Towards Causal Model-Based Engineering in Automotive System Safety

Robert Maier(✉), Lisa Grabinger, David Urlhart, and Jürgen Mottok

Regensburg University of Applied Sciences, 93053 Regensburg, Germany
{robert.maier,lisa.grabinger,david.urlhart, juergen.mottok}@oth-regensburg.de

Abstract. Engineering is based on the understanding of causes and effects. Thus, causality should also guide the safety assessment of complex systems such as autonomous driving cars. To ensure the safety of the intended functionality of these systems, normative regulations like ISO 21448 recommend scenario-based testing. An important task here is to identify critical scenarios, so-called edge and corner cases. Data-driven approaches to this task (e.g. based on machine learning) cannot adequately address a constantly changing operational design domain. Model-based approaches offer a remedy – they allow including different sources of knowledge (e.g. data, human experts) into safety considerations. With this paper, we outline a novel approach for ensuring automotive system safety. We propose to use structural causal models as a probabilistic modelling language to combine knowledge about an open-context environment from different sources. Based on these models, we investigate parameter configurations that are candidates for critical scenarios. In this paper, we first discuss some aspects of scenario-based testing. We then provide an informal introduction to causal models and relate their development lifecycle to the established V-model. Finally, we outline a generic workflow for using causal models to identify critical scenarios and highlight some challenges that arise in the process.

Keywords: SOTIF · Causality · Probabilistic reasoning · Model-based engineering · Scenario identification

1 Introduction

"A picture is worth a thousand words", this saying best describes the idea behind Model-based Testing (MBT) [16]. In MBT, it is essential to abstract a system, process, or any other part of reality in a structured and comprehensible manner. This can be achieved with models. They provide a common language for communicating assumptions, relationships, and concepts among experts [5,16]. MBT is assumed to increase both, the efficiency and effectiveness of test case specification, through a high degree of automation. It also benefits from the fact that models can be reused or split into sub-models. This allows to add further levels of detail iteratively while remaining transparent and modular.

In the automotive industry, the increasing complexity of modern vehicles challenges both, engineers and authorities. One of these challenges is how to ensure system safety. There are standards for managing Functional Safety (FS) [10] or the Safety Of The Intended Functionality (SOTIF) [11], but their application in practice is not trivial. In SOTIF, safety assurance is based upon scenarios. Creating, detecting, or specifying a sufficient set of those scenarios is an active area of research [1,8,13,17–19,26]. The difficulty lies primarily in the open-context environment that comes with real-life situations. A test set of scenarios should include so-called corner and edge cases, critical combinations of environmental and vehicle conditions. Discovering them by chance from real-world test drive data is very unlikely. This creates the need for a new approach to uncover scenarios, which takes multiple sources of knowledge into account.

MBT is very useful for describing and managing test cases during the development of a system. Nevertheless, it is missing some capabilities required for scenario-based approaches. One such capability is being able to reason under uncertainty. Probabilistic reasoning makes it possible to partially compensate for imperfect knowledge about an Operational Design Domain (ODD) by providing likelihoods for a conclusion. Vice versa, statistical learning uses data generated by real systems to build or improve models [22]. Probabilistic reasoning and statistical learning, show that models do not only provide a compact representation of a system. Instead, they also serve to gain valuable insights into a system. Note that, in both cases, human experts are needed to specify goals or provide missing domain knowledge.

An implicit assumption for building most models is causality. Many interactions among random variables encoded in a model can be interpreted as cause-effect pairs. Depending on the use case, these relationships represent triggering conditions or temporal dependencies. If models obey causality, the insights derived from them also become causal. This is crucial in the context of ensuring system safety, where one of the main interests is to gain causal insights, e.g. about causes of malfunctioning behaviour, triggering events, or the nature of influences between system components.

If the relationships in a model are deterministic and stochastic, it can be framed as a Probabilistic Graphical Model (PGM) [12,20]. Throughout many domains, Bayesian Networks (BNs) are the most widely used PGMs. As [3,20] show, BNs can also represent causality and can be linked to a more generic representation called Structural Causal Models (SCMs) [20]. Both approaches are based on sound formalism and an intuitive interpretation of the underlying formal concepts, such as independence among variables.

In this paper, we propose a novel approach for ensuring automotive system safety: causal model-based engineering. We use SCMs to specify an ODD and derive scenario parameters for testing a system according to SOTIF. The paper is structured as follows: we start by reviewing selected contributions to scenario-based testing. Thereby, we take a closer look at problems that arise in the open-context domain of automotive system safety. Next, we focus on the theoretical foundation of our method. We present how causal probabilistic models are built and what constitutes an SCM. With that, we create an intuition for causal model-based engineering in the automotive area. We proceed by discussing several touchpoints of our method with FS and SOTIF and give a high-level view of a modelling process. After outlining a workflow for identifying edge and corner case scenarios, we point out some challenges in implementing our approach in practice. Finally, we conclude with a summary as well as suggestions for future research.

2 Related Work

The development of new techniques for (partial) autonomous driving vehicles has made enormous progress in the last decade. This is attributable mostly to the extensive use of software and, more recently, Machine Learning (ML) methods. It is possible to build a vehicle operating on state-of-the-art technology. To actually deploy such a system, it has to be proven to be functionally safe [10]. As autonomous driving cars face a versatile and constantly changing environment [13], the plain validation of technical requirements is not sufficient. In response, ISO 21448 proposes the concept of desired and predefined intended functionality and advocates scenario-based testing to complement requirements-based testing [11]. This renders the definition of appropriate scenarios a key element for many verification and validation approaches.

2.1 Terminology of Scenarios

Bagschik et al. [2] suggest dividing the level of detail of scenarios into three categories. Functional scenarios describe the objects and environmental conditions considered in a scenario – its parameters. Logical scenarios enhance the semantic definition of these parameters by specifying their potential values. Finally, concrete scenarios constitute a specific instantiation of the parameter space defined by a logical scenario. The particular scenario parameters can be linked to different layers of abstraction as discussed in [1,24,28].

Based on a test objective, scenarios can be further categorized into common, edge, and corner cases [9,14]. The latter two are usually distinguished by their predictability. While edge cases are rare but often known in advance (e.g. boundary values), corner cases depict unforeseen combinations of several parameters with non-extreme values. In the context of SOTIF, it is necessary to consider all three scenario categories. Whereas common scenarios can be easily captured by a test drive or requirements, the specification of edge and corner cases is more challenging.

In practice, scenarios are represented in a suitable data format such as traffic sequence charts [6], ontologies [18], or object-oriented classes [25]. Due to their widespread use in industry, the *ASAM e. V.* standardized exchange formats, OpenDRIVE[1] for managing road networks, and OpenSCENARIO[2] for describing dynamic properties of scenario entities, are of particular interest.

[1] https://www.asam.net/standards/detail/opendrive/.
[2] https://www.asam.net/standards/detail/openscenario/.

2.2 Sources of Knowledge

A lot of work in scenario-based testing focuses on ML techniques and available real-life data [4,8,9,26]. These approaches provide promising results, yet they do not meet the challenge of an open-context environment. If the available data are too sparse (in either scope or diversity), edge and corner cases can hardly be detected. Moreover, the identified scenarios may only be valid for existing systems. When new vehicles with different sensors or algorithms are deployed, the available data may become obsolete. In other words: data alone is not sufficient for scenario-based testing.

Neurohr et al. [19] outline, that a reliable safety-case argumentation should involve human experts and data-driven methodologies. Experts are well suited to specify causal relationships between scenario parameters on a qualitative level. These informal descriptions can then be structured using linguistic methods such as ontologies [1]. In contrast, data constitutes an objective source for a reliable, quantitative system parametrization. Sampling, clustering, or counterexample selection among many others are established methods for drawing information from data [23]. As described before, due to a highly complex ODD, recorded data will lack information, which can be provided by experts.

Only by relying on both, data and domain experts, a comprehensive test database can be created. Ideally, these two sources of knowledge should be consulted iteratively. As either semantic frameworks or plain data suffer from including complementary information, a new methodology for describing a large and constantly evolving ODD is needed.

3 Causal Models

All model-based approaches use *some* notation as a common language to convey information about a system. In PGMs, this information consists of a set of random variables, their stochastic information (e.g. probability distribution) as well as their probabilistic relationships graphically represented as nodes and edges [12,20]. PGMs like BNs not only provide a graphical notation, but also define a mathematical framework for making use of the structural information encoded in the model.

3.1 Terminology of Causal Models

The edges in a BN represent relationships of random variables. Thereby, they encode additional information, such as dependencies and (conditional) independencies among model elements. If two random variables are directly connected by an edge, they are dependent. For non-adjacent nodes, the structure of subgraphs (so-called junctions, i.e. chains, forks, and colliders) needs to be considered. The corresponding concept is referred to as d-separation [7]. In short: the random variables in a BN influence each other based on the respective graph topology. If data is consistent with this topology, dependence between random variables found as correlation can also be identified graphically (i.e. d-separation). Any inference (i.e. calculating probabilistic information) in BNs is based on Bayes' rule [12]. Independencies among random variables may simplify inference. Because of the link between graph topology and independencies, the joint probability distribution $P(\mathbf{X})$ entailed by the model may be factorized as a special case of the chain rule of probability as shown in formula 1. The operator pa_i ("parents of the variable x_i") refers to all random variables, which are directly connected to x_i by an edge pointing into x_i in the graph.

$$P(\mathbf{X}) = \prod_i P(x_i|pa_i) \tag{1}$$

BNs can be created by algorithms working on data [27] or by human experts [15]. In the case of algorithmic model creation, the link between (in)dependency assumptions in the data and their interpretation as junctions is exploited (i.e. d-separation).

The graphical representation of a BN is built upon directed edges and represents a Directed Acyclic Graph (DAG). Whereas BNs by definition only model associational relationships [3], they can also be used for modelling cause-effect relations. In this case, the directed edges are given a causal interpretation [20] and their graphical representation is called a causal diagram. Stochastic information linking parent nodes to their children (i.e. the configuration of the parent states affects the resulting conditional distribution of the child node) is then considered as a probabilistic causal mechanism [3,20].

SCMs formalize a causal mechanism between a random variable x_i and its parents pa_i by so-called structural equations $x_i := f_i(pa_i, u_i)$ [3,20]. A formal definition of SCMs can be given as:

Definition 1 (Structural Causal Model (SCM) [20, p. 203]). *A SCM is a triple $M = \langle \boldsymbol{U}, \boldsymbol{V}, \boldsymbol{F} \rangle$ where:*

1. $\mathbf{U}$ *is a set of background variables (also called exogenous), that are determined by factors outside the model;*
2. $\mathbf{V}$ *is a set $\{V_1, V_2, ..., V_n\}$ of variables, called endogenous, that are determined by variables in the model – that is, variables in $\mathbf{U} \cup \mathbf{V}$; and*
3. $\mathbf{F}$ *is set of functions $\{f_1, f_2, ..., f_n\}$ such that each f_i is a mapping from (the respective domains of) $U_i \cup PA_i$ to V_i, where $U_i \subseteq \mathbf{U}$ and $PA_i \subseteq \mathbf{V} \setminus V_i$ and the entire set $\mathbf{F}$ forms a mapping form $\mathbf{U}$ to $\mathbf{V}$. In other words, each f_i in $v_i = f_i(pa_i, u_i), \quad i = 1, ..., n$,assigns a value to V_i that depends (on the values of) a selected set of variables in $\mathbf{V} \cup \mathbf{U}$, and the entire set $\mathbf{F}$ has a unique solution $V(u)$.*

Figure 1 exemplary visualizes the graphical notation of a causal diagram: three junctions, an exemplary DAG, and its corresponding representation by structural equations.

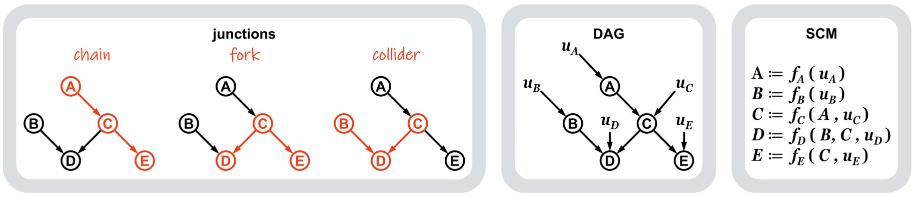

Fig. 1. On the left side, the junctions chain, fork, and collider are depicted. In the middle, an exemplary causal diagram (i.e. a BN structure) is shown including exogenous influences u_i of each random variable. On the right side, this diagram is represented by its structural equations.

3.2 Inference in Causal Models

The probabilistic definition of a system (e.g. by an SCM) together with the causal diagram (implied by this definition) forms a causal model. Causal models define how the probability distributions of random variables change, if causal mechanisms are modified. With that, causal models enable causal reasoning (i.e. estimating causal effects of random variables among each other) [3, 20].

Depending on the type of model used (e.g. BN, causal BN, or SCM), different levels of causal expressiveness [3] can be addressed. This is commonly pictured as the "ladder of causation" [21]. The three distinct levels are defined as associational (i.e. insights gathered by "seeing"), interventional (i.e. insights gathered by "doing"), and counterfactual (i.e. insights gathered by "imagining") reasoning. For our application, results gained by counterfactual queries are of great interest. Counterfactual metrics such as the probability of necessity (PN) or the probability of sufficiency (PS) allow distinguishing the *kind* of cause-effect relationship between (non-adjacent) random variables based on a likelihood [20, p. 283].

4 Causal Models and Scenario-Based Testing

In the following, we examine the link between the normative regulations ISO 26262 and ISO 21448, causal models, and scenario-based testing. Moreover, we present a possible approach for deriving corner and edge cases.

4.1 Models in Automotive Safety Engineering

The term *model* can be interpreted differently depending on the context of use. In model-based design, models specify the functional properties of a system and therefore are the central element of a development cycle. In software development, models are often associated with MBT. There, one of the central goals "is to formalize and automate as many activities related to test case specification as possible" [16, p. 2]. In automotive system safety, models are often used to describe the dynamic behaviour of a component (e.g. Markov-Model, Petri Net)

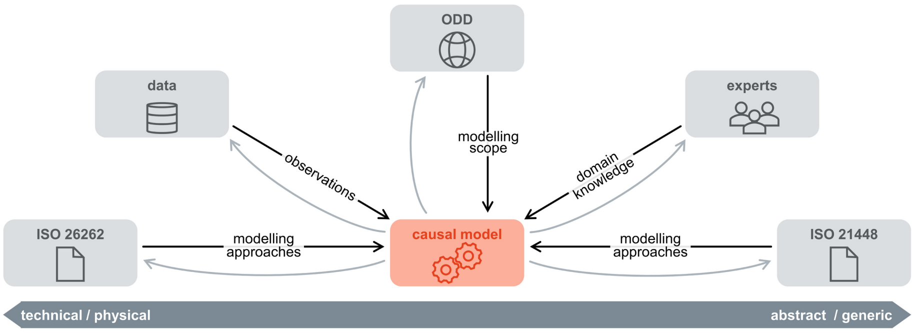

Fig. 2. A causal model interacts with different knowledge bases: data, experts, and normative regulations such as ISO 26262 or ISO 21448. Their modelling approaches and promoted artefacts contribute to building a model of an ODD. On the other hand, the knowledge bases benefit from the causal insights gained through the model.

or a logical connection of events (e.g. Fault Tree Analysis (FTA), Event Tree Analysis (ETA)). Thereby, the focus is on decomposing a complex system or process into tractable components or events and their interactions. Across all areas, models can be viewed as a standardized way of describing a use case. As a common language (i.e. visual representation and/or mathematical specification), they simplify and improve engineering.

While standards like ISO 26262 rely on requirement-based testing and analysis methods such as Hazard Analysis and Risk Assessment (HARA) or Failure modes, effects, and diagnostic analysis (FMEDA), ISO 21448 advocates giving attention specifically to the ODD. Systems are typically designed, developed, and tested with an ODD in mind. Yet, there is no model-based framework for explicitly specifying elements of an ODD and their interactions. Following the considerations in Sect. 2, such a model-based framework should take into account different sources of knowledge and allow for an iterative construction and continuous adaptation of models. Causal models as described in Sect. 3 meet those demands.

Figure 2 visualizes the different knowledge bases that interact with a causal model. These interactions are two-sided. Data, expert knowledge, ODD specifications, and artefacts from standard-encouraged analysis approaches (e.g. results of an FMEDA) can help to build a causal model. A model may then be used as an approximation of an ODD, or provide insights to improve data collection, fault management, or the design of associated systems.

4.2 Development of Causal Models

Causal models follow an iterative development lifecycle [15], as outlined in Fig. 3. This process can be roughly divided into the following steps:

Desired Insight (step 1): The intended use of the causal model is specified.

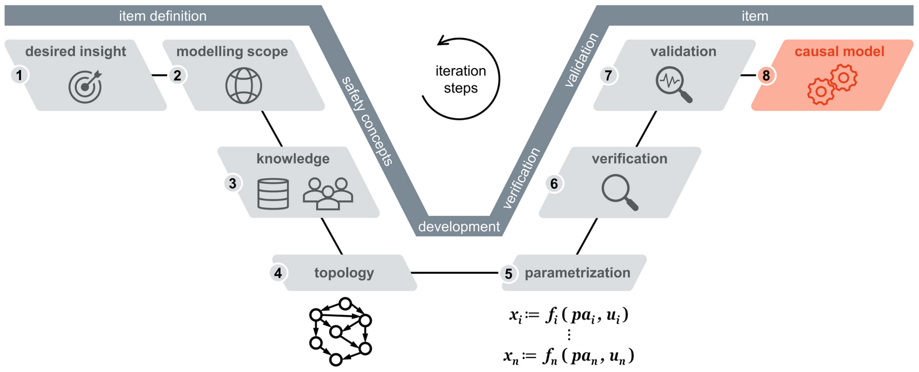

Fig. 3. On a high-level view, the iterative development lifecycle for causal models can be divided into distinct steps. The outlined process can be related to the established V-model (grey envelope) approach as specified by ISO 26262 and ISO 21448.

Modelling Scope (step 2): Depending on the use case, the required expertise of domain experts and potential data sources are identified. Target variables (i.e. effects of interest) are identified.

Knowledge (step 3): Domain experts are introduced to the use case and the basic concepts of causal modelling. This includes organizing an introductory session, creating a first model draft through guided creativity techniques (e.g. brainstorming), and enabling access to relevant data. As mentioned in Sect. 4.1, many artefacts of normative modelling approaches (e.g. results of an FMEDA, HARA, or FTA) might be re-used as sources of knowledge. Already generated insights can speed up the process of causal model creation. Moreover, they allow uncovering gaps in the system and ODD specifications.

Topology (step 4): Relevant random variables, their properties (e.g. value types, units, or ranges), and their relationships with each other are identified. A causal diagram representing this information is created either through expert elicitation, algorithmic approaches [27], or a combination of both.

Parameterization (step 5): The causal diagram is enhanced by the probability distributions of its random variables and by structural equations as a formalization of causal mechanisms. This is done based on either expert knowledge, data, or a combination of both. Since any change in the model topology requires a re-evaluation of the parameters, the causal diagram should be developed as far as possible before starting the current step.

Verification (step 6): The current version of the causal model is tested for internal consistency. This includes checking the (in)dependence assumptions implied by the model against the data and comparing inference estimates to expected results.

Validation (step 7): The current version of the causal model is tested for its fit to the use case. With regard to the intended use, the completeness and level of detail are examined.

Causal Model (step 8): Based on the findings of the verification and validation step, a new iteration of the entire process may be triggered. When a causal model adequately addresses a use case, it becomes the artefact of the development lifecycle and is ready for productive use.

In the automotive industry, an established development process is given by the V-model [10,11]. Creating causal models as described above can be easily adapted to fit into this established scheme, as suggested in Fig. 3.

4.3 Towards Discovering Edge and Corner Cases

As described in Sect. 2, identifying edge and corner cases is important for scenario-based testing. A common approach to find suitable scenario candidates is to use ML methods [9,29]. Due to their data-driven nature, three major problems arise. First, because of an open-context environment, a full description of an ODD is not possible. Second, even with a constrained model, the number of potentially relevant combinations of scenario parameters is intractable. Finally, the available data may not be suited to the use case: for example, real-world test drives carried out by human drivers may encounter critical situations, which are not necessarily relevant for autonomous systems. Those problems already show that purely data-driven methods are not enough to identify edge and corner cases. Instead, different sources of domain knowledge are to be used.

As argued before, SCMs constitute a suitable framework for modelling an ODD from different knowledge sources. Above that, SCMs allow for causal reasoning (e.g. computing counterfactual estimates such as PS and PN) [20]. Recall that those causal metrics can provide information about the causal relationships of random variables in the model. This allows exploring, which random variables (i.e. scenario parameters) influence a variable of interest (i.e. the effect we care about).

Edge or corner cases are not limited to one influencing factor, but can result from a combination of two or more parameters. Moreover, only certain value combinations of contributing parameters may result in a critical scenario. Because of that, we need to consider each configuration of parameters (tuple) in the model. By using causal metrics, all tuples and their respective parameter spaces are evaluated. The resulting hot spots (i.e. local and global extrema) can be interpreted as candidate areas for edge or corner case parameter combinations. Figure 4 builds intuition for the overall process.

To better understand the potential of our approach, suppose we are interested in whether a sensor system can detect an obstacle or not. Based on domain knowledge, we can build a causal model that describes the impact of environmental effects, occlusions, or hardware failure rates on this system. Finding a critical scenario (i.e. detection failed) amounts to finding a configuration of model parameters that causally affect the modelled detection capability.

The presented approach can be framed by the engineering process shown in Fig. 5. The individual steps are defined as follows:

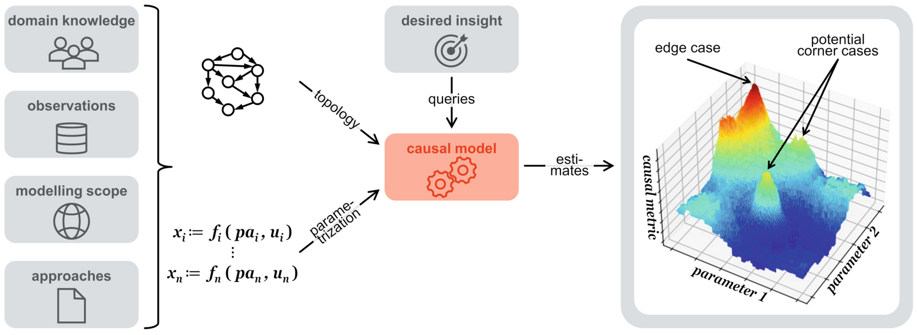

Fig. 4. A fully specified causal model (i.e. an SCM) is queried with regard to a parameter of interest (e.g. ability to detect an obstacle). The estimates of these queries relate to a parameter space for every combination of parameters. Hot spots in these parameter spaces can be interpreted as candidates for edge or corner cases.

Causal Model (step 1): Based on a use case (e.g. investigating detection issues), a relevant part of the related ODD is modelled in an iterative process (see Sect. 4.2) or an already available causal model (i.e. an SCM) is selected.

Target Variable (step 2): According to the present use case, a suitable random variable (i.e. the parameter of interest) is chosen. This random variable will serve as a query target, for which all causal metrics are evaluated.

Tuples (step 3): A sub-set of random variables (i.e. parameter tuple) is chosen based on the causal diagram and the testing scopes. By selecting the parameter combinations to be considered, we implicitly specify functional scenarios. Since an underlying SCM assigns value ranges for the selected parameters, we additionally gain potential logical scenarios [2].

Causal Metrics (step 4): Causal metrics like PN and PS are computed for the selected parameter tuples (i.e. potential causes) with regard to the selected target variable (i.e. effect of interest).

Relevant Tuples (step 5): The evaluated tuples are post-processed. This includes the detection of edge and corner cases (see Fig. 4) as well as the evaluation of the involved parameters in the context of safety (e.g. SOTIF relevance). Critical candidates implicitly describe concrete scenarios, as they represent a specific instantiation of the modelled parameter space.

In the automotive domain, elements of a scenario (as defined by [11]) can be broken down into several layers [2]. Each layer comprises different scenario entities, ranging from road networks to environmental influences. Various exchange formats for automated testing (e.g. [18,25]) build on this decomposition. Ideally, we want to provide scenario parametrizations for use in e.g. simulations. The results obtained with the above methodology must therefore remain compatible with the layer decomposition. This can be achieved by adjusting the modelling scope.

In general, the random variables in a causal model can be specified arbitrarily. Because of that, parameter configurations discovered from causal models can

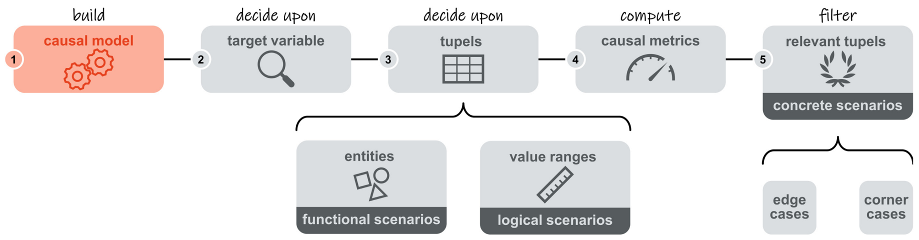

Fig. 5. The causal model-based scenario parameter detection can be represented as an engineering process. Relevant parameter tuples (i.e. sub-sets of random variables) are evaluated and post-processed. Several intermediate process artefacts can be related to functional, logical, or concrete scenarios.

usually not be linked directly to a specific exchange format (e.g. OpenDRIVE or OpenSCENARIO). This is in large part a semantic problem. To solve it, causal models need to be developed with the scenario layer entities in mind.

As exchange formats usually define a set of parameters, it is enough to simply include those defined parameters as required building blocks of the causal model. The parameter configurations resulting from such scenario-centric causal models can then be mapped automatically to a common scenario exchange format.

With that, our approach bridges the gap to model-based testing by providing a model-driven parametrization for executable scenarios (e.g. for simulation). This allows a joint evaluation and verification of a system. Figure 6 depicts the resulting adjustment of the engineering process with a focus on executable scenario exchange formats.

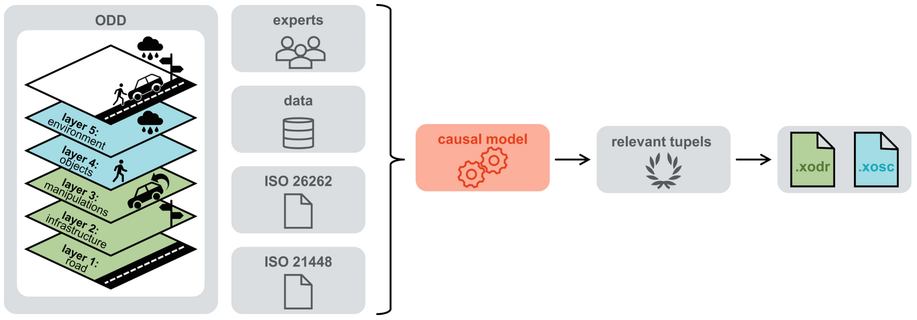

Fig. 6. In the automotive domain, scenario entities can be categorized into several layers [2]. To allow scenario-based testing according to [11], causal models need to include those entities. This allows mapping generated results to established exchange formats, which can then be used for system simulation and verification.

5 Conclusion

Causal models can be used to abstract an ODD. When they are described with probabilistic frameworks like SCMs, causal inference is enabled. Various sources of information (e.g. data, experts) can be jointly used for the creation of causal models, compensating for the limitations of using a sole knowledge source. As the associated development lifecycle can be related to the V-model, causal model-based engineering can be integrated into established industrial processes.

Scenario-based testing is considered an integral part of evaluating SOTIF. Scenarios can be described and abstracted in different ways. Besides the demand to cover a wide range of common situations (e.g. by a scenario database), critical scenarios are of great interest. Discovering such critical scenarios (i.e. edge and corner cases) is non-trivial.

In this paper, we outline an approach for identifying edge and corner cases based on causal models. We propose to use causal metrics such as the probability of necessity or the probability of sufficiency to identify critical parameter configurations. Ideally, discovered candidate tuples can serve as an input for simulation environments. This requires that the causal model itself contains scenario-centric entities. Once these random variables match with the parameters defined in model exchange formats like OpenDRIVE or OpenSCENARIO, an automated mapping of results becomes feasible.

When using causal models, the common problem of test case explosion based on potential parameter combinations remains. Even though concepts like d-separation seem promising to restrict some combinatorics, an evaluation of candidate tuples still suffers from the curse of dimensionality. Furthermore, in our approach, the decision for or against critical candidate scenarios is based on thresholds for causal metrics. The exact definition of these thresholds is an open research topic. Above that, in the context of automotive systems, additional safety considerations (e.g. severity, exposure, controllability, or potential resulting loss) of contributing factors must be taken into account. Additional research is needed to refine the proposed process with regard to these open challenges.

With this paper, we hope to encourage researchers to participate in the development of causal model-based engineering – a methodology where all types of knowledge contribute to a better understanding of safety issues.

Acknowledgment. The present paper is supported by *Bayerisches Staatsministerium für Wirtschaft, Landesentwicklung und Energie* through the granting of the funding project *HolmeS*^{3} (FKZ: DIK0173/03). Moreover, we thank Daniel Ebenhöch from *e:fs TechHub GmbH*, Tilo Linz from *imbus AG*, and their respective team members for valuable insights and discussions.

References

1. Bagschik, G., Menzel, T., Maurer, M.: Ontology based scene creation for the development of automated vehicles. In: 2018 IEEE Intelligent Vehicles Symposium (IV), pp. 1813–1820 (2018). https://doi.org/10.1109/IVS.2018.8500632

2. Bagschik, G., Menzel, T., Reschka, A., Maurer, M.: Szenarien für Entwicklung, Absicherung und Test von automatisierten Fahrzeugen. In: 11th Workshop Fahrerassistenzsysteme (Uni-DAS e. V.), pp. 125–135 (2017)
3. Bareinboim, E., Correa, J.D., Ibeling, D., Icard, T.: On Pearl's hierarchy and the foundations of causal inference, 1st edn., pp. 507–556. Association for Computing Machinery, New York (2022). https://doi.org/10.1145/3501714.3501743
4. Bogdoll, D., et al.: Description of corner cases in automated driving: goals and challenges. In: 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Montreal, BC, Canada, pp. 1023–1028. IEEE (2021). https://doi.org/10.1109/ICCVW54120.2021.00119
5. Bringmann, E., Kramer, A.: Model-based testing of automotive systems. In: 2008 1st International Conference on Software Testing, Verification, and Validation, pp. 485–493 (2008). https://doi.org/10.1109/ICST.2008.45
6. Damm, W., Kemper, S., Möhlmann, E., Peikenkamp, T., Rakow, A.: Using traffic sequence charts for the development of HAVs. In: European Congress on Embedded Real Time Software and Systems 2018. 9th European Congress on Embedded Real Time Software and Systems (ERTS 2018) (2018). https://hal.archives-ouvertes.fr/hal-01714060
7. Geiger, D., Verma, T., Pearl, J.: d-Separation: from theorems to algorithms. In: Machine Intelligence and Pattern Recognition, vol. 10, pp. 139–148. Elsevier Science Inc. (1990). https://doi.org/10.1016/B978-0-444-88738-2.50018-X
8. de Gelder, E., et al.: Scenario parameter generation method and scenario representativeness metric for scenario-based assessment of automated vehicles. arXiv:2202.12025 [cs] (2022)
9. Heidecker, F., et al.: An application-driven conceptualization of corner cases for perception in highly automated driving. In: 2021 IEEE Intelligent Vehicles Symposium (IV), pp. 644–651 (2021). https://doi.org/10.1109/IV48863.2021.9575933. arXiv:2103.03678
10. ISO Central Secretary: Road vehicles - Functional safety. Standard ISO 26262-2:2018, International Organization for Standardization, Geneva, CH (2018)
11. ISO Central Secretary: Road vehicles - Safety of the intended functionality. Standard ISO 21448:2022, International Organization for Standardization, Geneva, CH (2022)
12. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. Adaptive Computation and Machine Learning. MIT Press, Cambridge (2009)
13. Koopman, P., Fratrik, F.: How many operational design domains, objects, and events? In: SafeAI@AAAI (2019)
14. Koopman, P., Kane, A., Black, J.: Credible autonomy safety argumentation. In: 27th Safety-Critical Systems Symposium (2019)
15. Korb, K.B., Nicholson, A.E.: Bayesian Artificial Intelligence, 2nd edn. CRC Press Inc., (2010). https://doi.org/10.1201/b10391
16. Kramer, A., Legeard, B.: Model-Based Testing Essentials: Guide to the ISTQB® Certified Model-Based Tester Foundation Level. Wiley, Hoboken (2016). https://doi.org/10.1002/9781119130161
17. Kramer, B., Neurohr, C., Büker, M., Böde, E., Fränzle, M., Damm, W.: Identification and quantification of hazardous scenarios for automated driving. In: Zeller, M., Höfig, K. (eds.) IMBSA 2020. LNCS, vol. 12297, pp. 163–178. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58920-2_11

18. Menzel, T., Bagschik, G., Isensee, L., Schomburg, A., Maurer, M.: From functional to logical scenarios: detailing a keyword-based scenario description for execution in a simulation environment. In: 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, pp. 2383–2390. IEEE (2019). https://doi.org/10.1109/IVS.2019.8814099
19. Neurohr, C., Westhofen, L., Henning, T., de Graaff, T., Möhlmann, E., Böde, E.: Fundamental considerations around scenario-based testing for automated driving. In: 2020 IEEE Intelligent Vehicles Symposium (IV), pp. 121–127 (2020). https://doi.org/10.1109/IV47402.2020.9304823
20. Pearl, J.: Causality: Models, Reasoning and Inference, 2nd edn. Cambridge University Press (2009)
21. Pearl, J., Mackenzie, D.: The Book of Why: The New Science of Cause and Effect, 1st edn. Basic Books Inc. (2018)
22. Peters, J., Janzing, D., Schölkopf, B.: Elements of Causal Inference - Foundations and Learning Algorithms. Adaptive Computation and Machine Learning Series. The MIT Press, Cambridge (2017)
23. Riedmaier, S., Ponn, T., Ludwig, D., Schick, B., Diermeyer, F.: Survey on scenario-based safety assessment of automated vehicles. IEEE Access **8**, 87456–87477 (2020). https://doi.org/10.1109/ACCESS.2020.2993730
24. Schuldt, F.: Ein Beitrag für den methodischen Test von automatisierten Fahrfunktionen mit Hilfe von virtuellen Umgebungen, Ph.D. dissertation (2017). https://doi.org/10.24355/dbbs.084-201704241210
25. Steimle, M., Menzel, T., Maurer, M.: Toward a consistent taxonomy for scenario-based development and test approaches for automated vehicles: a proposal for a structuring framework, a basic vocabulary, and its application. IEEE Access **9**, 147828–147854 (2021). https://doi.org/10.1109/ACCESS.2021.3123504
26. Steimle, M., Weber, N., Maurer, M.: Toward generating sufficiently valid test case results: a method for systematically assigning test cases to test bench configurations in a scenario-based test approach for automated vehicles. IEEE Access **10**, 6260–6285 (2022). https://doi.org/10.1109/ACCESS.2022.3141198
27. Vowels, M.J., Camgöz, N.C., Bowden, R.: D'ya like DAGs? A survey on structure learning and causal discovery. CoRR abs/2103.02582 (2021)
28. Weber, H., et al.: A framework for definition of logical scenarios for safety assurance of automated driving. Traffic Injury Prev. **20**(sup1), S65–S70 (2019). https://doi.org/10.1080/15389588.2019.1630827. pMID: 31381437
29. Xinxin, Z., Fei, L., Xiangbin, W.: CSG: critical scenario generation from real traffic accidents. In: 2020 IEEE Intelligent Vehicles Symposium (IV), Las Vegas, NV, USA, pp. 1330–1336. IEEE (2020). https://doi.org/10.1109/IV47402.2020.9304609