequence of interactions between 3D elements is formed and described textually. In order to differentiate more precisely, each interaction relationship is assigned to one of the four relationship types of CONSENS (see Section 2.1). Figures 8 and 9 show an environment model and a behaviour model. Both models were designed using 3D Engineer and the CONSENS 3D-Modeling Method and were automatically generated in the form of SysML models. For this purpose 3D Engineer generates a file. This file is processed using the CMS plugin we developed, so that SysML models are automatically created from it.

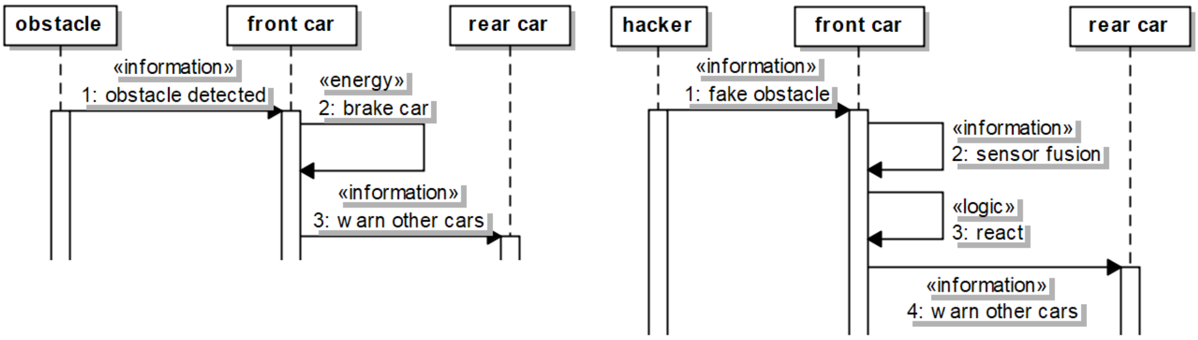

Figure 9. Generated behavior diagrams

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press

5. Discussion

In this paper, we have presented an requirements engineering approach which simultaneously improve the elicitation of domain knowledge, can be applied domain-independently and allow the automatic derivation of a system model to reduce the manual modeling effort.

We have identified the following challenges: The initial effort for our approach is relatively high. For preparation, a suitable selection of 3D models must be determined. Missing models must be created manually or represented by dummy objects. Furthermore computer hardware like tablets is necessary. The use in an industrial environment requires the exact knowledge of the corresponding license for free 3D objects. For example, in design thinking, sticky notes, pens and cardboard are sufficient. For using CONSES, a whiteboard and pens are sufficient. Classical elicitation techniques such as interviews or questionnaires require only a written preparation.

We have identified the following benefits: Compared to CONSENS, our approach supports the elicitation of the stakeholders' domain knowledge through an interactive visualization in a 3D environment. Furthermore, the additional manual effort of digitizing domain knowledge in the form of digital models is not required. Created digital models and relationships can be reused in 3D Engineer regardless of location and in subsequent projects. In contrast, analog approaches such as design thinking or CONSENS. Here, the extracted domain knowledge for example only exists in the form of labeled whiteboards, walls with sticky notes or in the form of cardboard prototypes. The consolidation of the domain knowledge in the form of digital models is still necessary in this context. Especially the creation of digital models based on design thinking results requires additional cognitive effort.

The direct creation of SysML models saves preparation effort, but the situational cognition regarding elicitation of domain knowledge from the stakeholders is not addressed. In particular, our approach enables the elicitation of domain knowledge without requiring detailed SysML knowledge from the stakeholders.

6. Conclusions and future work

The development of intelligent technical systems requires the cooperation of stakeholders from different disciplines. Model-based Systems Engineering (MBSE) addresses this challenge through methods like CONSENS for the creation of a common system model. The precise and early elicitation of the domain knowledge of the stakeholders represents a challenge for the creation of the system model and, in the negative case, poses risks for the discipline specific design.

The CONSENS 3D-Modeling Method addresses this challenge. Compared to existing approaches, this method is domain independent. For this purpose, in order to increase the situational cognition of stakeholders, the method provides a relation to 3D environments. To ensure consistency in engineering, the method allows an automatic derivation of a SysML system model. We have illustrated our approach by means of the platooning application example. For this purpose, we applied the method on the basis of the platooning example by our prototypical implementation - 3D Engineer.

Part of future work is the validation of the method and evaluation of other application examples like robot production cell. For this purpose, a comparison with two groups of people is suitable, in which one group uses e.g. established approaches of requirement engineering and the other group uses the CONSENS 3D method. In order to increase situational cognition, the use of the CONSENS 3D method in combination with virtual reality will be investigated.

Acknowledgement

I like to thank my students Oliver von Heißen and Sebastian Schmidt for they support in implementing 3D Engineer. This research was funded by the German Federal Ministry of Education and Research (BMBF) in the project SecForCARs, grant number 16KIS0790. The contents of this publication are the sole responsibility of the authors.

References

Abad, Z.S.H. et al. (2018), "ELICA: An Automated Tool for Dynamic Extraction of Requirements Relevant Information", *Proceedings of 5th International Workshop on Artificial Intelligence for Requirements Engineering (AIRE)*, Banff, AB, pp. 8-14. https://doi.org/10.1109/AIRE.2018.00007

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press

Alessio, F., Gnesi, S. and Spoletini, P. (2016), “Ambiguity and tacit knowledge in requirements elicitation interviews”, *Requirements Engineering*, Vol. 21 No. 3, pp. 333-355. https://doi.org/10.1007/s00766-016-0249-3

Asger, M. and Yousuf, M. (2015), “Comparison of Various Requirements Elicitation Techniques”, *International Journal of Computer Applications*, Vol. 116 No. 4, pp. 8-15. https://doi.org/10.5120/20322-2408

Atukorala, N.L. and Chang, C.K. (2018), Situation-Oriented Software Requirements Specification and Model Generation, Iowa State University, Iowa, USA.

Bhimani, A. and Spolentini, P. (2017), “Empowering Requirements Elicitation for Populations with Special Needs by Using Virtual Reality”, *ACM SE ‘17 Proceedings of the SouthEast Conference*, Kennesaw, GA, USA, April 13-15, 2017, ACM, New York, pp. 268-270. http://doi.acm.org/10.1145/3077286.3078467

Bhimani, A. (2017), Feasibility of Using Virtual Reality in Requirements Elicitation Process, [Master Thesis], Kennesaw State University.

Brenner, W. and Uebernickel, F. (2016), Design Thinking for Innovation: Research and Practice, 1st ed., Springer International Publishing, Cham.

Brown, R. et al. (2015), “Virtual Business Role-Play: Leveraging Familiar Environments to Prime Stakeholder Memory During Process Elicitation”, *27th International Conference, CAiSE 2015*, Stockholm, Sweden, June 8-12, 2015, Springer International Publishing, pp. 166-180. https://doi.org/10.1007/978-3-319-19069-3_11

Brown, R., Harman, J. and Johnson, D. (2017), “Improved Memory Elicitation in Virtual Reality: New Experimental Results and Insights”, *16th IFIP TC 13 International Conference*, Mumbai, India, September 25-29, 2017, Springer International Publishing, pp. 128-146, https://doi.org/10.1007/978-3-319-67684-5_9

Brown, J.S., Collins, A. and Duguid, P. (1989), “Situated Cognition and the Culture of Learning”, *Sage Journals*, Vol. 18 No. 1, p. 11. https://doi.org/10.3102/0013189X018001032

Coulin, C. and Zowghi, D. (2005), “Requirements Elicitation: A Survey of Techniques, Approaches, and Tools”, Engineering and Managing Software Requirements, Springer, Berlin Heidelberg, pp. 19-46. https://doi.org/10.1007/3-540-28244-0_2

Dumitrescu, R. et al. (2013), “Automatic Verification of Modeling Rules in Systems Engineering for Mechatronic Systems”, *ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, Portland, Oregon, USA, August 4-7, 2013, ASME, New York, p. 9. https://doi.org/10.1115/DETC2013-12330

Dori, D. (2016), Model-Based Systems Engineering with OPM and SysML, Springer, New York. https://doi.org/10.1007/978-1-4939-3295-5

Florides, C. et al. (2015), “Driving simulator for discovering requirements in complex systems”, *SummerSim ‘15 Proceedings of the Conference on Summer Computer Simulation*, Illinois, Chicago, July 26-29, 2015, pp. 1-10.

Gausemeier, J., Ramming, F.J. and Schäfer, W. (2014), Design Methodology for Intelligent Technical Systems: Develop Intelligent Technical Systems of the Future, Springer, Berlin/Heidelberg. https://doi.org/10.1007/978-3-642-45435-6

Japs, S. (2020), Prototypical implementation of 3D Engineer, [online] Japs, S. Available at: https://gitlab.cc-asp.fraunhofer.de/mbseguy/3d_engineer (accessed 10.02.2020).

Lewrick, M., Patrick, L. and Leifer, L. (2018), The Design Thinking Playbook: Mindful Digital Transformation of Teams, Products, Services, Businesses and Ecosystems, John Wiley and Sons, Hoboken, New Jersey.

No Magic (2019), Cameo Systems Modeler, [online] No Magic. Available at: https://www.nomagic.com/products/cameo-systems-modeler (accessed 02.11.2019).

OMG (2015), System Modeling Language V.1.4, OMG, Object Management Group, Needham, Massachusetts, USA.

Smart Mechatronics (2020), CONSENS, [online] Smart Mechatronics. Available at: https://smartmechatronics.de/consens (accessed 30.01.2020).

Tekaat, J. et al. (2019), “Potentials for the Integration of Design Thinking along Automotive Systems Engineering Focusing Security and Safety”, *Proceedings of the 22nd International Conference on Engineering Design (ICED19)*, Delft, The Netherlands, 5-8 August 2019. https://doi.org/10.1017/dsi.2019.295

Tomita, Y. et al. (2017), “Applying design thinking in systems engineering process as an extended version of DIKW model”, *Proceedings of the 27th Annual INCOSE International Symposium (IS 2017)*, Adelaide, Australia, July 15-20, 2017.

UNITY Innovation Alliance (2020), Project References of the UNITY Innovation Alliance, [online] UNITY Innovation Alliance, available at: https://www.unity-innovation-alliance.com/en/ (accessed 30.01.2020).

Unity Technologies (2019a), Unity Asset Store, [online] Unity Technologies. Available at: https://assetstore.unity.com (accessed 02.11.2019).

Unity Technologies (2019b), Unity Game Engine, [online] Unity Technologies. Available at: https://unity.com (accessed 02.11.2019).

https://doi.org/10.1017/dsd.2020.41 Published online by Cambridge University Press