FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Dissertation in Informatik

Accountability for Cyber-Physical Systems

Severin Kacianka

FAKULTÄT FÜR INFORMATIK
DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Accountability for Cyber-Physical Systems

Severin Kacianka

Vollständiger Abdruck der von der Fakultät für Informatik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzende: Prof. Gudrun J. Klinker, Ph.D.
Prüfer der Dissertation:
1. Prof. Dr. Alexander Pretschner
2. Prof. Dr. Stefan Leue,
Universität Konstanz

Die Dissertation wurde am 07. 03. 2022 bei der Technischen Universität München eingereicht und durch die Fakultät für Informatik am 31. 05. 2022 angenommen.

Acknowledgments

First and foremost I would like to thank my mother Andrea and my sister Nike for their unwavering support throughout my whole life and especially during the writing of this thesis. Thank you for lifting me up when I was down, for pushing me when I was lazy and for proof-reading every word I ever wrote.

Next and most importantly, I want to thank my supervisor Alexander Pretschner. Through this joint journey, you taught me how to get to the point, to tell a good story, and to be a decent researcher and human being. You had my back when I did mess up and supported me when I tried something new. I am especially grateful for the honest and tough discussions – they made this text and all my work so much better. Thank you for shaping my view of science, good work, and my philosophy of life.

A special thanks goes to all my colleagues past and present. You made the chair something I was looking forward to almost every single day. We made so many wonderful memories, shared stories and insights. All of you shaped my thinking and attitude and I am a better man for it. I am proud to have worked with you and grateful to be able to consider you friends. Thank you.

Last, but not least, I want to thank my friends for their help, support and especially for the time not spent on the thesis. Anastasia, for making this last leg of my PhD so special; Celia, Eva, Evi, Hannes, Luis, Marci, Maresa, Michael, Peer, and Philip for those countless hikes and climbs that kept me sane; Niina and Jan for our project and the discussions that inspired me so much; Anne, Corinna, Luisa, Markus, and Nikolaus for those long nights and deep talks.

Abstract

Accountability was originally a means to restrict the power of officials by threatening consequences for their actions. This concept has recently been repurposed to increase the trust in technical systems and especially algorithms. If cyber-physical systems (CPS) such as unmanned aerial vehicles or autonomous cars cause damage or harm, accountability helps to understand why this happened and to pinpoint who is accountable for it. The core problem is that there is no single universally accepted definition of accountability that applies to all situations and social contexts. On the contrary, there exist many different definitions that are suitable in different situations. However, current literature offers no means to compare them and no guidance for developers on which definition to use or how to apply them. To address these gaps, we (1) offer a method to model accountability definitions for CPS and (2) show how that method can be used to assess the accountability of CPS, understand trade-offs in the system design, and choose between different definitions of accountability.

First, to model accountability definitions, we leverage the facts that power is the capacity to influence other people or systems and that accountability is a means to restrict such power. In this thesis, we interpret this as a causal influence and consequently propose Structural Causal Models (SCMs) to model accountability definitions.

To close the second gap, we show how technical systems and their social context can be modeled as SCMs. As a means to restrict power and hence causal influence, accountability structures are visible in the SCM of the system. Thus, our method makes it possible both to understand the impact of accountability on the system design as well as to justify that specific changes to a system are necessary to comply with a given definition of accountability.

Finally, we illustrate our method with the real life accident of an autonomous vehicle. We show that our approach enables us to identify and resolve ambiguities in the accountability of the system. Furthermore, our method can prove that certain agents did not cause specific outcomes thus excluding them from any accountability.

Zusammenfassung

Accountability beschreibt ursprünglich Mechanismen die die Macht von Amtsträgern beschränken sollen. Neuere Forschung wendet dieses Konzept auch auf technische Systeme und Algorithmen an. Diese Dissertation untersucht inwieweit sich Accountability auch auf Cyber-physikalische Systeme (CPS), wie Drohnen oder autonome Fahrzeuge, übertragen lässt. Es gibt jedoch keine allgemeingültige Definition von Accountability, sondern eine Vielzahl an Definitionen, die in unterschiedlichen sozialen Kontexten verwendet werden. Aktuell bietet die Literatur keine Möglichkeiten diese zu vergleichen und auch keine Richtlinien für Entwickler*innen, wie diese im Systemdesign anzuwenden sind. Deshalb erweitert diese Dissertation den Stand der Forschung um eine Methode, die verschiedene Definitionen von Accountability für CPS modelliert. Darüber hinaus wird gezeigt, wie diese Methode dazu verwendet werden kann die Accountability von CPS zu beurteilen.

Um Definitionen von Accountability zu modellieren, stützt sich die vorliegende Arbeit darauf, dass (1) Macht die Fähigkeit ist, andere Menschen oder Systeme zu beeinflussen, und dass (2) Accountability ein Mechanismus ist, um Macht zu beschränken. Wir interpretieren diesen Einfluss kausal und nutzen strukturierte kausale Modelle (engl. Structural Causal Models (SCMs)) zur Modellierung dieser Definitionen.

Des Weiteren zeigt diese Dissertation, wie technische Systeme und ihr sozialer Kontext als SCM modelliert werden können. Als Mittel zur Einschränkung von Macht und damit von kausalem Einfluss werden im SCM eines Systems Accountabilitystrukturen sichtbar. Somit ermöglicht diese Methode, sowohl die Auswirkungen von Accountability auf das Systemdesign zu verstehen, als auch zu begründen, warum bestimmte Änderungen an einem System notwendig sind, um einer bestimmten Definition von Accountability zu entsprechen.

Abschließend wird die vorgestellte Methode anhand eines tödlichen Unfalls eines autonomen Fahrzeugs illustriert. Es wird gezeigt, wie man Definitionen von Accountability als SCM formuliert und dadurch Unklarheiten in der Accountability des CPS und seines sozialen Kontextes identifizieren kann. Schließlich wird bewiesen, dass gewisse Agenten für den Unfall keine Verantwortung tragen.

Contents

List of Figures

1 Introduction

1.1 Motivation

Rooted in Liberalism, Accountability is a means to constrict power.[1] As a concept it was introduced by political philosophers like John Locke,[2] and Adam Smith[3] who used it in the 17th and 18th century to describe the fact that official representatives will have to justify their actions to someone, ultimately their sovereign.[4] The core idea here is that some official is given power by a principal, for example their king or the voters, and that this power is then limited by the fact that they have to account for their use of power to their principal. This gives the principal the option to evaluate the performance of that official and take measures to punish or reward them. Thus accountability is an alternative to other means of constricting power such as violence, public shaming or economic pressure.

With the rise of computer systems, this general idea was applied to such systems as well. However, a core problem in the literature is that the term *accountability* is not well defined and the high level goal, restricting power, is usually not explicitly stated. Instead, many publications and implementations take accountability as an end in itself, usually relying on a vague dictionary meaning of the term. In our review of technical implementations [Kacianka et al., 2017] we found that systems will often implement "something" and then just call it "accountability", without trying to ground that in any definition or understanding of the term. In a recent systematic literature review of algorithmic accountability, [Wieringa, 2020] writes that many organizations are "advocating for more algorithmic accountability, yet a thorough and systematic definition of the term lacks, and it has not been systematically embedded within the existing body of work on accountability." This finding now begs the question what

[1] The term "power" is extremely complex. In this thesis power means the ability to act, and more specifically the power to affect a specific causal chain of events.

[2] "All I need for my purposes is to point out that none of them has any authority beyond what is delegated to it by positive grant and commission, and are all of them are accountable to some other power in the commonwealth." [Locke, 1690]

[3] In the first edition of the "Theory of Moral Sentiments" Smith wrote "A moral being is an accountable being."; this was removed in later editions. See [Dubnick, 2010] for a detailed discussion.

[4] See [Bovens, 2007] or [Lindberg, 2013] for a more detailed history.

exactly these organizations are advocating if there is not even a definition of the concept. This feeling of ambiguity is enforced when looking at the concrete examples given in the appendix of [Wieringa, 2020]. The first is an automatic system that checks if people repay their debt, the second one is a system that automatically anonymizes permits, the third and fourth check for fraudulent social benefit claims. All examples have in common that a computer system is given the power to influence the lives of people, but no one fully understands how it does so and thus seeks to somehow limit the system to "acceptable" actions.

When we now look at socio-technical Cyber-Physical Systems (CPS), such as autonomous cars, unmanned aerial vehicles (UAVs) or smart power grids, we see a rising number of systems which have significant power[5] but are at the same time highly complex and hard to understand. For example, [Elish, 2019] points out that in complex systems responsibility[6] is often deflected and hard to pinpoint, which leads to blame being assigned to humans. As she lays out, often a "pilot error" is not just an error of the pilots, but a complex interplay of the humans with the technical systems. Similarly, [Fosch-Villaronga et al., 2018] describe how complex and interconnected health care robots already are and [Felzmann et al., 2019] suggest that such systems need to be more transparent in order to make them "accountable".

These examples point us to the actual real world problem that accountability tries to solve. In its classic meaning, accountability is a means "to constrain the (mis-)use of power" [Lindberg, 2013]. This abstract notion is now applied to any context in which people feel powerless towards the actions of machines. The idea is that someone transferred their power to the machine and we now want to ensure that this power is not misused. If we think a machine did something we dislike, we have several options other than accountability available to constrain its power. For example, we might resort to violence and just beat up whoever we deem responsible for the actions of the machine. Another option is to resort to public shaming and start a mob to get the machine taken out of commission[7] or we could boycott the operator of a machine we dislike, thus exerting economic pressure. Furthermore, we could technically limit the capabilities of a machine; if a car cannot go faster than 1kph, it can never break the speed limit.

Accountability now aims to avoid these chaotic and debilitating responses and put in place a process to constrain the power of the system while preserving its usefulness. The exact nature of the process is described in accountability definitions (see Chapter 2.1).

[5] Often also in the physical sense that they are capable of transferring huge amounts of kinetic energy.

[6] [Elish, 2019] here uses the term "responsibility" synonymous with "accountability". The terminology is tricky, we go into more details in Chapter 2.1.

[7] A classic example are the luddites who destroyed textile machinery to protest against industrialization.

These definitions describe how a principal will limit the power of an agent and what tools the principal has available. To do that, we need a precise understanding of the actual powers and capabilities of the agent that is to be made accountable. Our argument is that because (1) accountability is a means to constrict the use of power, (2) power is the capacity to influence individuals or outcomes, and (3) influence is a causal concept, Structural Causal Models are a suitable means to express the capabilities of systems and understand the underlying accountability relations.

In our work, we do not aim to "punish the machine". We fundamentally believe this does not work, because machines cannot feel pain or show remorse. Instead we aim to identify the humans, as well as legal persons like companies that are part of a socio-technical system and their influence on a given course of events. Our goal is to make these connections transparent and understandable. If then an unwanted event occurs, we can find accountable entities and use societal instruments like courts to punish them. Conversely, our approach also allows us to show that entities cannot be accountable for events, thus absolving them for any consequences for a specific event.

1.2 Overview

In [Kacianka et al., 2017] we conducted a systematic mapping study of the computer science literature in search for a common definition of accountability. However, we found that few papers even rely on a definition at all; instead they will often just implement something and claim it makes a system accountable. Similarly, [Wieringa, 2020] did a systematic literature review on algorithmic accountability with the aim of systematizing the field. She then gives a definition of algorithmic accountability of her own, but misses to clearly state what algorithmic accountability is supposed to achieve. Just like the finding in our mapping study, she provides a series of steps that do "something" and then this something is called "accountability". So far all the literature in computer science seems to miss the ultimate goal of accountability: it is just a means of restricting the use of power. Without stating that goal explicitly, approaches in computer science are inherently vague and range from sophisticated logging systems (e.g., [Ko et al., 2011]) to systems that confuse causality and accountability (e.g., [Künnemann et al., 2019]). Looking at accountability through the lens of transferring power, makes it much easier to distinguish accountability from other means to limit power and to argue what form is appropriate in a given context.

Gap: The field of computer science has so far failed to look at the actual goal of accountability, namely that it is a means to limit power. While the literature offers us many definitions of the term, it offers us no method to model the accountability of a system, especially across system boundaries. It offers no way to quantify or qualify the accountability of a system, nor even a precise language to reason about it or compare implementations. The current state of the art does not go beyond giving differing definitions of steps that might make a system "accountable" without specifying the underlying goal.

In our initial research proposal, we started with the hypothesis "that significant features of accountability can be captured in a unified model, regardless of the concrete implementation or domain." This led us to formulate three research questions. The first asked about commonalities of accountability mechanisms, the second one about how to model accountability mechanisms and the final one asked how to utilize this model in the system design. Here, we chose to slightly reformulate and condense them:

RQ1 How can we express what it means for a system to be accountable?
RQ2 How can we measure and assess the accountability of a system?
RQ3 How do we adapt the design of a system to a specific notion of accountability?

RQ1: We suggest to use Structural Causal Models (SCMs) to capture a system's technical functionality and connect it to its social context. An SCM expresses what a socio-technical system can do, i.e., what power it has, and what entity caused an outcome. As such, accountability, that limits a system's power must be visible in the SCM, as the actual power of the system changes. Furthermore, causality itself is a necessary prerequisite for accountability [Kacianka and Pretschner, 2021], because accountability works by controlling the entities causing an outcome. The interesting aspect of computer systems is that we often have models of them available that can be converted into SCMs. In [Kacianka and Pretschner, 2018, Kacianka et al., 2019a, Kacianka et al., 2019b] we have shown that this works in principle.

RQ2: Building on our premise that accountability can be expressed with SCMs, we can rely on work on how to assess, measure and compare SCMs. Here we use work by [Halpern, 2016, Chapter 4] who gives different examples and guidelines on how to structure models, and our own work [Kacianka et al., 2020, Kacianka and Pretschner, 2021] on how to identify and leverage accountability patterns in SCMs.

RQ3: Since the SCM of a system derives from the system design, we can also reverse that process and design a system so that it conforms to a specific SCM. We show how the accountability patterns from RQ2 can be used to assess the accountability model of a system and then adapt the system design such that causes of effects are easy to identify [Kacianka et al., 2020, Kacianka and Pretschner, 2021].

Solution: In this thesis, we argue that the Structural Causal Models (SCMs) of a system allow us to express the power (i.e., influence) of a system and thus accountability, a means to restrict power, must manifest itself in the SCM. We argue that the difference between two SCMs, one which is restricted and one that is not, is the essence of a given definition of accountability. Computer systems are unique in the fact that we usually have models of different facets[8] of them available and can often use them to derive SCMs [Kacianka et al., 2019a]. These technical models can then be connected to their socio-technical context [Kacianka et al., 2019b, Kacianka et al., 2020, Kacianka and Pöchhacker, 2020], providing holistic causal models that span system boundaries. Of course, the quality of any analysis depends on the accuracy of the models. SCMs offer us mathematical clarity of expression, but the correctness of a model still needs to be agreed upon by humans. As [Pearl, 2014] puts it: "The question arises whether it is realistic to assume that investigators would possess such certainties in real-life applications. Here we should recall that anchoring one's analysis in specific causal scenarios does not imply a commitment to the validity of those scenarios." So our point is not that a model will be correct from the start, we would contend that no model will be correct right away, but that causal models offer the best way to communicate our assumptions. With these forward looking kind of model, we can compare accountability definitions. Additionally, using causal models, we can use the accompanying mathematical machinery and reason over them. This can be used to find accountable entities for specific events.

Once we have an adequate model of a system, we can leverage the fact that causality is necessary for any attribution of accountability, because it shows what a system can do, i.e. what power it has, and verify if a system provides the necessary causal structures for a given definition of accountability. This way, we can show if a system does not fulfill any necessary condition. This, however, is not sufficient for accountability. Once the basic causal structure is in place, we need to ensure that the surrounding social context of the system provides the facilities required by the accountability definition.

Contributions:

1. We are the first to provide a mathematical language to express different accountability definitions and identify them in systems. While formalizations for specific notions of accountability do exist, our approach can express a multitude of definitions and identify if the model of a given system conforms to a specific definition. We do this by relying on the premise that accountability, as a means to restrict power, requires causality and it can thus be expressed using SCMs.

[8] For example, Fault and Attack Trees or UML diagrams.

2. Our approach can be used to compare definitions of accountability as well as the accountability between systems. Furthermore, we can mathematically prove that certain agents in a model can never be accountable for a certain outcome. Here, again, we leverage SCMs and use their ability to show the absence of causality to proof the absence of any form of accountability.

1.3 Structure of this Thesis

In Chapter 2 we first give an overview of different definitions of accountability taken from fields such as sociology, psychology, the organizational sciences and, cover, of course, related work from computer science. With an understanding of the wide variety of definitions, we then give a brief introduction into the mathematical foundations for causality in Chapter 3. As a prerequisite for our CPS use case, Chapter 4 shows how we can model them using SCMs.

Having presented these foundations, we come back to the research questions. Chapter 5 shows how the accountability definitions presented in Chapter 2 can actually be expressed as SCMs, thus provide answers to RQ1, how we might express the accountability of a system in a formal model. This chapter also picks up RQ2 and discusses means to evaluate and compare causal models. The penultimate chapter, Chapter 6, presents a use case based on autonomous cars. Here we show how expressing accountability as SCM can be useful in practice and show how we can reflect accountability definitions in the system design. As such we answer RQ3, the implications on the system design, by giving an example. Finally, this thesis concludes in Chapter 7 with a discussion of our work, an outlook on future work and some final remarks.

This thesis is built on ideas that were first published in several papers. [Kacianka et al., 2016] outlined our initial idea, [Kacianka et al., 2017] systematically mapped the literature on accountability, [Kacianka and Pretschner, 2018] outlined our initial thoughts on formalizing definitions of accountability, [Kacianka et al., 2019a] showed how to derive causal models from system models, [Kacianka et al., 2019b] showed how we can transform models of human behavior to causal models, [Kacianka and Pöchhacker, 2020] discussed the social problems in constructing causal models and [Kacianka and Pretschner, 2021] presented the core idea of this thesis, using structural causal models to express accountability definitions and compare them to system models. Chapter 5 is unpublished, but under review at the submission of this thesis. Some passages of text are taken with permission from the publisher.

2 Accountability

2.1 Introduction

As explained in the introduction, accountability is a concept rooted in Liberalism and was conceived as a means to check the power of officials. This core idea was then picked up and refined by other political and, later, social scientists. In a survey, [Lindberg, 2013] gives the central idea as “when decision-making power is transferred from a principal (e.g. the citizens) to an agent (e.g. government), there must be a mechanism in place for holding the agent accountable for their decisions and tools for sanction”. [Bovens, 2007] writes that “[t]he most concise description of accountability would be: ‘the obligation to explain and justify conduct’ ”, while also warning that “[a]s a concept, however, ‘accountability’ is rather elusive. It has become a hurrah-word, like ‘learning’, ‘responsibility’, or ‘solidarity’, to which no one can object.”

This core idea is, with some variations, deeply embedded into the fabric of liberal democracies. From the idea that voters will hold politicians accountable for their performance, to companies that are accountable to their shareholders, to the legal systems, where wrongdoing is discouraged by the possibility of being held accountable for one’s actions. As such, accountability rose to prominence in computer science together with the tight integration of computers into our societies and their increasing effect on daily life, for example by managing medical records or controlling vehicles.

This long history of accountability has led to many different definitions and meanings of accountability. [Lindberg, 2013] identified twelve different sub types of accountability and also cautions us that [i]t cannot be assumed that findings in the area of one subtype of accountability are relevant for another.” For example, if we find an implementation of accountability that works well in a societal setting, it is not a given that it will also work in a legal setting. In our own survey on the topic, [Kacianka et al., 2017], we have found a tendency in computer science to not worry much about the underlying definitions of accountability. Even recent works, for example [Wieringa, 2020], usually pick some definitions and declare it as “typical”. This is, in our opinion, wrong. Computer science should not try to pick winners, and push one theory over another. [Wieringa, 2020] pushes a theory by [Bovens, 2007], that is however hotly debated.[9] When talking about accountability, it is important to be precise about the exact meaning. As [Lindberg, 2013] puts it, "everything is not accountability: it is but one of many possible ways to constrain the (mis-)use of power." Other means of limiting power are the "devolution of power, violence, economic pressure, public shaming, and anarchy." However, since accountability is a very old concept, it has multiple meanings that often have subtle differences. This is why, when talking about "accountability" or "making systems accountable", we should always first try to define what we actually mean. For example, in some definitions sanctioning an actor for their action is considered part of accountability, while in others a principal can only sanction an agent if they do not provide an account. Such differences have a huge impact on the underlying system design and should thus be made explicit, and not left ambiguous by just using the term "accountability". In this chapter we will give an overview of some popular definitions of accountability. In our view, all definitions of accountability have some merit in a specific context, and computer science should strive to offer ways to implement any definition. It is the purview of fields like sociology or the political sciences to debate the intricacies of the definitions themselves. They have accumulated experience in debating these finer points and computer science should rely on their output and offer ways to realize the theories developed there. Here we will present some approaches to accountability as an example of the wide variety. The goal of this thesis is to show how any definition of accountability can be formalized and then implemented for CPS.

In the following sections, we will introduce different notions of accountability. The goal is to show how differently accountability can be defined and that each definition has different advantages. They all aim to restrict the power of some agent, but do so in different ways.

2.2 Lindberg

[Lindberg, 2013] surveyed the literature in the social sciences and distilled the following definition of accountability:

1. *An agent or institution who is to give an account (A for agent);*

[9] For example, [Lindberg, 2013] has to say about it that its "main achievement is to obfuscate the distinctiveness of accountability from other types of constraints on actors' power to act autonomously. When the term 'sanction' finally is misunderstood to denote only punishment (deviating from the proper meaning of the word in English), the paraphrasing becomes misleading." So following this definition will entice us to connect accountability with punishment, even thought it is not necessary and, in the context of machines, will often be nonsensical.

2. *An area, responsibilities, or domain subject to accountability (D for domain);*
3. *An agent or institution to whom A is to give account (P for principal);*
4. *The right of P to require A to inform and explain/justify decisions with regard to D; and*
5. *The right of P to sanction A if A fails to inform and/or explain/justify decisions with regard to D.*

The first two points mean that there is an agent that has some power in a certain domain and knows that they need to give an account for their actions. The third and the forth condition imply that there is a third party that has the right to require A to explain and justify their decisions. The last condition requires that P can sanction A. Here, [Lindberg, 2013] adds an important restriction, often lost in other definitions: "But an important distinction should be made between the right to sanction A for failure to provide the information requested and justifications for decisions and actions taken, and the right to sanction agents or institutions (A) for the content or effects of such decisions and actions. At its core, accountability only necessitates the right to sanction A for failure to provide information and justify decisions. The right of P to also sanction the content of decisions and actions by A is a possibility that if present adds additional leverage for P but is not strictly necessary for the concept of 'accountability'." Another important implication is that there needs to be *standard or measurable expectations* to have accountability. Without a clear idea of what is acceptable and unacceptable behavior, it cannot be evaluated and sanctioned.

His contribution is notable, because he establishes accountability as a *classical concept*, meaning that its subtypes are complete instances of their parent category. As an example, he gives accountability, which is a "method of limiting power" and is on the same level as other subtypes such as "devolution of power, violence, economic pressure, public shame, and anarchy". [Lindberg, 2013] categorizes subtypes of accountability along three axes. The first is whether the principal is external or internal to the system or organization, the second is the degree of control that the principal has over an agent and the final dimension characterizes if the accountability relation is horizontal or vertical, i.e., if the principal and the agent are equals in some social hierarchy.

Table 2.1 provides an overview of the twelve subtypes. Here, we will not go into details about the different subtypes, but want to point out that [Lindberg, 2013] cautions us of the "inappropriateness of using the findings from one subtype of accountability to another." The reason for this is that every subtype is very much dependent on its

Table 2.1: The twelve sub types of accountability according to [Lindberg, 2013].

	Strength of	Vertical		
Source of Control	Control	Upward	Downward	Horizontal
Internal	High	*Business*	*Bureaucratic*	*Audit*
	Low	*Client-patron*	*Patron-client*	*Peer Professional*
External	High	*Representative*	*Fiscal*	*Legal*
	Low	*Societal*	*Political*	*Reputational*

context.[10] This means that "we cannot scale or even rank order the various types of accountability", because often their differences are nominal and cannot be mapped to quantitative or qualitative scales. Furthermore, every subtype is only suitable for certain situations. No sub type is a silver bullet suitable for all contexts. For us this means that, to reuse concepts of accountability, we need a formalization that is capable of expressing and abstracting the context of specific definitions of accountability.

2.3 Bovens

[Bovens, 2007] definition became popular recently in computer science, because it was used as the definition in the systematic literature review conducted by [Wieringa, 2020]. He finds that accountability is hardly defined and that it "has become a hurrah-word, like 'learning', 'responsibility', or 'solidarity', to which no one can object" and "the concept has become less useful for analytical purposes, and today resembles a garbage can filled with good intentions, loosely defined concepts, and vague images of good governance." With his definition, he tries to counteract this vagueness and make it "more amendable to empirical analysis". He focuses on *public accountability* and gives a short definition, "the obligation to explain and justify conduct", before giving the following more detailed one:

1. *There is a relationship between an actor and a forum*
2. *in which the actor is obliged*
3. *to explain and justify*
4. *his conduct,*
5. *the forum can pose questions,*

[10] [Lindberg, 2013] gives the following example: "the question of what is required for a society to imbibe a general culture of expectations of accountability from the state and its political institutions, is very different from the specific issues relating to vertical political accountability between citizens and representatives; or what factors and incentives make for effective vertical bureaucratic accountability."

6. *pass judgement,*
7. *and the actor may face consequences.*

Following his definition, actors can be individuals or organizations, and a forum can also be a specific person, an organization, or even the general public. The relationship between an actor and a forum will often, but not always, be a principal-agent relation in which the forum delegates power to the agent, who is then held to account. The obligation of the actor might be formal or informal. The act of giving an account consists of three stages. First, "the actor is obliged to inform the forum about his conduct, by providing various sorts of data about the performance of tasks, about outcomes, or about procedures." Second, "there needs to be a possibility for the forum to interrogate the actor and to question the adequacy of the information or the legitimacy of the conduct." Finally, "the forum may pass judgement on the conduct of the actor." [Bovens, 2007] also requires the possibility of consequences for the actor if they do not comply with the requests of the forum. There are two fundamental differences between the definitions of Lindberg and Bovens. The first is that Bovens requires the actor to regularly update the forum, whereas Lindberg suggests that the principal can demand an account from the agent at any time. The second is that Lindberg suggests the option that the principal cannot sanction an agent for the content of an account, but just if the agent does not provide an account. According to Bovens the forum can judge the actor for their actions.

He makes the point that "transparency" is a requirement for accountability, but that transparency lacks the debate that accountability requires and that "controllability" differs from accountability, because it implies that the principal has direct control over the agent. [Bovens, 2007] then identifies several types of accountability and delineates them along four questions. The first, to whom one is accountable, considers to how many different fora an actor is accountable to. The second one, who is the actor, considers who actually contributed to a course of events. The third, about what, limits the domain for which an actor is accountable to a forum. Fourth, and finally, why an actor is accountable, considers if the actor is hierarchical different or the same as the forum.

2.4 Hall

[Hall et al., 2017] look at accountability from the perspective of psychology. As such they focus on what it means for an individual to *feel accountable*. Their exact field of study is *felt accountability*. In their overview, they find that at the core of accountability is the expectation that ones' actions will be evaluated. They emphasize that it is not necessary that this evaluation does occur, but that the possibility that an evaluation occurs must be present. Furthermore, the actor needs to believe that an account-giving (i.e., an explanation) might be required. This account is then given to a salient audience that might reward or sanction the agent's behavior.

In their review of models of accountability, they find that accounts are often used by agents to protect their self-image and develop their social identity. This underscores the important role accountability has in a society and supports the assumption that complex societies and social order necessarily need accountability to augment the reduced level of personal trust between individuals. In their survey, they describe four essential features of accountability. First, the *accountability source* describes to whom one feels accountable. Second, the *accountability focus* captures how things get done and how they relate to the results. Third, *accountability salience* expresses how important the task is for which an agent might be held accountable; the idea is that agent will be more careful if their action is more significant. Fourth, and finally, *accountability intensity* captures for how many things an agent is accountable; here it is thought that being responsible for multiple things increases stress.[11]

One important finding of [Hall et al., 2017] is that "leaders who are trusted (...) will be held to lower levels of accountability". They elaborate on the interplay of accountability and trust, by explaining that "the level of confidence one may have in another to perform to some desired standards may be a product of (1) the trust one has based on a relationship or prior performance or (2) accountability mechanisms that are designed for monitoring and/or controlling others' behaviors." This finding seems to agree very much with recent discussions in the field of algorithmic accountability, where accountability is seen as a way to control or limit the power of algorithms that are too complex to be understood and are thus often not trusted.

2.5 RACI

The organizational and management sciences have developed several practical frameworks to understand accountability in an organization. Here, we present the Responsible-Accountable-Consult-Inform (RACI) framework [PMI, 2017, Chapter 9] [Smith et al., 2005]. These frameworks, sometimes called tools, are used in practical settings to expli-

[11] Interestingly, [Hall et al., 2017] also give some examples of how accountability is measured in psychology. Here, they cite scales used in surveys that consists of questions like "I often have to explain why I do certain things at work" or "Others in my organization can observe the outcome of my work performance in terms of achieving unit goals". While these scales are clearly aimed at humans, it might be interesting to use them as a basis to assess CPS as well.

cate accountability relationships in teams and organizations. While these tools do not build on a sophisticated body of scientific literature, they are nonetheless important in practical settings, because they allow people to express their understanding and perception of accountability in a given setting. We assume that such practical approaches will often be the basis for accountability expectations of CPS and thus need to be considered in any attempt to formalize accountability for them. [Smith et al., 2005] define the following four aspects:

- ***R**esponsible: The individual who completes a task. Responsibility can be shared.*
- ***A**ccountable: The person who answers for an action or decision. There can be only one such person.*
- ***C**onsult: Persons who are consulted prior to a decision. Communication must be bidirectional.*
- ***I**nform: Persons who are informed after a decision or action is taken. This is unidirectional communication.*

RACI, specifically, tries to explicate the roles of people in an organization and helps to reconcile the conception of a role, i.e., what a person thinks they are doing, with the expectation of a role, i.e., what others think the person is doing, and with the behavior of a role, i.e., what the person actually does. Following [Smith et al., 2005], having a RACI matrix, helps to align these three aspects. However, they also point out that this is an on-going process that needs to constantly realign those three aspects, whenever they drift apart. They list a few typical signs, such as "questions over who does what" or "concern over who makes decisions" that arise regularly during the design of systems. Among other things, one major difference of this definition with the others is the aspect of consultation, once again underlining our finding in the introduction that definitions are manifold and different.

2.6 Computer Science

A landmark publication on accountability in computer science was published by [Weitzner et al., 2008]. They provide a definition for "Information Accountability", as an improvement on classic preventive data control measures. Classically, systems protect a user's privacy by ensuring that data cannot be accessed by unauthorized personal and thus prevent data leaks. [Weitzner et al., 2008] changed this premise and, drawing parallels to law enforcement, suggest to build systems in such a way that it is easy to trace data leaks and then leverage the existing legal system to punish misbehavior. [12] This idea was later refined and formalized by [Feigenbaum et al., 2011, Feigenbaum et al., 2012], albeit with a focus on security and not privacy. Coinciding with the discussion on e-voting systems, [Küsters et al., 2010], formalized accountability in relation to verifiability. Here, the main question is how to design an e-voting system such that the results can be trusted and any attempts to falsify the vote count or the votes will be detected and the perpetrator held to account. The A4cloud project [Felici and Pearson, 2014], coinciding with the spread of cloud services into society and questions about data protection, has done extensive work on accountability in cloud environments, with a focus on data protection and privacy. They offer a reference architecture, tools to complete certifications and risk assessment. The exact definition of accountability is deferred to contracts or service level agreements.

Building on this foundation in the literature, we conducted a systematic mapping study, to understand how accountability is understood and implemented in research tools [Kacianka et al., 2017]. In this study, we could identify a steady rise in publications on the subject and found that most research was either a solution proposal or an evaluation of an approach. Analyzing the prominent application domains, cloud computing was clearly dominant, followed by distributed data sharing and web applications. The most prominent use cases were privacy focused, in line with [Weitzner et al., 2008] and the most popular techniques were cryptographic and network protocols, with some dedicated accountability protocols as well.

Lately, research seems to gravitate towards the term "Algorithmic Accountability", which makes it easier to find related work, yet this term is disconnected from important work like [Weitzner et al., 2008] and the work building on it. To us it is interesting that the strong focus on cloud computing and data privacy seems to be completely eclipsed by research on accountability of (machine learning) algorithms. We moved from caring about the data to the actual decisions based on the data. This seems to support [Bovens, 2007] argument that accountability "has become a hurrah-word" and, to us, it appears that accountability is just used to mean "make technology do what I want it to do".

In [Kacianka et al., 2017], we also analyzed the definitions used in the identified papers, and found five major themes:

1. *Accountability should associate (or* link*) actions to entities (often individuals).*
2. *This link should then be used (often by a neutral third party) to hold the entity* responsible

[12] The core problem, however, was discussed much earlier. Notable contributions are Lamport's logical time stamps [Lamport, 1978] and Nissenbaum's discussion of the "eroding accountability in computerized societies" [Nissenbaum, 1996, p. 25]. [Povey, 1999] presented the idea of "optimistic access control" that enforced rules retrospectively.

for that action (often the terms blame and punish are used).

3. *All definitions implicitly rely on some notion of* log *that is complete, tamper-proof and available to the neutral third party.*[13]
4. *Another implicit assumption is that the log data can be used to* reason *about the events that have transpired.*
5. *All definitions only consider* single *systems. There is no notion of "distributed" accountability in those definitions.*

Based on these insights, we gave our own first definition of accountability:

1. Accountability *is a property of a system or a collection of systems and is ensured by an* Accountability Mechanism.
2. *An* Accountability Mechanism *is part of an* Accountable System *and reasons over a tamper-proof log to link effects of that system to entities.*
3. *An entity is (partially)* accountable *for a given effect if an* Accountability Mechanism *can prove a causal link between the entity's action and the given effect.*
4. *The set of entities* accountable *for a given effect is the set of all entities for which an* Accountability Mechanism *can prove a causal link between the entities' actions and the given effect.*

However, as is evident in the context of this chapter, this is just one definition amongst many and its suitability greatly depends on the context. While we originally hoped to identify one definition of accountability that is suitable for all contexts, we changed our focus and focused on finding a method to express any given definition of accountability and then compare it to other definitions. One obvious shortcoming in this definition is that we did not distinguish between terms like responsibility and blame and were very fuzzy on the definition of a socio-technical system. Furthermore, we did not yet explicitly consider power and how it is transferred in this definition.

2.7 Algorithmic Accountability

The term algorithmic accountability first gained prominence with the paper by [Diakopoulos, 2015] where he discussed how journalists might investigate algorithms that started to make more decisions that affected human lives. In this vein, the literature on the subject usually focused on the understanding of machine learning algorithms.[14]

[13] In hindsight, we should also have added that logs need to be adequate in the sense that they contain enough information to meaningfully reason over them.

[14] Although in many papers any algorithm, no matter how trivial, will be considered an "AI".

Examples include algorithms used in court decisions [Angwin et al., 2016], policing [Kaufmann et al., 2018] and similar settings where human lives are directly affected by opaque computer systems. Most of the literature is highly critical of these systems, using terms like "Weapons of Math Destruction" [O'Neil, 2016] or "Algorithms of Oppression" [Noble, 2018]. The general approach to counter the power of algorithms is to make the decisions of algorithms transparent and explainable. Recently, [Wieringa, 2020] surveyed the literature on algorithmic accountability and found that "[w]hat is denoted with algorithmic accountability is this kind of accountability relationship where the topic of explanation and/or justification is an algorithmic system." Her survey also suggests algorithmic accountability follows the definition of [Bovens, 2007] given above.

In earlier works, transparency was often seen as a solution, but [Ananny and Crawford, 2018] have shown that transparency alone is not sufficient for accountability. Amongst other reasons, a main point is that we also need someone to understand the output of such an transparency mechanism.[15] To alleviate this problem, [Wachter et al., 2017], and later [Miller, 2018] as well as [Mittelstadt et al., 2019], propose to use contrastive explanations to make decisions understandable for humans. [Miller, 2018] gives the example of a machine learning classifier that categorizes an insect as a spider or a beetle. As an explanation it would give that a result is categorizes as a spider, because it has eight legs instead of six. In contrast to the weights in a neural net or the layout of a decision tree, such an explanation would be useful and understandable for a human.

2.8 Link to Causality

What is especially useful and interesting for the results in this thesis is that accountability is deeply intertwined with causality. First, as we mentioned in the introduction, causal models show the powers of the system and since accountability restricts the power of a system, it must also change the SCM. This explains the intuitive notions, that one can only be held accountable for actions that one also causes. You cannot be held accountable for outcomes that are beyond your control or influence. Realizing that fact, we began to set out to define accountability in terms of causality [Kacianka et al., 2016], and were surprised to find that causality was not a more prominent topic in our systematic mapping study [Kacianka et al., 2017]. This was especially surprising as, for example, [Lindberg, 2013] already remarked that "accountability theory assumes

[15] Interestingly, understanding the output is not important in the definitions given so far. We assume this is due to the fact that these definitions consider humans or organizations and that in this context it is simply assumed that any misunderstanding can be "talked out".

linear cause–effect relationships, as well as rational and informed decision-making with the aim of producing collective goods." Similarly, [Hall et al., 2017] wrote that "These external factors include role/task responsibility (roles or tasks actors acquire or choose to accept), which leads to normative responsibility (created by tacit rules), which leads to causal responsibility (whether a person contributed to an action or decision). Causal responsibility is modeled to lead to judged responsibility (the extent to which a person is held responsible for an action or decision), which, in turn, leads to external accountability (the extent to which an accountee is held answerable for his or her behaviors)."

In [Kacianka and Pretschner, 2018], we propose a formalization of accountability that looks at different notions of accountability and tried to define a *responsibility* relation that was distinct from a *causality* relation. At the time, we thought that it would make sense to distinguish these two and postulated that "(t)he notion of responsibl(ity) is closely linked to the notion of cause(ality) (...) however, while causality is a much broader concept, responsible means actual people who can stop someone or something from doing something." So, we tried to construct responsibility as causality with the addition of the ability to change the course of events. In [Kacianka et al., 2020] and more elaborate in [Kacianka and Pretschner, 2021] we realized that "causality is a prerequisite for accountability" and that causality can be used as a common language to express different accountability definitions, capture their context and then use this formalization to compare and evaluate them. Chapter 5 has our latest definition of responsibility based on causality.

This thesis will show how accountability can be expressed as a causal model and then be identified in the causal model of a system. We explicitly do not take a stance on how to define accountability and recognize that most definitions are slightly different and sometimes incompatible. For some authors even terms like responsibility and accountability can be used interchangeably.[16] Despite the ambiguity of the terms, we feel it valuable to explicate our own understanding of these terms, so that the reader might pinpoint our own bias in dealing with different definitions. This is our understanding of the main terms and their relation:

A **Structural Causal Model** (SCM – see Chapter 3) is a model of the relevant causes and effects that a system can exhibit. A causal relation is necessary for accountability, but in itself it is not sufficient to ensure accountability.

A **cause** is the *actual cause* in an SCM as determined by the HP definition of actual causality (see Chapter 3). Similarly, **to cause** means that an endogenous variable in an SCM is the (or part of the) *actual cause* of another endogenous variable. Causes are purely

[16] See [Kacianka et al., 2020] for a discussion.

technical without any notion of intent or other social attributions.[17]

Power is the (causal) influence one node has on another, which means it is a causal relation between two variables in an SCM.

Responsibility is the commitment of an entity to act a certain way and the ability to affect or change an outcome.[18] This entity is **responsible** for a certain outcome. As such we have explicit notions of *normality* that are used to determine responsibility. This entity is then responsible for an outcome when, had it acted *normally* the outcome would not have happened.

Blame is a social process in which an agent violates a specific notion of normality and thus causes an unwanted event.[19]

A **transparent** system will have an SCM available and log enough data to set the context of the causal model after some event. **Transparency** indicates that an SCM is available, although it makes no statement about the quality of the SCM.

Liability means that we have a natural or legal person,[20] called an agent, that is *responsible* for some outcome. This *responsibility* is made *transparent* with an SCM and thus allows a principal to ask this agent for its account and *blame* it for an unwanted outcome. This means we can actively question the agent and understand it so that our attribution of blame is no longer purely subjective, but derived from objective facts.

Accountability implies that we have an agent that was given some power by a principal, i.e., it requires a causal relation between the principal and the agent. This principal does not fully trust[21] the agent to behave according to their wishes and thus limits the agent's power by requiring the agent to account for its use. In this thesis, we use SCMs to model the possible power of a system and thus make it *transparent*. An **accountable system** is a socio-technical system in which the power of the system is restricted with an **accountability mechanism**. The **accountability definition** describes the means by which the power is limited. These means will show up as structures in the SCM.

[17] However, it is important to note that causes are always relative to a causal model. The causal model might be biased and thus social attributions can leak into the model.

[18] When this commitment is derived from moral reason, the term "duty" will often be used. For technical systems, the commitment would be their specification that describes their "normal" behavior.

[19] [Chockler and Halpern, 2004] were the first to look at representing responsibility and blame in SCMs. In this thesis, we use the version given by [Halpern, 2016]

[20] In principle machines can be assigned responsibility and be liable. However in practice, we sue a natural or legal person. One idea to resolve this is to give legal standing to machines.

[21] [Hall et al., 2017] find that lower levels of trust demand higher levels of accountability and vice versa.

3 Causality

3.1 Introduction

The study of causality[22] is of a specific interest to the study of accountability, our main subject matter, because causality is a prerequisite for accountability. In understanding how causal effects and influences work, we can improve the design of our systems to make sure that effects of causes are clearly understood and then in turn ensure that the causes of effects are easy to identify. The first notion is prospective and the second one retrospective.[23] Both are deeply intertwined, but often only studied separately. In this thesis, we combine the study of both and build on the assumption that a good prospective causal model is also a good retrospective causal model. For prospective models, we can find structures and patterns that allow us to show that some variables are not relevant to certain outcomes and once a specific outcome comes to pass, we can use retrospective reasoning to identify the concrete cause, relative to the given context.

Historically, causality and causal relationships are tightly connected to statistics. However, as laid out by [Pearl, 2000, p. 25], whereas statistical relations are epistemic and describe what we know or believe about the world, causal relationships are ontological, meaning that they describe objective constraints on the world. This means that causal relationships are much more stable and should not change if the environment changes. Still, causality is tightly linked to statistics as many "causal statements are uncertain" [Pearl et al., 2016, p. 7] and are thus often only true with some probability.[24] As such, many prospective causal models will answer questions with a certain probability. For accountability, we often need exact and retrospective answers. For instance, the question "Did Alice cause the crash?" should have a clear retrospective answer. For this we use Actual Causality [Halpern, 2016]. It allows us to use a (prospective) causal model and reason about it in a specific context. For example, we might have a prospective model

[22] In this thesis, we closely follow the definition of Structural Causal Models (SCMs), as introduced and refined over the years by Judea Pearl: [Pearl, 2000, Pearl, 2009, Pearl et al., 2016, Pearl and Mackenzie, 2018] and Joseph Halpern [Halpern, 2016]. Other definitions of causality exist and might sometimes even be more suitable, but Pearl's SCM approach is the most widely adopted notion.

[23] See also the notes in the first chapter of [Halpern, 2016].

[24] For example, "smoking causes cancer" is true, but a single cigarette is very unlikely to cause cancer.

that shows that texting-while-driving causes accidents. Then we might have the specific context of an accident in which we know that Alice was distracted, because she was texting. We can then set the context for this accident and use actual causality reasoning to find the cause of the accident; in this case Alice caused the accident by being distracted.

In this chapter we will first show how probabilities and causality are connected. This will give us an understanding of type causal models. They are interesting, because only if there is a causal link between two nodes, there can be some form of accountability. Conversely, if we can show that certain parts of a system are causally independent, there can be no accountability relation between them. SCMs now give us the mathematical toolkit to proof this causal independence. In Chapter 3.8, we will then show how we can reason over events that have already happened. This allows us to understand what actor contributed to a cause and how they might be accountable for the result.

3.2 Foundations in Statistics

Table 3.1: Observations of the UAV's behavior.

Observation	Stick moves	UAV moves
1	Yes	Yes
2	No	No
3	No	No
4	Yes	Yes
5	Yes	Yes

Looking at Table 3.1, we can see five observations. Each record shows two observations: if the control stick on the remote control moves and if the UAV moves. Looking at the data, we can see that $3/5$ of the times the stick moves and $3/5$ of the time the UAV moves. Moreover, we can see that whenever the stick moved, the UAV moved. So seeing the stick move leads with certainty to seeing the UAV move (or vice versa!). We can express this as a probability. $P(S)$ is the probability that the stick moves and $P(U)$ the probability that the UAV moves. $P(S \mid U)$ is the probability that the stick moves given that we see the UAV move and $P(U \mid S)$ the probability that the UAV moves given that we see the stick move. Figure 3.1 depict this model graphically.[25] Formally, they are expressed [Pearl et al., 2016, p. 12]:

[25] Graphical models have emerged as the standard representation of SCMs. They are based on Bayes Networks and have been found to be a convenient, economical and efficient representation [Pearl, 2000, p. 13]. In the literature, the models usually just use single letter variable-name and rounded nodes.

Formula 1 (Bayes Rule)
$P(S \wedge U) = P(S \mid U)P(U)$
$P(U \wedge S) = P(U \mid S)P(S)$
and logic now gives us the Bayes rule:
$P(S \mid U)P(U) = P(U \mid S)P(S)$

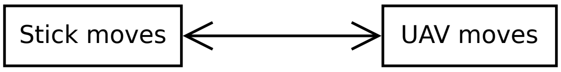


Figure 3.1: The stick's movement and the UAV's movement will be correlated.

The data in Table 3.1 is perfectly correlated, which means that whenever we see the stick move, we will see the UAV move, and vice versa. In this case, the Bayes rule is not particularly useful, as the inverse probability is always 1. To see its use, Table 3.2 has line 6 in which the UAV moves, but the stick does not. The UAV might be affected by a gust of wind or a similar event.

Table 3.2: Line 6 has an event in which the stick does not move, but the UAV does.

Observation	Stick moves	UAV moves
1	Yes	Yes
2	No	No
3	No	No
4	Yes	Yes
5	Yes	Yes
6	**No**	**Yes**

Here we can see movement of the UAV $4/6 = 2/3$ of the time, if we see the stick move, we still see the UAV move 100% of the time. However, we only see the stick move $3/4$ times when we see the UAV move. So $P(U \mid S) = 1$ and $P(S \mid U) = 3/4$.

At some point we might wonder if the stick influences the UAV or if the UAV influences the stick.[26] So we will try to hold one constant and change the other. This is called an *intervention* [Pearl et al., 2016, p. 53ff]. Formally we can use the do-operator to express that we manipulate one variable. $P(X \mid do(Y = 1))$ can be read as "probability of seeing X given that Y is set to 1". $do(Y)$, without a value, denotes that Y will be set to an arbitrary, but fixed, value. In a graphical model, an intervention is expressed

[26] In this example, simply recording the time series would also suffice. However, if you had two potential causes this would no longer work: As an example, imagine that the moment the pilot moves the stick, they also also tilt their head in the direction they want to go. In a such a case analyzing the time series alone will not yield a clear result.

by removing the inbound edges to a node, thus making it independent of any other variables.

Formula 2 (Cause vs. Probability)
If S is a cause, we expect the probability of U to change given that we see S:
$P(U \mid S) \neq P(U \mid do(S))$
As U is not a cause for S, we will see:
$P(S \mid U) = P(S \mid do(U))$

3.3 Structural Causal Models

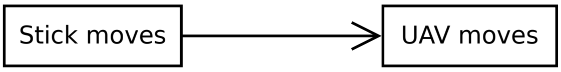


Figure 3.2: A graphical model of the SCM that expresses that the stick's movement *causes* the UAV's movement.

Causality can be formalized in so called *Structural Causal Models* (SCMs) [Pearl et al., 2016, p. 26f.]. They are are derived from structural equation models (SEMs) (e.g., [Lomax and Schumacker, 2004]), but their relations have a direction. Following [Pearl et al., 2016, p. 26f.], an SCM consists of two sets of variable, $\mathcal{U}$ and $\mathcal{V}$ and a set of functions, $\mathcal{F}$, that assigns each variable in $\mathcal{V}$ a value based on the value of the other variables in the model. Formally [Pearl et al., 2016, p. 26f.],

Definition 1 (Structural Causal Model)
A structural causal model M is a tuple $M = (\mathcal{U}, \mathcal{V}, \mathcal{F})$, where

- *$\mathcal{U}$ is a set of* exogenous *variables,*
- *$\mathcal{V}$ is a set of* endogenous *variables,*
- *$\mathcal{F}$ associates with each variable $X \in \mathcal{V}$ a function that determines the value of X given the values of all other variables.*

Every SCM is associated with a *graphical causal model* called the "graphical model" or the "graph" (see Figure 3.2). The semantic of an edge $A \to B$ in the graph is that A is a cause of B; we say that alone or together with other variables A *causes* B. Any further semantics on how exactly A influences B are given in the model's structural equations. These equations describe the causal mechanism. They might model that A is a stone that hits B, they might be a number that gets added to B, or anything else that we wish to express. There are no limitations or constraints on the kind and type of mathematical functions. Equations however, are not directed. This means that the equations alone can show correlations, but not causal connections. One advantage of SCMs is that the structure of the graph alone is enough to identify patterns and causes.

In the graphical model, we adopt the standard graph-terminology and use genealogical terms to describe relationships. If a directed edge starts in node A and its arrow ends in B, we say that A is the parent of B and, conversely, that B is a child of A. A (directed) path[27] is a sequence of distinct edges that are directed in the same direction and join a sequence of distinct nodes. The first node in the path is the ancestor of every node in the path and every node on the path is a descendent of the first node. In an SCM, exogenous variables, denoted by $\mathcal{U}$, are external to the model, meaning that we chose not to explain how they are caused. They are the root nodes of the *causal graph* and are not descendant of any other variable. Endogenous variables, denoted by $\mathcal{V}$, are descendants of at least one exogenous variable and model components of our system and the world for which we want to explain causes. $\mathcal{F}$ describes the relationships between all those variables. If we knew the value of every exogenous variable, we could use $\mathcal{F}$ to determine the value of every endogenous variable. In a graphical model every node represents an *endogenous* variable and arrows represent functions from $\mathcal{F}$ between those variables; the functions are the structural equations.

In the example in Figure 3.2, $\mathcal{V}$ would contain two variables: *Stick* and *UAV* and the graphical model would tell us that we, as modelers, assume that the *Stick* causes the *UAV* to move and not the other way around.[28] $\mathcal{F}$ would contain a function that describes how changes of *Stick* affect *UAV*; one example would be a linear function where *Stick* (S) changes *UAV* (U) by a factor of 0.5: $U = 0.5 \times S$; however, it could also be a boolean function that just states that S moves U. The equations have no direction, so we could also transform them to say $S = 2 \times U$. This would mathematically be fine, but violate our graphical model.[29] Exogenous variables are usually not explicated in the graphical model, and $\mathcal{U}$ would contain at least two variables $\mathcal{U}_{Stick}$ and $\mathcal{U}_{UAV}$. Each of them would model external influences on the variable. $\mathcal{U}_{Stick}$ would, for example, model that the pilot moves the stick.

In a causal model, a variable is the *direct cause* of another variable, if it appears in the function that assigns its value [Pearl et al., 2016, p. 26]. In a graphical model, it holds that if B is a child of A, A is a direct cause of B. However, if B is a descendent of A, i.e.,

[27] Since we only talk about directed graphs, every path is a directed path.

[28] As shown above, the data alone would also allow us to conclude that the *UAV* causes the *Stick* to move.

[29] The graphical model should conform to reality. If arrows are wrong, in principle, this can be uncovered with experiments.

a structure like $A \rightarrow C \rightarrow B$, then A is a potential cause of B. This means that causality is not always transitive.[30] It is important to note that the absence of arrows means that there is no causal relationship between two variables.

Definition 2 (Variable)
Variables in an SCM are the result of random events and can take multiple values. They can be discrete or continuous and will, especially in examples, often be binary. In the graphical model, they are depicted as nodes.

Typical examples are *age* or the probability for something. Especially when taking about socio-technical systems, we will deal with *categorical variables* where a variable can take one of a finite number of possible values. Typical examples are *Suzy throws* that can either be *true* or *false* or more abstract variables such as *UAV manufacturer* that can have values such as *Build the system correctly*, *Did not build the system at all*, *Paid for better Q&A*.

Finding the right granularity or level of abstraction for a variable in a given context is an open problem. Coming back to the variable *UAV manufacturer* above, it might make more sense to split it into several variables that model the inner workings of the company and help us to understand the interaction of departments and individuals.

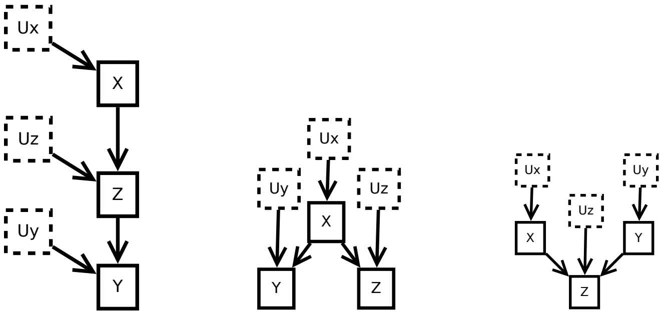
Definition 3 (Event)
An event *is any assignment of a value or set of values to a variable or set of variables [Pearl et al., 2016, p. 8].*

This means that *Suzy throws = true* is an event, but also that, counter to our every day use, *Peter = 25 years old* as well. Similarly, *UAV manufacturer = Build the system correctly* would be an event. Here, again we run into the problem of granularity: of course the manufacturer can *build the system correctly* and *pay for better Q&A*. Depending on the focus of the model it makes sense to either model this in more detail or add more possible values such as *Build the system correctly and pay for better Q&A*.

3.4 Chains, Forks, and Colliders

In graphical causal models, some basic structures will arise repeatedly. We can use them to analyze causal and, by extension, accountability relations and, for example, show that a certain variable cannot be the cause for an event and can thus also not be accountable for it. Chains, forks and colliders, exhibit very specific rules for causal (in-)dependence [Pearl et al., 2016, p. 35ff]. Figure 3.3 depicts these three structures. In

[30] See [Pearl et al., 2016, p. 38] for an example.



(a) Chain. (b) Fork. (c) Collider.

Figure 3.3: Three common structures in causal models. Ux, Uy, and Uz denote the exogenous variables. X, Y, and Z are the endogenous variables.

contrast to the usual convention, we here also model the exogenous variables Ux, Uy, and Uz. In most causal models, only the endogenous variables, here X, Y, and Z are modeled. Exogenous variables "stand for any unknown or random effect that may alter the relationship between endogenous variables" [Pearl et al., 2016, p. 36] and they are assumed to be independent of each other.

The important advantage of causal models is that in these structures several (in-)dependencies hold, regardless of the function between those variables. If we now look at Figure 3.3a, Z is always dependent on X. So, if we see the value of X change, we will usually also see the value of Z change. Next, Y is *likely* dependent on X. The reason for this is that Y depends on Z for its value and Z depends on X for its value. However, there are rare cases where changes in X will not affect Y.[31] And finally, X and Y are independent, conditional on Z. The reason is that conditioning on a variable means that we fix its value. If we had a dataset (such as Table 3.3 below), we would only look at the values of X and Y, where Z has a specific value. What happens here is that Uz changes to keep Z at this specific value. So whenever X changes, Uz would compensate for that change to keep Z constant and as Y only depends on Z and not on Uz, its value is independent of the changes in X.

Figure 3.3b depicts a fork. Here, Y and Z both depend on X and this also means that Y and Z are *likely* dependent. The reason is that, since both depend on X, a change in one will inform us that the other will also likely change. However, there are also cases where this is not the case. Finally, Y and Z are independent, conditional on X. Similar to the chain above, once we hold X constant, a change in either Y or Z does no longer indicate a change in the other. In this structure, X is called a *common cause*.

[31] See [Pearl et al., 2016, p. 38] for an example.

The third basic structure, a collider, is depicted in Figure 3.3c. Here, we can see that Z is dependent on X and Y and X and Y are independent, because they are in no parent-child relationship and their exogenous variable Ux and Uy are assumed to be independent as well. The most surprising property of a collider is that X and Y become dependent, conditional on Z. While it might be surprising that two independent variables can suddenly become dependent, it can be illustrated with a simple example [Pearl et al., 2016, p. 41]: If, we assume $X + Y = Z$ and X and Y are independent, knowing that $X = 3$ does not tell you anything about Y. However, the moment you know that $Z = 10$, knowing that $X = 3$ lets you deduce that $Y = 7$.[32]

This concept of (in-)dependence can now be generalized for all graphs with the concept of d-separation. This means that two variables are independent if every path between them is *blocked*. Formally [Pearl et al., 2016, p. 46],

Definition 4 (d-seperation)
A path p is blocked by a set of nodes Z if and only if

1. *p contains a chain of nodes $A \rightarrow B \rightarrow C$, or a fork $A \leftarrow B \rightarrow C$, such that the middle node B is in Z (i.e., B is conditioned on), or*

2. *p contains a collider $A \rightarrow B \leftarrow C$ such that the collosion node B is not in Z, and no descendant of B is in Z.*

If Z blocks every path between two nodes X and Y, then X and Y are d-separated, conditional on Z, and thus are independent conditional on Z.

Intuitively, d-separation shows when two parts of a directed graph (hence the d-), are separated by a set of nodes Z. Applied to a causal model, Z is the set of variables we need to condition on, i.e., fix their value, to and make these two sets causally independent. This is useful when we think about how a given model should look like in order that two nodes do not affect each other. If we manage to change the model by either removing the nodes in Z or ensuring that they behave a certain way, we can prove that there is no causal influence and thus that there can be no accountability. For example, if we have a chain $A \rightarrow Z \rightarrow B$ and we can prove that while Z is influenced by A, the influence of Z is independent of A, we can proof that A has no influence on B and thus that A has no power over B and so is not accountable for B.

[32] Of course this would violate the causal direction, see Chapter 3.3.

3.5 Confounder

Confounders are a term used for variables that confound our understanding of a causal relation. Figure 3.4 depicts the scenario in Figure 3.2, but with the addition of a so called *confounder* or *covariate* [Pearl, 2000, p. 78]. This is a variable that will affect both the (suspected) cause and the effect and will confound our understanding of the effect. Put another way, confounders are all factors that obscure the root cause. Unfortunately the term itself is hardly even defined; [VanderWeele and Shpitser, 2013] provide an overview of the intuition and propose several possible definitions in terms of counterfactuals. In our work, we use the following definition:

Definition 5 (Confounder)
A covariate C is a confounder for the effect of A on B if it blocks a backdoor path (see Chapter 3.7.1) from A to B (Adapted from [VanderWeele and Shpitser, 2013, p. 5]).

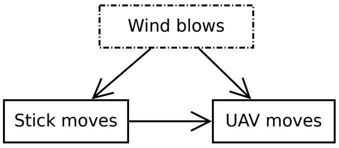


Figure 3.4: The wind (as covariate or confounder) will affect both the stick and the UAV. In this setting it is impossible to disentangle the effects from the wind and the stick.

If we look at the data in Table 3.2, the "confounding effect" would be the last row, that leads to a discrepancy between $P(U \mid S)$ and $P(U \mid do(S))$.[33] In Figure 3.4, we have an arrow from wind to UAV and also from wind to stick. The first arrow is quite intuitive and will account for the discrepancy in the data; the model expresses that the wind will directly move the UAV. The arrow from wind to stick indicates that we assume that the wind also affects the stick commands, for example by making the pilot more careful or overcompensating for expected gust of wind.

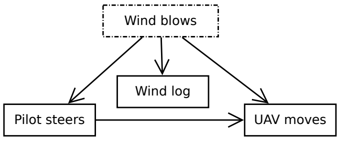


Figure 3.5: If we have a proxy for the confounder, we can use the proxy to control for it.

[33] See also [Pearl and Mackenzie, 2018, p. 150ff.].

In Figure 3.4, wind is depicted in a dotted-dashed box[34] which symbolizes that we cannot directly measure the wind that affects the UAV. However, without knowing the wind, we cannot control for it and try to estimate the causal effect from the stick on the UAV. Figure 3.5 depicts a model in which we have a *proxy* for the confounder. It might be a wind speed sensor that we set up near the pilot or wind data from computer models. This proxy variable will not give us the real value of the confounder, but should provide a useful estimate.

Table 3.3: The fourth column indicates if we observed a strong wind.

Observation	Stick moves	UAV moves	Strong Wind
1	Yes	Yes	No
2	No	No	No
3	No	No	No
4	Yes	Yes	No
5	Yes	Yes	No
6	No	Yes	**Yes**

Table 3.3 includes a column for the wind. Going back to Definition 4 above, we can see that we have a fork, $Stick \leftarrow Wind \rightarrow UAV$, in our model making $Stick$ and UAV marginally dependent.[35] This means that both are affected by $Wind$ and that just looking at one would tell us something about the $Wind$ and thereby also tell us something about the other.[36] However, now that we know the value of the wind, we can condition on it and thereby block this information flow (or d-separate the nodes), making $Stick$ and UAV marginally independent [Pearl, 2000, p. 17]. In practical terms, to condition on the wind, we would remove all rows where there was a strong wind. So we would remove line 6 and end up with Table 3.1. We can then again use this table to estimate the causal effect of the stick on the UAV.

Confounders will often feature in SCMs and are important for accountability, because they make it impossible to understand the influence one node has over another. When designing accountable systems, confounders should be avoided wherever possible. If we cannot avoid them, we can look for ways to de-confound the effect of one variable on another. However, this will typically require additional logging facilities.

[34] In contrast to exogenous variables, which are just dashed.

[35] More precisely, that our data is consistent with a causal model such as Figure 3.4. Data alone is not sufficient to build a causal model, it needs to be enriched with the modeler's understanding of causal connections. However, if the data is not consistent with a causal model, we know that the model is wrong.

[36] The effect of $Wind$ will likely be very indirect, for example by making the pilot jittery and try to anticipate gusts of wind.

3.6 Mediator

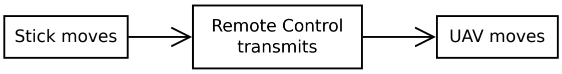


Figure 3.6: The stick will not directly move the UAV; the remote control is a *mediator*.

Definition 6 (Mediator)
A mediator, sometimes also called an indirect effect*, is a variable between a cause and an effect. [Pearl et al., 2016, p. 114].*

To better model the real world, Figure 3.6 includes the remote control as a *mediator*. It is a mediator, because might amplify or modify the initial input. [Baron and Kenny, 1986] define that a "variable functions as a mediator when it meets the following conditions: (a) variations in levels of the independent variable significantly account for variations in the presumed mediator (. . .), (b) variations in the mediator significantly account for variations in the dependent variable (. . .), and (c) when [the mediator is] controlled [for], a previously significant relation between the independent and dependent variables is no longer significant (...)" In this example, the stick is the independent variable, and the UAV is the dependent variable. The Stick does not influence the UAV directly, but uses a certain mechanism, here remote control, to do so. Would we control for the remote control, we would screen off any causal effect between the stick and the UAV.[37] In many real world applications we will see such mediators.

Table 3.4: The remote control is a mediator.

Observation	Stick moves	UAV moves	Remote turned on
1	Yes	Yes	Yes
2	**No**	**No**	**No**
3	No	No	Yes
4	Yes	Yes	Yes
5	**Yes**	**No**	**No**

Table 3.4 is an example where the remote control is a mediator for the stick. So the stick affects the remote control, which then in turn moves the UAV. If we were now to condition on the remote control, for example by only looking at the lines where it is turned off, we would only look at lines 2 and 5. These lines show that no matter what we do with the stick, the UAV will not move. We would need to look at the complete picture to understand the causal relationship.

[37] See the definition of a chain in Chapter 3.4

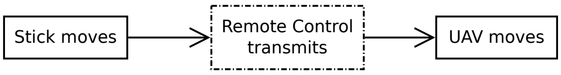


Figure 3.7: Often we cannot directly observe the mediator.

Figure 3.7 depicts the same model, but the mediator is denoted by a dotted-dashed box. This indicates the fact that we will often not be able to observe the mediator directly. In our example, we can observe what the stick does and we can observe what the UAV does. We have no knowledge of the inner workings of the remote control and cannot tell if it works “properly”. From an explanatory point of view, this model is similar to the model in Figure 3.2. However, by indicating that we assume there to be a mediator, we can work on trying to measure it.

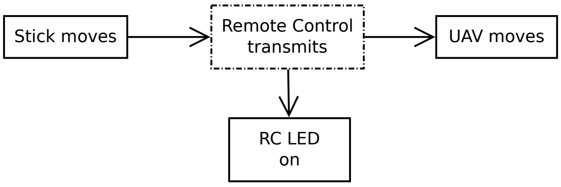


Figure 3.8: If we cannot observe the mediator, we might be able to introduce a *proxy mediator*. Here it is an indicator LED for the remote control.

To understand the remote control, we can use a *proxy mediator* [Pearl and Mackenzie, 2018, p. 153f]. In Figure 3.8 an indicator LED for the remote control fulfills that role. It is important that the proxy mediator is only influenced by the mediator and not by the cause or the effect. If we have such a proxy mediator, we can use it in our causal analysis. This proxy mediator should be affected by the mediator and allow for a more precise estimation of the mediator. However, since it is not exactly the mediator, it too will contain some error. In our example, the indicator LED might be broken and sometimes not glow when the RC is turned on. Similar to confounders, mediators are a common feature in SCMs and the real world. They are relevant in accountability contexts, when an entity exerts its power to indirectly cause some effect, for example by using a CPS to reach some goal or ordering a person to fulfil some task.

3.7 Analyzing Causal Models

[Pearl and Mackenzie, 2018, p. 157] presents four rules that hold in SCMs:

1. *In a chain junction, $A \to B \to C$, controlling for B prevents information about A from getting to C or vice versa.*

2. *Likewise, in a fork or confounding junction, $A \leftarrow B \rightarrow C$, controlling for B prevents information about A from getting to C and vice versa.*

3. *In a collider, $A \rightarrow B \leftarrow C$, exactly the opposite rules hold. The variables A and C start out independent, so that information about A tells you nothing about C. But if you control for B, then information starts flowing through the "pipe", due to the explain-away effect*[38].

4. *Controlling for descendants (or proxies) of a variable is like "partially" controlling for the variable itself. Controlling for a descendant of a mediator partly closes the pipe; controlling for a descendant of a collider partly opens the pipe.*

In this quote, a "pipe" refers to a path through a causal graph. What is interesting is that, even if we have a long causal path, it is enough that one junction blocks the information flow. To deconfound two variables we need to block any noncausal path while not blocking any causal path. This leads to two prominent criteria to identify causal independence: The Back-Door and the Front-Door Criterion.[39]

These rules hold in any SCM, so any graph that describes causal relationships. To illustrate that, imagine $A \rightarrow B \rightarrow C$ describes how billiard balls interact. If B is not controlled for (i.e., fixed), C "knows" that A must have hit B before B could have hit C. If we move B regardless of the fact that A hit it, this connection is broken. Similarly, in a junction ($A \leftarrow B \rightarrow C$), A and C "know" about each other's changes through B. In the example one ball would hit two other balls and thus each ball would know that the other is moving as well and with what velocity. Were we to replace B with a disconnected impulse on A and C, this information would no longer be available to them. In a collider ($A \rightarrow B \leftarrow C$) A and C affect a third variable, B, and without knowing B's value they have no idea of the others' value. A simple example is an addition, like $B = A + C$. The moment we know B, both A and C can compute the value of the other by subtracting their own value from B (i.e., $C = B - A$). The forth rule specifies that a proxy for a variable is not as precise as the variable itself, but will still provide some information related to the actual variable, because it depends on that variable.

[38] This is rooted in the nature of a collider. Knowing one cause and the common result of two independent causes will give us information about the other cause. See [Wellman and Henrion, 1993] for a detailed explanation.

[39] All of the following examples are based on examples given by [Pearl and Mackenzie, 2018]; [Tikka and Karvanen, 2018] provide a tool to automate the analysis.

3.7.1 The Back-Door Criterion

Definition 7 (The Back-Door Criterion)

Given an ordered pair of variables (X, Y) in a directed acyclic graph G, a set of variables Z satisfies the back-door criterion relative to (X, Y) if no node in Z is a descendant of X, and Z blocks every path between X and Y that contains an arrow into X.

Intuitively, the back-door criterion [Pearl et al., 2016, p. 61] ensures that (1) all spurious (i.e., associated, but not causally related) paths between X and Y are blocked, (2) all directed paths from X to Y are not perturbed (i.e., not affected by confounders), and (3) no new spurious paths are added.[40]

In the following, we show a few examples where the backdoor criterion is used to simplify the causal reasoning and sometimes ignore the effect of some variables altogether. In these examples, the nodes represent variables that might take arbitrary values and arrows represent the fact that there exists a causal relation between two variables, i.e., $A \to B$ means that A is an input in the structural equation that defines B. In these examples the possible values for the variables are not relevant. A node named Pilot might have a categorical value such as "starts the UAV" or "pays attention" or even a continuous variable like 0.5. To understand the causal effect and possibly rule out some causal effect we do not need this level of detail.

Example 1

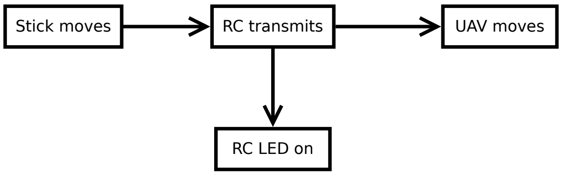


Figure 3.9: A pilot issues remote control commands to the engines of a UAV. The LED indicates if the RC is on or not.

Figure 3.9 depicts a simple causal model similar to the examples given above. There is a causal effect of the stick on the UAV and the RC status is indicated with a LED. The stick is the cause, the RC is the mediator, the UAV movement is the effect and the RC LED is a proxy for the mediator [Pearl and Mackenzie, 2018, p. 153f]. The RC LED is not a confounder and does not need to be controlled for. On the contrary, were we to control

[40] [Greenland et al., 1999] give a detailed and in-depth explanation of the Back-Door Criterion and why it works on graphical models.

for it, we would break the causal connection between the stick and the UAV. Here, we do not need to use the back-door criterion, because we have no possible confounders.

Example 2

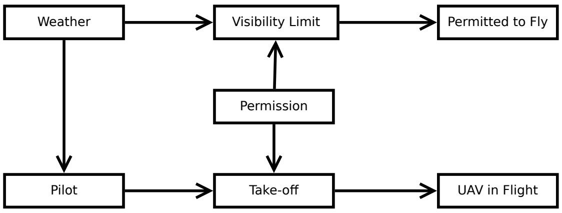


Figure 3.10: Conditions that influence the legality of a UAV's flight.

The scenario in Figure 3.10 models decisions on whether a UAV should be allowed to take off. The pilot is the cause, take-off is the mediator and UAV in flight is the effect. Weather, Visibility Limit, Permission and Permitted to Fly are, in the terms of Pearl, "pretreatment variables"; in our example the term "environmental variables" might be more suitable. The weather will affect the pilot as well as general visibility conditions. The permit (e.g., to fly in restricted airspace) specifies if a pilot is allowed to take off, but might also allow pilots to fly in low-visibility conditions. If a flight is permitted depends on the visibility and the obtained permit, so a specific permission will reduce the allowed visibility limit. Since there are no open back door paths, we do not need to control for any variable to understand the causal effect of the pilot on the UAV. Another option would be to control for Permission and Permitted Fly. In both variants, the pilot's decision is the sole cause for the effect of the drone being airborne.

Example 3

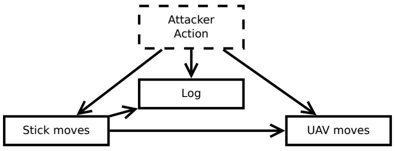


Figure 3.11: An attacker affects the stick and the UAV.

In Figure 3.11, the attacker is a confounder, as it is on the back-door path Stick ← Attacker → UAV. If we cannot control for it, we cannot ascertain whether the stick caused the UAV's direction or not. The log, however, may serve as a proxy for the unobserved attacker. Controlling for the log can eliminate some confounding bias, but might also introduce a collider bias.

3.7.2 The Front-Door Criterion

Definition 8 (The Front-Door Criterion)

A set of variables Z is said to satisfy the front-door criterion relative to an ordered pair of variables (X, Y) if

1. *Z intercepts[41] all directed paths from X to Y.*

2. *There is no unblocked path from X to Z.*

3. *All back-door paths from Z to Y are blocked by X.*

Intuitively, the front-door criterion [Pearl et al., 2016, p. 69] relies on the fact that one can identify the effect of X on Z and the effect of Z on Y separately. This works, because Z, so the mechanism (or mediator) by which X affects Y, is not affected by any unobserved confounders.[42] Having identified the separate effect, we can then calculate the effect from X on Y.[43]

Once some set Z satisfies the front-door criterion, we can use the front-door adjustment to identify the causal effect of X on Y. Formally [Pearl et al., 2016, p. 69],

Definition 9 (The Front-Door Adjustment)

If Z satisfies the front-door criterion relative to (X, Y) and if $P(x, z) > 0$, then the causal effect of X on Y is identifiable and is given by the formula

$$P(y \mid do(x)) = \sum_z P(z \mid x) \sum_{x'} P(y \mid x', z) P(x')$$

Example 6

Figure 3.12 is different from the other examples, as we cannot use the back-door adjustment to control for the confounder wind. Instead we can use the front-door criterion [Pearl and Mackenzie, 2018, p. 224]. Using the RC commands, the Engine RPM and the battery use, which we can log and observe, we can estimate the effect of the RC commands on the battery use, without explicitly controlling for the wind. Were the wind

[41] This means that all paths from X to Y contain at least one node that is an element of Z.

[42] The Front-Door Criterion will also work if Z is only weakly affected by a confounder. The results will, however, get more imprecise the bigger Z is influenced.

[43] See [Bellemare and Bloem, 2019] for an in-depth discussion of the application of the front-door criterion.

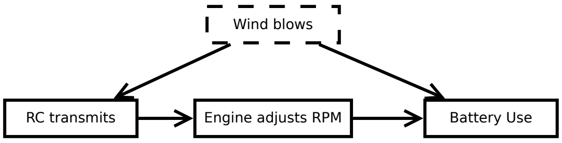


Figure 3.12: Wind influences the RC commands (to counteract the wind force) and thereby the battery use. It does not influence the engine RPM.

to affect the RPM only very weakly, we could still use the front door criterion, albeit with a slight bias.

To illustrate the use, we present the example with some numbers.[44] To simplify the example, we will assume the model to be linear. Here, we assume that we do not or cannot measure the effect of the $Wind$, but still want to understand the effect of the RC on the $battery$ use. For this to work, the $wind$ must not influence the RPM in any way.[45] Here, we have the SCM for the example:

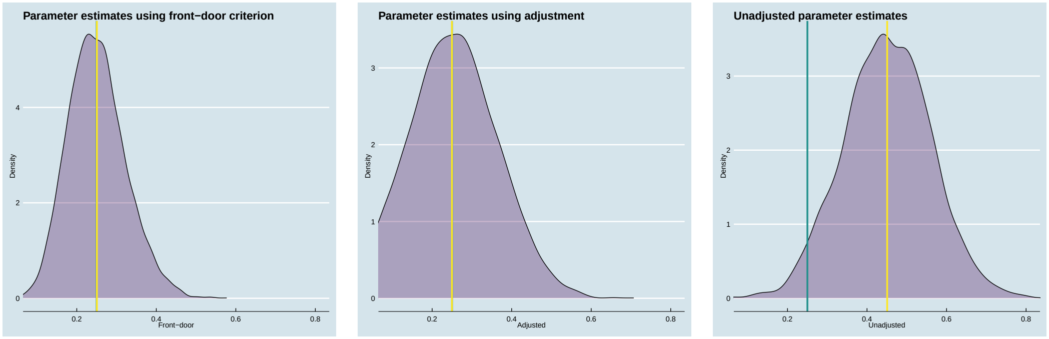
$$\begin{aligned} Wind &\sim N(0,1) \\ RC &= \mathcal{U}_{\text{Input}} + .5 * Wind + \epsilon_{RC} \\ RPM &= \mathcal{U}_{\text{Base}} + .5 * RC + \epsilon_{RPM} \\ Battery &= .5 * RC + .5 * Wind + \epsilon_{Battery} \\ \epsilon_{RC}, \epsilon_{RPM}, \epsilon_{Battery} &\sim N(0,1) \end{aligned}$$

In this model, the strength of $wind$ is normally distributed and will range from no wind (0) to 1 (a hurricane).[46] The variables ϵ_{RC}, ϵ_{RPM} and $\epsilon_{Battery}$ are independent error terms and model any error in the measurement we might have; we assume them to be normally distributed as well. The variable RC indicates how much thrust the pilot needs to exert to keep the UAV on course. We call the pilot's input $\mathcal{U}_{\text{Input}}$. However, to do so it needs to counteract $.5 * Wind$ force. Changes in RC have some effect on RPM, however just half of RPM depends on RC, the rest of the power is drawn by the system's base load ($\mathcal{U}_{\text{Base}}$) that is required to keep the UAV airborne. The battery drain now causally depends on both the commands of the RC that might cause faster spinning propellers, and the $Wind$ that might cause more battery drain by requiring more power to just hover

[44] Our example largely follows [Thoemmes, 2018].

[45] If the influence is only very minor the front-door criterion will still yield results, just less precise ones [Pearl and Mackenzie, 2018, p. 230].

[46] We do this for simplicity, Wind could also be a speed in m/s and would in reality not be normally distributed.



(a) The front-door adjustment fit almost perfectly.

(b) Knowing the effect allows us to be precise.

(c) A naive regression will give a wrong result.

Figure 3.13: Three plots of the estimate of the true effect of RC on $Battery$; the green bar is at the true effect of 0.25. The plots are re-created after the example given by [Thoemmes, 2018].

in place. For simplicity, we $\mathcal{U}_{\text{Input}}$ assume to be 0, indicating that the pilot just wants the UAV to hover in place and $\mathcal{U}_{\text{Base}}$ to also be 0, indicating no base load.

The true total effect of RC on $Battery$ is 0.25, but without being able to measure $Wind$ we do not know this. If we used an linear regression model to estimate the effect from RC on $Battery$, we would get a result similar to Figure 3.13. Here we can see that adjusting using the front-door criterion gives almost the exact effect on RC on $Battery$, even without taking any information on $Wind$ into account. Knowing this, we can now construct our UAV in such a way that we do not log anything about $Wind$. This would still allow us to ensure the accountability of the RC, according to Lindberg's definition (see Chapter 2.2).

In our example, we used a linear model and normally distributed variables. It is important to highlight that the front-door criterion works, regardless of the underlying distributions or the linearity of the model. So if we design an architecture where the causal effect of one component on the other is important, such as for accountability, we can decide not to log the $Wind$. If we, however, log this data anyway, for example for debugging purposes, we can of course use it to determine the cause. This then allows us to forgo the requirement that RPM is independent of the $Wind$.

3.8 Actual Causality

Up until this point, we looked at causality to make predictions how events will unfold. This is useful, because we want to build systems in a way that they *will probably be* accountable. To achieve this, we try to make sure that the causal effects within a system are clearly understood and that the structure ensures that as many components as possible are causally independent. However, what if an unwanted event has already happened? In this case we do not care about the probabilities of events, we know it happened, but we want to find out why exactly it did happen. For this we need the concept of *actual causality* [Halpern, 2016].[47] It is backward-looking and stands in contrast to *type causality* which is forward-looking[48]. Actual causality talks about specific instances and is thus less useful for making predictions.[49]

The classic way to illustrate actual causality is the Suzy-Billy rock throwing example:[50] Imagine two kids, Suzy and Billy, each throwing a rock at a bottle. Suzy's rock hits the bottle first, shattering it. However, because both kids have perfect aim, Billy's stone would have shattered the bottle had Suzy not thrown. Figure 3.14 depicts this example graphically. This example is designed to show that the simple but-for test[51] is not adequate to attribute causality. In the Suzy-Billy example, the but-for test fails because the bottle would have shattered, even if Suzy had not thrown the stone.[52] This would yield that Suzy is not a cause for the bottle to shatter, which is counter to our understanding of causality. The goal of the Halpern-Pearl (HP) definition of causality is now to find a precise mathematical definition to enable algorithmic reasoning over such examples such that the result confirms with our human understanding of causality.

As a first step, the HP definition[53] uses the following formalization of a causal model,

[47] This chapter is mainly based on [Halpern, 2016], which in turn draws heavily on [Halpern and Pearl, 2005a], [Halpern and Pearl, 2005b] and [Halpern, 2015].

[48] Note that actual causality can also deal with probabilities for retrospective events; see [Halpern, 2016, Chp. 2.5].

[49] [Halpern, 2016] writes that *"the distinction between actual causality and type causality has been related to the distinction between causes of effects—that is, the possible causes of a particular outcome—and effects of causes—that is, the possible effects of a given event. Some have claimed that reasoning about actual causality is equivalent to reasoning about causes of effects, because both typically focus on individual events (e.g., the particular outcome of interest), whereas reasoning about type causality is equivalent to reasoning about effects of causes, because when thinking of effects of causes, we are typically interested in understanding whether a certain type of behavior brings about a certain type of outcome."*

[50] Here we use the formulation given by [Halpern, 2016, Example 2.3.3].

[51] The but-for test is a simple understanding of causality that reads *"A is a cause of B if, but for A, B would not have happened"* [Halpern, 2016]. It is often used in the legal context, called by its Latin name *sine qua non* test. In this domain, several improvements were developed such as the INUS (an Insufficient but Necessary element of an Unnecessary but Sufficient set) or the NESS (Necessary Element of a Sufficient Set) test. For a detailed overview, see [Moore, 2019].

[52] This problem is called overdetermination; see [Halpern, 2016, Example 2.3.2]. for more details.

[53] It is important to note that the HP definition is just one possible way to define causality. As [Halpern, 2016, Chp. 2.2.2] puts it so eloquently after introducing all the details of the HP definition: *"At this point, ideally, I would prove a theorem showing that some variant of the HP definition of actual causality is the "right" definition of actual causality. But I know of no way to argue convincingly that a definition is the "right" one; the best we can hope to do is to show that it is useful."*

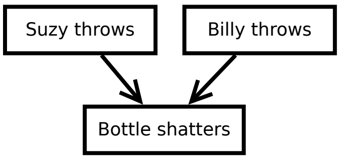


Figure 3.14: The basic model of the Suzy-Billy rock-throwing example.

based on Pearl's work on type causality [Halpern, 2015]:

Definition 10 (Actual Causal Model)
A causal model M is a tuple $M = ((\mathcal{U}, \mathcal{V}, \mathcal{R}), \mathcal{F})$, where

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

A *primitive event*, given $(\mathcal{U}, \mathcal{V}, \mathcal{R})$, is a formula of the form $X = x$ for $X \in \mathcal{V}$ and $x \in \mathcal{R}(X)$. *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 $\mathcal{V}$, and $y_i \in \mathcal{R}(Y_i)$. 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 $\overrightarrow{u}$ (i.e., the specific values of the variables). 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 $(M_{\overrightarrow{Y} \leftarrow \overrightarrow{y}}, \overrightarrow{u}) \models \varphi$ [Kacianka et al., 2019a]. One important difference between type and actual causality is also that in actual causality we need to know the range of possible values for a variable (c.f. Definition 1). Only if we know the range, we can compute alternative courses of events and determine an actual cause, which is defined as [Halpern, 2015]:

Definition 11 (Actual Cause)
$\overrightarrow{X} = \overrightarrow{x}$ 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 $\mathcal{V}$ 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, i.e., no subset of* $\overrightarrow{X}$ *fulfills AC1 and AC2.*

Informally[54] AC1 just says that a specific event $\overrightarrow{X} = \overrightarrow{x}$ actually happened, otherwise it cannot be a cause. The minimality condition AC3 ensures that only relevant events are part of a cause. In the Suzy-Billy example it would remove the detail that Suzy had a green hat on from the cause. AC2 is the most complex condition and is thus traditionally explained last in any text. Here the idea is that we can show that ϕ depends on $\overrightarrow{X}$ as long as we keep the variables in $\overrightarrow{W}$ fixed. This allows us to find that only the variables in $\overrightarrow{X}$ are affecting the outcome.

Coming back to the Suzy-Billy rock-throwing example depicted in Figure 3.14, we would have three endogenous variables: ST for "Suzy throws", BT for "Billy throws", and BS for "bottle shatters". For simplicity, all variables can be true (1) or false (0). Beside these endogenous variables, this model is usually constructed to have a single exogenous variables, U, that determines whether Suzy or Billy threw the stone.[55] U is usually constructed to have two components U_1 and U_2, each saying if Suzy or Billy threw. It would thus have four possible values $(0, 0), (1, 0), (0, 1), (1, 1)$ The model would then consist of three structural equations:

1. $ST = U_1$ indicates that Suzy threw.
2. $BT = U_2$ indicates that Billy threw.
3. $BS = ST \vee BT$ to indicate that the bottle shattered if either Suzy or Billy threw a stone.[56]

In this model, $BT \wedge ST$ would be considered a cause for the bottle shattering. This makes sense if the rocks really hit the bottle at the same time, however what if Suzy's stone hit the bottle just a fraction of a second before Billy's? This model cannot capture this difference. While there are ways to "salvage" the original model,[57] the recommended way is to extend the original model with two variables, *SH* and *BH*, for Suzy and Billy hits respectively, that indicate if either of them hit the bottle. We then would need to change the structural equations in the following:

[54] For an in-depth discussion, of all conditions and especially AC2,[Halpern, 2016, Chp. 2.2.2] is the most thorough resource.

[55] Note that the exogenous variables are usually omitted in discussions of causal models. Often times there might be one exogenous variable for each endogenous leaf variable. In the case of Suzy and Billy, for example, you could have two exogenous variables, U_s and U_b setting the value of ST and BT.

[56] These examples assume that Suzy and Billy never miss and that there are no other factors that could influence the chain of events. If there were, the model would be wrong and would need to be improved.

[57] See [Halpern, 2016, Chp. 2.3] for a discussion.

1. $BS = 1$, if $SH = 1$ or $BH = 1$
2. $SH = 1$, if $ST = 1$
3. $BH = 1$, if $BT = 1$ and $SH = 0$

Figure 3.15 depicts the resulting model graphically. In this model, Suzy's throw hitting the bottle will *preempt* Billy's throw, thus allowing us to ensure that Billy is not the cause of the bottle shattering.

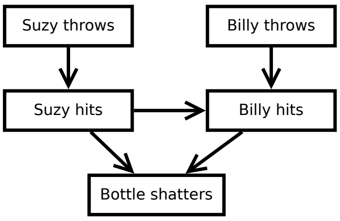


Figure 3.15: A more elaborate model of the rock throwing example, expressing the preemption relation between Suzy's and Billy's throws.

The most important take-away is that it is highly important how we exactly structure our model. Slight differences in the language used to describe them as well as their formalization can have big impacts on the causal reasoning. Even the choice of definition of (actual) causality can alter the attribution of causes significantly. It is thus important that each causal model is discussed and ideally peer reviewed. It is not enough to have a model, but we also need a clear rational for that model. Of course this will then cause second order questions of the correctness of models, the bias of the modelers and even who is allowed to create the models.[58] Here we simply assume that a causal model is a correct and detailed enough representation of reality.

3.9 Assessing the Structure of SCMs

First of all it is very important to note that SCM cannot be proven correct in any theoretical way. They are models of the world and, as aphorism goes, "All models are wrong, but some are useful".

In [Halpern, 2016, Chapter 4], Halpern discusses the *"Art of Causal Modeling"* and is very clear that he does not believe there to be a single "right" model. He illustrates this with an instructional example:

> *For example, suppose that we ask for the cause of a serious traffic accident. A traffic engineer might say that the bad road design was the cause; an educator*

[58] See [Kacianka and Pöchhacker, 2020] for some discussions of this topic.

> *might focus on poor driver education; a sociologist might point to the pub near the highway where the driver got drunk; a psychologist might say that the cause is the driver's recent breakup with his girlfriend. Each of these answers is reasonable. By appropriately choosing the variables, the structural-equations framework can accommodate them all.*

3.9.1 Adding Variables

When we analyze a causal model to identify the cause of some event, the answer is always relative to the given causal model. If the actual cause is not part of the model, it can also not be discovered by any reasoning mechanism. It is important that the variables help us distinguish between possible scenarios. However, too many superfluous variables simply bloat the model and hinder understandability. Unfortunately there is no "sweet spot" or independent criterion to identify the correct number of variables in a causal models. As such each model has to be discussed and every modeling choice has to be argued.

3.9.2 Extending Models

As it would be easy to add variables to an model to get any desired result, Halpern requires extensions to causal models to be "conservative", meaning that new variables might refine the model, but not fundamentally change it.

Formally, [Halpern, 2016, Chapter 4.2]

Definition 12 (Conservative extension of a causal model)
A causal model $M' = ((\mathcal{U}', \mathcal{V}', \mathcal{R}'), \mathcal{F}')$, is a conservative extension of $M = ((\mathcal{U}, \mathcal{V}, \mathcal{R}), \mathcal{F})$, if $\mathcal{U} = \mathcal{U}'$, $\mathcal{V} \subset \mathcal{V}'$, and for all contexts $\vec{u}$, all variables $X \in V$, and all settings $\vec{w}$ of the variables in $\vec{W} = V - \{X\}$, we have $(M, \vec{u}) \models [\vec{W} \leftarrow \vec{w}](X = x)$ iff $(M', \vec{u}) \models [\vec{W} \leftarrow \vec{w}](X = x)$. That is, no matter how we set the variables in M other than X, X has the same value in context $\vec{u}$ in both M and M'.

3.9.3 Choice of Causality Definition

Halpern makes a point that the current definition of actual causality [Halpern, 2015] can be expressed using the old [Halpern and Pearl, 2005a] and add some variables to it. We would like to extend this point and highlight that there are different definitions of causality in the literature. In this thesis, we use the Halpern and Pearl definition, because we believe it conforms best to human intuition and because it can be checked by automatic tools [Ibrahim and Pretschner, 2020]. To the best of our knowledge, the relationship between all those different definitions has not been analyzed in depth, but we assume that, similar to the old and modified HP definition, they can be made to mimic each other given enough additional variables.

3.9.4 Stability

The problem here is that adding variables to a causal model can turn causes into non-causes and vice versa. The examples are intricate,[59] but the core idea is that while this can happen it usually only happens in rare and contrived cases it generally does not happen in real world scenarios, especially when we have a notion of normality.

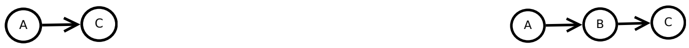
3.9.5 Variable Ranges

Another problem for creating causal models is the range of the variables. In our examples, we will often use binary variables because they are easier to handle in examples and because we can solve them automatically. However, often times other ranges will be appropriate. From a computational point of view, the bigger the ranges, the more complex the calculation of causes, because we would need to check for every counterfactual. This is relatively easy in the binary case, because we can just flip a variable from true to false and vice versa. However, the moment you handle anything infinite, like numbers, in theory your search space explodes to infinity. So we should try to partition the search space and provide upper and lower bounds for variables.

3.9.6 (In-)dependence

Since the analysis of causal models requires us to be able to set variables independently, it is important that two variables are not logically connected. [Halpern, 2016, Chapter 4.6] gives the example of Martha saying "hello". If we now had a second variable that represented the fact that "Martha said 'hello' loudly", these variables would be dependent. A world in which "Martha says 'hello' " is false and "Martha said 'hello' loudly" has no meaning. Such contradictions will, according to Halpern, be quite rare, because most modellers will be able to spot them. However, he cautions that often times variables should be related, particularly if they are mutually exclusive, but this is not represented in the model. As an example, Halpern illustrates this using the classic Suzy-Billy example: "Although, in the actual context, Billy's rock will hit the bottle if Suzy's doesn't, this is not a necessary relationship. Suppose that, instead of using two variables SH and BH, we try to model the scenario with a variable H that takes

[59] See [Halpern, 2016, Chapter 4.4] for details and a formal treatment.



(a) $\mathcal{F} = \{C = A\}$

(b) $\mathcal{F} = \{C = B, B = A\}$

Figure 3.16: Two models that are equivalent with regards to A and C.

the value 1 if Suzy's rock hits and 0 if Billy's rock hits. If $BS = 1$ no matter who hits the bottle (i.e., no matter what the value of H), then it is not hard to verify that in this model, there is no contingency such that the bottle's shattering depends on Suzy's throw. The problem is that $H = 0$ and $H = 1$ are not mutually exclusive; there are possible situations in which both rocks hit or neither rock hits the bottle. ”

3.9.7 Normality

If models use normality, it is possible to define normality such that specific causes are not identified as causes. For example, were we to say that it is normal for brakes to fail, a brake failure in a car would never be seen as a cause for an accident. Here it is very important that any assumption of normality is well argued and criticized. Here again the advantage of causal models is that they make it easy to have a rational discussion about the model and any premises, such as normality.

3.9.8 The Equivalence of Causal Models

Unfortunately it is not straight forward to determine when two causal models are equivalent. Just comparing the structure of the causal graphs will usually not give a correct result. To illustrate this, Figure 3.16 depicts two causal models. In Figure 3.16a, A causes C directly, whereas in Figure 3.16b A causes C via a mediator B. The question now is if those two models are equivalent and if so what this means. If we just look at the graph structure, those two models are different. One has three endogenous variables and the other only two. However, if we look at structural equations of this model, we can see that B does not affect the influence of A on C. So any useful notion of equivalence would need to find that those two models are equivalent. If it would not, this notion of equivalence would be next to useless, because one can always include additional intermediate variables in any causal model. Figure 3.17 now depicts two models that have a similar structure, but are functionally their complete opposite. Any notion of equivalence should find those two models distinct.

[Beckers, 2021b] now splits the question of equivalence into two sub problems. First, he shows that two models are *structurally equivalent*. This notion consideres ancestral relationships and is tolerant of “meaningless” nodes on a path or network between two

(a) $\mathcal{F} = \{C = A\}$

(b) $\mathcal{F} = \{C = \neg\, A\}$

Figure 3.17: Despite their similar structure, those two models are not equivalent.

variables. Additionally, it respects the context of the causal model, by ensuring that the equivalence also depends on the actual setting of the variables. Second, [Beckers, 2021b] defines *functional equivalence* that ensures that the actual values of the functions will also be equivalent.

3.10 Causal Models of Complex Entities

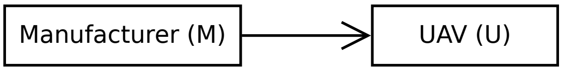


Figure 3.18: The manufacturer has explicit causal influence on the UAV.

When we think of accountability problems, we often do not deal with simple causal relations. For example, the manufacturer has a lot of direct influence on how the UAV will behave and on what it can do. This relation is inherently causal, but also highly complex. It is also on such a high level that we need to ask ourselves what the nodes and edges actually mean. If we just look at the graphical model in Figure 3.18, it will tell us that *Manufacturer* (M) is the cause of the *UAV* (U) and more precisely, that we can compute the value of U by knowing the value of M (if we, for now, ignore any exogenous variables and error terms). This raises two questions: what is a meaningful value range for U and how would U be computed from M. Up until now, we implied very simple semantics for the variables such as "the stick moves" or "increases by 0.5". This need not be the case. U can, for example, be the position of the UAV in 3D-space. Or it could be an n-tuple representing any properties of U that is relevant to the modeler.

The structural equations would then describe how all elements of U get their value and they would only take M as input, no other variable (again, we are ignoring exogenous variables and error terms for now). So if U is the UAV's position in 3D-space, the position could be set by M by physically putting U on that spot. One option to model this would be to make M a tuple (M_1, M_2, M_3) and have a structural equation that assigns $U_1 = M_1$, $U_2 = M_2$ and $U_3 = M_3$.

Both M and U are random variables, which means that they can have a range of possible values and which are distributed in some manner. So if M_1 were continuously and uniformly 5, U_1 would also always be 5. If M_1 were normally distributed in the interval $(0, 100)$, we would see values close to the mean (50) much more often than values on the fringes (0 and 100). It is important to note that when we are looking at the model ex-post, so after-the-fact, the values are already known and no longer a distribution. In this sense, the prospective model shows us possible worlds, whereas the retrospective model allows us to reason about one actual world.

3.10.1 The Abstraction Problem

With this understanding in mind, we still have the problem that *Manufacturer* and *UAV* are highly abstract concepts. M causes the color of U, its design, the source code of the autopilot, and so forth. U can be arbitrarily complex as well; in theory the manufacturer "caused" every molecule of U. In this sense the chosen names of the variables are not ideal, and we could maybe have called them M_{position} and U_{position}. This would improve the comprehension of this particular model, but also require us to build a new model, if we wanted to talk about the color of the UAV. So we would have a model with the exact same structure, just slightly differently named nodes. For example $M_{\text{position}} \to U_{\text{position}}$, $M_{\text{color}} \to U_{\text{color}}$, $M_{\text{developers}} \to U_{\text{software}}$, etc.

The first two at least are intuitively clear, while the mapping from developers to software is again open to interpretation. It tries to express that the manufacturer hired developers that build the UAV's software. This model hides the internal workings of the company, so it ignores managers, hiring decisions, etc. and reduces it to a variable named "developers". This variable could now be a vector with all the code contributions of the single developers. Software could now be, for example, encoded as a single integer value that stands for the binary that came out of the development process.[60] If we wanted to understand the causal contribution of single developers, we could model every developer as an independent node, that contributes to the UAV's software (see Figure 3.19). The structural equation for the software could then look something like $S = D_1 + D_2 + ... + D_n$. The structural equation for a developer could be indicate that the contribution of that developer depends on if the manufacturer hired them or not. So if M is a vector, where each element is 1 if the developer was hired and 0 otherwise, and the contribution is an exogenous variable called $\mathcal{U}_{D_1}$, the structural equation for the first developer would be $D_1 = M_1 \times \mathcal{U}_{D_1}$, ensuring that the developer's contribution would be 0, if D_1 was not hired. The problem with SCMs is that we have no formal means to abstract models and ensure that two nodes are on similar levels of abstraction. SCMs

[60] And strictly speaking, a binary file is nothing more than a very long number that happens to be executable by a CPU.

express the modeler's understanding of the world.

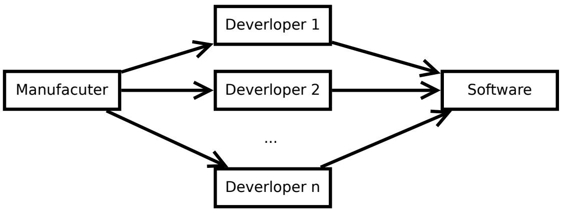


Figure 3.19: Modeling that Developers contribute to a piece of software.

3.10.2 Absent Arrows

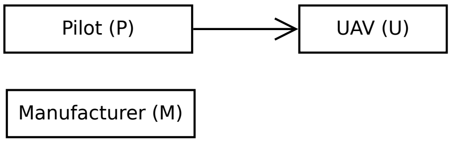


Figure 3.20: The manufacturer has no causal influence on the UAV.

Looking at even very abstract causal models can, however, be very useful if we do not look at the edges that indicate some causal relation, but at the edges that are not there and that indicate a causal independence. Figure 3.20 does not tell us how exact the pilot influences the UAV (or: causes the value of the variable called *UAV*). To understand this, we would need to look at the structural equation and potentially even talk to the modeler. However, what is immediately useful is the *lack of* an edge $M \rightarrow U$, because this indicates that manufacturer has no causal influence whatsoever on U. This means that either U does not represent anything M causes (for example, the likelihood that P sells U) or, if U might be influenced by P, that in this specific model M does not do so. This means that, if this model represents reality, nothing that the manufacturer does (or: regardless of the value of M), the value of U is determined by P (and potentially other exogenous variables or error terms).

4 Modeling CPS

4.1 Introduction

So far, we have introduced the concepts of accountability and causal models. Here we will introduce two more building blocks. First, we will give a quick introduction into Cyber-Physical Systems (CPS) and show some modeling techniques. Finally, we will show that many of the models can be transformed into causal models, which we can then use to reason about the accountability of the system.

4.2 Cyber-Physical Systems

"Cyber-Physical Systems (CPS) are integrations of computation with physical processes" [Lee, 2008]. This means that they use sensors to perceive their environment and actuators to affect it. This concept originally only considered the technical systems. Lately the term Cyber-Physical-Social Systems (CPSS) has gained some traction, explicitly considering the humans as part of the system and its environment [Wang, 2010]. In a recent survey, [Zeng et al., 2020] give a detailed overview of the state of the art in CPSS. It is important to be aware that CPSS often encompass large scale systems like transportation systems, water management or disease control systems. In this thesis, however, we mainly focus on robots and drones, which are a subclass of CPS that are easier to describe and comprehend. While robots, drones and autonomous cars might be part of a transportation system, they are also CPSS in their own right. While all of our arguments apply to systems of any scale, the smaller systems are easier to understand and are thus a better target to introduce accountability and the relevant concepts.

4.3 Modeling CPS

While there are many methods to model CPS,[61] in this thesis we focused on three specific techniques that lend themselves well to be translated into causal models. It is our assumption that most other techniques, especially when they can be translated to graphs

[61] [Graja et al., 2018] provide a recent survey.

and have a graphical representation, can be translated to causal models. However, to show this is still open work. Namely, we will present *Timed Failure Propagation Graphs (TFPGs)*, *Fault Trees (FTs)* and *Attack Trees (ATs)* to model technical properties of a system and *Hierarchical Task Analysis (HTA)* and cognitive architectures like *ACT-R* to represent human behavior. As with technical models, there are a myriad of ways to model human behavior and, as with technical models, we expect that most such models can be translated to causal models, simply because such models describe basic cause-effect relationships. But, again, this has to be shown in future work.[62]

4.3.1 Technical Models

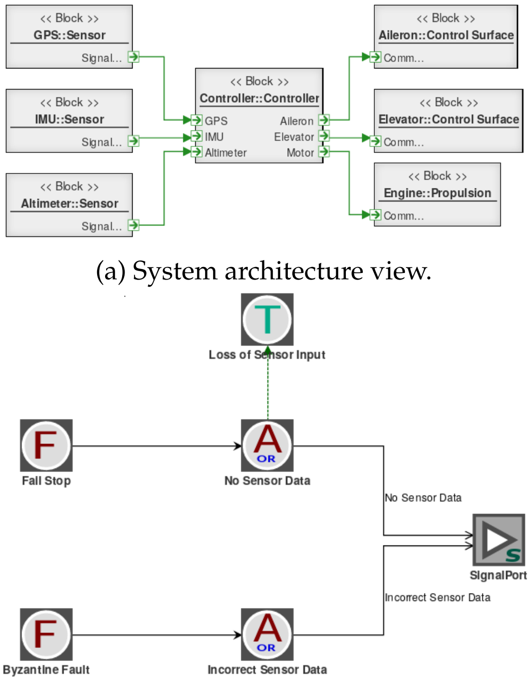
Timed Failure Propagation Graphs

As we have published in [Kacianka et al., 2019a], TFPGs [Abdelwahed et al., 2011, Abdelwahed and Karsai, 2009, Abdelwahed et al., 2005] are, similar to causal models, directed acyclical graphs that model the propagation routes of typical failures in systems. In contrast to many other techniques, they also take modal and temporal constraints into account. "Nodes within TFPGs represent either *failure modes* (i.e., root causes that are not directly observable) or *discrepancies* (i.e., off-nominal effects resulting from failure modes which may be observable). Discrepancies that can be detected at run-time (e.g., with a dedicated sensor) are associated with an *alarm* node. *Edges* in TFPGs represent the cause-effect relationships in failure propagation, are directional, and may be parameterized with activation conditions based on the current system *mode* and propagation time limits" [Kacianka et al., 2019a]. Figure 4.1, taken from [Kacianka et al., 2019a], depicts such an TFPG modeling a UAV. On the left, Figure 4.1a, shows how TFPGs model a system architecture. In the example, a UAV consists of several sub-systems such as a GPS sensor or a propulsion system. Figure 4.1b shows how one such node, the GPS node, can model two different faults and how they would propagate through the system.

Fault Trees

Fault Trees originate in the safety domain [Vesely et al., 1981] and are a, usually graphical, representation of possible component faults leading to system failures, and their relationship to each other. They contain two elements: events and gates. "Events represent happenings (faults) that can possibly result in an undesired event in the root of the fault

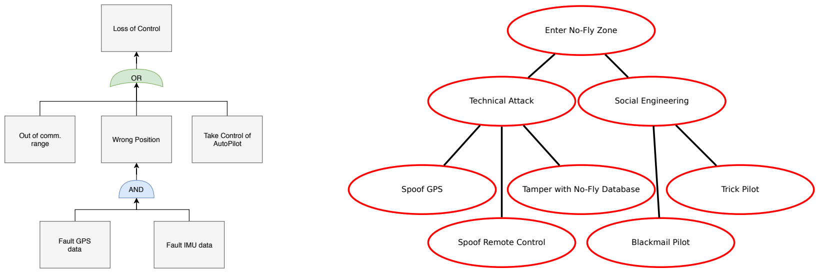
[62] Here, we also have to point out that the presented models are not the best models of the architecture of the system. While especially TFPGs give us some information about a system's architecture, using actual models of a system's architecture should be preferred in the future.



(a) System architecture view.

(b) The detailed view of the UAV's GPS module. **F** - Failure Mode, **A** - Anomaly/Discrepancy, **T** - Test/Alarm.

Figure 4.1: An example of a UAV's timed failure propagation graph, taken from [Kacianka et al., 2019a]. Figure 4.1a is the high level system view that shows how the physical hardware modules are connected. Figure 4.1b shows the detailed failure modes of the GPS sensor. The signal port of the GPS sensor is then connected to the output port in the system architecture view. More details and the modeling tool can be found online `https://modelbasedassurance.org`.



(a) Fault Tree.

(b) Attack Tree.

Figure 4.2: Part of the UAV's fault and attack tree. Taken from [Kacianka et al., 2019a].

tree, e.g., a system failure. In [Vesely et al., 1981], the authors distinguish between *basic*, *conditioning*, *undeveloped*, *external*, and *intermediate* events. Gates allow for connecting two or more events or other gates to express particular relationships between events. For instance, it may be the case that two events must occur together to cause a third event. This would be expressed using an *AND* gate. Additionally, [Vesely et al., 1981] defines *OR*, *EXCLUSIVE OR*, *PRIORITY AND*, and *INHIBIT* gates. A different symbol is used for different types of gates and events." [Kacianka et al., 2019a]. Figure 4.2a, taken from [Kacianka et al., 2019a], depicts a fault tree for a UAV.

Attack Trees

Attack Trees are similar to fault trees, but come from the security domain and describe the necessary steps to conduct an attack on a system [Schneier, 1999, Schneier, 2004]. "Similar to fault trees, a hierarchical tree representation is used in attack trees. The root node contains the ultimate goal of an attack tree (e.g., violate the NFZ) and sub-nodes describe activities that are necessary to accomplish the respective parent activity/goal (e.g., spoof GPS). The relationship between a parent node and its children can be either *OR* or *AND*, i.e., either any child activity/goal will fulfill the parent activity, or all child activities/goals together are required." [Kacianka et al., 2019a]. Figure 4.2b, also taken from [Kacianka et al., 2019a], shows such an attack tree for a UAV.

4.3.2 Models of Human Behavior

In [Kacianka et al., 2019b], we investigated how we can use models of human behavior to bridge the gap between the social system and the CPS. While, as with modeling techniques for CPS, there are many different techniques to model human behavior,[63] we

[63] See for example [Stanton et al., 2013] for an introductory textbook or [Zielinska et al., 2020] for quick reference website.

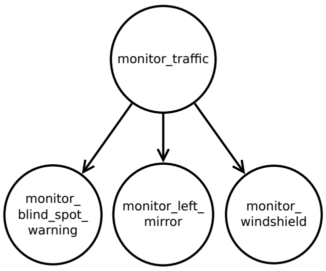


Figure 4.3: HTA example given in [Kacianka et al., 2019b] that models a driver changing a lane.

focused on Hierarchical Task Analysis (HTA) [Annett, 2003], because it is fairly detailed and translates into tree-like structures that lend themselves well to being translated to causal models. Additionally, our project partner at the time had models and a use case available to use.

Once we have a detailed HTA, we can use this to derive cognitive architectures like ACT-R [Anderson et al., 1997] or CASCaS [Lüdtke et al., 2010]. As we detailed in [Kacianka et al., 2019b], the architectures "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. (...) Sub-goals defined by an HTA can be formulated as procedures in this format."

In [Kacianka et al., 2019b], we give the example of a lane change, depicted in Figure 4.3. It shows the goal, *monitor_traffic*, depends on three sub goals *observe_blind_spot_warning*, *observe_left_mirror* and *observe_windshield*. This could then be translated into CASCaS rules and be used in a simulation, as depicted in Figure 4.4.

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

Figure 4.4: CASCaS rules as presented in [Kacianka et al., 2019b].

We use these cognitive architectures, because they codify human behavior and are widely used in the field of human factors. As we wrote in [Kacianka et al., 2019b]: "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 [Liu and Wu, 2006], [Salvucci, 2006] and [Lüdtke et al., 2009]. Another important application of these cognitive models is the modelling of human error. [Lüdtke et al., 2009] investigated human error in lane merging tasks and aviation and [Lüdtke et al., 2010] 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'."

4.4 Translating Models to Causal Models

Fundamentally anything that has a clear definition of the dependence of variables can be called a causal model. The big contribution of SCMs is to move the conversation from directionless correlation that used the be the gold standard in statistics, to directed causes. As such most models used by humans to explain the behavior of systems are already causal in nature. Fault Trees tell us how failures cause undesired events and attack trees enumerate the possible causes of security violations. The core question now is if we can turn such an implicit causal model into an explicit SCM.

In [Kacianka et al., 2019a] and [Kacianka et al., 2019b] we presented a methodology to FTs, ATs, TFPGs and HTAs into causal models. We based our methodology on the fact that these models resemble trees. Treelike structures are easy to translate to causal models, however, "such transformations entail assigning new, possibly different, semantics to those utilized within the source models." In other words, we have not shown that the causal model is semantically equivalent to the actual source model. Here, as usual, we argue that the causal model needs to be reviewed and agreed upon by experts. In the case of these technical models these arguments should usually be very straight forward as we can translate both the structure of the models as well as the semantics of the functions directly into causal models. Especially for FTs and ATs

Table 4.1: Comparison of Formats as given by [Kacianka et al., 2019a].

	Fault Tree	Attack Tree	TFPG	Causal Model
Inner Element or Gate	Describes a failure which occurs as a consequence of other failures connected by an operator.	Describes an attack/sub-goal achieved by executing its child-attacks.	Describes off-nominal conditions resulting from a set of failure modes being present.	An endogenous variable which is defined by other endogenous variables composed to a formula.
Leaf Elements	Describes a basic failure which is not a consequence of other failures.	Describes a basic attack which does not rely on others and can be executed as is.	Describes failure modes of a component at the lowest level of abstraction which is useful at run-time.	An endogenous variable defined by an exogenous variable. Describes whether or not a specific event occurred.
Operators	OR, AND, XOR, PRIORITY_AND, PRIORITY_OR, INHIBIT, ORMORE, ORLESS	conjuctive (i.e., AND), disjunctive (i.e., OR)		AND, OR, NOT, XOR, IFF, NAND, NOR, ATLEAST(min), CARDINALITY(min,max)

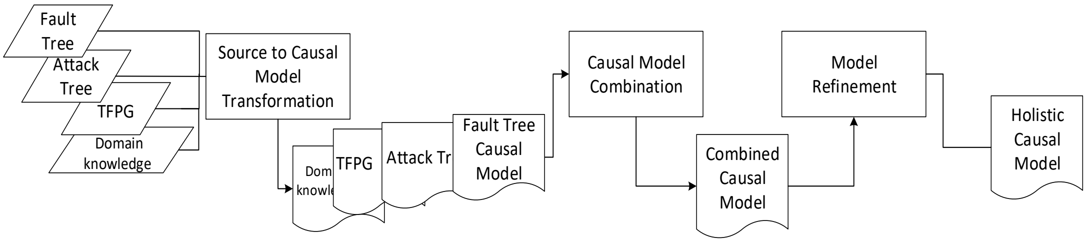


Figure 4.5: Process Diagram of the Methodology, taken from [Kacianka et al., 2019a].

that have a simple boolean logic the translation should not cause many problems. The argument is not so simple for TFPGs, as they contain the element of time, which is normally not represented in causal models. While there are ideas to represent time in causal models,[64] this is not yet a common practice and we would first need to develop a sound extension to the mathematical framework of causality.

Figure 4.5, taken from [Kacianka et al., 2019a], shows the entire methodology. First, we transform the source models into causal models. Second, we join those causal models into a combined causal model, encompassing all sub-models. Finally, this model is then refined into a holistic causal model.

4.4.1 Translating Trees

As in [Kacianka et al., 2019a], we here also use the propositional semantics in attack and fault trees, because they align well with causal models. So, every "non-leaf node in the tree is expressed with a propositional formula of its parents, e.g., $out = in_1 \wedge in_2$."

Following the definition of actual causality in Chapter 3, we consider each node as an *endogenous variable* that can either be true or false, depending on weather a specific

[64] [Halpern, 2016, Chp. 2.3] suggest using time-index variables. In general time is an interesting topic in causal models, but generally only realized in the form of an implicit happens-before relation, as we assume that causes precede effects.

failure or attack occurred or not. We can then represent the structure of the tree in SCMs.

Leaf nodes represent basic, atomic events that are not refined any further. They might be a failure in an FT or an attack in an AT. However, since each endogenous variable is defined by either an other endogenous variable, or an exogenous variable, we create an artificial extra node that sets the value of the leaf nodes. This allows us to also blame the leaf nodes, which is not possible for exogenous variables. This extra node, has the suffix _exo. Additionally, multiple occurrences of a leaf node in a AT or FT, will be matched to a single exogenous variable in the causal model. In [Kacianka et al., 2019a], we used the following definition of a Fault or Attack Tree:

Definition 13

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 an acyclic relation.

Using this definition, in [Kacianka et al., 2019a] we developed the following formalization of the translation from Fault or Attack Trees to causal models:

Definition 14

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

- *$\mathcal{U} = E(T, \_\text{exo})$, where $E(T, \text{suffix})$ returns a renamed copy of the leaf nodes of a tree T with a suffix _"exo".*
- *$\mathcal{V} = N$.*
- *$\mathcal{R} = \{true, false\}$.*
- *$\mathcal{F}$ 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 $X = X\_exo$.*

In this paper we evaded the question of semantics. Of course it is trivial to convert a tree into a graph, the core question is if we can also transfer the semantics of a fault tree into an SCM. We never did a formal proof of this proposition, so our answer is tentative. However, since the causal relations $\mathcal{F}$ can have any mathematical form or semantic, and fault trees fundamentally represent causal relations this should not pose a problem. A problem is that even the semantics of fault trees are often not clearly specified [Schellhorn et al., 2002]; for example, time is not explicitly modeled in fault trees. A decomposition of the form $A = true \vee B = true$ does not explicate the fact that the subevent A and B must happen before the disjunction. In this notation is unclear if A and B must happen simultaneously for the fault to emerge, if there is a specific time window, or some other temporal condition that needs to happen. So to fully translate the semantics in an actual project, we would first need to ensure that the used fault tree follows a well defined semantic.

4.4.2 Translating TFPGs

We follow [Kacianka et al., 2019a] and use the definition of TFPGs as a tuple (F, D, E, M, ET, EM, DE), where F is a non-empty set of failure modes which are always root nodes, D is a non-empty set of discrepancy nodes, $E \subseteq V \times V$ is the set of all edges; $V = F \cup D$ and M is a non-empty set of system modes. Furthermore, $ET : E \to I$ associates every edge with a time interval $[t_1, t_2] \in I$ and $EM : E \to \mathcal{P}(M)$ associates every edge with a system mode.

Since F are leaf nodes, they are equivalent to $\mathcal{U}$ in a causal model; however, since we want to be able to blame failure modes, we use a similar technique to above and create extra nodes with the *_exo* suffix. D cannot be leaf nodes and are thus equivalent to the endogenous variables $\mathcal{V}$. TFPG often include time, which we usually do not model in causal models. It is however possible to model time in SCMs using time indexed variables. The functional relations will then be modeled as part of the structural equations. Definition 15, taken from [Kacianka et al., 2019a], gives the full definition:

Definition 15
TFPG To Causal Model
$TFPG = (F, D, E, M, ET, EM, DE)$ is mapped to a $M = (\mathcal{U}, \mathcal{V}, \mathcal{R}, \mathcal{F})$ i.e., $TFPG \to M$ as follows: $\mathcal{U} = rename(F, \text{\_exo})$, $\mathcal{V} = F \cup D$, $\mathcal{R} = \{true, false\}$, $\mathcal{F}$ associates with each $X \in \mathcal{V}$ a propositional formula based on DC which maps a $D \to \{AND, OR\}$; and with each $X \in F$ a formula of the form $X = X\_exo$.

4.4.3 Translating Human Models

Following [Kacianka et al., 2019b], in contrast to many technical models, human models differ in that they usually model positive behavior, whereas FTs, ATs, and TFPGs only model wrong behavior. So, as a first step, we need to translate a positive model into a negative model. For example, instead of "looking into the mirror", which would prevent an accident, we need to ask if the driver "did not look into the mirror", thereby causing an accident. The reason for is that when we automatically reason over such models we look for events that caused an accident and events that comply with the "perfect" behavior by definition do not qualify.[65] Second, translating the structure of the models of human behavior is often not feasible. Most models of human behavior are recursive and causal models are directed acyclic graphs.[66] Third, the semantics are often impossible to directly translate and leave room for interpretation. For example, what does "looking into the mirror" actually mean? Does it imply that the driver also perceived what was in the mirror? Does it imply they also recognized an object in the mirror? All these things are open for debate. In [Kacianka et al., 2019b], we simply created the models by hand. And, as usual, recommend that experts develop and debate the final model.

4.4.4 Semantic Translations

All the translations up until this point were of a syntactic nature. We exploit the fact that trees are DAGs and thus syntactically are a causal model. In doing so, we sidestepped the much harder question about semantic translations. One problem is that there is no universal semantic for any of these models, so different authors will define the semantics of fault tree in different ways. However, usually there are ways to translate the different versions of a model into another one. The much harder problem is that although authors will state that, for example, "(t)he fundamental concept in fault-tree analysis is the translation of a physical system into a structured logic diagram (fault tree), in which certain specified causes lead to one specified TOP event of interest" [Lee et al., 1985], they are not explicit about their understanding of causality. This problem is now compounded by the fact that there is not one true definition of causality. It is an open question if a single definition can work for type and actual causality or if separate definitions are needed.[67]

This leaves us with two options. First, we can analyze definitions of source models and derive their notion of causality from the definitions. Second, we can simply assume that a source model uses the Halpern-Pearl definition of causality and reason under this premise. The second option makes a semantic translation trivial: The SCM will take the form of the tree, and the structural equations will be set to the formal meaning of the modeling framework (in fault trees, for example a logical and operator). Related to this second option, but much more interesting, [Leitner-Fischer and Leue, 2013] have

[65] In general, SCMs need not model "negative" behavior. In this specific case having a normal form made it easier for us to join the models and then reason over them with SAT solvers, see [Ibrahim and Pretschner, 2020].

[66] We can always "unroll" a recursive human model, but then we have to justify the "depth level".

[67] [Halpern, 2016] writes that "(his) current feeling is that type causation arises from many instances of actual causation, so that actual causation is more fundamental (...)". In a recent paper, [Beckers, 2021a] gives twelve new definitions of actual causality.

shown that it is possible to use counterfactual computation on SCMs to derive fault trees from SCMs. Similarly, [Kölbl and Leue, 2020] present two algorithms that together can translate a set of action traces that form a causality class into a fault tree. Taken together these two results suggest that it is possible to transform causal knowledge (formalized as SCM or otherwise) into fault trees.

For our work, however, we need to go the other direction and understand the notion of causality in a source model. For this, we can use the work of [Beckers, 2021a], who observes that we can use Pearl's definition of causal sufficiency to derive definitions of causality. If we have a causal model $M = ((\mathcal{U}, \mathcal{V}, \mathcal{R}), \mathcal{F})$, we say that a setting $\vec{X} = \vec{x}$ is sufficient for a setting $\vec{Y} = \vec{y}$ if $\vec{Y}$ "follows" from $\vec{X}$. This means that our model has some form of causal direction. Using the Halpern-Pearl terminology, setting $\vec{X} = \vec{x}$ is an intervention and $\vec{Y} = \vec{y}$ is the consequence or result of that intervention. All tree-like structures would fulfill this minimal demand and can be considered a causal model.

However, as [Beckers, 2021a] points out, the question is what happens to all other endogenous variables in that model. In his paper, he provides six different ways the other variables can be treated. First, if $\vec{X} = \vec{x}$, $\vec{Y} = \vec{y}$ is independent of all other variables; this is called *directly sufficient*. Second, if $\vec{X} = \vec{x}$ is directly sufficient for a set $\vec{N} = \vec{n}$ and $\vec{Y}$ is a subset of $\vec{N}$; this is called *strongly sufficient*. Third, if we do not intervene on any other variables, we say that $\vec{X} = \vec{x}$ is *weakly sufficient* for $\vec{Y} = \vec{y}$. These three definitions can then be applied to specific contexts and give us three more definitions: *actually directly sufficient*, *actually strongly sufficient*, and *actually weakly sufficient*.

If we now look at a recent fairly popular work on fault trees, [Xing and Amari, 2008] define four elements of a fault tree:

1. *A TOP event: represents the undesired event, usually the system failure or accident.*

2. *Basic events: represent basic causes for the undesired event, usually the failures of components that constitute the system, human errors, or environmental stresses. No further development of failure causes is required for basic events.*

3. *Undeveloped events: represent fault events that are not examined further because information is unavailable or because its consequence is insignificant;*

4. *Gates: are outcomes of one or a combination of basic events or other gates. The gate events are also referred to as intermediate events.*

For a syntactic translation, we can turn all three types of events into exogenous variables. We do not refine their causes, and just work with their values. Depending on the semantic, some basic events might be seen as endogenous events, especially if we want them to be possible causes in an automated analysis. At each gate we would have an endogenous variable whose value would be determined by a structural equation that takes several events as input. For example, an AND-gate will be the conjunction of several input events.

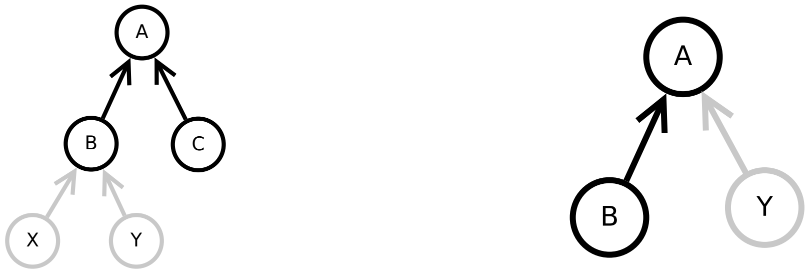
So at this point we have the structure of the fault tree and we need to understand its underlying definition of causality. Fault Tree Analysis (FTA) uses the notion of a minimal cutset that causes an event. From [Xing and Amari, 2008]: "Qualitative analysis usually consists of studying minimal cutsets. A cutset in a fault tree is a set of basic events whose occurrence leads to the occurrence of the TOP event. A minimal cutset is a cutset without redundancy. In other words, if any basic event is removed from a minimal cutset, it ceases to be a cutset." This means that we are trying to find a set where the TOP event is caused, independent of all other endogenous variables. This indicates that the authors assume *directly sufficient* of the cutset $\vec{X} = \vec{x}$ for the TOP event $Y = y$.

Going back to [Beckers, 2021a], this narrows it down to two possible definitions of causality: Contrastive directly sufficiency and Minimal directly sufficiency. Beckers shows that both definitions are equivalent, so we can just look at contrastive directly sufficiency. Again following [Beckers, 2021a], this is equivalent to the to Pearl's first definition of actual causality [Pearl, 1998], which again is similar to his idea of a causal beam [Pearl, 2009, p. 318]. The only problem is that this definition has been found lacking, because causality is not restricted to just parent-child pairs. This means that the notion of causality used by [Xing and Amari, 2008] (as far as we can infer it!) in fault trees is equivalent to an early definition of causality, but is limited in the types of causes it can express. This is fine, because fault trees are not meant to be full fledged causal models, their main use is to provide a quick overview of causal links in a technical system. It also means that we can indeed transform this understanding of causality to SCMs, although we might need to adapt the understanding of causality. [68]

4.5 Combining Causal Models

At some point in the development process we will end up with several causal models of the same systems. Some will describe its technical behavior, such as fault or attack trees, while others will describe how humans will react and interact with the system. We now have to find a way to meaningfully combine these causal models into a single

[68] Similar analysis would now need to be done for other types of models; however unfortunately they are outside the scope of this text.



(a) Refine.

(b) Extend.

Figure 4.6: Merging Causal Models as presented by [Kacianka et al., 2019a].

holistic model. The core problem is that often times it will not be clear what nodes are identical, how we can merge functional relations, and that models might also contradict each other. So at the core, creating these holistic models is, again, a task for a group of experts that look at the source models and then join them manually.

In [Kacianka et al., 2019a], we identified two scenarios where causal models can be combined. In the first one, *refine*, one model adds more details to another model. In this case, the two models share one common node and the second model simply adds more information about that node. In Figure 4.6a node B would be shared by both models and the second model would add information X and Y to B. The second scenario, *extend*, describes the case where two models agree on a common node, and the second models adds an alternative cause. In Figure 4.6b both models agree on node A, but the second model propose Y as as alternative cause. In [Kacianka et al., 2019a], we borrowed the notions of *dominance, compatibility,* and *combination* from [Alrajeh et al., 2018]. As we stated there, this "allow[s] us to combine models where the modelers agree on the causal relationships among the common variables but use different levels of detail to describe how the variables are affected by other variables."

Formally, [Alrajeh et al., 2018] first define a domination relation:

Definition 16
Domination relation *[Alrajeh et al., 2018] Let* $M_1 = ((U_1, V_1, R_1), F_1)$ *and* $M_2 = ((U_2, V_2, R_2), F_2)$*. Let* $Par_M(C)$ *denote the variables that are parents of C in* M*.*[69] M_1 *strongly dominates* M_2 *with respect to C, denoted* $M_1 \succeq_C M_2$*, if the following conditions hold:*
$MI1_{M_1,M_2,C}$ *The parents of C in* M_2 *are the immediate* M_2*-ancestors of C in* M_1*.*[70]
$MI2_{M_1,M_2,C}$ *every path from an exogenous variable to C in* M_1 *goes through a variable in* $Par_{M_2}(C)$*.*

[69] M is any causal model.

[70] Here we assume that we can identify nodes that are the same in M_1 and M_2.

$MI3_{M_1,M_2,C}$ Let $X = ((U_1 \cup V_1) \cap (U_2 \cup V_2)) - \{C\}$ then for all settings $\vec{x}$ of the variables in $\vec{X}$, all values c of C, all contexts $\vec{u_1}$ for M_1, and all contexts $\vec{u_2}$ for M_2 $(M_1, \overrightarrow{u_1}) \models [\overrightarrow{X} \leftarrow \overrightarrow{x}](C = c)$ iff $(M_2, \overrightarrow{u_2}) \models [\overrightarrow{X} \leftarrow \overrightarrow{x}](C = c)$.

Building upon it, they then define how to combine two causal models:

Definition 17
Combination *[Alrajeh et al., 2018] If $M_1 = ((U_1, V_1, R_1), F_1)$ and $M_2 = ((U_2, V_2, R_2), F_2)$, then M_1 and M_2 are compatible if (1) for all variables $C \in ((U_1 \cup V_1) \cap (U_2 \cup V_2))$, we have $R_1(C) = R_2(C)$ and (2) for all variables $C \in (V_1 \cap V_2)$, either $M_1 \preceq_C M_2$ or $M_2 \preceq_C M_1$. If M_1 and M_2 are compatible, then $M_1 \oplus M_2$ is the causal model $((U, V, R), F)$, where $U = U_1 - U_2 - (V_1 \cup V_2)$; $V = V_1 \cup V_2$; if $C \in U_1 \cup V_1$, then $R(C) = R_1(C)$, and iff $C \in U_2 \cup V_2$, then $R(C) = R_2(C)$ if $C \in V_1 - V_2$ or if both $C \in V_1 \cap V_2$ and $M_1 \preceq_C M_2$ then $F(C) = F_1(C)$; if $C \in V_2 - V_1$ or if both $C \in V_1 \cap V_2$ and $M_2 \preceq_C M_1$, then $F(C) = F_2(C)$.*

In [Kacianka et al., 2019a], we gave more detailed explanations of these two definitions that we will quote here for the reader's convenience:[71]

> Informally, the work by [Alrajeh et al., 2018] provides conditions for the *compatibility* of causal models as a prerequisite for *combining* them. To that end, they introduce the notion of a *dominant* relation, essentially comparing the information expressiveness of two models about a variable C. One model, M_1, *dominates* another, M_2 in expressing variable C, denoted as $M_1 \succeq_C M_2$, if they agree on the causal dependence of C, but M_1 provides a more detailed picture. For each common variable, if the two models have a dominance relation (regardless of which model is dominating the other), the models are *compatible.* Only compatible models can then be combined, denoted as $M_1 \oplus M_2$.
>
> The idea of refinement as shown in Fig. 4.6a can be considered as a special case of the combination step. We limit our work to *refining* causal models on the leaf nodes. That is, we only consider a detailed model M_1 to be appended to a more abstract model M_2 if the root, or the root of a sub-tree, of M_1 is identical to one of the leaf-nodes of M_2 (of the causal graph with exogenous variables omitted). In this specific case, according to the *strong domination* definition in [Alrajeh et al., 2018] (Definition 16), the two causal models are

[71] The full-quote is edited slightly for brevity and type-setting.

always compatible. Hence, they can be combined resulting in a new model M as defined in Definition 17.

Combining Extensions (Fig 4.6b) is, unfortunately, more complex than merging refinements. In general, extensions will be *incompatible* and cannot be merged automatically. If two models disagree on the variables and their causal relationships, we have to defer the merging decision to an expert. (...)

For some specific models, Friedenberg and Halpern [Friedenberg and Halpern, 2018] provide an automatic way of joining the models. Such FH-compatible models are causal models that are extended with a focus function $\mathcal{G} : \mathcal{U} \cup \mathcal{V} \to 2^{(\mathcal{U} \cup \mathcal{V})}$ that, given a variable C, provides the set of variables that the modeler considered as having an effect on C. Using this additional information, if one model "can explain and has everything considered by" some other model, these two models can be merged. The intuition is that if both models consider the same variable C, they consider the same parents of a variable C, and, thus one model can explain all of the other model's observations, they are compatible in a similar sense as [Alrajeh et al., 2018] consider a model to be compatible. If one of the two models provides additional information, this information can be merged into a joint model. (...)

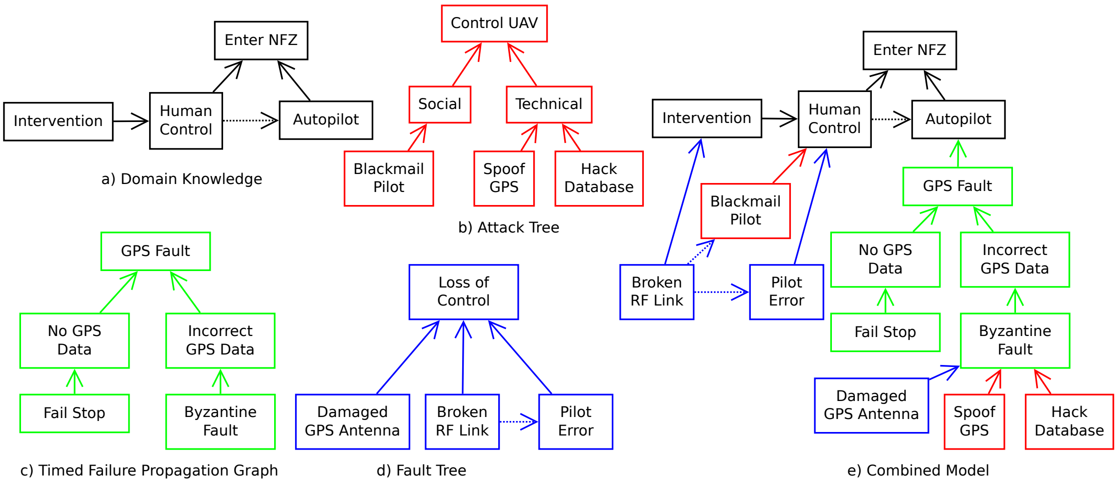


Figure 4.7: All causal models color coded by their source model. Dashed edges represent preemption relations, and solid edges are normal causal relations. Taken from [Kacianka et al., 2019a].

To illustrate this process, in [Kacianka et al., 2019a] we gave the example of different source models for an Unmanned Aerial Vehicle (UAV). Figure 4.7 illustrates the source models (a - d) and the combined model (e) graphically. In this example, the first model encodes explicit domain knowledge, here that if a UAV enters a No-Fly Zone (NFZ) it is either controlled by a human pilot or by the system's autopilot. Model b shows an attack tree that considers different ways an UAV might be attacked such as blackmailing the pilot or spoofing the UAV's GPS signal. The third model consists of a TFPG that provides details of failures related to the UAV's GPS data. Model d finally depicts the causal model resulting from a fault tree showing reasons for a potential loss of control.

The combined model shows one way that these sub-models can be combined into a holistic model. Having such a holistic model then allows us to reason across the boundaries of singular models and try to identify reasons for some unwanted behavior.[72]

This chapter offers three key insights. First, in the everyday development of technical systems many prospective models are created and are available to inform causal models. While it is true that translating these models to SCMs is not well researched, we are positive that this can be done efficiently and be automated to a large degree. Second, while models of human behavior are a lot harder to come by and inherently riddled with uncertainty they do exist and can also, in principle, be turned into SCMs. Third and finally, these different types of model can, again in principle, be merged and used to reason over causality (and by extension accountability) across system boundaries and, even more importantly, across the boundaries of social and technical elements. What this chapter does not offer is concrete proof that this works. At best it offers some well founded intuitions and shows the need for future research.

[72] [Kacianka et al., 2019a] provide some more details about this examples and also shows how the reasoning could be done. We go into more details about this in Chapter 6.

5 Accountability and Structural Causal Models

5.1 Introduction

To define what responsibility and accountability mean, we build on the foundations of structural causal models in Chapter 3. Additionally, here we assume that we have adequate SCMs given, derived in a processes described in the previous chapter. Chapter 6 will then tie all the chapter together in a single example.

5.1.1 Structural Causal Models

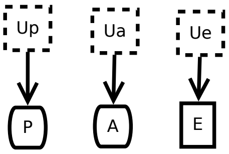


Figure 5.1: A causal model, without any causal links. Up, Ua and Ue stand for the exogenous variables.

Figure 5.1 depicts a structural causal model (SCM) with three endogenous variables, P, A, E, and three exogenous variables, $\mathcal{U}_P$, $\mathcal{U}_A$, $\mathcal{U}_E$. We assume that the range of all six variables is $\mathbb{N}$. In this model the values of every endogenous variable would be determined by its exogenous parent. This means that the structural equations, $\mathcal{F}$, look like this:

$$\begin{aligned} P &= \mathcal{U}_P \\ A &= \mathcal{U}_A \\ E &= \mathcal{U}_E \end{aligned} \tag{5.1}$$

This means that in full it reads $\mathcal{M} = (\{\mathcal{U}_P, \mathcal{U}_A, \mathcal{U}_E\}, \{P, A, E\}, \mathcal{F})$. In this model there is no causal influence, so every endogenous variable takes on the value set by exogenous factors outside the model, so the context.

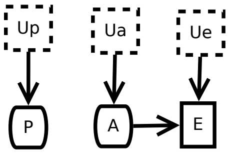


Figure 5.2: We add the causal relation $A \rightarrow E$.

Figure 5.2 depicts a causal model where A has causal influence on E. This means that the value of E depends on the value of A (but P still has no influence). So we need to change the structural equations $\mathcal{F}$ so that A appears on the right hand side of E. For simplicity we assume addition, but in general structural equations can be any mathematical expression; they do not need to be linear or adhere to any other requirements.

$$\begin{aligned} P &= \mathcal{U}_P \\ A &= \mathcal{U}_A \\ E &= A + \mathcal{U}_E \end{aligned} \tag{5.2}$$

If we now look at the context where $\mathcal{U}_P = 1, \mathcal{U}_A = 1$ and $\mathcal{U}_E = 1$, we get the following structural equations:

$$\begin{aligned} P &= 1 \\ A &= 1 \\ E &= A + 1 = 1 + 1 = 2 \end{aligned} \tag{5.3}$$

If we now want to compute the counterfactual $E_{A=2}$, so asking the question "what would the value of E be, had A been 2", we would change the model and let $A = 2$:

$$\begin{aligned} P &= 1 \\ A &= 2 \\ E &= A + 1 = 2 + 1 = 3 \end{aligned} \tag{5.4}$$

5.1.2 Probability in Causal Models

If we try to determine a cause, we might be uncertain about the exact model or the context. This can be handled by giving a probability on different models and contexts and then reason over them. For example, we might have a model $\mathcal{M}_1$ and a model $\mathcal{M}_2$

that both explain why an apple falls towards the earth. If $\mathcal{M}_2$ has some very specific requirements (e.g., it is the result of magic energy), while $\mathcal{M}_1$ has a simple straight forward explanation (e.g., that objects want to move towards the center of the universe), we might say that $\mathcal{M}_1$ is correct with probability $.9$ and $\mathcal{M}_2$ with probability $.1$. Similarly, we might not be able to always observe if the apple falls to the ground or not. So the context in which the apple falls on the ground might be assigned probability 0.7 and the context in which it floats might get probability 0.3. The third source of uncertainty can be in the equations themselves. For example, $E = A + U_E$ might only hold with probability $.9$ and not always. Currently, there is no definite way of incorporating probability into causal models. Here we will use a technique by Halpern that converts probabilistic equations into a probability over causal settings, where each setting is deterministic.[73]

If we look at $\mathcal{M}$, we see that it is deterministic. If we can specify the context $\mathcal{U}$, we can compute the values for all endogenous variables.[74] This means that we also assume the world itself to be deterministic and that, given all the relevant details, we can predict what will happen.[75] So if we say that $E = A + U_E$ with probability 0.9, there must be some reason why this might not be the case. [Halpern, 2016, Chp. 2.5] now "packages up" all these uncertainties and makes them exogenous. So we have an additional variable in U that is 1, if $E = A + U_E$ holds and 0 otherwise. $U = 1$ now has probability 0.9 and $U = 0$ has probability 0.1. If we do this with all probabilities, we have removed all uncertainty and non-determinism from the structural equations. This then gives us two separate models, $\mathcal{M}_1$ and $\mathcal{M}_2$ that are similar, except in the structural equation for E. In $\mathcal{M}_1$ $E = A + U_E$ and in $\mathcal{M}_2$ it is something else. Unfortunately, specifying this "something else" is only easy in binary models. In non binary models, would have a sub model for every possible value that E can take.[76] There is to date no proof that this is the best approach or that it is equivalent to approaches that understand causality as an effect on the probability distribution of another variable. As a consolation, the questions of responsibility and accountability that this text is concerned with are usually asked in a retrospective manner and about an individual of a population. In these scenarios, we can use evidence from the context to reduce the possible models to a manageable number.[77]

[73] Pearl uses a similar method, see [Pearl et al., 2016, p.97f].

[74] As [Pearl et al., 2016, p. 92] note, $\mathcal{U} = u$ identifies exactly one individual in a population or situation.

[75] As Halpern points out, this is probably wrong on the quantum level, but it holds in the macroscopic world for which we apply causal models.

[76] See [Halpern, 2016, Chp. 5.1] on a discussion on how to handle such cases.

[77] Or as [Halpern, 2016, Chp. 2.5] summarizes: "Perhaps the key message here is that there is no need to work hard on getting a definition of probabilistic causality, at least at the macroscopic level; it suffices to get a good definition of deterministic causality." See also [Pearl et al., 2016, p. 92ff].

5.2 Responsibility

When we ask questions of responsibility, we ask about an attribution of the use of power to a social actor. An SCM, such as $\mathcal{M}$ shown in Figure 5.2, depicts things that we can observe and the structural equations encode our understanding of how the world works. So if A is the observation of a social actor's action it means that an actor did something so that we observe $A = a$. Since we cannot observe the way that social actors make decisions, we cannot add this to the model and have to model this as exogenous variables. This has implications on the kind of questions an SCM can answer. While we can answer the questions "What would have happened in our model, had the social actor acted so that $A = b$ instead of $A = a$", we cannot answer questions about the motives of A; for example, $\mathcal{M}$ cannot answer the question "What would A have done had $\mathcal{U}_A = x$ instead of $\mathcal{U}_A = y$". However, what we can do is compare the observations of the actor's behavior to an ideal or law-abiding agent. So we can create a model of a "normal actor" that will produce a set of normal actions ($\mathcal{N}_A$) and answer the question "What would A have been, had it been the result of the actions of a normal actor given that $\mathcal{U}_A = x$" and then compare this to the action A. This is important, because people are usually only responsible for a subset of all the actions they can take. Children, for example, can cause many things for which we do not hold them responsible; precisely because we do not expect them to understand all the ramifications of their actions. Similarly, if a person causes an explosion by flipping a switch, he might not be held responsible if he had no reason to expect that the flipping of the switch would cause an explosion [Talbert, 2019].

So a theoretical agent gives us a baseline for responsibility. If we encounter an unwanted event, and a social actor acted "as a normal agent would", it might be a cause for the unwanted event, but it would not be *blamed* for it. Symmetrically, if we encounter a desired event that only happened because a social actor acted "as a normal agent would", the actor would not necessarily get any special *praise*. However, if an unwanted event could have been prevented had the social actor acted "as a normal agent would", we would *blame* the actor for not preventing it. Similarly, if a desired event only happens because a social actor did not act "as a normal agent would", the actor would be *praised* for doing something exceptional. Table 5.1 summarizes these four meanings of responsibility.

Our distinction does not cover cases where an agent does something outside its commitment which has no influence on E. For example, an agent might do something outside its commitment, and an unwanted event happens. So we would have $E =$ unwanted and $A \notin \mathcal{N}_A$. Following our definition the agent would be to blame, because

Table 5.1: The meaning of praise and blame. Here we presuppose that A is a cause for E, which means that changing A has a causal effect on the value of E. The first column indicates if the event was desired or unwanted, the second column indicates whether the actor's agent where "normal" or not and the third column indicates how the outcome would have changed, had the agent acted "normally" (or not). Every second row summarizes the meaning of the previous row.

<table>
<tr><td>$E =$ unwanted</td><td>$A \in \mathcal{N}_A$</td><td>$E =$ desired, for some $A \notin \mathcal{N}_A$</td></tr>
<tr><td colspan="3">"$Actor_A$ did all it was supposed to do."</td></tr>
<tr><td>$E =$ desired</td><td>$A \in \mathcal{N}_A$</td><td>$E =$ unwanted, if $A \notin \mathcal{N}_A$</td></tr>
<tr><td colspan="3">"$Actor_A$ did its job."</td></tr>
<tr><td>$E =$ unwanted</td><td>$A \notin \mathcal{N}_A$</td><td>$E =$ desired, if $A \in \mathcal{N}_A$</td></tr>
<tr><td colspan="3">"$Actor_A$ failed to do its job and is to be blamed for E."</td></tr>
<tr><td>$E =$ desired</td><td>$A \notin \mathcal{N}_A$</td><td>$E =$ unwanted, if $A \in \mathcal{N}_A$</td></tr>
<tr><td colspan="3">"$Actor_A$ went above and beyond to make the impossible happen; it is to be praised for E."</td></tr>
</table>

$A \notin \mathcal{N}_A$; however additionally $E =$ unwanted would also happen if $A \in \mathcal{N}_A$. This means that no matter what $Actor_A$ does, $E =$ unwanted. Similarly, $E =$ desired and $A \notin \mathcal{N}_A$; however $E =$ desired even if $A \in \mathcal{N}_A$. So no matter what $Actor_A$ does, $E =$ desired. So the actions of $Actor_A$ have no influence on the outcome. This means that E is causally independent from A and thus violates our precondition for responsibility that A needs to be a cause of E.

In most cases, we do not need to rely on a hypothetical "normal" actor to use as a reference. Most often agents will explicitly or implicitly commit themselves to certain actions. Such a commitment is similar to $\mathcal{N}_A$ but derived ex ante and part of the design of a system. So actors know what they commit themselves to and can prepare to meet those expectations. "Normal actors" are only used in ex-post analysis, when no commitment is defined.

Definition 18 (Commitment)
A commitment is the explicit or implicit dedication of an actor $Actor_A$ to only take on some of its values, called $\mathcal{C}_A$, connoting it is a commitment. $\mathcal{C}_A$ is a subset of the range of A, so $\mathcal{C}_A \subset range(A)$. This is often done to achieve a certain goal, i.e. ensure that some other variable E does (not) take on a certain value. Without a commitment, we can use a "normal and law-abiding" agent as a reference to generate $\mathcal{N}_A$ which can serve as a baseline for $\mathcal{C}_A$. While it will often be the case that $\mathcal{C}_A \subseteq \mathcal{N}_A$, the commitment can also be completely different from what a normal agent would do, i. e., $\mathcal{C}_A \cap \mathcal{N}_A = \varnothing$.

Examples of commitments are the implicit adherence to laws, mores and expectations or the explicit promise to do something or a contractual obligation to prevent something from coming to pass.

With this we can now give a definition for responsibility in SCMs:

Definition 19 (Responsibility)
If a variable E has an (often: unwanted) value e, social actor $Actor_A$ is responsible for $E = e$, iff

1. *A is a cause for $E = e$,*
2. *A was committed to take on a value in its range, $A \in \mathcal{C}_A$, so that $E \neq e$ (or $E = e$), and,*
3. *$Actor_A$ can freely choose the value of A.*

If $E = e$ is unwanted and $A \notin C_A$, while $A \in C_A$ would change E to a desired value, we say that $Actor_A$ is to be blamed *for $E = e$. If $E = e$ is desired and $A \notin C_A$, while $A \in C_A$ would change E to an unwanted value, we say that $Actor_A$ is to be* praised *for $E = e$.*

The first condition ensures that A is a cause for E, because responsibility for things that one did not cause makes little sense. The second condition is the core difference between responsibility and causality, namely that $Actor_A$ is only responsible for some values and not all. Finally, the third condition requires that $Actor_A$ sets A with its "free will" and without external force. This condition is hotly debated, but for us responsibility without choice is meaningless.

5.2.1 Identify Responsibility in an SCM

Given an SCM $\mathcal{M} = (\mathcal{U}, \mathcal{V}, \mathcal{F})$ any cause can potentially also be a responsibility relation. Since exogenous variables, $\mathcal{U}$, cannot be causes, they can also not be responsible for any events. The structural equations, $\mathcal{F}$, are needed to determine causes, but aside from this do not influence responsibility assessments.[78]

To understand responsibility, we need to analyze every endogenous node, $v \in \mathcal{V}$ and determine (1) if they are the result of a social actor's decision and (2) understand their commitment, $\mathcal{C}_v$. Both of these properties are not directly part of the SCM, but are part of the context of the SCM. We consider them as meta-properties of the SCM. To make this distinction clear, we use boxes with rounded corners for variables that indicate actions of social actors and rectangular boxes for all other variables; see Figure 5.3. An SCM

[78] However, if we try to determine the degree of responsibility $\mathcal{F}$ is essential because the structural equations define how much influence variables have upon each other. See [Halpern, 2016, Chp. 6] for some work on this.

$\mathcal{M}$ can now be used in two different methodological ways. We can either use it in a prospective manner and use it to predict the future, or we can use it a retrospective manner to explain the past.

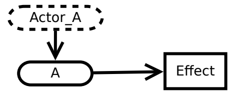


Figure 5.3: Actions of social actors, here node A, are indicated with rounded boxes and all other variables with rectangular boxes. These action have some commitment, here $\mathcal{C}_A$, associated with them. The decision process of a social actor is always exogenous. Edges indicate causal influence.

Prospective Models

In prospective models, we can compute potential causes and, more importantly, show causal independence. So we can compute for which events an endogenous variable $v \in \mathcal{V}$ that is the result of a social actor's decision can potentially be responsible and for which it will never be responsible. Additionally we can analyze the commitment, $\mathcal{C}_v$, to design the system that is modeled by $\mathcal{M}$ so that commitments are clearly understood and everyone knows their space of options. Furthermore, we can use the prospective model to compute responsibilities for specific contexts and use this to determine if this agrees with our intended responsibilities and especially if we can ensure that no unwanted events happen if every social actor acts according to its commitment.

Retrospective Models

Here we have a concrete setting of the context, so $\mathcal{U} = u$ and an event $E = e$ for which we want to understand responsibility. This means that we can determine causes and, more importantly, conduct counterfactual analysis to understand alternative courses of events. A node $v \in \mathcal{V}$ that is the result of social actor $Actor_v$'s decision will usually have a commitment $\mathcal{C}_v$ associated with it. If not, we can construct a hypothetical normal and law-abiding agent to generate the set of normal actions $\mathcal{N}_v$. We can then analyze $\mathcal{M}$ and determine which actors' actions are causes for $E = e$, which actors acted normally, and which failed their commitments. With this information we can determine if a social actor is to be blamed or praised for $E = e$.

5.2.2 Examples

Forest Fire

The forest fire example, is a classic thought experiment which we take from [Halpern, 2016, Chp. 2.1]. In this example, a forest fire (FF) can either be caused by a lightning strike (L) or by an arsonist who drops a match (MD).[79] This model has two exogenous variables $\mathcal{U}_1$ and $\mathcal{U}_2$ that determine all other factors outside the model, such as the conditions for a lightning strike, the dryness of the wood or the arsonist's choice. Beside the assignment of exogenous variable, this model only has one structural equation, $FF = L \vee MD$ indicating that the forest fire will start if either a lightning strikes or a match is dropped. Formally, $\mathcal{M}_{FF} = (\{\mathcal{U}_1, \mathcal{U}_2\}, \{L, MD, FF\}, \{L = \mathcal{U}_1, MD = \mathcal{U}_2, FF = L \vee MD\})$ and the ranges of all variables is binary. Figure 5.4 depicts the causal graph for $\mathcal{M}_{FF}$.

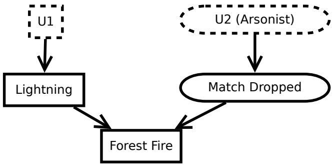


Figure 5.4: The graphical model for the forest fire example.

If we look at the model in a prospective manner, both, L and MD can be causes for the forest fire. However, if we look at it from a responsibility perspective, L cannot be responsible for the fire because lightning is not a social actor. The arsonist, however, is a social actor and should be committed to not drop matches in a dry forest. This means that while the range of MD is $\{0, 1\}$, the commitment is $\mathcal{C}_{MD} = \{0\}$, not dropping a match.

If we now use $\mathcal{M}_{FF}$ to analyze a forest fire, we have four possible worlds, depending on $\mathcal{U}_1$ and $\mathcal{U}_2$: $\{\mathcal{U}_1 = 0, \mathcal{U}_2 = 0\}, \{\mathcal{U}_1 = 1, \mathcal{U}_2 = 0\}, \{\mathcal{U}_1 = 0, \mathcal{U}_2 = 1\}, \{\mathcal{U}_1 = 1, \mathcal{U}_2 = 1\}$

Looking at the first world, $\{\mathcal{U}_1 = 0, \mathcal{U}_2 = 0\}$, we do not see a forest fire, which is our desired event. Additionally, $MD = 0$, which means that $MD \in \mathcal{C}_{MD}$, so the arsonist fulfilled his commitment. This means there is no special blame or praise for the arsonist.

In the second world, $\{\mathcal{U}_1 = 1, \mathcal{U}_2 = 0\}$, we do see a fire, however there was no match dropped, so the arsonist is not a cause for the fire and thus also not responsible for it. Here no one is responsible, a lightning strike simply is an "act of god".

In the third world, $\{\mathcal{U}_1 = 0, \mathcal{U}_2 = 1\}$, we also see a fire and the dropped match was the cause for the fire. Here, we have an unwanted event ($FF = 1$) which is caused by an

[79] Halpern also discusses a conjunctive from, where both L and FF are needed for FF.

action committed by a social actor ($MD = 1$). Additionally this action was against the commitment $\mathcal{C}_{MD} = \{0\}$, so $MD \notin \mathcal{C}_{MD}$. This means that the arsonist is responsible for the forest fire and should be blamed for it.

To expand on this scenario, suppose that $MD = 1$ is not caused by an arsonist, but by a firefighter who tries a controlled burn to prevent the fire; so $\mathcal{C}_{MD} = \{1\}$. However, something goes wrong, inadvertently causing the forest fire that the burn was meant to prevent. In this case, $MD = 1$ would still be a cause for the fire, but we would not blame the firefighter for it – after all she did her best to prevent the fire ($MD \in \mathcal{C}_{MD}$).

Finally, in the fourth world, $\{\mathcal{U}_1 = 1, \mathcal{U}_2 = 1\}$, the responsibility of the arsonist depends on our understanding of causation. This case is –by design– notoriously tricky to handle and [Halpern, 2016, Chp. 2.6] goes into some depth of how the different HP-definitions handle this case. In short, in the disjunctive model we use here both L and MD are sufficient for the forest fire, so they are independent causes. Other definitions disagree, because the fire would have happened regardless of the arsonist's actions. So under an HP definition, the arsonist would be responsible for the fire and receive the blame for it.

Rock Throwing

Another classic example is the Suzy and Billy rock-throwing example [Halpern, 2016, Chp. 2.3]. Here two kids, Suzy and Billy, throw rocks at a bottle. Suzy's rock hits just before Billy's and we now want to express that Suzy's throw is a cause for the bottle shattering and Billy's is not. In Halpern's formulation with preemption $\mathcal{M}_{SB} = (\{\mathcal{U}_1, \mathcal{U}_2\}, \{ST, BT, SH, BH, BS\}, \{ST = \mathcal{U}_1, BT = \mathcal{U}_2, SH = ST, BH = BT, BH = BT \land \neg SH, BS = SH \lor BH\})$ and the range for all variables is again binary. The interesting part of the structural equation is $BH = BT \land \neg SH$ which says that Billy can only hit, if Suzy misses; this encodes the preemption. Figure 5.5 shows the graphical model for $\mathcal{M}_{SB}$.

Both BT and ST are actions of social actors and we assume that they are committed to not throw stones at bottles, so $\mathcal{C}_{BT} = \mathcal{C}_{ST} = \{0\}$.

Similar to the forest fire example above, we have four possible worlds.

First, in world $\{\mathcal{U}_1 = 0, \mathcal{U}_2 = 0\}$, the bottle stays whole and both kids acted according to their commitment, so there is no praise or blame.

Second, in world $\{\mathcal{U}_1 = 1, \mathcal{U}_2 = 0\}$, only Suzy throws and the bottle shatters. Here Billy is not a cause for the bottle shattering and additionally, Billy acted according to his commitment $\mathcal{C}_{BT}$, so he is not responsible for the fate of the bottle. Suzy, however, broke her commitment and $ST = 1$, which causes the bottle to shatter. So Suzy is to be blamed

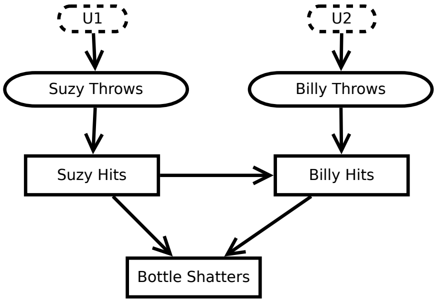


Figure 5.5: The graphical model for the Suzy-Billy rock-throwing example.

for the bottle shattering.

Third, $\{\mathcal{U}_1 = 0, \mathcal{U}_2 = 1\}$, is similar to the second, except that Billy instead of Suzy is to blame for the bottle shattering.

Fourth and finally, in world $\{\mathcal{U}_1 = 1, \mathcal{U}_2 = 1\}$, the bottle shatters and both, Suzy and Billy, violated their commitments. Since only Suzy is a cause, she is responsible for the bottle shattering and will be blamed. Billy is not a cause and thus also not responsible.

Pollutant

Another example given by Halpern is the pollutant example: We have two companies, A and B, that both dump pollutant into a river [Halpern, 2016, Chp. 2.3]. If the pollutant in the river crosses the threshold k, all the fish in the river die. In this model, the actions of the companies are the result of decisions taken by social actors, probably the CEO. $\mathcal{M}_{Pollutant} = (\{\mathcal{U}_{CEO\_A}, \mathcal{U}_{CEO\_B}\}, \{A, B, F\}, \{A = \mathcal{U}_{CEO\_A}, B = \mathcal{U}_{CEO\_B}, F = A + B > k\})$. The last structural equation, $F = A + B > k$ indicates that the fish die if $A + B$ are greater than the threshold k. The range of A and B is $\mathbb{N}$ and F is a binary value. Figure 5.6 depicts the graphical model.

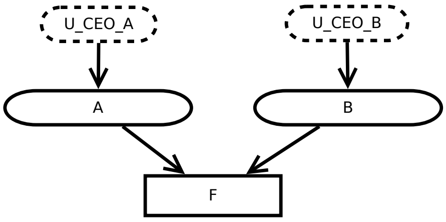


Figure 5.6: The graphical model for the pollutant example.

The example now focuses on different values for k and how they affect causality assessments. In Halpern's example $A = 100$ and $B = 60$. If $k = 120$, both companies are a cause for the polluted river. If $k = 50$ both companies are a cause under the original HP-definition of causality and no single company is a cause in the modified HP-definition. If $k = 80$ A is always a cause. If B is a cause depends on whether A can only dump 0 or 100 kilograms of pollutant.

If we want to analyze the model for responsibility, the commitment becomes particularly important. If polluting is not allowed, so $\mathcal{C}_A = \mathcal{C}_B = \{0\}$, a company that is a cause is also responsible and to blame for the pollution. However, what if $k = 120$, $\mathcal{C}_A = \{0, \ldots, 100\}$ and $\mathcal{C}_B = \{0\}$, while $A = 100$ and $B = 60$? So company A is allowed to dump 100kg of pollutant into the river, while company B is not allowed to dump any pollutant at all. In this case, a causal analysis still give us both A and B as causes. However, $A \in \mathcal{C}_A$ which means that A is fulfilling its commitment. This means that while A is a cause and responsible, A is doing what it should be doing and thus should not be blamed for the death of all the fish. B, on the other hand is not meeting its commitment, i. e., $B \notin \mathcal{C}_B$, and should be blamed for the death of all the fish.

If we set $k = 80$, and $B = 0$, both companies would fulfill their commitment, yet the fish still die. No company would be to blame, but here we have a case where following the commitments does not give a desired outcome. This should ideally be avoided at the system design stage and can be uncovered when analysing the prospective model.

If we keep $k = 80$, $\mathcal{C}_A = \{0, \ldots, 100\}$ and $\mathcal{C}_B = \{0\}$, but set $B = -40$,[80] so company B is removing pollutant from the river, we get another interesting situation. First, the total pollutant is $A + B = 100 + (-40) = 60$, so below the threshold $k = 80$, and the fish live – which is the desired event. In this case company A followed its commitment, so it is not to blame for the result. Company B, however, violated its commitment $\mathcal{C}_B$ by polluting less than is allowed. However, if B followed its commitment, we would get an unwanted event. In this case B is responsible for saving the fish and should be praised for its extra effort.

5.3 Transfer of Power and Accountability

Accountability now differs from responsibility in that it is a means to restrict the use of power. Additionally, a common aspect of accountability is a transfer of power from a principal to an agent. To prevent the agent from abusing this power the agent is required to give an account to the principal. Furthermore, there is not a single definition of accountability; depending on the context and the requirements, different definitions might be applied.

[80] Note: here we are cheating. The range of B is $\mathbb{N}$ which prohibits negative numbers; the range should be changed to $\mathbb{Z}$. We will return to this problem in Section 5.5.3.

For a system $\mathcal{S}$ this means it can either be accountable according to a specific definition of accountability $\mathcal{D}$ or that it can be transformed to a new system $\mathcal{S}_\mathcal{D}$ that incorporates all the changes that accountability definition $\mathcal{D}$ requires. Symbolically we write $\mathcal{S} \xrightarrow{\mathcal{D}} \mathcal{S}_\mathcal{D}$. Different accountability definitions require different causal structures, so when $\mathcal{M}$, the SCM representing $\mathcal{S}$, does not exhibit certain causal structures we know that $\mathcal{S}$ is not accountable according to a given definition.[81] Similarly, if we change $\mathcal{S}$ so that $\mathcal{M}$ complies with the definition, we can show that $\mathcal{S}$ is indeed accountable according to a given definition of accountability.

5.3.1 Change of Power

In SCMs the power of a variable A (or node A in the graphical model) is represented by its range. It describes all possible values that A can take. If A is now indicating the actions that a social actor takes, its range describes all possible actions that this actor can take. In the Suzy and Billy example above, the range of "Suzy throws" is binary, which means that Suzy can either throw or not throw. She cannot do a third thing, for example throwing her stone to intercept Billy's stone. So a change of power can alter the range of a variable. This change of range implies that we potentially need to adapt the structural equations $\mathcal{F}$.[82]

Furthermore, a change of power can mean that what an agent can do, becomes allowed, or part of the commitment. If we go back to the pollutant example, company A originally is not allowed to pollute at all, $\mathcal{C}_A = \{0\}$. However, A can get a permission to dump up to 100kg of pollutant into the river, changing $\mathcal{C}_A$ to $\mathcal{C}_A = \{0, \ldots, 100\}$. Here the change of power affects the commitment of a variable.

Definition 20 (Change of Power)
The change of power is a transformation of an SCM $\mathcal{M}$ to a new SCM $\mathcal{M}'$. In this transformation the exogenous and endogenous variables do not change, so $\mathcal{U} = \mathcal{U}'$ and $\mathcal{V} = \mathcal{V}'$, however the range of nodes that represent actions of social actors can change. If the change of power affects the semantics of $\mathcal{F}$, we need to adapt the affected structural equations. Additionally, a change of power can change the commitment, $\mathcal{C}$, of social actors.

To give an example, if we want to change the SCM for Suzy and Billy so that Suzy's stone can intercept Billy's stone, we need to extend the range of what Suzy can do and

[81] Here we assume that $\mathcal{M}$ is a correct model of $\mathcal{S}$. Creating correct causal models is its very own problem, see Chapter 4.

[82] However, not all changes of power require changes to $\mathcal{F}$. Often the structural equations are robust enough to handle small changes in the range of an input variable.

then adapt $\mathcal{F}$. The original model is:

$$\begin{aligned}\mathcal{M}_{SB} = (\{\mathcal{U}_1, \mathcal{U}_2\},\\ \{ST, BT, SH, BH, BS\},\\ \{ST = \mathcal{U}_1, BT = \mathcal{U}_2, SH = ST, BH = BT,\\ BH = BT \land \neg\, SH, BS = SH \lor BH\})\end{aligned} \tag{5.5}$$

And the range of all variables is binary. The new model will look like this:

$$\begin{aligned}\mathcal{M}'_{SB} = (\{\mathcal{U}_1, \mathcal{U}_2\},\\ \{ST, BT, SH, BH, BS\},\\ \{ST = \mathcal{U}_1, BT = \mathcal{U}_2, SH = ST, BH = BT,\\ BH = BT \land (\neg\,(SH = 1) \land \neg\,(SH = 2)),\\ BS = (SH = 1) \lor BH)\})\end{aligned} \tag{5.6}$$

Here we change the range of SH and ST from binary to three values $\{0, 1, 2\}$, where 0 means that Suzy misses, 1 means that Suzy hits and 2 means that Suzy's rock intercepts Billy's rock. To account for the new range, we need to change the two equations where SH appears on the right hand side, i.e., all the variables on which SH has causal influence. First, the equation for BS is changed to reflect the fact that SH is no longer binary valued and explicitly checks for $SH = 1$, instead of just a truth value. Second, the equation for SH has to be changed to reflect Suzy's new power to intercept Billy's throw; we do this by only letting Billy break the bottle if Suzy does not throw and also does not intercept Billy.

5.3.2 Transfer of Power

A transfer of power is a specific change of power in which the powers need to be part of the original SCM $\mathcal{M}$. The transfer copies powers and commitments of one variable to another. Usually a transfer of power only happens between variables that represent the actions of social actors.

Definition 21 (Transfer of Power)
A transfer of power from a variable $A \in \mathcal{V}$ to a variable $B \in \mathcal{V}$ means that we transform an SCM $\mathcal{M}$ to a new SCM $\mathcal{M}'$ and add some sub range of A, $\mathcal{P}_A \subseteq range(A)$, to the range of B so that $range(B) = range(B) \cup \mathcal{P}_A$.

Additionally the commitment $\mathcal{C}_{B'}$ will be extended with certain options from A, $\mathcal{O}_A \subseteq \mathcal{C}_A$, so that $\mathcal{C}_{B'} = \mathcal{C}_B \cup \mathcal{O}_A$. In other words a transfer of power is a change of power, including potential changes to $\mathcal{F}$, transforming $\mathcal{M}$ to $\mathcal{M}'$, where the power needs to exist in the prior model $\mathcal{M}$.

The powers that are transferred can be removed from the range of A or be duplicated in B. It usually depends on whether a power is linked to some physical property (e.g., a tool) or some allowed action (e.g., issuing commands). The first cannot be duplicated while the second is immaterial and can exist an unlimited amount of time.[83]

To give an example, in one version of the pollution example above, $\mathcal{C}_A = \{0, \ldots, 100\}$, so company A is allowed to dump 100kg of pollutant into the river, while $\mathcal{C}_B = \{0\}$. $\mathcal{U}_{CEO\_A}$ could now decide to transfer some of their credits to company B. This would result in a transformed model $\mathcal{M}'_{Pollutant}$ that would be similar to the original, except that $\mathcal{C}_A$ would be smaller and $\mathcal{C}_B$ bigger. If $\mathcal{U}_{CEO\_A}$ decided to transfer the power to pollute 40kg from company A to company B, $\mathcal{C}_A = \{0, \ldots, 60\}$ and $\mathcal{C}_B = \{0, \ldots, 40\}$. These changes are external to the SCM and the new SCM can then be used to assess questions of responsibility.[84]

Here it is important to note that a transfer of power can also change the responsibility relations. If the whole range of A is transferred to B this means that A no longer has any causal influence. And since causality is a requirement for responsibility, $Actor_A$, whose actions are represented by A, can no longer be responsible for any effects.

5.3.3 Accountability

While responsibility is the attribution of the use of power, accountability is a means to restrict the use of power. In contrast to other means, such as violence, public shaming or economic pressure, it is well structured and based on transparency, which means that the person wielding the power knowingly (and usually also willingly) submits to certain conditions in order to gain access to this power.

Definition 22 (Restriction of Power)
In an SCM $\mathcal{M}$, the power of an exogenous social actor $Actor_A$ is called restricted, if there are some influences –possibly external to $\mathcal{M}$– that prohibit $Actor_A$ from using the full range of its powers, i.e. A, that models the actions of $Actor_A$, will never reach some possible values in its range.

For example, if a person is threatened with violence when they follow a certain course of action, their free choice is limited and they will –most likely– not follow that path.

[83] We do not consider the case where a principal might remove power from an agent. While this case is relevant in the whole "accountability life cycle", i.e., how power is transferred between social actors over time, this is beyond the scope of this text.

[84] This example does not require adaptations to $\mathcal{F}$; more complex transfers of power, such as adding the intercept action to Suzy in the previous example, might need some adaptions to $\mathcal{F}$.

Similarly, if a person is expecting to be punished for abuses of power, it is less likely that they will abuse it.

Accountability is a structured means to restrict the use of power and usually affects the system that the SCM represents in two ways that go beyond the SCM: It will prescribe reporting requirements from the recipient of power to the originator and it will give the originator the option to sanction the recipient, if it does not use the power according to its expectations.

Definition 23 (Accountability Relation)
An accountability relation in a system $\mathcal{S}$ is a tuple $(Actor_P, Actor_A, R, S)$ where $Actor_A$ has power and $Actor_P$ restricts the free use of this power by $Actor_A$. $Actor_P$ can do so by imposing reporting requirements, R, on $Actor_A$ and having the ability to impose sanctions, S, on $Actor_A$.

Accountability relations are often coupled to transfers of power. A transfer of power implies that there was a system $\mathcal{S}_{\text{prior}}$ in which $Actor_A$ had less power than it has in the current system $\mathcal{S}$. However, accountability can also exist without such a transfer of power. It can happen that accountability is used to restrict existing or developing powers. In this case there are no prior models to compare to. Since an SCM $\mathcal{M}$ of a system $\mathcal{S}$ will reflect the powers of social actors, accountability relations will also be visible in $\mathcal{M}$.

Accountability is only defined for social actors, so $Actor_A$ must always be a social actor, whereas $Actor_P$ can –in very rare cases– be non-human and even imaginary. For an example see Chapter 5.4.1. The exact form of R and S depends on the context and can range from oral reports to computer logs and from the threat of prison to just a stern frown. In some special cases, R and S can also be non existent.

This context specificity of accountability relations has given rise to different accountability definitions that solve the core problem, restricting the power of $Agent_A$, in slightly different ways.

Definition 24 (Accountability Definition)
An accountability definition $\mathcal{D}$ is an accountability relation with additional demands on the nature of the principal, the way reporting needs to be done and if and how $Actor_P$ can sanction $Actor_A$.

These demands induce patterns, $\mathcal{M}_P$, in $\mathcal{M}$, the SCM describing the system $\mathcal{S}$, that are distinguishing, but not necessarily unique, for $\mathcal{D}$. Formally an accountability definition is a tuple $(Actor_P, Actor_A, R, S, \mathcal{M}_P)$. Valid examples of accountability definitions are the ones describes by Bovens, Hall, Lindberg and RACI; however, this list is open to additions.

Our definition of accountability is casuistic in that it explicitly names four definitions of accountability that we consider as valid. The reason for this is that accountability is an open concept that is socially constructed which means that accountability is whatever we agree it to be. In an ideal world, we would like to have a generic definition of what an accountability definition is, but even just the four definitions here vary widely in their demands on the actor. There is no common thread, other in that they aim to restrict the use of power, by generating the feeling of oversight.

$\mathcal{M}_P$ describes in the abstract what causal influence (if any) the principal retains on the agent and the course of events. Additionally many accountability definitions have special reporting requirements and/or give the principals means to sanction the agent. These we capture as informal descriptions that need to be validated manually (for examples see below).

The fact that $Actor_A$ expects to be held accountable for action A will affect its reasoning process. However, we model $Actor_A$ as an exogenous variable, precisely because we cannot model how a human's reasoning will change in a different context. So accountability indirectly (via $Actor_A$) influences A. In some cases we might be able to model the influence of properties of accountability on A directly. An example might be the influence of frequency of speed traps on a highway on the driving behavior (see also Hall accountability, Section 5.4.1, below). Accountability also affects responsibility by changing the SCM $\mathcal{M}$ of a system. If, for example, the principal has no more causal influence on a variable V in $\mathcal{M}$, the principal can also no longer be responsible for its value. Similar an agent's greater power leads to greater responsibility.

5.4 Definitions of Accountability

5.4.1 Hall

At the core of the definition given by [Hall et al., 2017] (see Chapter 2.4) is an agent's expectation that their action will be monitored by a third party. However, it does not explicitly consider the fact that power is transferred. It focuses on the feelings (or perception) of the agent. As such, there is no formal transfer of power. We do not need to have a principal (so we do not have $Actor_P$ and $P = \varnothing$), we just need to introduce the possibility of an evaluation. As long as the agent *believes* it might be monitored, it will feel accountable and should constrain its use of power.[85] In other words, the idea that there is accountability will lead to better behavior. In extremis, a system does not have to do

[85] That is the power they already posses. In Hall's definition agents do not acquire new powers.

any logging, as long a person using it ($Actor_A$) does not know this. Random evaluation systems, like systems that only audit some (randomly chosen) transactions, are an example of this. Most of the time a transaction will not be evaluated, but the fact that some are evaluated is enough to deter misbehavior and abuse of power.[86] The graphical model in Figure 5.7 depicts this belief of being evaluated as an exogenous variable. Modeling it as an exogenous variable has the advantage that it can never directly be the cause of something. In reality *Belief* might also influence $Actor_A$, however in SCMs we cannot model the causal relations between exogenous variables and in practice we cannot model the decision process of a human. So while this influence is relevant, we cannot model it on the level of the individual. To give an example, if we find that the fear of speed traps reduces the average speed by 10%, the new model must reflect this in any probabilities. This means that the new model describes a different world than the original one: the new world contains an accountability mechanism and its causal effects that the old world did not contain.

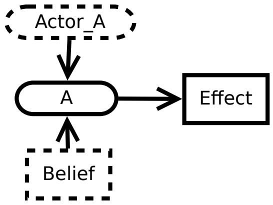


Figure 5.7: $\mathcal{M}_P$ for Hall: It does not does require a principal. It is enough if the agent believes there to be one; thus we model this belief as an exogenous variable.

So formally, $\mathcal{D}_{\text{Hall}} = (\varnothing, Actor_A, \varnothing, \varnothing, \mathcal{M}_P)$, with $\mathcal{M}_P = (\mathcal{U}, \mathcal{V}, \mathcal{F})$, where $Actor_A \in \mathcal{U}$, $Belief \in \mathcal{U}$, $A \in \mathcal{V}$, $Effect \in \mathcal{V}$ and $\mathcal{F} \in \mathcal{M}_P$ has at least two functions, $\mathcal{F}_A$ that depends at least on $Actor_A$ and *Belief* and $\mathcal{F}_{\text{Effect}}$ that depends at least on A.

Here it is important to note that Hall's notion of accountability is almost indistinguishable from our notion of responsibility (Definition 19). This is not an accident – in their systematic literature review Hall et al. used the search terms "accountability" and "responsibility" and they even note that "(...) responsibility and accountability have been used interchangeably in some of the literature". Their distinction is that accountability requires an external audience. So if the audience is missing (or just imaginary) their notion of accountability is indistinguishable from responsibility.

[86] Jeremy Bentham's Panopticon is a well known thought-experiment using the same effect; the fear of divine punishment is another.

5.4.2 Lindberg

As explained in detail in Chapter 2.2, [Lindberg, 2013] provides the following definition of accountability after a transfer of power:

1. *An agent or institution who is to give an account (A for agent);*
2. *An area, responsibilities, or domain subject to accountability (D for domain);*
3. *An agent or institution to whom A is to give account (P for principal);*
4. *The right of P to require A to inform and explain/justify decisions with regard to D; and*
5. *The right of P to sanction A if A fails to inform and/or explain/justify decisions with regard to D.*

In contrast to Hall above, we have a principal with very clearly defined roles: It transfers power to $Actor_A$ and is then in turn empowered to ask $Actor_A$ for explanations and justifications. Additionally, $Actor_P$ has the right to sanction $Actor_A$ if it does not provide an explanation. Formally, $\mathcal{D}_{\text{Lindberg}} = (Actor_P, Actor_A, R, S, \mathcal{M}_P)$, with $\mathcal{M}_P = (\mathcal{U}, \mathcal{V}, \mathcal{F})$, where $Actor_P \in \mathcal{U}$, $Actor_A \in \mathcal{U}$, $P \in \mathcal{V}$, $A \in \mathcal{V}$, and P depends at least on $Actor_P$, A depends at least on P and $Actor_A$, and effect depends at least on A. The causal link $P \to A$ captures that the actions of $Actor_A$ will often depend on some prior action of $Actor_P$. For example, $Actor_P$ might issue a command or provide some input for A, such as an essential document or advice. Figure 5.8 depicts the graphical model. The exact means in which $Actor_A$ has to inform $Actor_P$ is specified in R; this might be in very informal ways. Similarly S contains a, possibly informal, descriptions of how $Actor_P$ can sanction $Actor_A$. In principle $Actor_P$ can also retain some influence on the effect. So an edge $P \to Effect$ does not violate Lindberg's pattern, but $Actor_P$ would be responsible for its influence on the effect.

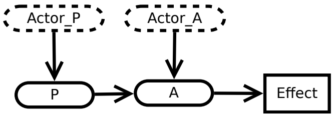


Figure 5.8: The causal graph for $\mathcal{M}$ suggested by Lindberg.

5.4.3 Bovens

Despite the mutual criticism, the definition of [Bovens, 2007] (see Chapter 2.3) is, seen as a causal model, very similar to the definition of [Lindberg, 2013] discussed above. His definition reads:

1. *There is a relationship between an actor and a forum*
2. *in which the actor is obliged*
3. *to explain and justify*
4. *his conduct,*
5. *the forum can pose questions,*
6. *pass judgement,*
7. *and the actor may face consequences.*

Where [Lindberg, 2013] calls for an *agent* giving an account to a *principal*, [Bovens, 2007] considers an *actor* that *explains his conduct* to a *forum*. Both the *principal* and the *forum* might sanction or judge the *agent* or *actor*. This similarity of definitions suggests to us that the causal models should also be similar. One pronounced difference is that Bovens explicitly suggests that the actor might be punished for the content of the account (so what it did) instead of just for the fact if it provided an account or not (as Lindberg suggests). Furthermore, Bovens requires the actor to regularly inform the principal about changes, whereas Lindberg sees the principal as asking for information.[87]

Formally, $\mathcal{D}_{\text{Bovens}} = (Actor_P, Actor_A, R, S, \mathcal{M}_P)$, with $\mathcal{M}_P = (\mathcal{U}, \mathcal{V}, \mathcal{F})$, where $Actor_P \in \mathcal{U}$, $Actor_A \in \mathcal{U}$, $P \in \mathcal{V}$, $A \in \mathcal{V}$, and P depends at least on $Actor_P$, A depends at least on P and $Actor_A$, and effect depends at least on A. Similarly to Lindberg above, the causal link $P \to A$ captures that the action of $Actor_A$ can depend on actions by $Actor_P$. Since Bovens is so similar to Lindberg, Figure 5.8 also depicts the graphical model for Bovens.

Bovens and Lindberg only differ in R and S. Boven requires regular reports from $Actor_A$ and Bovens allows for $Actor_P$ to sanction the agent for the content of its actions, not just if it does not give a report.

5.4.4 RACI

The *Responsible-Accountable-Consult-Inform (RACI)* framework definition of accountability [Smith et al., 2005], see Chapter 2.5, is as follows:

- ***R**esponsible: The individual who completes a task. Responsibility can be shared.*
- ***A**ccountable: The person who answers for an action or decision. There can be only one such person.*
- ***C**onsult: Persons who are consulted prior to a decision. Communication must be bidirectional.*

[87] Lindberg also allows for the principal to punish the agent. This version would be equivalent to Boven's.

- ***I**nform: Persons who are informed after a decision or action is taken. This is unidirectional communication.*

What is interesting about RACI is that it is very explicit about the transfer of power. There explicitly is only one "accountable agent" ($Actor_P$ in our terminology) and it has a veto power. So it can prevent the "responsible agent(s)" ($Actor_A$) from doing the task. This $Actor_P$ is then "ultimately answerable for the activity or decision." [Smith et al., 2005].

Formally, $\mathcal{D}_{\text{RACI}} = (Actor_P, Actor_A, R, S, \mathcal{M}_P)$ with $\mathcal{M}_P = (\mathcal{U}, \mathcal{V}, \mathcal{F})$, where $Actor_P \in \mathcal{U}$, $Actor_A \in \mathcal{U}$, $Consultants \in \mathcal{U}$ and $P \in \mathcal{V}$, $A \in \mathcal{V}$, $D \in \mathcal{V}$, $C \in \mathcal{V}$ $I \in \mathcal{V}$, and C depends at least on $Consultants$, P depends at least on $Actor_P$, A depends at least on D and $Actor_A$, D depends at least on C and P, the effect depend on A and also directly on P, and finally, I depends at least on the effect.

By introducing the informed agents, I, RACI models the reporting directly as part of the SCM. So here R is not informal, but needs to be part of the SCM via the edge $Effect \to I$. Similar to the other definitions, S is also not further specified here. Figure 5.9 depicts the graph.

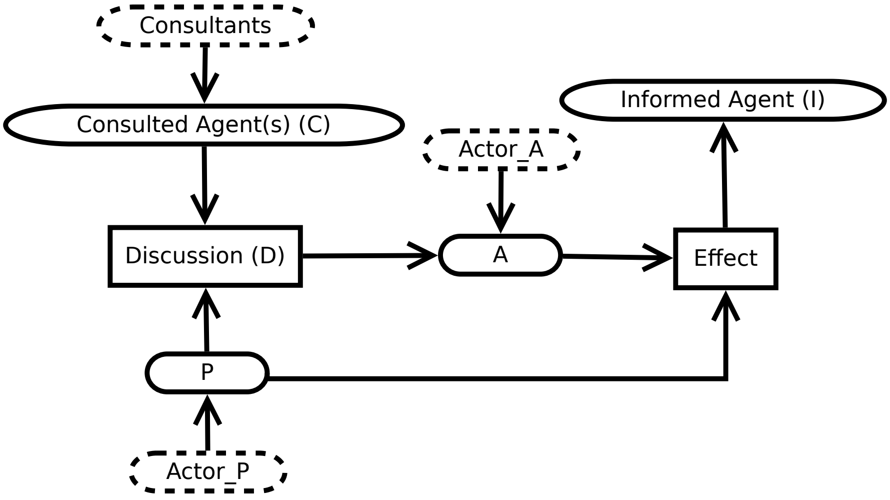
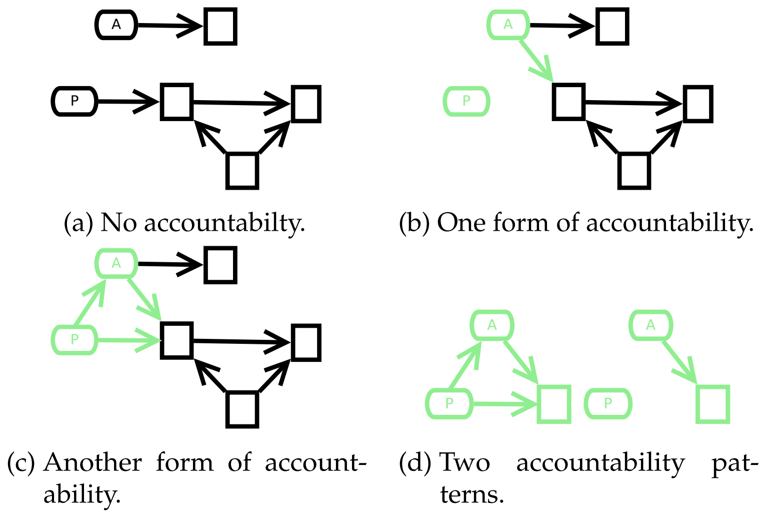


Figure 5.9: The RACI accountability pattern.

5.5 Accountability in Systems

The final goal now is to take an SCM $\mathcal{M}$ that models the causal relations of a system and evaluate if it fulfills a given definition of accountability $\mathcal{D}$ and find the necessary changes to make $\mathcal{S}$ and thus $\mathcal{M}$ fulfill $\mathcal{D}$. Figure 5.10 depicts this process.



(a) No accountabilty.

(b) One form of accountability.

(c) Another form of accountability.

(d) Two accountability patterns.

Figure 5.10: Figure 5.10a depicts the SCM of a system without any accountability. Figure 5.10b depicts the same SCM, with the power of $Actor_P$ transferred to $Agent_A$. In Figure 5.10c $Actor_P$ transfers power, but still retains some causal influence. Figure 5.10d depicts the SCMs required by two fictitious accountability definitions.

5.5.1 Evaluating a System for Accountability

To evaluate if $Actor_A$ is accountable to $Actor_P$ according to accountability definition $\mathcal{D}$, we need two inputs SCM $\mathcal{M} = (\mathcal{U}, \mathcal{V}, \mathcal{F})$, the SCM of the system, and accountability definition $\mathcal{D} = (Actor_P, Actor_A, R, S, \mathcal{M}_P)$ and then need to follow these steps:

E-1 If we cannot identify $Actor_A$ and, if required, $Actor_P$ as exogenous variables in $\mathcal{M}$, the answer is no.

E-2 If we are not convinced that the SCM $\mathcal{M}$ fulfills all the causal dependencies required by $\mathcal{D}$, the answer is no.[88]

E-3 If the reporting requirements, R, required by $\mathcal{D}$ are not met, the answer is no.

E-4 If there is no means for $Actor_P$ to sanction $Actor_A$ as specified by S, the answer is no.

E-5 Otherwise the answer is yes.

Here it is important that the original system, $\mathcal{M}$, does not necessarily need to exist.

[88] Ideally we would be able to have a formal proof, but evaluating the equivalence and similarity of causal model is still an open problem, see Chapter 3.9.8. However, since accountability is a social construct, it really is enough if we are convinced by the similarity of two models. This conviction could, of course, be challenged by others and we might be called upon to make an argument for our belief.

In many cases it will be a hypothetical system, for example in the case of parents and children, there is no real "system" without the children.

Examples for accountability follow in Section 5.5.3.

5.5.2 Transform a System to contain an Accountability Relation

For this we need two inputs: SCM $\mathcal{M} = (\mathcal{U}, \mathcal{V}, \mathcal{F})$, the SCM of the system, and accountability definition $\mathcal{D} = (Actor_P, Actor_A, R, S, \mathcal{M}_P)$ and then need to follow these steps to get SCM $\mathcal{M}_D$ as output that describes how the SCM of the accountable system should look like; symbolically $\mathcal{M} \xrightarrow{\mathcal{D}} \mathcal{M}_{\mathcal{D}}$:

T-1 Identify the principal and the agent in $\mathcal{M}$. For that, find two exogenous variables in $\mathcal{M}$ that represent the reasoning process of a social actor.

T-2 Change the range and/or commitment of A to reflect the transfer of power from $Actor_P$ to $Actor_A$ (Definition 21). The new SCM is called $\mathcal{M}_D$.

T-3 Check that the causal dependencies in $\mathcal{M}_D$ comply with the requirement of $\mathcal{M}_P$ in $\mathcal{D}$. If they do not comply, adapt $\mathcal{M}_D$ to comply.

T-4 Ensure that the reporting requirements, R, required by $\mathcal{D}$ are met.

T-5 Ensure that the system has the ability to facilities sanctions for $Actor_A$ according to the requirements S given in $\mathcal{D}$.

5.5.3 Examples

Forest Fire

If we go back to the example of the forest fire (Section 5.2.2), we can easily see that lacking a second social actor and thus violating condition E-1, $\mathcal{M}_{FF}$ cannot fulfill $\mathcal{D}_{\text{Lindberg}}$, $\mathcal{D}_{\text{Bovens}}$, or $\mathcal{D}_{\text{RACI}}$. It also does not fulfill $\mathcal{D}_{\text{Hall}}$, because MD does not depend any form of $Belief$ in an external audience, violating condition E-2.

If $\mathcal{M}_{FF}$ was a system that we develop, we could, for example, aim to make $Actor_A$ (called U_2 in the model) accountable according to $\mathcal{D}_{\text{Hall}}$. To achieve this, we first need to identify the agent (Hall does not require a principal) to fulfill condition T-1 and condition T-2. Next, for condition T-3, we would need to change the system so that the system is an SCM with an edge from $Belief \rightarrow A$. A simple way doing this in a real world setting would be to lie to $Actor_A$ and make it believe that someone is watching. This means that without any additional technical system and only minimal cost, we would be compliant with this specific accountability definition. In this case the structural equation for A would change from $MD = Actor_A$ (i.e. to drop a match or not) to $MD = Actor_A \land \neg\ Belief$ meaning that a match is only dropped if $Actor_A$ decided to drop a match and it does not believe there to by anyone watching.[89] Since Hall does not require any reporting or sanctioning, condition T-4 and condition T-5 are trivially met.

Another goal could be to aim for $\mathcal{D}_{\text{Lindberg}}$. However, for this to make sense we change the example's backstory slightly.[90] First, as Lindberg requires a principal and to satisfy condition T-1, we need to add $Actor_P$ to the model. In our modified example, this will be the chief of the fire brigade, so P models the action of the chief. Because she knows that a wild fire is about to start, she instructs a firefighter to start controlled burns to reduce the hazardous fuel in the area. This means she sets the firefighter's commitment from $\mathcal{C}_{MD} = \{0\}$ to $\mathcal{C}_{MD} = \{1\}$,[91] so this means that $Actor_P$ transfers the power to start controlled burns to fight the fire to the firefighter (condition T-2). However, of course the chief would not blindly trust anyone with that power. To counterbalance the firefighter's new power to freely start fires, she sets up R so that the firefighter has to wear a body cam whenever he is in the forest (condition T-4). Additionally, S specifies that the firefighter will have to go to jail if he is found starting fires without permission or wearing an active body cam (condition T-5). Figure 5.11 depicts the new SCM $\mathcal{M}_{FF\_\text{Lindberg}}$ that satisfies condition T-3 .

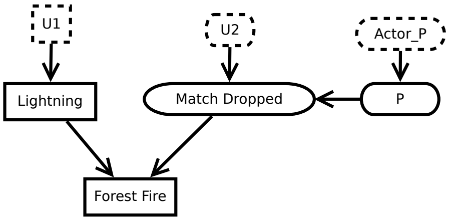


Figure 5.11: The graphical model for the forest fire example with Lindberg accountability. Here, $U2 = Actor_A$ and $MD = P$.

Rock Throwing

Looking at the Suzy-Billy rock-throwing example (Section 5.2.2), we do see two social actors, however the models do not fulfill the causal dependencies required by any accountability definition (condition E-2). This means that the causal graph of $\mathcal{M}_{SB}$ is not semantically equivalent to the M given by any accountability definition. For example, to

[89] Of course we could also keep this as part of the decision process of $Actor_A$, but making it explicit makes the model more understandable.

[90] To be honest, we are stretching this example almost to the limit.

[91] Setting it to $\mathcal{C}_{MD} = \{0, 1\}$ would allow the firefighter to not do the burns, i.e., not do his job, which is not what we want to model.

be Bovens accountable, we lack a principal, as well as any reporting duties or sanctioning options.

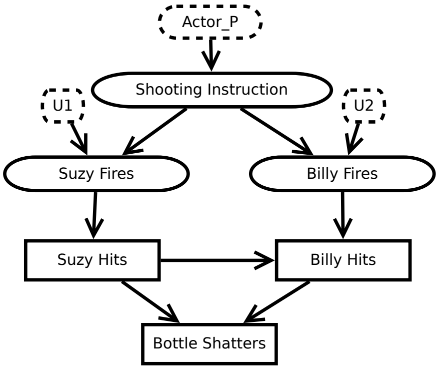


Figure 5.12: Transforming the Suzy-Billy example into $\mathcal{M}_{SB\_\text{Bovens}}$.

If we now want the system described by $\mathcal{M}_{SB}$ to exhibit, for example, the $\mathcal{D}_{\text{Bovens}}$, it again helps to slightly adapt the story of the example. If we change the setting and see Suzy and Billy as soldiers that shoot at a target, we can add a principal $Actor_P$ that is giving them permission to shoot (condition T-1). Adding this principal would change the commitment of Suzy and Billy from not shooting to shooting, so $\mathcal{C}_S = \mathcal{C}_B = \{1\}$ meaning that in the context of the shooting range it is normal for them to shoot (condition T-2).[92] For R we would have them report each hit or miss to $Actor_P$ (condition T-4) and S might be negative points for missed shots (condition T-5). The structural equations would change to include $Actor_P$ and the Shooting Instruction (SI): $SI = Actor_P$, $SF = \mathcal{U}_1 \wedge SI$, $BF = \mathcal{U}_2 \wedge SI$[93] and the other equations would be as before: $SH = SF, BH = BF \wedge \neg\, SH, BS = SH \vee BH\}$. Figure 5.12 depicts the graphical model ratifying condition T-3.

This model is interesting, because it shows that we can have two accountability relation in a single model. Here $Actor_P$ is, in a sense, multiplying power by sharing it between multiple agents. All agents have to report back to $Actor_P$, so in the system design, we need to take special care that R is able to handle this many reports. Another interesting aspect in this model is that the effect is mediated by a third variable. So we do not have a link $SF \rightarrow BS$, but the effect of Suzy's shot is mediated by a third variable SH, giving us the causal chain $SF \rightarrow SH \rightarrow BS$. In general causes can often be split into multiple variables and it is up to the modeler to find a meaningful decomposition. This

[92] Depending on the context, their commitment could also be {0,1}, indicating that not shooting is also acceptable. However, in, for example, a shooting competition, not shooting would be abnormal in the sense that it you get no points and fail.

[93] $U1$ and $U2$ are the traditional names for the exogenous variables modeling Suzy and Billy's decision process. In our terminology we would call them $Actor_S$ and $Actor_B$.

also means that when comparing the semantics of $\mathcal{M}$ to the model prescribed by $\mathcal{D}$ we cannot use simple pattern matching to evaluate if $\mathcal{M}$ is semantically equivalent to the M in $\mathcal{D}$. This is the problem of showing the equivalence of causal models and is beyond the scope of this text.[94]

Pollutant

As with the previous examples, the pollutant example (Section 5.2.2) does not fulfill any definition of accountability, because we lack principals and agents (condition E-1) and the SCM is not semantically equivalent to any accountability definition (condition E-2). And similar to the other examples, we need to slightly change the backstory for accountability to make some sense. Here we change it so that the example becomes RACI accountable. First, we assume that $\mathcal{U}_{CEO\_B}$ is enabling an operator to also remove pollutant from the river, for example by buying some new machinery (condition T-1). This now changes the range of B from $\mathbb{N}$ to $\mathbb{Z}$, which allows negative numbers and thus us to model the removal of pollutant (condition T-2). Furthermore, the amount that is removed depends not just on $Actor_A$, so the agent, but also on a discussion between $\mathcal{U}_{CEO\_B}$ and some consultants, C (condition T-3). Finally, the state of the river is actively reported to some environmental agency, giving us R (condition T-4). Here it is important to note that RACI explicitly requires a social actor that gets informed about decisions or actions. So here $Actor_A$ does not need to report informally, but there is a node (I) that is part of the SCM that if affected by the Effect For S we have the standard means to sanction someone in a company (condition T-5).

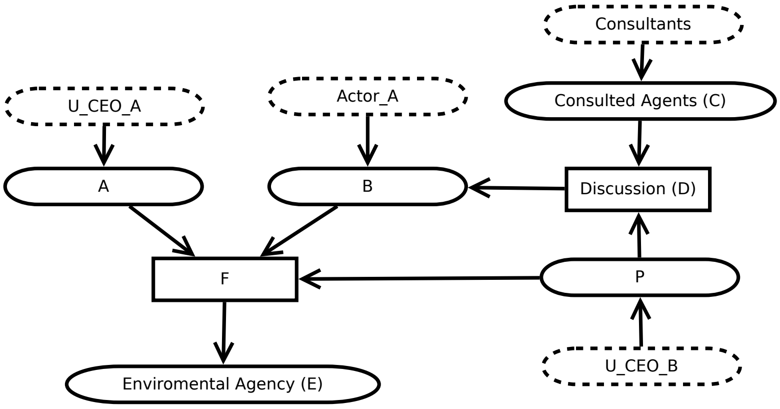


Figure 5.13: Transforming the pollutant example into a model that exhibits RACI accountability.

[94] See [Beckers, 2021b] for more details.

This now heavily changes the SCM $\mathcal{M}_{Pollutant}$, see Figure 5.13. Where the original model only had two exogenous variables ($\mathcal{U}_{CEO\_A}$ and $\mathcal{U}_{CEO\_B}$), the new model, $\mathcal{M}_{Pollutant\_RACI}$ has four: $\mathcal{U}_{CEO\_A}, \mathcal{U}_{CEO\_B}$, $Actor_A$ and $Consultant$ modeling the decisions of the respective entities. Similarly, instead of just three endogenous variable (A, B, F), it has seven: A (the actions of $\mathcal{U}_{CEO\_A}$), B (the actions of the operator $Actor_A$), F (the state of the river), E (the data reported to the environmental agency), D (the decision about the amount of pollutant), C (the input from the consultants), P (the actions of $\mathcal{U}_{CEO\_B}$). For B the range changes from $\mathbb{N}$ to $\mathbb{Z}$ and $\mathcal{C}_B$ will also change to negative numbers. While the structural equations for A stays the same ($A = \mathcal{U}_{CEO\_A}$), they change for all other nodes. First, $E = F$, indicating the report of the state of the river; that the reporting shows up in the SCM is unique to RACI. Next, $C = Consultants$ is will be a number of the suggested amount. P is now a tuple, (V, N), where V indicates that the CEO vetoes the decision of $Actor_A$ and sets N as the amount for the pollution of plant B. D is now the average between the suggestion of the CEO and the consultants, so $D = (C + P.N)/2$. B is now the decision of $Actor_A$ averaged[95] with D, i.e., $B = (Actor_A + D)/2$. F now finally is $A + B > k$ if $P.V = 0$, so there is no veto, or $A + P.N > k$ otherwise.

What is interesting here is that in the new model the range of B changes, not just the commitment. Another interesting aspect is the interplay of accountability and responsibility. If $P.V = 0$, $Actor_A$ is responsible for F just like before, however this responsibility is shared with C and P. If however, the CEO overrides $Actor_A$ with $P.V = 1$, B no longer has any causal influence in F and $Actor_A$ as well as the consultants are also no longer responsible for the result. This agrees very much with our intuition that a CEO will always bear some responsibility and that an actor that has no causal influence because the decision is made “over its head” bears no responsibility.

[95] We use averages for simplicity; in reality this would be a more complex mathematical expression.

6 Use Case

6.1 Introduction

In this chapter we take a real world use case and show how we can derive SCMs of a system and then use this SCM to reason over the system's responsibility and accountability relations. As a use case, we analyze the 2018 deadly crash of an Uber car [Elish, 2019].[96] In this accident, an autonomous vehicle developed by Uber crashed into a pedestrian, Elaine Herzberg, crossing a road and is regarded as the first accident in which a pedestrian was killed by an autonomous vehicle. Ms. Herzberg was pushing a bicycle while crossing a dimly lit road and the software of the car repeatedly misclassified her, ultimately hitting and killing her. The safety driver on board the vehicle was distracted at the time and also did not brake in time. Using this accident as a seed, we will develop a scenario that illustrates how formal definitions of accountability can be used to specify accountability expectations for systems a priori, and avoid any confusion about questions of accountability.

In the aftermath of the crash the accountability of the parties was hotly contested. At first the police claimed it was the pedestrian's fault, because at the site of the accident, crossing the road was illegal. Next, the car's safety driver was blamed, because she did not pay attention to the road. The manufacturer of the car's chassis, Volvo, was quick to distance themselves from any blame, arguing that their chassis had a collision avoidance system which would have prevented the crash, but it was turned off by Uber to test their own software. Velodyne, the manufacturer of the car's LiDAR, also pointed out that their system was capable of detecting a pedestrian, but that their system does not take the decision to brake. The search for reasons went as far as criticizing Uber's development process, the testing process of having only one driver in the car and even the car-friendly (and pedestrian-hostile) layout of the road in Arizona or the point of autonomous cars in general.

All these questions have in common that they ask about counterfactuals in a causal model. The problem now is to find rules or patterns to create these causal models.

[96] See for example [Harris, 2019]; a detailed write-up can be found here: `https://en.wikipedia.org/wiki/Death_of_Elaine_Herzberg`

We need to find ways to integrate lessons learned from ex-post assessments into the ex-ante design of systems. This will allow us to derive clear accountability structures that manufacturers can follow and compare their designs to. Having such clear structures allows us to exclude some actors (e.g., pedestrians) from accountability and explicate who has what powers and who has delegated power to whom.

For example, if we follow a very simplistic understanding of accountability and say that the manufacturer of an autonomous car is giving power to a car[97] and should thus ensure that this power is not abused, the manufacturer would be liable for any damages caused by the car. If this is explicit, companies can adapt to this rule and only build cars that go 2 km/h[98] and can stop the moment their sensors recognize anything in their sensor range.[99] In most jurisdictions today, the driver is liable for anything their car causes. However, if there is an accident and a driver kills another person to save their own life, this is usually excused if there was no (reasonable) way to avoid the death.[100] So the driver is still accountable, but no longer liable and will not be punished, because they are *without guilt*.

In the following, we will derive the SCM for an autonomous car from its technical architecture and then show how we can identify definitions of accountability in it and how to alter it to confirm to some specific notion of accountability.

6.2 Technical Architecture

First, we look at a technical architecture. Since only little is known about the exact architecture used in the Uber car, we use the architecture described by [Taş et al., 2016] and assume that Uber did not do something radically different from the state of the art.[101] Figure 6.1 depicts this functional architecture. On the highest level, an autonomous car has sensors that will record their environment and output raw sensor data. This raw data is then fed into a sophisticated module to "perceive and understand" this data and generate a model of the environment. Finally, this model is used to decide on the next action of the car (e.g., steer or speed up) which then is realized by affecting the actuators

[97] So the manufacturer has causal influence on the car and this influence enables the car to affect the world around it.

[98] In other words: they limit the power of the car to influence the physical world.

[99] This approach is, for example, used in automatic doors: They close slowly and will stop the moment they detect an obstacle.

[100] German law, for example, has the concept of "Entschuldigender Notstand" (§35 StGB), roughly translates to "necessity".

[101] To support that, a recent survey by [Badue et al., 2021] finds that in the literature, the fundamental design of the architecture for self-driving cars has not changed and a recent blog post by Uber [Guo et al., 2020], suggests that they use functional blocks similar to the presented architecture.

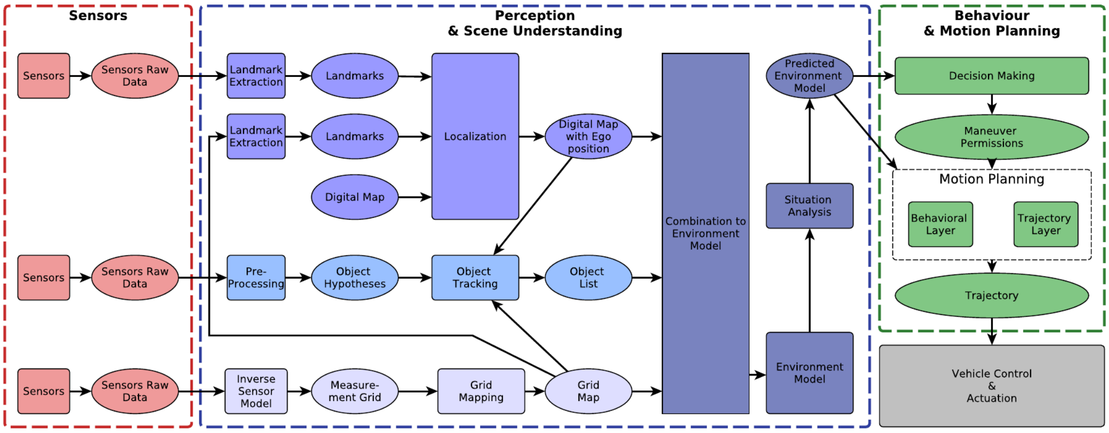


Figure 6.1: The functional architecture of a contemporary autonomous car as given by [Taş et al., 2016].

of the car (e.g., changing the steering angle or increasing the engine speed).

For similar autonomous cars [Bhavsar et al., 2017] conducted a risk analysis and published fault trees. Figure 6.2 depicts a simplified version that we reduced only to hardware, software, and mechanical failures. This fault tree informs us that, perhaps trivially, each of these failures can lead to a failure of the vehicle similar to the crash in the Uber accident. So if this fault tree was complete[102], it could inform our causal model and rule out other possible causes for the accident.[103] In Figure 6.3, we aggregate this knowledge in the form of the node *Autonomous Vehicle Failure* that represents the unwanted event we want to have accountability for.

We apply the methodology described in Chapter 4, to convert technical models like these to causal models. The only problem is that the process is not deterministic, so any technical model must be discussed and experts need to agree on weather a specific model makes sense or not. Figure 6.3 shows one possible SCM derived from the technical architecture in Figure 6.1 and the fault tree in Figure 6.2.[104]

Here, each node in the graph stands for something we can observe (i.e., log). For example, *Vehicle Control* might be an n-tuple that represents all the necessary inputs to steer the car into a certain direction. *Decision Making* might contain an explanation for the control algorithm's choice of action. The edges in the graph denote causal connections. This means that, for example, the value of *Situation Analysis* depends on

[102] The depicted version is already shortened compared to the original, so it most definitely is not complete!

[103] See Chapter 4 for a discussion on translating models such as fault trees to SCM.

[104] There are many other SCMs that can be derived from such an technical architecture. Our goal is not to find the best SCM for that particular architecture, but illustrate what can be done with the SCM once we have it.

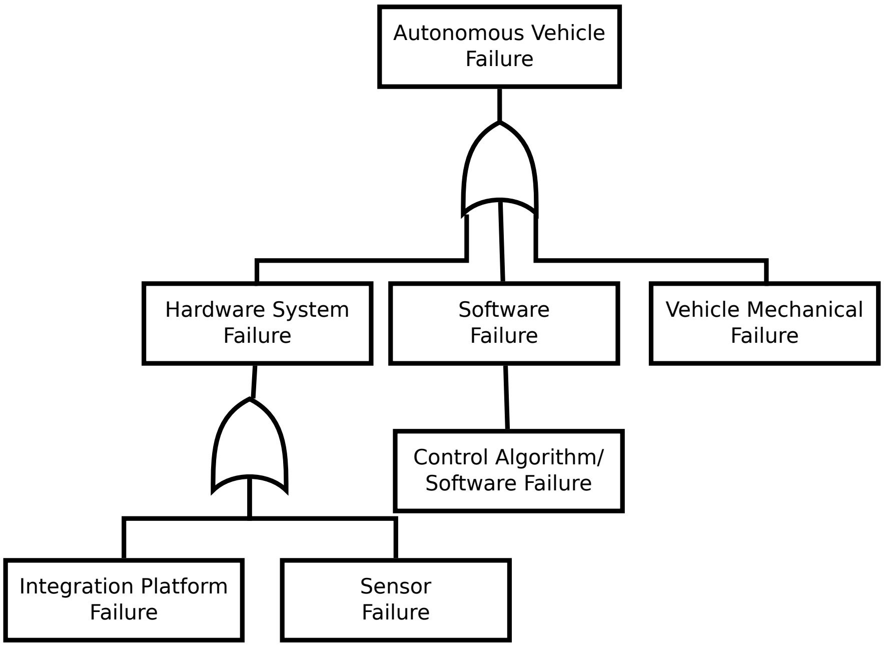

Autonomous Vehicle Failure
Hardware System Failure
Software Failure
Vehicle Mechanical Failure
Control Algorithm/ Software Failure
Integration Platform Failure
Sensor Failure

Figure 6.2: A simplified version of the fault tree for an autonomous vehicle given by [Bhavsar et al., 2017].

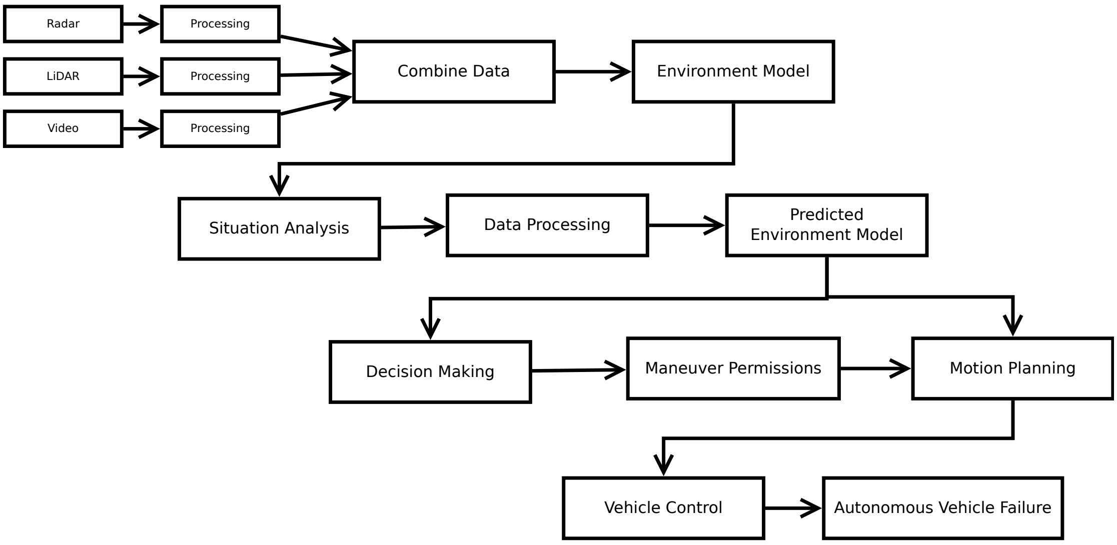


Figure 6.3: One possible SCM for the technical architecture.

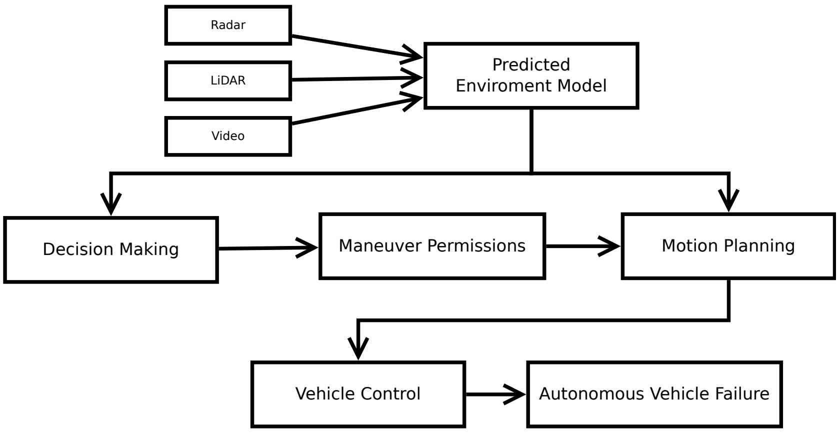


Figure 6.4: A simplified, more abstract version of the model in Figure 6.3.

the *Environmental Model* or that the only input to *Vehicle Control* is *Motion Planning* and potentially an exogenous variable. All these details are in principle knowable, however it is sometimes also possible to analyze a model when data is missing. In Figure 6.3 we can see a front-door structure (see Chapter 3.7.2) between *Predicted Environmental Model*, *Decision Making*, *Maneuver Permissions* and *Motion Planning*. Leveraging the front door structure allows us to discern the effect of *Decision Making* on *Motion Planning* without the need to know the value of *Predicted Environmental Model*.

Since this model quickly becomes complex, we follow [Beckers and Halpern, 2019] and abstract parts of the model to make it easier to understand. Figure 6.4 depicts a model that is based on Figure 6.3, but has some fewer nodes. In this example, we removed the causal chain $CombineData \rightarrow EnviromentModel \rightarrow SituationAnalyis \rightarrow DataProcessing$ and connect the three sensors directly to the Predicted Environment Model. This is fine in this model, because the chain has no side effects, but in the general case always we need to make sure that no important transformations get lost.

6.3 Modeling the Social Context

It is important to note that technical components have to be treated differently from humans when it comes to questions of responsibility, accountability, and liability. While they can be accountable in the sense that their power can be limited, holding them to account for their actions does not mean the same as it means for a human. A human might be punished, but a machine has no concept of pain and can at best be turned off or be reprogrammed. Because our intuitive understanding of accountability is deeply linked to concepts of responsibility, liability, and ultimately free will it is important to

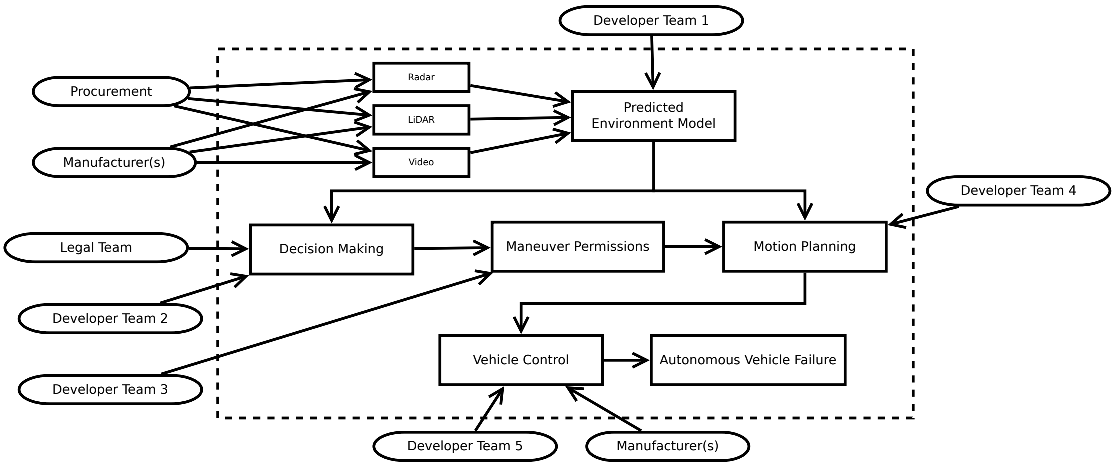


Figure 6.5: Extending the model in with the actions of some social actors. The dashed frame indicates the boundaries of the model in Figure 6.4.

connect any technical model with a model of the social world around it. Every machine is built by humans, put into action by humans, and interacts with humans. Without these humans in the causal models, notions of accountability, responsibility and liability make no sense. [105]

If we go back to the facts of the Uber accident, we have several social actors that are relevant for this scenario: Uber, Volvo, and Velodyne all contributed parts of the technology. The pedestrian and the safety driver were part of the socio-technical system while it was in operation. Some managers might have decided to use only a single safety driver, whereas city planning officials have influence on the road design. We can extend this list almost without end. However, the goal of a good accountability model is to capture the relevant entities and help understand their accountability. As with the technical model, there is no unique or correct solution, different models suit different purposes. Again, experts need to agree on the correctness of a model and the SCM is merely a tool to facilitate communication. However, once we agree on a model, we can then check if it confirms to our requirements and expectations.

Figure 6.5 extends the technical model with the actions of social actors and their influence (for readability we follow the convention of not modeling exogenous variables explicitly). This SCM models the manufacturers that build the sensors, the procurement department that decides on which sensors to buy, different developer teams that deliver the components, as well as a legal team that gives input on how to make maneuver decisions so that they comply with existing laws and regulations. However, so far we

[105] See Chapter 2.8 for a detailed discussion.

do not yet include details from the actual accident such as the pedestrian or the safety driver.

To illustrate the semantics, the node *Legal Team* might simply stand for the decision of a specific algorithm that tries to resolve the Trolley problem. This algorithm might be part of the *Decision Making* and be implemented by *Development Team 2* under guidance from the *Legal Team*. This means that any decision of this algorithm not only depends on the *Predicted Environmental Model*, but also on the values of *Legal Team* and *Development Team 2*. Here, possible values for *Legal Team* might be possible ways to resolve the Trolley problem, such as *prefer young people over old*, *flip a coin*, or *brake and don't steer*. After an unwanted event (e.g., an accident), we could analyze the model and consider alternative decisions by the *Legal Team*. Similarly, the value range of *Development Team 2* could be all possible lines of source code. In principle we could then test if a specific change to the source code would have avoided an unwanted event.[106]

To analyze the actual accident, we first simplify the example by abstracting the model in Figure 6.5 and then extending it with some additional nodes to model the accident scenario.[107] In Figure 6.6, the model shown in Figure 6.5 is abstracted to a single node called *Software*. Additionally, we add a node *Emergency Brake* that could have prevented a collision with a pedestrian, as well as the safety driver and the pedestrian. Figure 6.6 shows three different versions of this new SCM in a single graph.[108] If there is a connection $SD \rightarrow T$, the human can take over at any point in time. If there is a link $SD \rightarrow S$ any input by the human can be overridden by the machine, and if SD is not connected to the graph, the human cannot influence the car at all. This example is highly simplified, but to illustrate the real-world use, we will provide realistic value ranges for the variables and examples for the structural equations. The meaning of the variables is a follows:

Collision with Pedestrian, CwP, has a binary range. It is true if a collision with a pedestrian occurs and false otherwise.

Trajectory, T, is a vector in 3D space that describes the heading and the speed of the car.

Safety Driver, SD, is a unity vector in 3D space that describes the steering input of the safety driver.

[106] Of course in practice this needs to be simplified, otherwise the combinations are virtually infinite!

[107] See [Beckers and Halpern, 2019] on how to abstract SCMs.

[108] A slightly different and binary version of this model was first presented in [Kacianka and Pretschner, 2021].

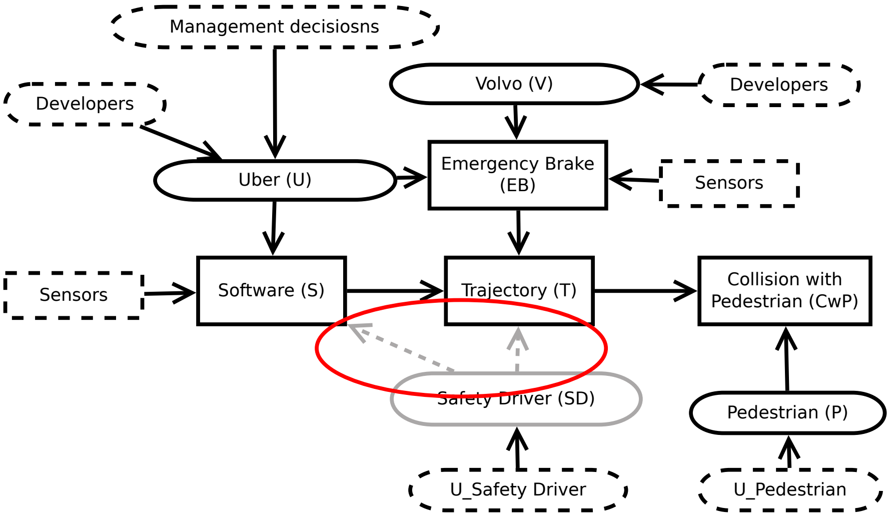


Figure 6.6: One possible SCM for the design of the Uber car. The red circle indicates that this SCM can be read in three ways: (1) $SD \rightarrow T$ holds, which implies that the human can take over, (2) $SD \rightarrow S$ is true, in which case the human influence is moderated by the machine, or (3) SD is not connected to the graph, meaning that the human has no causal influence (and thus power) whatsoever.

Pedestrian, P, is a unity vector in 3D space that describes the movement of the pedestrian.

Software, S, is also a unity vector in 3D space that describes the steering input from the car's control software.

Emergency Brake, EB, is a scalar that describes the deceleration of the car in m/s. In the example, we assume that -5 would be enough to avoid the accident.

Volvo, V, is an n-tuple that describes Volvo's contribution to the source code of EB. In our example, we assume that element 42 needs to read `brake-if-radar-ping = true` needs to be set to avoid a collision.[109]

Uber, U, is also an n-tuple that describes Uber's contribution to the source code of the car's control software and the software of the emergency brake. In the example, we assume that if element 13 reads `disable-emergency-brake = true` the emergency brake is disabled. If element 64 reads `improved-object-detection`

[109] 42 is a randomly chosen number without any special significance. As an analogy we could think of source code lines. For simplicity, we ignore the hardware here.

`= true`, the car would be able to detect the pedestrian and brake in time.[110]

Formally, our model $\mathcal{M} = ((\mathcal{U}, \mathcal{V}, \mathcal{R}), \mathcal{F})$ looks like this:

$\mathcal{U} = \{MD, DE_U, DE_V, SE_S, SE_{EB}, U_{SD}, U_P\}$, for management decisions, developers, sensors, the safety driver, and the pedestrian; in short $\mathcal{U}$ would contain all the exogenous variables that we care about.

$\mathcal{V} = \{CwP, T, SD, S, EB, V, U, P\}$ are the eight endogenous variables.

Now we need to define $\mathcal{F}$, so the structural equations for every endogenous variable:[111]

U, P, V, SD get their values from the exogenous variables. For example, we could have $\mathcal{F}_V(DE_V) = DE_V$, for the developer output, $\mathcal{F}_{SD}(U\_SD) = U\_SD$ for the safety driver and $\mathcal{F}_P(U\_P) = U\_P$. U would be some combination of DE_U and MD, without giving the exact details on how they are combined, we write the combination of the developers and management at Uber is given by $\mathcal{F}_U(DE_U, MD) = DE_U \otimes MD$.

EB is defined by three input variables: U, V and SE_{EB}. $U \otimes V$ describes the control flow when joining the control software from Uber and Volvo and SE_{EB} describes the sensor input. Together the joint control flow and the input map to a scalar that describes the car's deceleration; so $\mathcal{F}_{EB}(U, V, SE_{EB}) = \text{exec}(U \otimes V, SE_{EB})$, where `exec` is a function that executes a control flow with a given input. For the example we assume that if `brake-if-radar-ping = true` is present, the car will brake in time, i.e., the value of EB will be ≤ -5, and if `disable-emergency-brake = true` is present it will preempt any control logic coming from V, thus rendering the brake inactive.

For S we have two equations, depending on whether or not our model has the link $SD \to S$. If there is no such link, i.e., the software does not mediate the safety driver's effect on T, the 3D vector for S would be the output of the control logic U with the sensor data SE_S as input; symbolically $\mathcal{F}_S(U, SE_S) = \text{exec}(U, SE_S)$. Here the simplified understanding is that if U contains `improved-object-detection = true` the accident could be avoided. If, however, there is a link $SD \to S$, i.e., the software mediates the user trajectory, the output of the control logic would be joined with the user vector, i.e., $\mathcal{F}_S(U, SE_S, SD) = \text{exec}(U, SE_S) \oplus SD$, where $\oplus$ denotes an operation to join two 3D vectors.

[110] Again, these numbers have no special meaning. In reality the granularity of single source code lines might also be too fine.

[111] For simplicity we ignore any uncertainty; see [Halpern, 2016, Ch. 2.5] for a detailed discussion.

T now also has two equations, depending on if there is a link $SD \rightarrow T$ or not. Without the link, $\mathcal{F}_T(S, EB) = EB \odot S$, where $\odot$ denotes an operation to scale the vector. With the link it would be $\mathcal{F}_T(S, EB, SD) = EB \odot (S \oplus SD)$.

CwP would now be the result of the intersection of the 3D vector T and P, the pedestrian's vector. If they do intersect, CwP is *true* and otherwise *false*: $\mathcal{F}_{CwP} = intersect(T, P)$.

6.4 Responsibility

To understand responsibility in this model, we need to understand the commitments of the social actors SD, P, U, and V. First, the range of SD is a unity vector in 3D space, capturing the steering input of the driver. Here the commitment $\mathcal{C}_{SD}$ will be any vector that will avoid a collision. However, some vectors that lead to a collision might also be a valid commitment. For example, if the car was driving on a highway within the speed limit and a pedestrian suddenly jumps in front of the car, this valid commitment will still lead to a collision. Here, we also need to understand the causal relation. If the driver had no chance to spot the pedestrian, she also had no way to avoid to collision and without a (credible) counterfactual that avoids the collision the driver also cannot be a cause for it. Similar reasoning applies to P.

For V the commitment is to ensure that the emergency brake is working as intended. So $\mathcal{C}_V$ would include the line `brake-if-radar-ping = true`. For U it is harder to understand the commitment. `disable-emergency-brake = true` means that the emergency brake is disabled, which is necessary for the autonomous features of the car to work. So provided that the software works, disabling EB is reasonable. It is up for interpretation if disabling EB is part of $\mathcal{C}_U$ or not. `improved-object-detection = true` should be part of $\mathcal{C}_U$, however showing that object detection indeed works is not trivial.

If we now analyze this model retrospectively, and try to understand who is responsible for $CwP = true$, we first need to understand if P is a cause for $CwP = true$. This depends of whether there was a reasonable and possible alternative course of action for P. While pedestrians are not supposed to cross that stretch of the road, Ms Herzberg had already crossed two of the three lanes and could expect the car to notice her. Using this reasoning, we assume that she had no counterfactual course of action at her disposal and is thus not a cause for the collision and thus also not responsible for it. Next, we can assess the responsibility of the safety driver. First, any assessment depends on the exact influence that SD has. If SD can influence T, the safety driver could have prevented the crash by steering or braking. This would also be the driver's commitment. However, unfortunately the driver was distracted by her phone and did not do so and thus violated $\mathcal{C}_{SD}$. This is why SD is a cause and also responsible.

Looking at V, we find that any influence of V is preempted by U, so V cannot be a cause for the behavior of EB and thus also not responsible for the crash.

The analysis for U now depends on our understanding of the commitment. Since there was a collision, we know that the trajectory chosen by the software lead to an unwanted event and that EB could not prevent the unwanted event. We also know that there was a counterfactual course of action that could have prevented the collision (braking or steering). Following the reporting of the accident, it seems that Uber used too few sensors to reliably detect the pedestrian, while turning off the emergency braking system to avoid unreliable braking. This suggests that U did violate the commitment to ensure the safety of its cars. Thus U is a cause and also responsible for the collision.

6.5 Accountability

So far we completed the first step in our methodology (see Chapter 5.5). We created $\mathcal{M}$, defined expected values for each node and specified the structural equations $\mathcal{F}$. The next step is to identify an event for which we define accountability. In this model, we chose CwP, so the collision with the pedestrian. For the third step, we need to choose a definition of accountability ($\mathcal{D}$). In this example, we will look at the definitions of Lindberg (Chapter 5.4.2) and Hall (Chapter 5.4.1).[112] To start step E-1 from our methodology, we need to identify potential social agents. In $\mathcal{M}$ we can identify four such agents: Uber (U), Volvo (V), the pedestrian (P) and the safety driver (SD).

Next (step E-2), we need to check if $\mathcal{M}$ fulfills the causal dependencies required by $\mathcal{D}$. This means that for Lindberg accountability we need to find $Principal \rightarrow Agent \rightarrow Effect$ and for Hall accountability $Agent \rightarrow Effect$. Looking at $\mathcal{M}$ (Figure 6.6), we can find the following causal chains leading to the crash: $U \rightarrow S \rightarrow T \rightarrow CwP$, $V \rightarrow EB \rightarrow T \rightarrow CwP$, and $SD \rightarrow T \rightarrow CwP$ (see Figure 6.7).

If we just compare the graphical patterns, no chain is Hall accountable, because no social actor directly causes CwP. However, if we analyze $\mathcal{F}$, we can see that, if U contains `disable-emergency-brake = true`, EB will be disabled, which means that the link $EB \rightarrow T$ is removed from the model. If we, furthermore, assume the model in which SD has no causal influence on S and T, we are left with the causal chain

[112] Making this model RACI accountable can be done, but would require many additional nodes, similar to Chapter 5.5.3

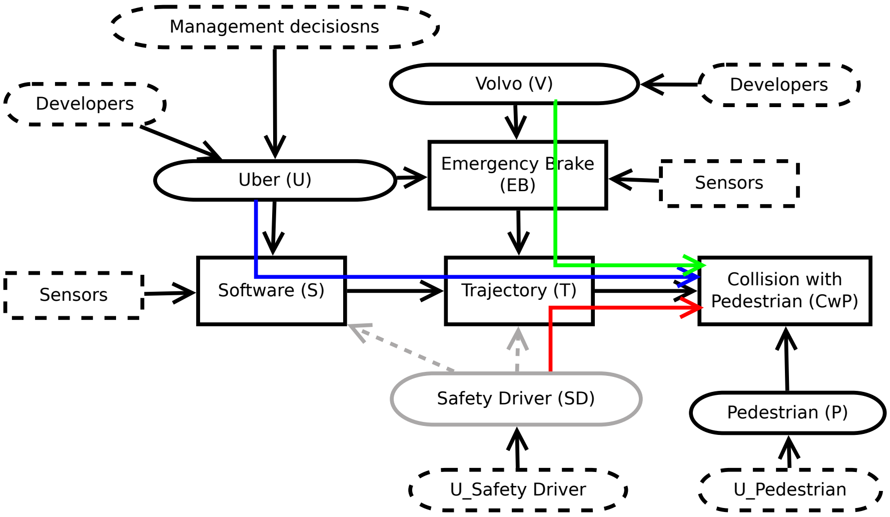


Figure 6.7: The three causal chains that lead to CwP: $U \rightarrow S \rightarrow T \rightarrow CwP$ (blue), $V \rightarrow EB \rightarrow T \rightarrow CwP$ (green), and $SD \rightarrow T \rightarrow CwP$ (red). We assume that the link $EB \rightarrow T$ is disabled by the value of U.

$U \rightarrow S \rightarrow T \rightarrow CwP$. While this chain is not equal to Hall's $\mathcal{D}$ in the graphical model, we can argue that S and T are merely mediators that transfer the causal influence from U to CwP. Because we assume the sensor input as an exogenous variable that cannot be changed and that P could not have acted differently, a change in the value of U would indeed have prevented the accident which means U is responsible for the crash. For a more detailed attribution, we would then need to look at an SCM of U to understand who exactly was responsible and resolve the accountability questions within Uber.

If we assume the link $SD \rightarrow T$, we can look at the chain $SD \rightarrow T \rightarrow CwP$. Because the value of SD affects T, we can no longer make the argument that T is just a mediator for U. In this case, we would need to conclude that accountability is ambiguous. To resolve it, we would either need to change the system so that, for example, $\mathcal{M}$ did not contain the link $SD \rightarrow T$ and Uber would again be Hall accountable.

Another interesting observation is that the chain $U \rightarrow S \rightarrow T \rightarrow CwP$ would not be Lindberg accountable, because S is not a social actor. If Lindberg accountability is our goal, we could either replace the S with a human driver, or change the regulatory landscape to give the software of the car some form of legal personhood. Another option to argue for Lindberg accountability would be to add a node for Society (SO) to the model and a link $SO \rightarrow U$ (so SO and U would be social actors). In this case, we could, again, join S and T into a mediator and see the chain $SO \rightarrow U \rightarrow (S + T) \rightarrow CwP$ as accountable according to Lindberg's definition.

If our model contained the link $SD \rightarrow S$, so that the input from the driver would pass through the software system (for example in a drive-by-wire setup), Lindberg accountability would be unclear, because both U and SD (provided we also had the link $SO \rightarrow SD$) could be accountable to society for the crash. In this scenario, we would need to take a closer look at the structural equation for S. So far, $\mathcal{F}_S(U, SE_S, SD) = \text{exec}(U, SE_S) \oplus SD$ gives us no indication if S or SD has more influence on the result. However, if we wanted U to be accountable, we could design $\mathcal{F}_S$ in such a way that the results from U always preempt SD. A real world example would be an emergency brake system that will always override the driver. So if $\mathcal{F}_S$ was designed similar to the link $U \rightarrow EB$, i.e., U preempts SD, we would again have an argument that U is Lindberg accountable to SO for CwP.

A final property of our model is that, if SD has no link to S and T, the safety driver has no causal influence. This also means that the safety driver cannot be held to account for the outcome, CwP. So if we can proof the absence of causality, we can also show the absence of accountability. We could use this in the design of the system and, for example, design a car without a steering wheel or other use input, thus turning the safety driver into a passenger. Causal independence is especially useful, because even in fairly big models, we can detect it automatically, see for example [Tikka and Karvanen, 2018].

Without adding society to it, $\mathcal{M}$ depicted in Figure 6.7 has no direct causal relations between social actors. The reason for this is that we abstracted most of Uber into a single highly abstract node U. To find the necessary social actors, we would need to take a closer look at U. Doing so will reveal causal links between humans. For example, most likely some manager instructed developers to do (not) implement a certain feature relevant to the accident. Similarly, the legal department probably gave the green light for the way the car was designed. All these causal connections and transfers of power are investigated by authorities after such a crash. Unfortunately, we have no specific details as Uber and the family of Ms Herzberg settled out of court and the trial of the driver has not yet begun.

Similarly, we have very little actual knowledge about steps E-3 and E-4, so the reporting requirements and the sanctions. Because autonomous cars are a very new problem for courts, there is currently no precedence based on which sanctions could be modeled on. As for the reporting, the current view is that Uber had so many logs and data recorders that there would have been few points of contention for a jury to decide (which probably partly explains the quick out-of-court settlement with the family of Ms Herzberg).

Finally, in Chapter 3.7, we introduced the Front- and the Back-Door criteria for causal models that help us identify variables that we do not need to measure, because their causal effect can be deduced from other variables. If we now look at Figure 6.6, we might just be interested in the effect of U on T. The question now is, what other values do we need to control for to calculate the effect of U on T? Employing the Back-Door Criterion, we can see that we do have an open back-door path, namely $S \leftarrow U \rightarrow EB \rightarrow T$, that will confound our estimate of the effect of U on T. To deconfound this reading, we could either control for S, EB, or both of them. This is interesting, because this allows us to justify not logging one of these variables, provided we are only interested in the effect of U on T. The advantage for the system design would be that we do not have to log any data that influences EB, such as V, CL and SE. Especially not having to log the sensor data might make the system easier to design. For example, if the sensor is a camera, we would not have to worry about any privacy implications invoked in handling and storing the data. However, as storing such data might be useful for general debugging, we could at least justify not logging the data in full detail and could, for example, anonymize them.

6.6 Summary

This chapter first gave an example on how we can construct the causal model $\mathcal{M}$ for an autonomous car. We then extended this model with social actors and showed how we can evaluate if this model is accountable according to some definition $\mathcal{D}$. Next, we showed that we can also make adapt this model to fulfill a specific $\mathcal{D}$ by, for example, adding nodes. Finally, we showed an example of how we can use the structural equations and the graph in $\mathcal{M}$ to compare the accountability in a system where the driver can influence the car compared to a system where the driver is just a passenger. In this example, we also show how we can leverage the structure of $\mathcal{M}$ to justify not logging specific values. While all this reasoning can, of course, be done without SCMs and definitions of accountability, our approach allows us to state our understandings unambiguously and follow a strict methodology.

7 Conclusion

In this thesis we have joined three fields of research that have not previously been in much contact. First, we have used the fact that accountability is a means to restrict power and used the assumption that power is the same as causal influence to model accountability definitions. Second, we gave an introduction into SCMs and showed how we can use their semantics to model socio-technical systems. Third, we proposed a methodology to evaluate if the SCM of a socio-technical systems is accountable according to a specific definition. If this is not the case, we can analyze the SCM and suggest modification to the system's model that would make it compatible with a given accountability definition. Finally, we presented a use case based on the domain of autonomous driving.

7.1 Discussion

The big question at the end of this thesis is if SCMs are indeed a useful tool to express and reason about the accountability of systems (or more precisely their models). Ideally we would have loved to show a formal proof that they do or do not work. Unfortunately, we learned very early on that the concept of accountability is hotly debated and as such offers no unified definition or consensus to work with. Similar to [Pearl and Mackenzie, 2018] and [Halpern, 2016] on causality, we are left with arguing our point and hoping that our reasoning and the examples are convincing. A major short coming in our work is that all our examples are comparably simple and not derived from a real use case. Integrating our notion of accountability into an actual development process would have given us valuable insights into the applicability and usability of our method. The main obstacle in doing such an evaluation is that the main building blocks for our idea, accountability and SCMs, are relatively unknown concepts. While accountability is at least known as a concept to pay lip service to, hardly anyone is familiar with the intricacies of accountability theories.[113] To complicate matters, causal modeling is a relatively new branch of mathematics and few developers are familiar with it.[114] So we

[113] See [Kacianka et al., 2017] for a detailed discussion.

[114] As anecdotal evidence, at the time of writing our university was not offering any courses teaching about it.

end up with two concepts that are alien to most people and would thus first need to develop a comprehensive idea of how to teach them to developers or, alternatively, try to package them into modeling tools.

Closely related to this problem is the general problem of getting useful causal models of systems. In the current literature causal models are usually generated as part of scientific studies. Such scientific models are usually pretty small and are used to communicate assumptions about mechanisms in a study. As such, experts spend a lot of time and care in getting their models right. And while we presented some ways to convert existing models, such as fault or attack trees, into causal models, this process is not automated and most of the time these source models will also not exists, or not exist in a decent quality. So overall, we would need to better understand how to convert system models to SCMs.

Another problem is the semantics of these causal models. The structural equations that are the backbone of an SCM can take any form, so in theory we can represent any relation between two variables in an SCM. However, in practice most modelers, especially those building models for actual causality reasoning, restrict themselves to binary models. The reason for this is that such models are easy to build, reasoning over them is much less complex and the semantic is usually reduced to something along the lines of "A has (or has no) a causal influence on B". However, if we want to use causal model to represent a concept like accountability, we need to think about how to handle scenarios like "A developer contributed a line of code" or "The pedestrian wore clothes that are hard to see". As far as we are aware, there are no other examples in the literature using such complex semantics and we assume that there might be better ways to model and reason over them.

If we look at accountability, we will face problems identifying principals and justifying any notion of accountability. For principals we cannot really give a clear cut answer who should be included and who should not be – this really needs to be decided on a case-by-case basis. The same is true for a specific notion of accountability. Each notion has advantages and downsides and there no simple algorithm to automate this process.

Despite all these shortcomings, we are convinced that SCMs offer the best means to express accountability in a well-defined formalism. The reason for this is that, at its core, accountability is about restricting the use of power. And since SCMs describe causal influences and what powers can be exerted, accountability must be visible in an SCM. This is so deeply ingrained into the concept, that in any definition of accountability an actor can only be held to account for an event that they caused (i.e., had power over) and without causality there can be no accountability. And in the current state of the art SCMs offer the best understood means to model such causality relations. It stands to reason that using SCMs to formalize accountability mechanisms can directly benefit from any developments in the SCM community. In this thesis we have shown how to leverage basic structures such as the Front- and Back-Door criterion to improve the system design and future improvements in the analysis and reasoning over SCMs could then be leveraged to analyze accountability structures. In a similar vein, we assume that improvements to the development of SCMs will help us create the models, even for more complex systems.

This leaves us with the final question if it makes sense to try and express a fuzzy and "social" concept such as accountability in formal models. In the legal systems the exact formulation of laws is usually open to some interpretation and we then have trails and judges to apply the laws to the exact context. This open approach has served humanity well so far, so why should we change this now? The main argument in favor of using formal models is that with the spread of automatic and autonomous systems there are just too many possible cases that would need to be manually analyzed. Having a formal definition of accountability forces us to resolve ambiguities and ultimately paves the way for (semi-)automatic reasoning in case of unwanted events. This would allow us to scale the number of investigations in line with the number of systems, something that is impossible if there are only a few experts who understand accountability relations.

7.2 Future Work

Following this thesis, we see many avenues for future research. In the area of causality, a focus should be put on developing methods to build causal models. Based on our own work in [Kacianka et al., 2019a], developing ways to automatically derive causal models from existing system models seems to be a promising approach. Instead of relying on models such as fault trees, we suggest to investigate how to derive causal models from source code directly. Especially the fact that source code can be represented as a graph[115] suggests that it should be possible to find some useful translation to causal models. If this level of granularity is too fine, it might be worth investigating other graphical representation of programs[116] or even UML diagrams. SCMs are flexible enough to represent all these structures, but finding a good translation is a hard problem.

Which leads us to the highly related problem of translating models of human behavior to causal models. We need this to sensibly reason over the interaction between humans

[115] For example, call graphs [Grove and Chambers, 2001].

[116] For example control flow graphs [Cota et al., 1994].

and machines, but unfortunately expressing human behavior formally is a much harder problem. In [Kacianka et al., 2019b] and later [Biebl et al., 2021], we made some first steps into this direction, but there is still too much that is not known. First, getting reliable models of general human behavior is hard and made worse by the fact that in a specific, or actual causal, scenario an individual human might stray very far from any norm. So in addition to good models, we also need sensors to instantiate these models. And while logging computer systems is in general an uncontroversial topic, tracking (or "logging") humans to the necessary degree comes with a host of ethical and, more specifically, privacy problems.

If we imagine that we actually have decent models of humans and technical systems, we need to work on ways to merge these models automatically. In [Kacianka et al., 2019a], we presented some initial ideas on how to do this, but as of now this problem is still hotly discussed in the SCM community. Similarly, developing ways to compare causal models is another open problem. While [Beckers, 2021b] presents one way to compare them for equivalence, there are no ways to rank them for quality or even to find a measure of similarly between different causal models.[117]

Looking at the engineering side of things, there are unfortunately also precious few tools to automate the development and reasoning of causal models. [Ibrahim et al., 2020] presented a framework to automate the reasoning over actual causality models, but, especially for accountability, we also need frameworks that can handle type causal models and help us design models that are suitable for technical systems. While we here fully commit to Pearl's notion of causality and the Halpern-Pearl notion of actual causality, if these notions are the best is still up for debate [Beckers, 2021a]. There have been multiple definitions of causality proposed over the years and just because SCMs are popular does not mean that they are also always the proverbial "best tool for the job". Understanding where they work best and where other notions might be an improvement is another important open problem.

Finally, our whole argument builds on the premise that a good forward looking type causal model is also a good backward looking actual causality model. To us this seems plausible and it seems to be true in all the examples we have looked at over the years. However, we have not proven this correct. It might very well be that this is only true for some type of models and this approach falls apart in certain classes of models. A detailed investigation of this question might yield interesting insights into how to best employ SCMs as models of socio-technical systems.

[117] While causal models can be represented as graphs, they are not graphs and thus measures of graph similarity are, unfortunately, only of limited use.

When we look into the field of accountability, the most pressing question is to understand what definitions of accountability make sense for technical systems and to understand their implications. Especially considering the discussions about bias in systems, finding definitions that conform to societal expectations seems like a hard problem. In general understanding how we derive accountability requirements and who is setting them is a complex topic that would require further study. Tightly connected to this is the question if SCMs are really a suitable formalization and if this formalization itself might introduce problems.[118] Once definitions of accountability and their impact are understood, we could develop them into distinct accountability patterns. In [Kacianka and Pretschner, 2021], we made some first steps into this direction, many of the fundamentals are still in their infancy and would need further study. In general, our focus was technical and we focused on how to use one method, SCMs, to develop a methodology to assess the accountability of a system. However, accountability is first and foremost a social concept, and it will pay off to spend much more time with researchers from the fields of sociology, philosophy and political sciences and law. These and similar fields have very detailed and often diverging understandings of the term accountability, as well as related terms like responsibility, liability, transparency, causality and free will. It would be immensely useful to systematize the knowledge there and revise our results with these findings.

When looking at computer science, compared to well understood properties like safety or security, accountability is still poorly understood and investigated. It would be interesting to look at methods from these fields[119] and understand if they can be used to express accountability properties for systems. In doing that, we could then also verify if our SCM approach is correct and if it work in practice. Following this, it would be interesting to see if SCMs might also be useful to express other types of system requirements and how SCMs relate to other modeling languages.

7.3 Final Remarks

This thesis started out with the goal of finding “the” definition of accountability that is suitable for socio-technical systems. It seemed obvious, that we could “simply” take a definition from computer science, do some adaptations and implement it in a technical system. In hindsight, this naivety seems silly, but as we have learned in [Kacianka et al., 2017] it is all too common. Unfortunately accountability is an elusive concept that

[118] This will be tightly connected to the problem of finding suitable semantics in causal models.

[119] For example, safety cases [Kelly, 1999].

requires a thorough interdisciplinary understanding. This thesis is an attempt at treating this problem from three angles, namely accountability, causality and CPS. Despite the complexities of each field, we hope that this thesis gives the reader a comprehensive overview of the intricacies of the problem.

In RQ1, we wanted to understand how we can express the accountability of a system in a formal model. Our expectation is that the analysis of a formal model can be automated and will thus keep pace with the increase of systems. After working on this thesis, we are convinced that SCMs are a great possible means to do that. They allow to connect technical and social models and reasoning over them can be automated. Furthermore, because causal influence can be equated to power, they describe the necessary core of accountability, causality. We expect that even in the future, no formal definition of accountability will be complete without an understanding of causality at its core.

For RQ2, we reduced the question of measuring accountability to the problem of measuring and comparing SCMs. This is problematic for two reasons: First, there is not that much literature on how to compare SCMs. This is still pretty much an open research problem. Second, we are not fully convinced that we can reduce all the details of accountability to an SCM. It seems to work for the general structure of an accountability definition, but often times these definitions are very fuzzy and by presenting one causal model, we simultaneously hide all ambiguity. So while agreeing on a graphical model seems straight forward, finding efficient ways to agree on all the details of the structural equations is still an open problem.

RQ3 and the question about the effects on the system design has the most potential for future work. While we could show some implications in our example, we assume that there probably are some more general rules or patterns underlying accountability in the system design. We assume that leveraging insights from other disciplines might help to identify and formalize these patterns.

Our final take-away from this thesis is that as a means to restrict power, accountability can be expressing with SCMs, because they model causal influence, which can be equated to power. This insight allows us use a formal language to express accountability and thus improves our ability to discuss it. While natural language discussions of the nature of accountability are very useful and yield deep insights, we need the precision of a formal notation to uncover discrepancies between definitions and then also realize them in technical machines. Despite their current shortcomings, we think that Structural Causal Models are the most suitable formalization available for this task.

Bibliography

[Abdelwahed et al., 2011] Abdelwahed, S., Dubey, A., Karsai, G., and Mahadevan, N. (2011). Model-based tools and techniques for real-time system and software health management. In Srivastava, A. and Han, J., editors, *Machine Learning and Knowledge Discovery for Engineering Systems Health Management*, chapter 9. Chapman and Hall/CRC.

[Abdelwahed and Karsai, 2009] Abdelwahed, S. and Karsai, G. (2009). Failure prognosis using timed failure propagation graphs. In *The International Conference of the Prognostics and Health Management Society*.

[Abdelwahed et al., 2005] Abdelwahed, S., Karsai, G., and Biswas, G. (2005). A consistency-based robust diagnosis approach for temporal causal systems. In *The 16th International Workshop on Principles of Diagnosis*, pages 73–79.

[Alrajeh et al., 2018] Alrajeh, D., Chockler, H., and Halpern, J. Y. (2018). Combining experts' causal judgments. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*.

[Ananny and Crawford, 2018] Ananny, M. and Crawford, K. (2018). Seeing without knowing: Limitations of the transparency ideal and its application to algorithmic accountability. *New Media & Society*, 20(3):973–989.

[Anderson et al., 1997] Anderson, J. R., Matessa, M., and Lebiere, C. (1997). ACT-R: A theory of higher level cognition and its relation to visual attention. *Human-Computer Interaction*, 12(4):439–462.

[Angwin et al., 2016] Angwin, J., Larson, J., Mattu, S., and Kirchner, L. (2016). Machine bias: there's software used across the country to predict future criminals. And it's biased against blacks. ProPublica 2016.

[Annett, 2003] Annett, J. (2003). Hierarchical task analysis. In *Handbook of cognitive task design*, pages 41–60. CRC Press.

[Badue et al., 2021] Badue, C., Guidolini, R., Carneiro, R. V., Azevedo, P., Cardoso, V. B., Forechi, A., Jesus, L., Berriel, R., Paixão, T. M., Mutz, F., de Paula Veronese, L., Oliveira-Santos, T., and De Souza, A. F. (2021). Self-driving cars: A survey. *Expert Systems with Applications*, 165:113816.

[Baron and Kenny, 1986] Baron, R. M. and Kenny, D. A. (1986). The moderator–mediator variable distinction in social psychological research: Conceptual, strategic, and statistical considerations. *Journal of personality and social psychology*, 51(6):1173.

[Beckers, 2021a] Beckers, S. (2021a). Causal sufficiency and actual causation. *Journal of Philosophical Logic*, pages 1–34.

[Beckers, 2021b] Beckers, S. (2021b). Equivalent causal models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6202–6209.

[Beckers and Halpern, 2019] Beckers, S. and Halpern, J. Y. (2019). Abstracting causal models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2678–2685.

[Bellemare and Bloem, 2019] Bellemare, M. F. and Bloem, J. R. (2019). The paper of how: Estimating treatment effects using the front-door criterion. Technical report, Working Paper.

[Bhavsar et al., 2017] Bhavsar, P., Das, P., Paugh, M., Dey, K., and Chowdhury, M. (2017). Risk analysis of autonomous vehicles in mixed traffic streams. *Transportation Research Record*, 2625(1):51–61.

[Biebl et al., 2021] Biebl, B., Kacianka, S., Unni, A., Trende, A., Rieger, J. W., Lüdtke, A., Pretschner, A., and Bengler, K. (2021). A causal model of intersection-related collisions for drivers with and without visual field loss. In *International Conference on Human-Computer Interaction*, pages 219–234. Springer.

[Bovens, 2007] Bovens, M. (2007). Analysing and assessing accountability: A conceptual framework 1. *European law journal*, 13(4):447–468.

[Chockler and Halpern, 2004] Chockler, H. and Halpern, J. Y. (2004). Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115.

[Cota et al., 1994] Cota, B. A., Fritz, D. G., and Sargent, R. G. (1994). Control flow graphs as a representation language. In *Proceedings of Winter Simulation Conference*, pages 555–559. IEEE.

[Diakopoulos, 2015] Diakopoulos, N. (2015). Algorithmic accountability: Journalistic investigation of computational power structures. *Digital journalism*, 3(3):398–415.

[Dubnick, 2010] Dubnick, M. (2010). 'A Moral Being is an Accountable Being': Adam smith and the ethical foundations of accountable governance. In *68th Annual Meeting of the Midwest Political Science Association, March*, pages 22–25.

[Elish, 2019] Elish, M. C. (2019). Moral crumple zones: Cautionary tales in human-robot interaction. *Engaging Science, Technology, and Society*, 5:40–60.

[Feigenbaum et al., 2011] Feigenbaum, J., Jaggard, A. D., and Wright, R. N. (2011). Towards a formal model of accountability. In *Proceedings of the 2011 Workshop on New Security Paradigms*, pages 45–56. ACM.

[Feigenbaum et al., 2012] Feigenbaum, J., Jaggard, A. D., Wright, R. N., and Xiao, H. (2012). Systematizing "accountability" in computer science (version of feb. 17, 2012). Technical report, YALEU/DCS/TR-1452, Yale University, New Haven, CT.

[Felici and Pearson, 2014] Felici, M. and Pearson, S. (2014). D: C-2.1 report detailing conceptual framework. *Deliverable D32*, 1:A4CLOUD.

[Felzmann et al., 2019] Felzmann, H., Fosch-Villaronga, E., Lutz, C., and Tamo-Larrieux, A. (2019). Robots and transparency: The multiple dimensions of transparency in the context of robot technologies. *IEEE Robotics & Automation Magazine*, 26(2):71–78.

[Fosch-Villaronga et al., 2018] Fosch-Villaronga, E., Felzmann, H., Ramos-Montero, M., and Mahler, T. (2018). Cloud services for robotic nurses? Assessing legal and ethical issues in the use of cloud services for healthcare robots. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 290–296. IEEE.

[Friedenberg and Halpern, 2018] Friedenberg, M. and Halpern, J. Y. (2018). Combining the causal judgments of experts with possibly different focus areas. In *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning*.

[Graja et al., 2018] Graja, I., Kallel, S., Guermouche, N., Cheikhrouhou, S., and Hadj Kacem, A. (2018). A comprehensive survey on modeling of cyber-physical systems. *Concurrency and Computation: Practice and Experience*, pages 48–50.

[Greenland et al., 1999] Greenland, S., Pearl, J., and Robins, J. M. (1999). Causal diagrams for epidemiologic research. *Epidemiology*, pages 37–48.

[Grove and Chambers, 2001] Grove, D. and Chambers, C. (2001). A framework for call graph construction algorithms. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 23(6):685–746.

[Guo et al., 2020] Guo, Y., Ashmawy, K., Huang, E., and Zeng, W. (2020). Under the hood of Uber ATG's machine learning infrastructure and versioning control platform for self-driving vehicles. `https://eng.uber.com/machine-learning-model-life-cycle-version-control/`. [Online; acc. 2020-03-07].

[Hall et al., 2017] Hall, A. T., Frink, D. D., and Buckley, M. R. (2017). An accountability account: A review and synthesis of the theoretical and empirical research on felt accountability. *Journal of Organizational Behavior*, 38(2):204–224. JOB-13-0646.R4.

[Halpern, 2015] Halpern, J. Y. (2015). A modification of the Halpern-Pearl definition of causality. In *International Joint Conference on Artificial Intelligence*, pages 3022–3033.

[Halpern, 2016] Halpern, J. Y. (2016). *Actual causality*. MIT Press.

[Halpern and Pearl, 2005a] Halpern, J. Y. and Pearl, J. (2005a). Causes and explanations: A structural-model approach. Part I: Causes. *The British Journal for the Philosophy of Science*, 56(4):843–887.

[Halpern and Pearl, 2005b] Halpern, J. Y. and Pearl, J. (2005b). Causes and explanations: A structural-model approach. Part II: Explanations. *The British Journal for the Philosophy of Science*, 56(4):889–911.

[Harris, 2019] Harris, M. (2019). NTSB Investigation Into Deadly Uber Self-Driving Car Crash Reveals Lax Attitude Toward Safety . `https://spectrum.ieee.org/ntsb-investigation-into-deadly-uber-selfdriving-car-crash-reveals-lax-a` [Online; acc. 2021-02-25].

[Ibrahim et al., 2020] Ibrahim, A., Klesel, T., Zibaei, E., Kacianka, S., and Pretschner, A. (2020). Actual causality canvas: A general framework for explanation-based socio-technical constructs. In *ECAI 2020*, pages 2978–2985. IOS Press.

[Ibrahim and Pretschner, 2020] Ibrahim, A. and Pretschner, A. (2020). From checking to inference: Actual causality computations as optimization problems. In *International Symposium on Automated Technology for Verification and Analysis*, pages 343–359. Springer.

[Kacianka et al., 2017] Kacianka, S., Beckers, K., Kelbert, F., and Kumari, P. (2017). How accountability is implemented and understood in research tools. In *International Conference on Product-Focused Software Process Improvement*, pages 199–218. Springer.

[Kacianka et al., 2020] Kacianka, S., Ibrahim, A., and Pretschner, A. (2020). Expressing accountability patterns using structural causal models. *arXiv preprint arXiv:2005.03294*.

[Kacianka et al., 2019a] Kacianka, S., Ibrahim, A., Pretschner, A., Hartsell, C., and Karsai, G. (2019a). Practical causal models for cyber-physical systems. In *NASA Formal Methods Symposium*, pages 211–227. Springer. Note: In this paper Amjad Ibrahim and Severin Kacianka contributed equally and the order of the authors is alphabetically in the actual paper. It is flipped here, to explicate that Severin Kacianka is also a first author.

[Kacianka et al., 2019b] Kacianka, S., Ibrahim, A., Pretschner, A., Trende, A., and Lüdtke, A. (2019b). Extending causal models from machines into humans. In *Proceedings Forth Workshop on Causal Reasoning for Embedded and safety-critical Systems Technologies (CREST 2019)*, pages 17–31.

[Kacianka et al., 2016] Kacianka, S., Kelbert, F., and Pretschner, A. (2016). Towards a unified model of accountability infrastructures. In *Proceedings First Workshop on Causal Reasoning for Embedded and safety-critical Systems Technologies (CREST 2016)*, pages 40–54.

[Kacianka and Pöchhacker, 2020] Kacianka, S. and Pöchhacker, N. (2020). Algorithmic accountability in context. socio-technical perspectives on structural causal models. *Frontiers in big Data*, 3. Note: In this paper Nikolaus Poechhacker and Severin Kacianka contributed equally, but the order of authors is reversed in the original paper. It is flipped here, to explicate that Severin Kacianka is also a first author.

[Kacianka and Pretschner, 2018] Kacianka, S. and Pretschner, A. (2018). Understanding and formalizing accountability for cyber-physical systems. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 3165–3170. IEEE.

[Kacianka and Pretschner, 2021] Kacianka, S. and Pretschner, A. (2021). Designing accountable systems. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 424–437.

[Kaufmann et al., 2018] Kaufmann, M., Egbert, S., and Leese, M. (2018). Predictive policing and the politics of patterns. *The British Journal of Criminology*, 59(3):674–692.

[Kelly, 1999] Kelly, T. P. (1999). *Arguing safety: a systematic approach to managing safety cases*. PhD thesis, University of York York, UK.

[Ko et al., 2011] Ko, R. K., Kirchberg, M., and Lee, B. S. (2011). From system-centric to data-centric logging-accountability, trust & security in cloud computing. In *2011 Defense Science Research Conference and Expo (DSR)*, pages 1–4. IEEE.

[Kölbl and Leue, 2020] Kölbl, M. and Leue, S. (2020). An algorithm to compute a strict partial ordering of actions in action traces. In *International Symposium on Leveraging Applications of Formal Methods*, pages 10–26. Springer.

[Künnemann et al., 2019] Künnemann, R., Esiyok, I., and Backes, M. (2019). Automated verification of accountability in security protocols. In *2019 IEEE 32nd Computer Security Foundations Symposium (CSF)*, pages 397–39716. IEEE.

[Küsters et al., 2010] Küsters, R., Truderung, T., and Vogt, A. (2010). Accountability: definition and relationship to verifiability. In *Proceedings of the 17th ACM conference on Computer and communications security*, pages 526–535. ACM.

[Lamport, 1978] Lamport, L. (1978). Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*.

[Lee, 2008] Lee, E. A. (2008). Cyber physical systems: Design challenges. In *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)*, pages 363–369. IEEE.

[Lee et al., 1985] Lee, W.-S., Grosh, D. L., Tillman, F. A., and Lie, C. H. (1985). Fault tree analysis, methods, and applications – a review. *IEEE transactions on reliability*, 34(3):194–203.

[Leitner-Fischer and Leue, 2013] Leitner-Fischer, F. and Leue, S. (2013). Probabilistic fault tree synthesis using causality computation. *International Journal of Critical Computer-Based Systems 30*, 4(2):119–143.

[Lindberg, 2013] Lindberg, S. I. (2013). Mapping accountability: Core concept and subtypes. *International Review of Administrative Sciences*, 79(2):202–226.

[Liu and Wu, 2006] Liu, Y. and Wu, Z. (2006). Driver behavior modeling in ACT-R cognitive architecture. *Zhejiang Daxue Xuebao (Gongxue Ban)*, 40(10):1657–1662.

[Locke, 1690] Locke, J. (1690). Second treatise of government: An essay concerning the true original, extent and end of civil government. `https://www.earlymoderntexts.com/assets/pdfs/locke1689a.pdf`. [Online; acc. 2020-01-02].

[Lomax and Schumacker, 2004] Lomax, R. G. and Schumacker, R. E. (2004). *A beginner's guide to structural equation modeling*. Psychology Press.

[Lüdtke et al., 2010] Lüdtke, A., Osterloh, J.-P., Mioch, T., Rister, F., and Looije, R. (2010). Cognitive modelling of pilot errors and error recovery in flight management tasks. In Palanque, P., Vanderdonckt, J., and Winckler, M., editors, *Human Error, Safety and Systems Development*, pages 54–67, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Lüdtke et al., 2009] Lüdtke, A., Weber, L., Osterloh, J.-P., and Wortelen, B. (2009). Modeling pilot and driver behavior for human error simulation. In Duffy, V. G., editor, *Digital Human Modeling*, pages 403–412, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Miller, 2018] Miller, T. (2018). Explanation in artificial intelligence: Insights from the social sciences. *Artificial Intelligence*.

[Mittelstadt et al., 2019] Mittelstadt, B., Russell, C., and Wachter, S. (2019). Explaining explanations in ai. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT* '19, pages 279–288, New York, NY, USA. ACM.

[Moore, 2019] Moore, M. (2019). Causation in the law. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, winter 2019 edition.

[Nissenbaum, 1996] Nissenbaum, H. (1996). Accountability in a computerized society. *Science and engineering ethics*, 2(1):25–42.

[Noble, 2018] Noble, S. U. (2018). *Algorithms of oppression: How search engines reinforce racism*. NYU Press.

[O'Neil, 2016] O'Neil, C. (2016). *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books.

[Pearl, 1998] Pearl, J. (1998). On the definition of actual cause. Technical report, University of California, Los Angeles, CA 90024.

[Pearl, 2000] Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

[Pearl, 2009] Pearl, J. (2009). *Causality*. Cambridge University Press.

[Pearl, 2014] Pearl, J. (2014). Interpretation and identification of causal mediation. *Psychological methods*, 19(4):459.

[Pearl et al., 2016] Pearl, J., Glymour, M., and Jewell, N. P. (2016). *Causal inference in statistics: A primer*. John Wiley & Sons.

[Pearl and Mackenzie, 2018] Pearl, J. and Mackenzie, D. (2018). *The Book of Why*. New York, NY: Basic Books.

[PMI, 2017] PMI (2017). *A guide to the project management body of knowledge (PMBOK guide)*. Project Management Institute, Inc.

[Povey, 1999] Povey, D. (1999). Optimistic security: a new access control paradigm. In *Proceedings of the 1999 Workshop on New Security Paradigms*, pages 40–45.

[Salvucci, 2006] Salvucci, D. D. (2006). Modeling driver behavior in a cognitive architecture. *Human factors*, 48(2):362–380.

[Schellhorn et al., 2002] Schellhorn, G., Thums, A., Reif, W., et al. (2002). Formal fault tree semantics. In *Proceedings of The Sixth World Conference on Integrated Design & Process Technology, Pasadena, CA*, pages 1–8.

[Schneier, 1999] Schneier, B. (1999). Attack Trees - Modeling security threats. *Dr. Dobb's Journal*.

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

[Smith et al., 2005] Smith, M. L., Erwin, J., and Diaferio, S. (2005). Role & responsibility charting (RACI). In *Project Management Forum (PMForum)*, page 5.

[Stanton et al., 2013] Stanton, N., Salmon, P. M., and Rafferty, L. A. (2013). *Human factors methods: A practical guide for engineering and design*. Ashgate Publishing, Ltd.

[Talbert, 2019] Talbert, M. (2019). Moral Responsibility. In Zalta, E. N., editor, *The Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, Winter 2019 edition.

[Taş et al., 2016] Taş, Ö. Ş., Kuhnt, F., Zöllner, J. M., and Stiller, C. (2016). Functional system architectures towards fully automated driving. In *2016 IEEE Intelligent vehicles symposium (IV)*, pages 304–309. IEEE.

[Thoemmes, 2018] Thoemmes, F. (2018). The front-door criterion in linear, parametric models. [Online; acc. 2019-08-10].

[Tikka and Karvanen, 2018] Tikka, S. and Karvanen, J. (2018). Identifying causal effects with the R package causaleffect. *arXiv preprint arXiv:1806.07161*.

[VanderWeele and Shpitser, 2013] VanderWeele, T. J. and Shpitser, I. (2013). On the definition of a confounder. *The Annals of Statistics*, 41(1):196–220.

[Vesely et al., 1981] Vesely, W., Goldberg, F., Roberts, N., and Haasl, D. (1981). Fault tree handbook.

[Wachter et al., 2017] Wachter, S., Mittelstadt, B., and Russell, C. (2017). Counterfactual explanations without opening the black box: Automated decisions and the gpdr. *Harv. JL & Tech.*, 31:841.

[Wang, 2010] Wang, F.-Y. (2010). The emergence of intelligent enterprises: From cps to cpss. *IEEE Intelligent Systems*, 25(4):85–88.

[Weitzner et al., 2008] Weitzner, D. J., Abelson, H., Berners-Lee, T., Feigenbaum, J., Hendler, J., and Sussman, G. J. (2008). Information accountability. *Commun. ACM*, 51(6):82–87.

[Wellman and Henrion, 1993] Wellman, M. P. and Henrion, M. (1993). Explaining 'explaining away'. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(3):287–292.

[Wieringa, 2020] Wieringa, M. (2020). What to account for when accounting for algorithms: A systematic literature review on algorithmic accountability. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 1–18.

[Xing and Amari, 2008] Xing, L. and Amari, S. V. (2008). Fault tree analysis. In *Handbook of performability engineering*, pages 595–620. Springer.

[Zeng et al., 2020] Zeng, J., Yang, L. T., Lin, M., Ning, H., and Ma, J. (2020). A survey: Cyber-physical-social systems and their system-level design methodology. *Future Generation Computer Systems*, 105:1028–1042.

[Zielinska et al., 2020] Zielinska, O. A., Welk, A. K., Furlough, C., Stokes, T., and Geden, M. (2020). Human factors methods. `https://hfacmethods.wordpress.com/`. [Online; acc. 2020-06-07].