Extending Causal Models from Machines into Humans

Severin Kacianka Amjad Ibrahim Alexander Pretschner
Technical University of Munich
Munich, Germany
firstname.lastname@tum.de

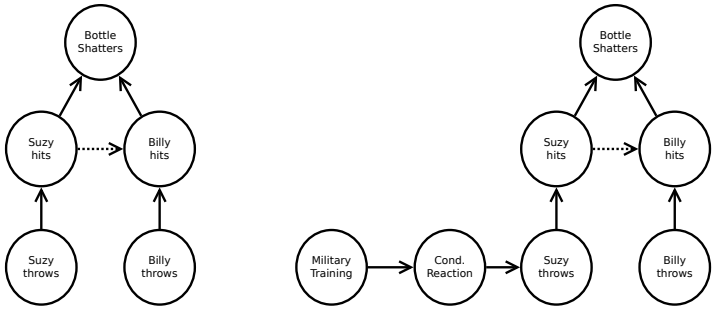
Alexander Trende Andreas Lüdtke
Offis
Oldenburg, Germany
firstname.lastname@offis.de

Causal Models are increasingly suggested as a means to reason about the behavior of cyber-physical systems in socio-technical contexts. They allow us to analyze courses of events and reason about possible alternatives. Until now, however, such reasoning is confined to the technical domain and limited to single systems or at most groups of systems. The humans that are an integral part of any such socio-technical system are usually ignored or dealt with by "expert judgment". We show how a technical causal model can be extended with models of human behavior to cover the complexity and interplay between humans and technical systems. This integrated socio-technical causal model can then be used to reason not only about actions and decisions taken by the machine, but also about those taken by humans interacting with the system. In this paper we demonstrate the feasibility of merging causal models about machines with causal models about humans and illustrate the usefulness of this approach with a highly automated vehicle example.

1 Introduction

Causality is an essential building block to answer complex questions of accountability [14]. It allows us to reason about different courses of actions and go beyond mere correlation based reasoning. While building causal models is still a major problem and technologies to automatically build causal models are still in their infancy, recent research has shown that it is possible to derive useful starting points for models from system descriptions [12]. One can, for example, use fault trees as a starting point for a causal model and then ask experts to refine it. While this still requires human intervention, it is a lot easier to improve models than to create them from scratch. However, most Cyber-Physical Systems (CPS) are embedded into larger Socio-Technical Systems (STS) and interact with people. In current research, automatically generated causal models only encompass the technical aspects of these STS. Models for the interaction between humans and machines are usually created by experts in an ad-hoc fashion. The field of human factors is concerned exactly with this problem (e.g., [27]): how to model the behavior of a human that interacts with a technical system (e.g., a pilot or a driver). The result of this research are different models of human cognition and human behavior. We show how these models can be converted to causal models and linked with the technical causal models. To illustrate the general idea, we use the classic rock throwing thought experiment from the causality literature. In this scenario, two persons, Billy and Suzy, will throw rocks at a bottle. If Suzy's rock hits the bottle slightly before Billy, who is the cause for the bottle to shatter? This example is constructed to show that simple counter factual reasoning is not enough to attribute causality: Had Suzy not thrown the rock, the bottle would still have shattered (because Billy also threw a rock), therefore Suzy cannot be the cause for the bottle to shatter. Finding

G. Caltais and J. Krivine (eds.): CREST 2019
EPTCS 308, 2019, pp. 17–31, doi:10.4204/EPTCS.308.2



(a) Classic version. (b) With military conditioning.

Figure 1: The rock-throwing example.

definitions of causality that will not lead to such counterintuitive results culminated in the Halpern-Pearl definition of causality [10]. It resolves the issues by introducing a so called preemption relation [12] that can express that Suzy's throw preempted Billy's throw (see Figure 1a).

While such a model expresses the objective facts very well, we cannot reason about why Suzy threw faster. To do so, we would need a model of Suzy's mind. If, for example, she was a soldier and, as part of her training, was conditioned to throw rocks the moment a bottle appeared in her field of view[1], we might not simply say that "Suzy throwing the rock caused the bottle to shatter", but extend our causal chain to "Her military training caused Suzy to automatically throw the rock at the bottle, shattering it". Instead of just blaming Suzy, we could also consider her military training.

Returning to CPS and their interaction with humans, we can now utilize existing models of human behavior, as in Figure 1b, transform them to causal models and link them with causal models of the technical systems. Instead of just saying "The car crashed, because the driver pressed the red button", we can say that "Drivers are conditioned to press the red button in an emergency; this lead to the driver pressing the button and the car crashing". While it is true that we might not have enough data in many cases, in the cases where we do have enough data, like the example in Section 3, and can actually extend the causal model into the human mind, we can gain valuable insights and avoid the unsatisfying generic answer "human error".

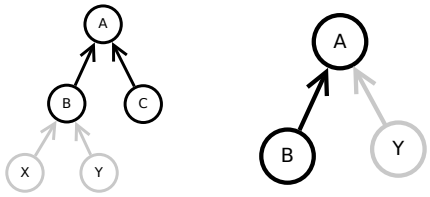
In this paper we investigate the *problem* of joining causal models of technical systems with causal models of their operators and people they interact with. As a *solution* we propose to extend current methods to derive causal models from system models to also include models of the human behavior. Our *contribution* is a detailed example of such an integrated socio-technical causal model based on an automotive use case, preliminary research into a process of converting human models to causal models and showing how these integrated models can be used to reason over accidents.

2 Background

2.1 The Preliminaries for Causal Models and Inference

In this present paper, as in previous work [12], we build on the Halpern and Pearl (HP for short) definition of causality [8, 10, 11, 21]. This definition models the world as two distinct sets, called *endogenous* and

[1] Soldiers are conditioned to shoot at human shaped targets without second thought. This increases their "efficiency" on the battlefield [7].



(a) Refine. (b) Extend.

Figure 2: Merging Causal Models [12].

exogenous variables. Their relation is described with a set of *structural equations*. Exogenous variables model factors outside of the model and endogenous variables model our understanding of the causal relations within our model. The value of endogenous variables is determined by the exogenous variables and the structural equations. [8] defines the conditions under which an endogenous variable can be the cause of a specific state of another variable in the model. Definition 1 provides the formal definition of binary causal models.

Definition 1 ***Binary Causal Model*** *[8]*
A binary causal model M is a tuple $M = ((\mathscr{U},\mathscr{V},\mathscr{R}),\mathscr{F})$*, where*

- $\mathscr{U}$ *is a set of exogenous variables,*
- $\mathscr{V}$ *is a set of endogenous variables,*
- $\mathscr{R}$ *: associates with every variable the nonempty set* $\{0,1\}$ *of possible values,*
- $\mathscr{F}$ *associates with each variable* $X \in \mathscr{V}$ *a function that determines the value of X (from the set of possible values* $\mathscr{R}(X)$*) given the values of all other variables*
  $F_X : (\times_{U\in\mathscr{U}}\mathscr{R}(U)) \times (\times_{Y\in\mathscr{V}-\{X\}}\mathscr{R}(Y)) \to \mathscr{R}(X)$.

A *primitive event*, given $(\mathscr{U},\mathscr{V},\mathscr{R})$, is a formula of the form $X = x$ for $X \in \mathscr{V}$ and $x \in \mathscr{R}(X)$. *A causal formula* is of the form $[Y_1 \leftarrow y_1,\ldots,Y_k \leftarrow y_k]\varphi$, where φ is a Boolean combination of primitive events. $Y_1,\ldots,Y_k$ (abbreviated $\overrightarrow{Y}$) are distinct variables in $\mathscr{V}$, and $y_i \in \mathscr{R}(Y_i)$. Intuitively, this notation says that φ would hold if Y_i were set to y_i for each i. $(M,\overrightarrow{u}) \models X = x$ if the variable X has value x in the unique solution to the equations in M in context $\overrightarrow{u}$. An intervention on a model is expressed either by setting the values of $\overrightarrow{X}$ to $\overrightarrow{x}$, written as $[X_1 \leftarrow x_1,..,X_k \leftarrow x_k]$, or by fixing the values of $\overrightarrow{X}$ in the model, written as $M_{\overrightarrow{X}\leftarrow\overrightarrow{x}}$. So, $(M,\overrightarrow{u}) \models [\overrightarrow{Y} \leftarrow \overrightarrow{y}]\varphi$ is identical to $(M_{\overrightarrow{Y}\leftarrow\overrightarrow{y}},\overrightarrow{u}) \models \varphi$.

Definition 2 ***Actual Cause*** *(latest/modified version [8])*
$\overrightarrow{X} = \overrightarrow{x}$ *is an actual cause of* φ *in* $(M,\overrightarrow{u})$ *if the following three conditions hold:*
AC1. $(M,\overrightarrow{u}) \models (\overrightarrow{X} = \overrightarrow{x})$ *and* $(M,\overrightarrow{u}) \models \varphi$.
AC2. *There is a set* $\overrightarrow{W}$ *of variables in* $\mathscr{V}$ *and a setting* $\overrightarrow{x}'$ *of the variables in* $\overrightarrow{X}$ *such that if* $(M,\overrightarrow{u}) \models \overrightarrow{W} = \overrightarrow{w}$*, then* $(M,\overrightarrow{u}) \models [\overrightarrow{X} \leftarrow \overrightarrow{x}',\overrightarrow{W} \leftarrow \overrightarrow{w}]\neg\varphi$.
AC3. $\overrightarrow{X}$ *is minimal.*

2.2 Joining Causal Models

For joining causal models, we rely on previous work [12]. To briefly summarize the method, it utilizes Alrajeh et al. [1] and Friedenberg and Halpern [5] to automatically combine models that are either

compatibility or extended with a *focus function*. Without going into the technical details here (see [12] for an in-depth discussion), if one model explains more about cause and effect relations than another model and both models talk about the same things, we can simply combine those models. In the simplest case, one model *refines* a leaf node of an existing model (see Figure 2a). In such a case, both models are compatible and can be automatically combined. *Extensions* (see Figure 2b) are unfortunately much more complex. In some scenarios they can be joined automatically, but often they will require the intervention from an expert.

2.3 Technical Source Models

In this paper, we focus on fault and attack trees as technical source models. Fault trees are used in the domain of safety, reliability, and risk assessment methodology [26, 16] and are a means to analyze a system's resilience against faults at design time [22]. They are usually presented graphically as trees that model component faults that might lead to system failure and their relationship among each other. While the exact syntax varies, fault trees definitions usually include *AND*, *OR*, *EXCLUSIVE OR*, *PRIORITY AND* and *INHIBIT* gates.

Attack Trees stem from the field of security research [24, 25] and are used to model potential security threats within a system and possible ways to exploit them. Similar to fault trees, they are graphically represented as trees. The ultimate goal of an attack is the root node of the tree and steps necessary to achieve that goal are notated in sub-nodes. Attack trees usually support *OR* and *AND* gates,

As in previous work [12], we follow Mauw and Oostdijk's [20] formalization of attack trees, but replace their multi-set semantics with Bruns and Anderson's propositional semantics for fault trees [4] and extend them with the semantics for fault trees. This semantic matches the semantics of binary structural equations used in causal models. Therefore, each non-leaf node in the tree is expressed with a propositional formula of its parents, e.g., $out = in_1 \wedge in_2$.

Definition 3 ***Attack/Fault Tree***
$A(F)T$ is a 3-tuple $A(F)T = (\mathcal{N}, \rightarrow, n_0)$ where $\mathcal{N}$ is a finite set of nodes, $n_0 \in \mathcal{N}$ is the root node and $\rightarrow \subseteq \mathcal{N} \times \mathcal{N}$ is a finite set of acyclic relations.

2.4 Human Behavior Source Models

In the following, Hierarchical Task Analysis (HTA) [3] will be explained and how it can be used to create models of human behavior. The HTA is used to partition tasks into sub-tasks and to demonstrate their dependencies. The partition into sub-tasks is done in a way that finishing the sub-tasks leads to completing the main task. Overall the abstract model consists of the main task, sub-tasks, plans and operations and is expressed in a tree diagram in most cases. An integral part of an HTA is the definition of the task and the data collection. Different types of data collection are possible: e.g., questionnaires, experiments, or safety records.

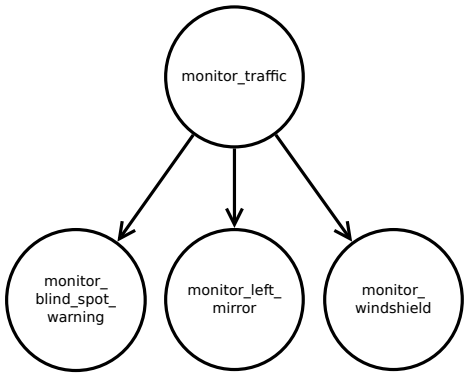


Figure 3: Sub-goal dependencies.

Models of human behavior can be constructed based on the results of an HTA. A first step could be to construct a state chart. This reflects the order and interplay of the previously defined tasks. Afterwards the HTA or state chart could

be translated into a model usable in a cognitive architecture. [15] performed an HTA for lane merging maneuvers onto freeways to define and improve the structure of the cognitive driver model defined in [27]. Cognitive architectures like ACT-R [2] or CASCaS [18] use a rule-based format for their model descriptions. In the domain of cognitive architectures the above described tasks and sub-tasks are called goals and sub-goals. This terminology will be used in the following sections. Sub-goals defined by an HTA can be formulated as procedures in this format. Every time step of the simulation the cognitive cycle will be simulated. In this process a currently active goal is being selected and executed. This will most likely trigger the next sub-goal in the model. We will show the structure and procedure of such a simulation while using four small sub-goals of an HTA: We assume that three sub-goals were defined as part of a model describing a successful merging maneuver: *observe_blind_spot_warning*, *monitor_traffic*, *observe_windshield* and *observe_left_mirror*. The corresponding dependencies between the sub-goals are shown in Figure 3.

This means that the sub-goal *monitor_traffic* can be separated into the sub-goals *observe_blind_spot_warning*, *observe_windshield* and *observe_left_mirror*. Translated into the rule-based format usable in CASCaS:

```
rule(goal=monitor_traffic){
  Condition(boolean expression)
 -->
  Goal(observe_blind_spot_warning)
  Goal(observe_windshield)
  Goal(observe_left_mirror)
}
```

This procedure is executed, if the sub-goal *monitor_traffic* is active. This leads to the addition of the sub-goals *observe_blind_spot_warning*, *observe_windshield* and *observe_left_mirror* to the active sub-goals. At this point *monitor_traffic* is completed and the two newly added sub-goals will be processed in the next simulation step.

One major advantage of cognitive architectures is that they provide a software framework to simulate models of human behavior. The structure and function of these architectures are based on findings from cognitive science. Additionally most cognitive architectures contain several modules like the memory which contains the above described procedures and the declarative memory for facts. Additional modules could be included to simulate human perception and motor control. In the above described example the perception module could also simulate the gaze direction from the windshield to the left mirror and back and the motor module could simulate the hand movement required for steering. Since the time required for such actions is also based on results from cognitive research these simulations can be used to predict the time needed for a given task. Such cognitive driver models comprising different abstraction levels have been investigated in [17], [23] and [19]. Another important application of these cognitive models is the modelling of human error. [19] investigated human error in lane merging tasks and aviation and [18] proposed a cognitive pilot model that simulates the process of Learned Carelessness. Learned Carelessness corresponds to “effort-optimizing shortcuts leading to inadequate simplifications that omit safety critical aspects of normative activities”.

3 Example

As an example, we will consider a freeway merging situation as depicted in Figure 4. The ego vehicle (A) is driving on a freeway ramp with one alter vehicle in front it (C). The distance between the ego

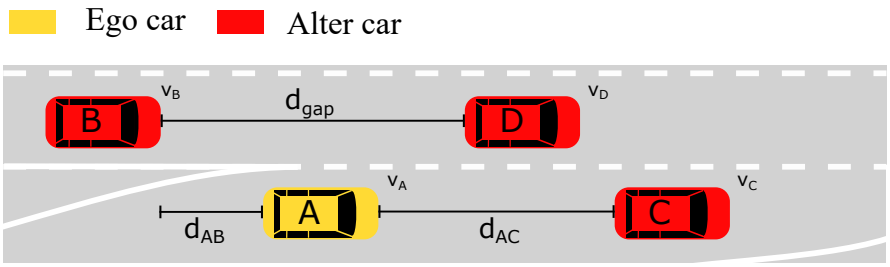


Figure 4: A simple freeway lane merging example. The ego vehicle A wants to merge onto the first lane of the freeway. A has to keep its distance to the car in front of it and simultaneously looking for a gap to merge onto the freeway.

vehicle and C is called d_{AC}. A wants to merge onto the first lane of the freeway. We consider two alter vehicles B and D driving on this lane, whereas B is the rear and D the front car. The distance between B and D is named d_{gap} and the distance between A and B is called d_{AB}. Every car in this situation can have a variable velocity which leads to difference velocities like $v_{diff_{AB}} = v_B - v_A$. For a successful merging maneuver the driver in the ego vehicle has to achieve several sub-goals like the selection of acceleration or check the distance to the front car. The relevant sub-goals and their interconnection will be described in Section 3.2. For this example we assume that the ego car will be equipped with a collision avoidance system (CAS).

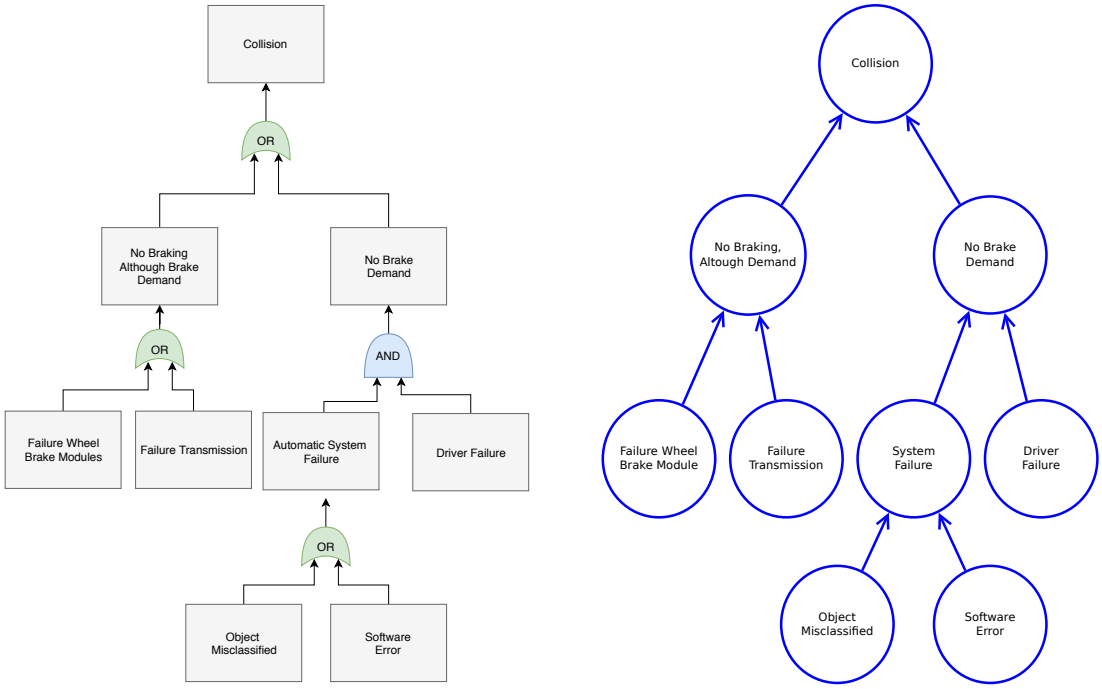
In the first scenario, the ego car A rear-ends the leading car C. For this scenario to happen, both the CAS needs to malfunction and the human driver needs to make a mistake. In this discussion we assume that these are the only two possible causes; so we assume that we can produce evidence that other causes, like a total brake failure, can be ruled out.

In the second scenario we assume that the ego car, A, collides with alter car B. In contrast to the scenario above, where the CAS can avoid a collision with the lead car, we have no pure technical means to avoid a collision. Since most highway laws require the merging car to yield, if car A collides with car B, it was always the fault of the driver of A. However, we assume that car A has a blind spot warning system, that should alert the driver to the presence of car B.

3.1 Technical Models

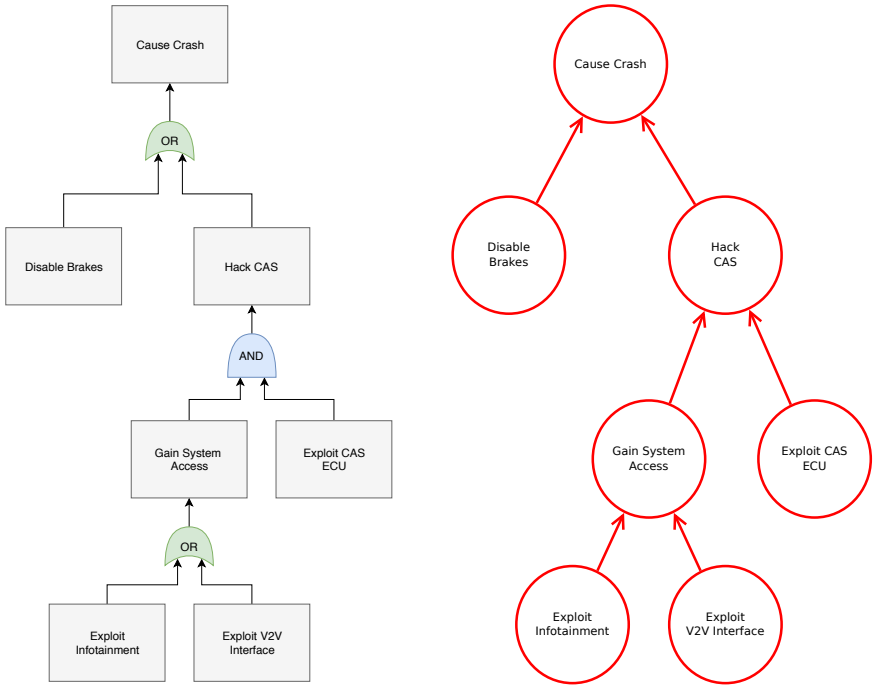
A major problem for causal reasoning is the lack of models. Creating causal models manually is a tedious process and impossible without detailed technical knowledge of the system that is being modeled. To kickstart this process, we have shown in [12] that it is possible to use existing models of systems to seed a causal model. Having an automatically generated model as a base or scaffold for causal models makes it easier for domain experts to build the models. Additionally, resulting causal models can then be reused in future investigations that analyze similar accidents. In [12] we have shown how attack trees, fault trees and timed failure propagation models can be converted to a causal model and be merged into a holistic causal model. To keep the paper focused, we will restrict the discussion to attack and fault trees. However, in practice many different system models could be used to seed the causal models.

Figure 5a shows the fault tree for our example. It is based on a fault tree provided by [13] and depicts reasons why a car might not brake and collide with another car. One branch, *No Braking Although Brake Demand* covers mechanical failures of the brake system, while the other branch, *No Brake Demand*,



(a) Technical. (b) Causal.

Figure 5: Fault tree.



(a) Technical. (b) Causal.

Figure 6: Attack tree.

covers cases in which the brakes are working, but no brake attempt is being made.

Figure 6a likewise present a simplified attack tree for our example. The left branch, *Disable Brakes*, covers cases where an attacker manages to disable the brakes. Here we do not detail possible ways to do so any further, but they can reach from software attack to the classic "cut the brake lines". The right branch covers some ways in which an attacker could attack the CAS subsystem and cause a crash. This attack tree is modeled on the widely publicized "Jeep Hack" [6].

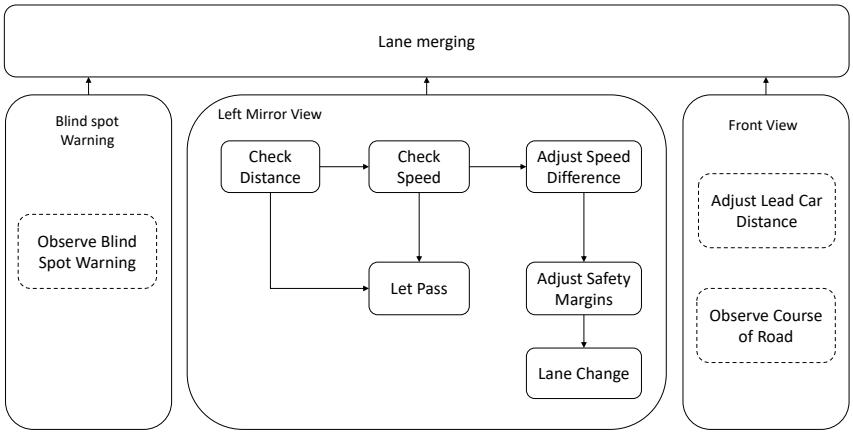
3.2 Driver Model

We use the structure of the driver model for freeway lane change tasks presented in [27] and [15] as a basis for our human driver model. A slightly modified version of the state chart from [15] can be found in Figure 7a. Here the parent goal of the Lane Change Manoeuvre Task has three main sub-goals: observing the blind spot warning, observing the left mirror view and observing the front view through the windshield.

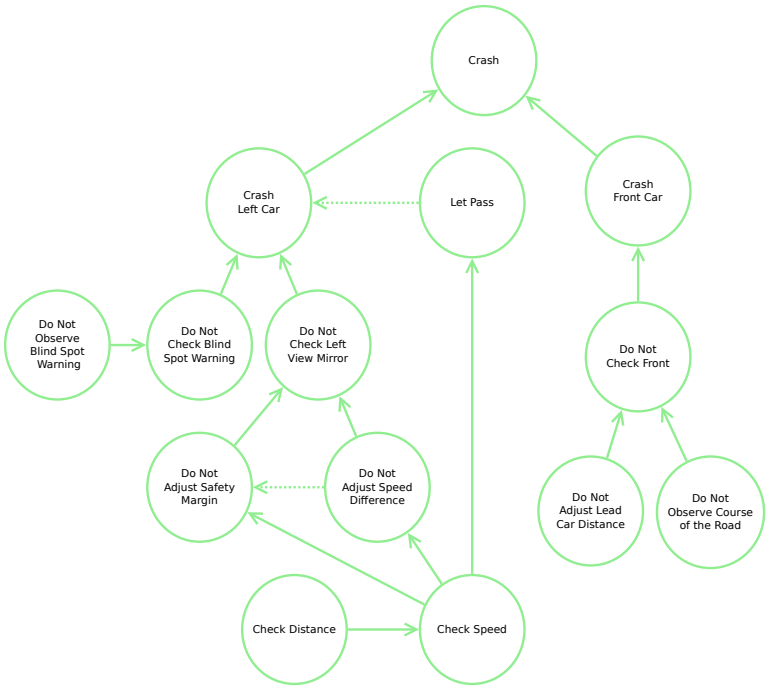
The blind spot warning sub-goal has one further sub-goal. During a lane merging maneuver the driver has to be sure that no car is driving in the blind spot. In this example we assume that the driver only uses the installed blind spot warning system for this goal. The front view sub-goal consists of two sub-goals, namely adjusting the distance to the lead car C and observing the course of the road. The second sub-goal describes the observation of the alter vehicles through the left mirror. The first sub-goal of the driver model is the estimation of the distance between the car B on the freeway and the ego vehicle A. If the distance d_{AB} is big enough the driver will check the speed v_B and estimate, based on a threshold if the relative velocity v_{AB} is low enough for a merging maneuver. If the conditions for one of these two goals are not fulfilled the driver will let B pass and start with the first goal again. If v_{AB} is lower than the threshold, the driver's confidence that the lane change will be successful rises and $AdjustSpeedDifference$ will be triggered. This means that the driver will accelerate or decelerate to have the optimal relative speed w.r.t. B and D for the lane change. The last sub-goal is the adjustment of the safety margin to all adjacent cars during the lane changing. This last sub-goal completes the lane change goal.

3.3 Setting the Context

When converting technical models, like fault- or attack trees, or human models, like HTA models, to causal models we need to be mindful of the fact that these models describe *type causality*, while we try to reason about *actual causality*. Following Halpern [9], type causality is "typically forward-looking and used for prediction", whereas actual causality "retrospectively ask[s] why [something] occurred". In our example, fault trees, attack trees, and HTA models are created from measurements, predictions and expert judgment. To reason about actual causality, however, we need to link those models to actual events and actors. In general, this is a lot easier for technical systems and often next to impossible for human beings. The reason is that we can instrumentalize and monitor technical systems with arbitrary resolution and precision, whereas observations of humans are incomplete and fuzzy. For example, we might have eye-tracking sensors that allows us to deduce that a driver was about to merge into the left lane; if we analyze the same scenario on a motorcycle, on the other hand, we most likely cannot see the rider's eye under the helmet and are unable to deduce any driver intentions at all. In a technical model, in contrast, we could in both instances detect if the turn indicator light was set. Both, the technical and the human model, are similar in structure, but very different in the available level of logging granularity. Highly automated vehicles (HAVs) are uniquely suitable to investigate the integration of human and technical



(a) HTA model.



(b) Causal Model.

Figure 7: Modified Driver Model based on [15].

models, because the gap between the log granularity is comparatively small. HAVs have access to a wide array of very precise sensors, are equipped with the computing power to process the data, have the storage capacity to record them, and, most importantly, have driver monitoring systems to gather reliable "log data" about the driver. An additional advantage in the HAV example is that the social context is very well defined. All drivers go to driving school and learn similar patterns of behavior there. In common situations the drivers will react highly similarly. Defining the context for a less standardized task, e.g., writing a poem, is a hard challenge for future research.

4 Transforming and Joining the Models

4.1 Transforming Trees into Causal Models

Comparing Definition 3 and Definition 1, the mapping is straightforward. Again following [12], Definition 4 shows that we transform each node in a tree into an *endogenous variable* that determines whether or not a specific event occurred. For now, we only consider binary causal models and express the operators of the tree in the structural equations.

To fulfill the requirement that each endogenous variable is defined by the other endogenous or exogenous variables [10], we define an exogenous variable (has the same name as the tree node with an "_exo" suffix) that will provide the value of the node's endogenous variable. This turns those leafs into endogenous and allows us to identify leaf nodes as causes of events.

Definition 4 ***Attack/Fault Tree To Causal Model***
$T = (\mathcal{N}, \rightarrow, n_0)$ is mapped to a $M = ((\mathcal{U}, \mathcal{V}, \mathcal{R}), \mathcal{F})$, i.e., $T \rightarrow M$ as follows

- *$\mathcal{U} = E(T, \_exo)$, where $E(T, suffix)$ returns a renamed copy of the leaf nodes of a tree T with a suffix "_exo".*
- *$\mathcal{V} = N$.*
- *$\mathcal{R} = \{0, 1\}$.*
- *$\mathcal{F}$ associates with each $X \in V - E(T)$ a propositional formula based on the tree gates; and with each $X \in E(T)$ a formula of the form $X = X\_exo$.*

4.2 Transforming Human Models into Causal Models

The presented model of human behavior differs from the technical models in one important way: It models positive behavior. The technical models in this paper, as well as in previous work [12], describe things that can go wrong. The presented human models, in contrast, describes the typical positive behavior. So if we want to analyze a crash, or other negative events, we first need to transform the positive model into a negative model. In Figure 7a, for example, a crash can happen if the driver does not check the blind spot warning and does not check the left mirror and does not check the front view.

Due to the intricacies of inverting only some nodes, we created the causal model in Figure 7b manually. To convert the HTA model to a causal model, we first started by modeling that the HTA actually shows actions to prevent two different crashes. The blind spot warning and the left view mirror prevent crashes with cars in the first highway lane and checking the front view prevents against read-ending leading cars on the ramp. The right hand sub-tree, that prevents rear-ending a lead car, is fairly easy to model. If the driver does not check the front view, the car will crash into any lead vehicle. This check is done by two sub-goals: adjusting the distance to any lead car and observing the stretch of road ahead.

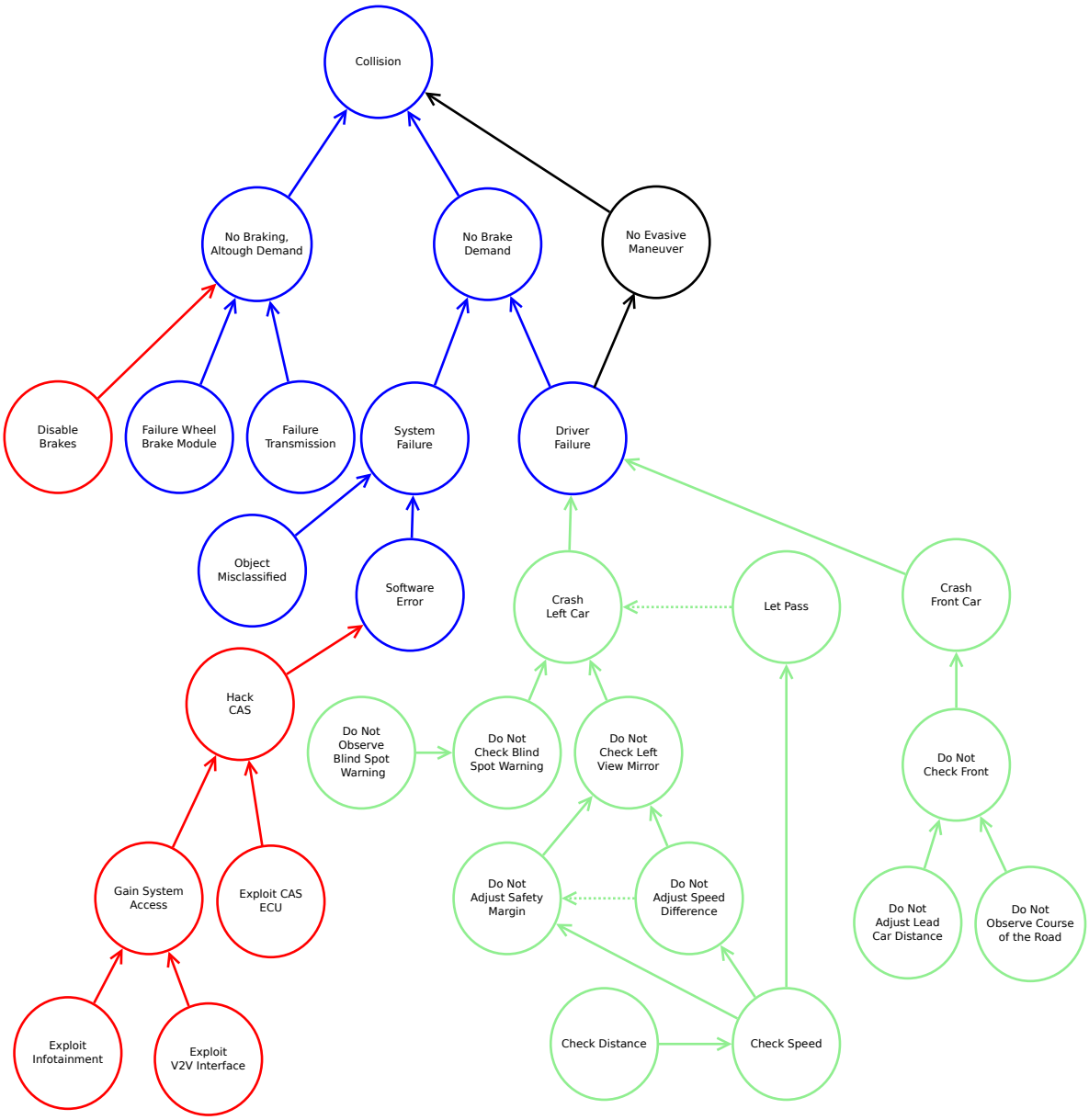


Figure 8: Integrated socio-technical causal model.

$$
\begin{aligned}
&Collision = NoBrakingAltoughDemand \vee NoBrakeDemand \vee NoEvasiveManeuver\\
&NoBrakingAltoughDemand = DisableBrakes \vee FailureWheelBrakeModule \vee FailureTransmission\\
&NoBrakeDemand = SystemFailure \vee DriverFailure\\
&NoEvasiveManeuver = DriverFailure\\
&DriverFailure = CrashLeftCar \vee CrashFrontCar\\
&SystemFailure = ObjectMissclassified \vee SoftwareError\\
&CrashLeftCar = (DoNotCheckBlindSpotWarning \wedge DoNotCheckLeftViewMirror) \wedge \boxed{\neg LetPass}\\
&LetPass = CheckSpeed\\
&CrashFrontCar = DoNotCheckFront\\
&DoNotCheckBlindSpotWarning = DoNotObserveBlindSpot\\
&DoNotCheckLeftViewMirror = DoNotAdjustSafetyMargin \wedge DoNotAdjustSpeedDifference\\
&DoNotAdjustSafetyMargin = CheckSpeed \wedge \boxed{\neg DoNotAdjustSpeedDifference}\\
&DoNotAdjustSpeedDifference = CheckSpeed\\
&CheckSpeed = CheckDistance\\
&DoNotCheckFront = DoNotAdustLeadCarDistance \wedge DoNotObserveCourseOfTheRoad\\
&SoftwareError = HackCAS\\
&HackCAS = GainSystemAccess \wedge ExploitCASECU\\
&GainSystemAccess = ExploitInfotainment \vee ExploitV2VInterface
\end{aligned}
$$

Figure 9: The equations for the integrated model. Dashed Boxes highlight preemption relations.

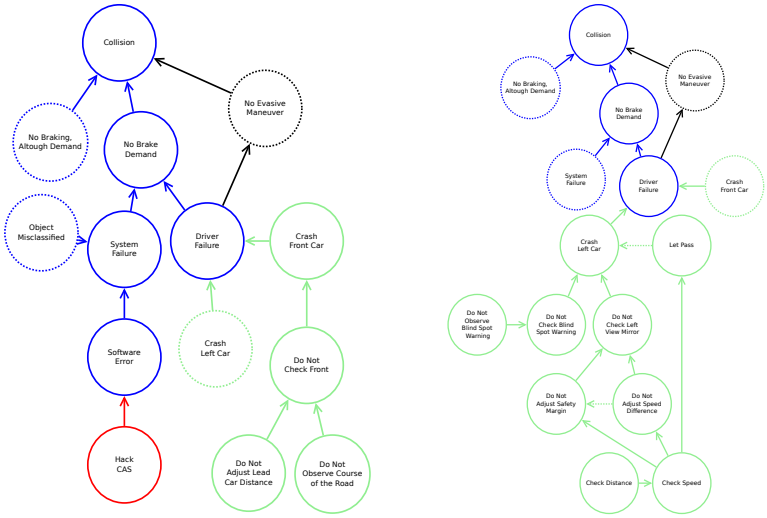
Avoiding a crash with a car in the first lane of the highway can be caused by omitting two actions: checking the blind spot warning and not checking the left view mirror. When checking the left view mirror, adjusting the speed will preempt (dotted lined arrow) adjusting the safety marking. Additionally, a crash with a car in the first lane of the highway can be preempted by simply letting it pass.

4.3 The Integrated Socio-Technical Model

Figure 8 depicts the combined causal model and Figure 9 the corresponding structural equations. The source models were joined similar to the procedure described in [12] and can now be used to reason about the actual causality in concrete accidents or events. In this concrete example, we started with the fault tree and then first joined it with the attack tree. In the process, we split the attack tree and connected *Disable Brakes* with the fault *No Braking Although Demand*. *Hack CAS* was then connected to the *Software Error* fault. To join the HTA model, we equated its top node, *Crash*, with the fault trees *Driver Failure*. The reason for this choice is that when viewed under the prospective of a crash, all human actions are seen as either correct, or as wrong and being a human error.

To set the context (i.e., the value of each variable), we of course need sensors and monitors to provide us with the values. The integrated model is useful in two ways: it helps us to decide what sensors and log data we need and it allows us to reason over the results. The first part is unfortunately less useful than might be at first expected. Since the causal model originates from technical models, it only considers states for which there anyway are sensors or at least expectations of problems. The same holds true for the human model: it generally only considers what can be measured in a lab. It is slightly more insightful, because it can inform developers of HAVs which driver sensors are relevant and which data is necessary.

The real benefit of the integrated socio-technical causal model is that it can be used by experts to guide their investigation. One advantage is that they can extend the model with knowledge from other sources. The model in Figure 8, for example, was extended with the node *No evasive maneuver* (colored



(a) Scenario 1. (b) Scenario 2.

Figure 10: Simplified causal models.

in black). The core advantage, however, is that we can reason across multiple models. For example, we can evaluate scenarios in which the driver did not check the left view mirror and at the same time the CAS did not detect a car in the highway lane. We can also express that in scenarios, like the Jeep hack, where an attacker has the ability to disable the brakes, both the CAS and the human driver could not have prevented an accident. In our experience, causal model are uniquely suitable for such cross domain modeling and reasoning.

4.4 Analyzing Scenarios

We can now use the integrated socio-technical causal model (Figure 8) to reason about the two scenarios of our example (Section 3). If there was a rear-end collision and we can confirm that there was a demand to brake, we can, in a first step, disregard the left most sub-tree under the node *No Braking, Although Demand*. Since we crashed into the lead car, we can also ignore the sub-tree *Crash Left Car* from the HTA model. If we can now find evidence for a hacking attempt against the CAS, we know that a combination of driver error and software error, inducted by the hacking is the cause of the event. Figure 10a depicts this causal model, with showing "pruned nodes" with dashed borders. We can now also model a preemption relation between the *system failure* and the *driver failure*. By choosing the direction of this preemption, we can model our view of autonomy: if the driver preempts the system, we are talking about non autonomous cars, in which the driver needs to always pay attention and if the system preempts the driver, we model fully autonomous vehicles that do not require driver attention. There is, of course, no right or wrong model – the correctness depends on the context. The great advantage of causal model is that they can capture those assumption and explicate them for further discussion.

Figure 10b depicts the causal model for the scenario in which the car crashes into the car on the first lane of the highway. In this scenario we have no active technological measure to avoid the crash – the fault arises from a human error. From the model, we can deduce that the driver did not look (or ignored) the blind spot warning light and did not check the left mirror. Also the driver did not let the other car pass, thus not preempting the crash.

5 Conclusion

In this paper, we have shown how we can use Halpern-Pearl causal models as a *lingua franca* to combine technical and social models of a single system into an integrated socio-technical causal model. The expressive power and versatility of causal models allows us to transform the source models and, in some cases even automatically, combine them into a single model. The major advantage of this approach is that we can reuse already existing models to seed the causal models and do not have to build them from scratch. The integrated model can then be revised by experts and used to reason over problems that cross multiple models.

Our work, however, is still in its infancy and poses several challenges for future work. While the joining of models can be done automatically in many cases, there are many instances where we still need expert intervention; especially joining positive and negative models automatically is a great challenge. Converting the human models into causal models is still a wide open problem. For one, developing those human models requires time consuming and expensive empirical studies. Since negative situations, such as crashes, are, even in simulations, relatively rare, building and validating negative models of human behavior is significantly harder than developing positive models. Another major issue is how to actually set the context for human models, or, in other words, how can we measure the human behavior in sufficient detail. While generalized, type causal, knowledge is useful in research, for actual causality we need to know exactly what a certain individual has done. While recording this data is a formidable technical challenge, it also raises very important issues about privacy and data ownership. One solution could be to rely on cognitive architectures, connected to detailed HAV simulators to analyze actual and possible accident scenarios.

Despite these open challenges, we are convinced that extending technical models with human models and use automatic tool to reason about these models is an essential building block for accountable CPS that integrate well into their societal context. The example given in this paper highlights the future potential and the general feasibility of our approach.

Acknowledgments. This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant no. PR1266/3-1, Design Paradigms for Societal-Scale Cyber-Physical Systems.

References

[1] Dalal Alrajeh, Hana Chockler & Joseph Y Halpern (2018): *Combining Experts' Causal Judgments*. In: *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

[2] John R Anderson, Michael Matessa & Christian Lebiere (1997): *ACT-R: A theory of higher level cognition and its relation to visual attention*. *Human-Computer Interaction* 12(4), pp. 439–462, doi:10.1016/0010-0285(80)90005-5.

[3] John Annett (2003): *Hierarchical task analysis*. In: *Handbook of cognitive task design*, CRC Press, pp. 41–60, doi:10.1201/9781410607775.ch2.

[4] Glenn Bruns & Stuart Anderson (1993): *Validating safety models with fault trees*. In: *SAFECOMP'93*, Springer, pp. 21–30, doi:10.1007/978-1-4471-2061-2_3.

[5] Meir Friedenberg & Joseph Y Halpern (2018): *Combining the Causal Judgments of Experts with Possibly Different Focus Areas*. Available at `http://www.cs.cornell.edu/home/halpern/papers/focus.pdf`.

[6] Andy Greenberg (2015): *Hackers Remotely Kill a Jeep on the Highway—With Me in It*. `https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/`. [Online; acc. 2019-01-24].

[7] Dave Grossman (2001): *On killing. II: The psychological cost of learning to kill*. *International Journal of Emergency Mental Health* 3(3), pp. 137–144.

[8] Joseph Y Halpern (2015): *A Modification of the Halpern-Pearl Definition of Causality*. In: *International Joint Conference on Artificial Intelligence*, pp. 3022–3033. Available at `https://www.aaai.org/ocs/index.php/IJCAI/IJCAI15/paper/view/11058/11085`.

[9] Joseph Y Halpern (2016): *Actual causality*. MiT Press, doi:10.7551/mitpress/10809.001.0001.

[10] Joseph Y. Halpern & Judea Pearl (2005): *Causes and Explanations: A Structural-Model Approach. Part I: Causes*. *The British Journal for the Philosophy of Science* 56(4), pp. 843–887, doi:10.1093/bjps/axi147.

[11] Joseph Y. Halpern & Judea Pearl (2005): *Causes and Explanations: A Structural-Model Approach. Part II: Explanations*. *The British Journal for the Philosophy of Science* 56(4), pp. 889–911, doi:10.1093/bjps/axi148.

[12] Amjad Ibrahim, Severin Kacianka, Alexander Pretschner, Charles Hartsell & Gabor Karsai (submitted): *Practical Causal Models for Cyber-Physical Systems*.

[13] Rolf Isermann, Ralf Schwarz & Stefan Stolzl (2002): *Fault-tolerant drive-by-wire systems*. *IEEE Control Systems* 22(5), pp. 64–81, doi:10.1109/MCS.2002.1035218.

[14] Severin Kacianka & Alexander Pretschner (2018): *Understanding and Formalizing Accountability for Cyber-Physical Systems*. In: *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, pp. 3165–3170, doi:10.1109/SMC.2018.00536.

[15] Astrid Kassner, Martin Baumann & Lars Weber (2011): *A Hierarchical Task Analysis of Merging onto a Freeway—Comparison of Driver's and Driver Model's Task Representation*, doi:10.1007/978-88-470-1821-1_31.

[16] Barbara Kordy, Ludovic Piètre-Cambacédès & Patrick Schweitzer (2014): *DAG-based attack and defense modeling: Don't miss the forest for the attack trees*. *Computer science review* 13, pp. 1–38, doi:10.1016/j.cosrev.2014.07.001.

[17] YF Liu & ZH Wu (2006): *Driver behavior modeling in ACT-R cognitive architecture*. *Zhejiang Daxue Xuebao (Gongxue Ban)* 40(10), pp. 1657–1662.

[18] Andreas Lüdtke, Jan-Patrick Osterloh, Tina Mioch, Frank Rister & Rosemarijn Looije (2010): *Cognitive Modelling of Pilot Errors and Error Recovery in Flight Management Tasks*, doi:10.1007/978-3-642-03655-2_59.

[19] Andreas Lüdtke, Lars Weber, Jan-Patrick Osterloh & Bertram Wortelen (2009): *Modeling Pilot and Driver Behavior for Human Error Simulation*. In Vincent G. Duffy, editor: *Digital Human Modeling*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 403–412, doi:10.1007/978-3-642-02809-0_43.

[20] Sjouke Mauw & Martijn Oostdijk (2005): *Foundations of Attack Trees*. In: *Information Security and Cryptology - ICISC 2005, 8th International Conference, Seoul, Korea, December 1-2, 2005, Revised Selected Papers*, pp. 186–198, doi:10.1007/11734727_17.

[21] Judea Pearl & Dana Mackenzie (2018): *The Book of Why*. New York, NY: Basic Books.

[22] Enno Ruijters & Mariëlle Stoelinga (2015): *Fault tree analysis: A survey of the state-of-the-art in modeling, analysis and tools*. *Computer science review* 15, pp. 29–62, doi:10.1016/j.cosrev.2015.03.001.

[23] Dario D Salvucci (2006): *Modeling driver behavior in a cognitive architecture*. *Human factors* 48(2), pp. 362–380, doi:10.1037/0096-1523.29.2.363.

[24] Bruce Schneier (1999): *Attack Trees - Modeling security threats*. *Dr. Dobb's Journal*. Available at `http://www.schneier.com/paper-attacktrees-ddj-ft.html`.

[25] Bruce Schneier (2004): *Secrets and lies - digital security in a networked world: with new information about post-9/11 security*. Wiley.

[26] W.E. Vesely, F.F. Goldberg, N.H. Roberts & D.F. Haasl (1981): *Fault Tree Handbook*.

[27] Lars Weber, Martin Baumann, Andreas Lüdtke & Rike Steenken (2009): *Modellierung von Entscheidungen beim Einfädeln auf die Autobahn*. *To appear: Fortschritts-Berichte VDI: Der Mensch im Mittelpunkt technischer Systeme* 8.