Design for Values and Conceptual Engineering

Abstract

Politicians and engineers are increasingly realizing that *values* are important in the development of *technological artefacts*. What is often overlooked is that different *conceptualizations* of these abstract values lead to different design-requirements. For example, designing social media platforms for deliberative democracy sets us up for technical work on completely different types of architectures and mechanisms than designing for so-called liquid or direct forms of democracy. Thinking about Democracy is not enough, we need to design for the proper conceptualization of these values. As we see it, we cannot responsibly engineer and innovate and shape technology in accordance with our moral values without engaging in systematic and continuous conceptual engineering: This is not only an academic, or theoretical issue, it is also not simply an issue for public policy or politics, or regulators, it has become a central problem for engineering and the world of technology.

In this paper, we present a framework for doing the necessary conceptual work in the context of requirement engineering. We draw on the literature on conceptual engineering to lay out a methodology to (1) assess different conceptions and (2) to develop new conceptions. Moreover, we integrate this methodology with extant approaches in the philosophy of technology which aim at designing technological artefacts ethically. In the final section we apply this integrated framework to freedom in the context of social media networks.

1. Introduction

Politicians and engineers are increasingly becoming aware that *values* are important in the development of *technological artefacts*. What is often overlooked, is that different *conceptualizations* of these abstract values lead to different design-requirements. Suppose for example that we set out to design a *democratic* social media platform. If we design this platform for *deliberative democracy, this* sets us up for technical work on completely different types of architectures and mechanisms than designing for so-called *liquid* or *direct forms of democracy*. If we think *voting* is central to democracy, we will design and develop efficient and secure voting technology, if we think *contestation* is central to democracy it is obviously the design of information provision and mechanisms to protest and contest that would be foregrounded. If *trust* is conceived in terms of epistemic reliability and confidence, different types of evidence need to be provided to trustors compared to when it is conceived in moral terms. As ever more powerful science and engineering are introducing ever new and miraculous possibilities into our life worlds, these conceptual issues will proliferate. Thinking about values such as Democracy or Trust is not enough, we need to design for the proper conceptualization of these values. That is, our value-based engineering work needs to be informed by our conceptual engineering.

We conceive of conceptual engineering as "the design, implementation, and evaluation of concepts and (…) includes or should include *de novo* conceptual engineering (designing a new concept) as well as conceptual re-engineering (fixing an old concept)" (Chalmers, 2020). We cannot responsibly engineer and innovate and shape technology in accordance with our moral values without engaging in systematic and continuous conceptual engineering: This is not only an academic, or theoretical issue, it is also not simply an issue for public policy or politics, or regulators, it has become a central problem for engineering and the world of technology.

In this paper we have two main goals. We aim to (1) convince the reader that conceptual engineering is crucial for value-based or value-sensitive requirement engineering and (2) present a framework for doing this kind of conceptual work in the context of requirement engineering. To reach these goals, we proceed as follows. We first discuss Design for Values, a methodology for doing value-based requirement engineering.

In section 3, we explain how we understand conceptual engineering. Subsequently, in section 4, we discuss designing for the value of Control, and show that conceptual engineering is needed to do this well. Next, we explain what we take to be a good approach to conceptual engineering (section 5), and show how this approach can be integrated into the Design for Values methodology. Finally, we discuss a case study where we apply the framework developed in this paper.

2. Requirement Engineering

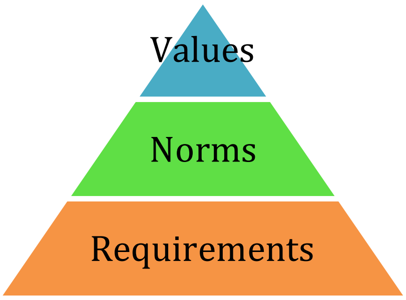


Figure 1

Designs and technical requirements should be assessed and criticized in light of human values. The IEEE, the European Commission, the WHO, UNESCO, and many others have come to appreciate that design thinking needs to play a bigger role in our ethical thinking about the problems of our present age. The idea that values can be designed for and inform our designs has been elaborated by Batya Friedman and others from the early 1990s onwards in what they refer to as *Value Sensitive Design* (VSD). VSD originated in computer science and has a strong focus on value elicitation and inclusion of stakeholders in design processes. Design for Values (DfV) was heavily influenced by VSD, but originated in the field of *ethics of technology* and has a strong pragmatic focus on making ethics work in a world of technology. It aims to help designers to put social and moral values at the heart of the design of new technologies (van de Poel, 2020; van den Hoven et al., 2015). Central to DfV is the idea that moral values can be systematically specified – and that this can be captured for practical purposes in a specification schema, a decomposition of moral considerations, a values hierarchy, a hierarchical structure of values, norms and design requirements (see figure 1). Such a perspicuous representation of values and implied requirements in a schema or hierarchy allows designers to translate abstract **values** or core moral concepts via context-sensitive **norms** into design **requirements** by means of specifications with decreasing levels of abstraction (van de Poel, 2013, 2020).

However, as we have noted above, it is not always obvious which concept invoked in the decomposition of requirements is the most appropriate in the relevant context of use. And although scholars working in the field of Design for Values have shown awareness of the conceptual challenges associated with the introduction of disruptive technologies, we think there are two aspects that are currently missing. Firstly, many of the conceptual analyses on offer focus on the different available conceptions of a specific concept, seemingly overlooking the potential need to engineer new concepts *de novo*. Secondly, there is at the moment no generic methodology for deciding which concept is most appropriate for a given technology. To remedy this, we suggest extending work in the field of Design for Values with a methodological framework for conceptual engineering. However, we'll start with some theoretical work to get a better grip on what we are doing when we are engineering concepts.[1]

3. Conceptual engineering: concepts, functions and inferential roles

Conceptual engineering has always been an important aspect of philosophy (although not under that name). As a systematic methodology, however, it has recently received much interest. As such, the field is very diverse and there is disagreement about some of the core issues associated with the method. For example, some argue that the target of conceptual engineering should be concepts (e.g., (Chalmers, 2020)), while others focus on expressions (e.g. (Cappelen, 2018; Thomasson, 2022)). A second, related, question is what we should engineer for. Cappelen (2018), for example, argues that we should focus on intensions and extensions, while others argue that we should engineer inferential roles (e.g. (Jorem & Löhr, 2022; Löhr, 2022; Veluwenkamp et al., 2022)), use-patterns (e.g. (Jorem, 2021)) or commitment and entitlement structures (e.g. (Löhr, 2021)). In this paper we will not try to argue for any of these positions. We will propose a set of assumptions that we take to be independently plausible, and show how conceptual engineering understood in this way can be employed by the philosopher of technology. It is our view, however, that most of what we argue for also holds for any of the alternative assumptions.

[1] In Delft we have been working for decades on translating ethical values into design requirements. At the end of 2020. though, we realized that because our current conceptual framework might be inadequate, we need to engage in conceptual engineering if we want to do requirement engineering correctly. With a group of philosophers of technology, we started mapping out the terrain; arguing that many debates in the philosophy of technology should be seen as conceptual engineering problems. This resulted in a popular science piece in February 2021 (Santoni De Sio et al., 2021) and a recent academic paper (Veluwenkamp et al., 2022). From September 2021 on, we started talking about the ideas presented in the current paper in workshops and conferences.

In this paper we will focus on engineering the content of expressions. We assume that in ordinary language terms are open-textured (Shapiro & Roberts, 2019) and the content of our expressions is to some extent indeterminate, such that there are many different ways to make that content precise *without a change of topic*. Take for example the expression “justice”. There are several ways of making this expression more precise. We can, for example, assign a content to “justice”, such that something is just if and only if Rawls’s two fairness principles are satisfied (1999). When we assign this content to “justice”, then we have plausibly not changed the topic. Moreover, there are different other ways of assigning contents to “justice” that do not change the topic. This does not, however, hold for all content assignments. If we assign a content to “justice”, such that something is just if and only if it is hot, then this constitutes a change of topic.

We will call any complete way of making the content of an expression precise without changing that expression’s topic *T* a *conception* of *T*. So, for example, the topic of “freedom” is freedom and Skinner’s (2008) account of neo-republicanism is one conception of freedom. Given this, we have to say something about when two ways of making the content of an expression more precise have the same topic. Roughly, we assume that two determinate contents are conceptions of the same topic if they fulfill the same function (see also (Prinzing, 2018; Sundell, 2020; Thomasson, 2020)). When Rawls, for example, proposed a new conception of justice, he was interested in conceptions that fulfilled a specific function: i.e., providing “principles for assigning rights and duties” (1999, p. 5). However, in different contexts we might have different functions, so what counts as a change of topic is context-dependent (see also (Eklund, 2021)).

We also need to say something about the way we understand expressions. We take expressions to have their contents in virtue of certain patterns of use, e.g., their conceptual or inferential role. On this approach, when we engage in conceptual engineering for engineering contents, we are aiming to engineer the expression such that it is characterized by a different conceptual or inferential role (i.e., the role it ought to play).[2] This entails that a complete way of making the indeterminate content of an

[2] Inferentialism about conceptual engineering has been defended in (Jorem & Löhr, 2022). It is important, however, to note that by claiming that inferential roles are a suitable place for conceptual engineering, one is not committed to metasemantic inferentialism, i.e., the position that inferential roles are explanatorily basic. The metasemantic referentialist (who holds the purported reference is explanatorily basic), for example, does not deny that expressions have inferential roles (as we have explained in (Veluwenkamp et al., 2022)). Moreover, the position is also compatible with both atomist and holist ways of concept individuation (as (Löhr, 2022) has argued).

expression precise should give us a determinate conceptual or inferential role for that expression. Two or more determinate conceptual or inferential roles provide us with *conceptions* of the same topic just in case, as suggested above, they fulfill the same function. This allows us to say what it is we think conceptual engineering should engineer: conceptual engineering should determine what determinate conception, understood in terms of topic-preserving conceptual or inferential role, ought to be associated with an expression.

4. CE and AWS

In what way is conceptual engineering relevant when designing socio-technical systems? Let us consider a simplified example concerning autonomous weapon systems (AWS). As we saw above, a Design for Values approach invites and allows designers to translate (a specification of) moral values into the much needed design requirements. One of the core values that have been related to AWS is the ability to hold someone responsible in cases of untoward outcomes. Opponents of AWS have argued that AWS which do not afford responsibility are in violation of international law. Sparrow (2007), for example, assumes in his argument that direct control over the outcome of an AWS is a necessary condition for responsibility. Moreover, he takes direct control over a system to entail some kind of ability to causally influence that system. A partial value hierarchy that can be derived from the sketched argument is the one depicted in Figure 2.

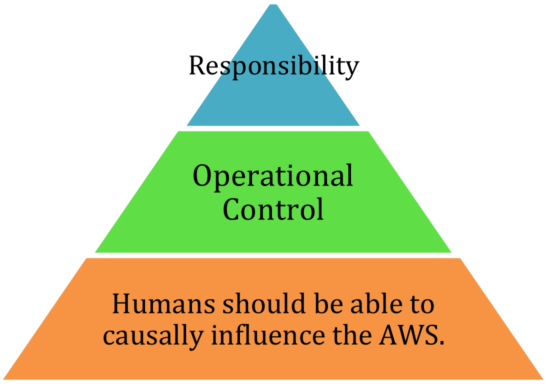


Figure 2

The conception of control that is used by Sparrow in this context is that of operational control. We can specify this conception as follows:

Operational control

Agent A is responsible for outcome O → A is in control of O

Agent A is in control of O → A is (or has been) able to causally influence O

Let X → Y mean that if someone utters that X, then she ought to accept that Y.[3] So, if someone utters that Yantha is responsible for the drone bombing the innocent people, then she ought to accept the utterance that Yantha is in control of the drone bombing the innocent people. When we look at the value hierarchy sketched above, it is clear why Sparrow concluded that designing for responsibility is incompatible with autonomous weapon systems: the autonomous nature of AWS precludes the possibility of direct human causal interaction.

However, some critics of Sparrow's argument have resisted his argument by introducing a different conception of control in the context of autonomous weapon systems. Johannes Himmelreich (2019), for example, introduced what he calls "robust tracking control" as a rival conception and argued that it is more appropriate than causal control in the context of autonomous systems.

Robust tracking control

Agent A is responsible for outcome O → A is in control of O

Agent A is in control of O → The outcome O tracks the decisions of A.

For Himmelreich, we say the outcome (x) tracks the decisions of a human agent if there is an order the human (a) can give for which it is the case that "if a were to give this order, then x would occur (in all relevantly similar situations), and [...] if a were not to give this order, then x would not occur (in all relevantly similar situations)" (2019, p. 736). When

[3] Inferential relation can be spelled out in terms of the utterances one is disposed to accept, or in terms of utterances one is committed to. Another point of contention is what the reference point for the inferential roles is. Individualists hold that it is the speaker's commitments or dispositions that determine meaning. Alternatively, one can hold that it is society that determines which commitments or dispositions are correct (see also (Sinclair, 2017)). For our purposes in this paper these distinctions do not matter.

we use this conception for our value hierarchy for AWS, we get something like the following schema:

Value hierarchy for weapon system:

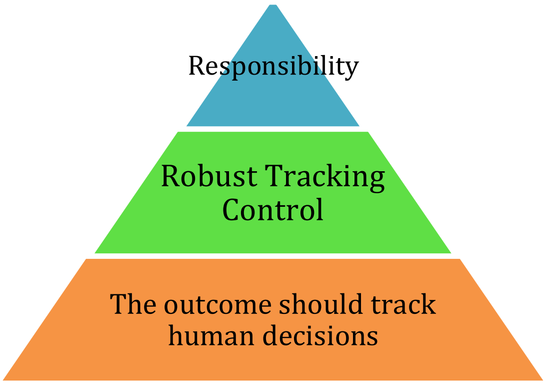


Figure 3

We can use this schema or value hierarchy to argue that AWS that can be shown to successfully implement the bottomline requirement are compatible with justified responsibility ascriptions. The details of Himmelreich's conception of robust tracking control are not relevant for this paper, but what is relevant is that he does not accept Sparrow's conception of control and proposes a different one. Moreover, several other authors on this topic have also proposed rival conceptions of control (Cavalcante Siebert et al., 2022; Hindriks & Veluwenkamp, forthcoming; Mecacci & Santoni de Sio, 2020; Santoni de Sio & Van den Hoven, 2018; Veluwenkamp, 2022). All these different conceptions of control entail different design requirements and point in the direction of different technical features.

What this shows is that design requirements depend crucially on conceptual choices. Moreover, these different design requirements can be incompatible: Sparrow's conception of control arguably entails that no autonomous AWS allows for appropriate responsibility attributions, while Himmelreich's conception tells us that it can be appropriate to hold someone responsible if an AWS causes harm. This means that we have to decide which conception we ought to use in a specific context.

This raises some questions; how do we decide which conception is best in a specific context? And, more fundamentally, what exactly do we mean with "best" here? In the next section we will answer these questions.

5. How to conduct conceptual engineering properly?

There currently is no consensus on the methodology of conceptual engineering. Matti Eklund, for example, calls the question what the proper methodology for conceptual engineering is one of the “big questions [that] remain entirely unresolved” . However, there is some consensus. Almost all participants in the debate (e.g., (Eklund, 2014, 2015; Isaac, 2021; Prinzing, 2018; Simion, 2018; Simion & Kelp, 2020)) hold that concepts serve different purposes; and serving their purposes is the function of the concepts. Many also hold that the function of a concept gives us all-things-considered reasons to opt for a specific concept (e.g., (Queloz, forthcoming; Simion, 2018; Simion & Kelp, 2020; Thomasson, 2020, 2022)).

One way of developing this idea is the so-called pragmatic approach (Thomasson 2020). If we want to decide which conception of a concept we should employ, we should first determine what function, or purpose, this concept should perform in the context that we are discussing.[4] Once we have determined what the function is, then the best conception is that conception that fulfills this function best. Sometimes the function that a concept ought to perform is the function that it has at the moment. Suppose for example that one had in one’s society a conception of marriage that precludes same-sex couples. Moreover, suppose, as is plausible, that the function of marriage is to afford a special legal and social status to a range of close relationships (Cappelen, 2018). In these circumstances we can come to see a conception of marriage which includes same-sex couples as better than the old one.

However, we do not have to uncritically adopt the current function of a concept as the function the concept ought to have. This is what makes this approach distinctly normative: there might be different normative reasons that determine what function a concept ought to play. There might, for example, be moral, epistemic or prudential reasons for preferring one function over another. In The Netherlands there has been a recent debate about the history of slave trading by the Netherlands in the 17th century and there is a broad appreciation in Dutch society that a better way of speaking about the victims of slave trade as the ‘enslaved’ (“tot slaaf gemaakten”) instead of ‘slaves’

[4] There currently is a debate on how to understand “function” in the context of conceptual engineering (Jorem, 2022; Queloz, forthcoming; Riggs, 2021). In this paper we will understand “function” as, roughly, the things that a concept allows us to do.

("slaven"). These fine distinctions can be highly relevant because, as Evitai Zerubavel and Ned Block have argued already some time ago, concepts co-determine what we see and perceive as salient and relevant, what stands out against a background, and in social contexts.[5] The empirical adequacy *and* the normative implications are therefore criteria for replacing one construal with another one. To give another example, Sally Haslanger argued that the function of our current conception of race and gender terms is to facilitate and legitimize discriminatory practices. We have therefore, Haslanger argues, moral reasons to use a concept with a different function. She suggests adopting conceptions of race and gender that serve as "effective tools in the fight against injustice" (2012, p. 226). Subsequently, Haslanger proposes conceptions of race and gender that she takes to be able to fulfill this function.

We see these strategies as suitable for engaging in conceptual engineering (Thomasson, 2020). We can distinguish three phases in the strategy (see also Figure 3):

1) a discovery phase: in this phase we determine the actual function of our current concept. The actual inferential role of the conception we are employing can be used to determine this function.
2) a justificatory phase: in this phase we determine whether the function our concept has can be improved upon.
3) a constructive phase: in this phase we determine which inferential role would be able to fulfill the function we want our concept to have. Even if we do not want to

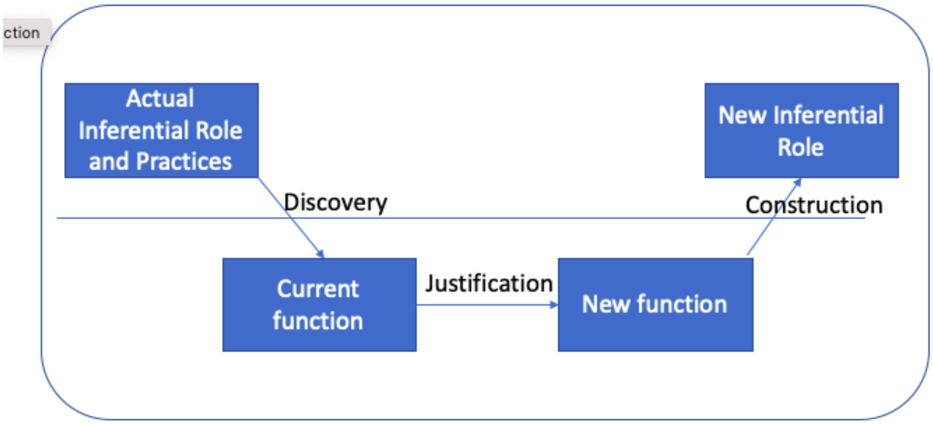


Figure 4

[5] Work in linguistics and sociology (e.g., by the sociologist Zerubavel, the linguists Lakoff and Sapir), but also research in cognitive psychology and the philosophy of mind (e.g., Ned Block 's work on access consciousness) have independently shown that having access to a concept or an effective prototype activation pattern may determine awareness perception and thought processes.

> change the function of our concept, it could be the case that a different inferential role fulfills the function better than our actual one in the context that we are interested in.

Having briefly sketched the pragmatist approach to conceptual engineering, we will show in the next section how this approach can be integrated with the Design for Values methodology.

Design for Values and Conceptual Engineering

The Design for Values methodology recognizes the need for a conceptual stage in which the different possible conceptualizations of the values or norms that are under discussion are investigated. In the past, we have contributed to this as well ((Van den Hoven, 2017; Vermaas et al., 2010)). However, we think that Design for Values - and the same applies to the Value Sensitive Design methodology proposed by Friedman and others - would benefit from a more structured and integrated approach to conceptual engineering. Moreover, we think that the task of the philosopher of technology is not limited to enumerating the different possible conceptualizations that are on offer in the literature. Sometimes novel contexts ask for novel conceptualizations, and we need to decide which conception we want to embed in our technologies. This means that it might be the job of the philosopher of technology to engage in conceptual design as well. Let us assume that one is applying a Design for Values approach, and during the process of specifying and decomposing abstract values in terms of specific design requirements one recognizes the possibility that one of the concepts is not the most appropriate in the given context. This means that it is unclear how to translate the values into design requirements and we have to engage in conceptual engineering.

Discovery phase

Above we have established that we should design for that conception of a concept that fulfills the function of that concept best. We, therefore, first have to determine what the current function of a concept is. Just as in other branches of engineering, this is usually done through a process called reverse engineering. The idea behind reverse engineering is that we discover the function of an object by disassembly and observation of its behavior in different contexts. This is not unlike what archaeologists do when they find puzzling objects or artefacts. They may to that end have to provide an account of a whole society and way of life, rituals, meanings and values in order to ascribe functions to an unfamiliar artefact.

There are several methods available for reverse engineering our current concepts. These methods can roughly be divided into two categories: historical and a-historical reverse engineering. In this section we will sketch versions of both variants and then briefly give some pointers which help determine which method is most suitable in a given context. A-historical variants of reverse conceptual engineering try to determine the function of a concept by looking at our current practices. Miranda Fricker, for example, defends what she calls a Paradigm-Based Explanation (2016). This method identifies a paradigmatic form of the concept that is being analyzed. The paradigmatic form is supposed to be explanatorily basic for the target concept. The goal is to use the paradigmatic form of a concept and determine the function of this form. The hypothesis that the chosen form is indeed an explanatorily basic form of the target concept can be tested by investigating whether and to what extent other forms of the concept can be seen as derivative of the paradigmatic form.

As an example, we can look at the concept that Fricker uses herself. Fricker is interested in the function of the concept Blame. The paradigm case she identifies is communicative blame: “I wrong you, and in response you let me know with feeling that I am at fault for it” (2016, p. 167). Communicative blame does not have to be verbal, as there are other ways of communication. However, what is characteristic of communicative blame is that someone accuses someone else of fault. Fricker subsequently analyses typical speech acts which express communicative blame to uncover its function. The function which Fricker discovers, is that it “aims at bringing the wrongdoer to see things in part from the wronged party's point of view, thereby enlarging her perception and altering her reasons” (2016, p. 173). To finalize her analysis, Fricker shows that forms of blame, such as first-person blame (self-blame) and third-personal cases of blame, can be derived from the paradigm case. Because this approach does not look at historical facts to determine the function of a concept, it is a version of a-historical reverse engineering.

Historical versions of reverse conceptual engineering are often known as variants of genealogy (e.g., (Dutilh Novaes, 2015; Queloz, 2021)). Philosophers engaging in genealogies see the (sometimes fictionalized) historical development of a concept as an essential ingredient for its analysis. Nietzsche argued, for example, that a proper understanding of moral terms is only possible when we closely investigate under which conditions we came to use moral concepts. And, although this method is usually associated with continental philosophers such as Nietzsche and Foucault, it is recently also adopted by philosophers working in the analytic tradition, such as Ian Hacking, Edward Craig, Bernard Williams, Catarina Dutilh Novaes and Simon Blackburn.[6]

Both the historical and the ahistorical approach to reverse conceptual engineering have their advantages. The upshot of the ahistorical approach is that it is often simpler and easier to apply. According to this approach, we don't have to look at (sometimes fictional) historical narratives in order to determine the function of a concept. For this reason Fricker takes her method to be superior to a historical method; she sees pragmatic genealogy 'as a more straightforward and transparent way of achieving the very same explanatory pay-off' (Fricker, 2016, p. 245).

However, the historical method can be useful in some circumstances. If there are no paradigmatic examples available, for example, the ahistorical method sketched by Fricker fails. It is not obvious that there is always a paradigmatic form that is able to explain the derivative instances of the practice, especially in those cases in which the conceptual practice is internally diverse and/or is held together by family resemblance. Moreover, Fricker's method explains the point of a practice given some generic or local

[6] Reverse conceptual engineering is supposed to be normatively neutral. This can be contrasted with the use in which genealogies are usually being deployed. Because genealogies are often presented as subversive or vindicatory. A subversive genealogy undermines a practice by showing that it has a disreputable history, while vindictive genealogies use the historical development to justify a practice. Nietzsche's moral genealogy is, for example, one of the paradigmatic examples of a subversive genealogy. Nietzsche argued against our conception of morality, exactly by exposing its disreputable origins: "the value of [moral] values should itself, for once, be examined – and so we need to know about the conditions and circumstances under which the values" . However, a worry with this kind of argument is that it seems to suffer from the genetic fallacy. This fallacy consists in a confusion of the origins of a belief with its justification (Klement, 2002). Whether Nietzsche's argument actually suffers from the fallacy is a matter of debate (Loeb, 2008), but it is important to see that this problem does not occur if we restrict ourselves in the discovery phase to a normatively neutral description of the historical development of a specific concept.

needs of a discursive community. However, if the needs that explain the point of a practice are not current, but historical needs, then the paradigm-based approach doesn't suffice. To explain such a practice, it is necessary to look at the historical development of a conceptual practice (see also (Queloz, 2020, 2021)).

Justificatory phase

Once the current function of a concept is identified, it is possible to reflect critically on that function in the context that we are interested in. One of the important questions in the context of Design for Values, is whether the concept can play the role in the value hierarchy that it is supposed to play. If we are evaluating a concept that figures in one of the norms, then we should verify that the concept with this function is a proper translation of the value. So, if we deem control to be important for responsibility, then the function of control should make clear that an agent who has control is an appropriate candidate for responsibility attributions.

If the concept we are reflecting on is itself one of the fundamental values, then we can use the function of that concept to assess whether this is a function we want to design for. Haslanger's discussion of race and gender concepts mentioned above is an example of this kind of criticism. If it can indeed be shown that these concepts function in practices that facilitate and legitimize discrimination, then we have moral reasons to use a concept with a different function. Haslanger's example is a case where we critically assess a concept and conclude that we should keep the concept but change its function.

In some circumstances, philosophers have concluded that it would be better to banish the concept altogether. Cora Diamond (2019), for example, discusses the possibility of 'losing our concepts'. Certain concepts have functions that are worth having, or functions that we want to promote (Teichmann, 2021). Other concepts, however, have features that "we could well do without, others may be disastrous, as elements in our lives" (Diamond, 2019, p. 212).

As an example, Diamond discusses in an earlier paper Elizabeth Anscombe's critique of our modern moral discourse. According to Anscombe, the original function of the moral 'ought' is to track the divine law. However, this concept is currently used without the necessary theological background. So the idea is that we talk and think as if "ought" implies some kind of obligation which has the force of a law, while the idea of being bound by something like the moral law does not make sense anymore. Anscombe concludes for this reason that our moral concepts "ought to be jettisoned if this is psychologically possible" (1958, p. 1).

Constructive phase

If we decide that it is best to discard a concept, then the constructive phase is obviously not necessary. However, in other instances we can use the function a concept ought to serve in order to construct a conception of that concept which performs this function best in the new context. Note that the fact that a conception performs a function well in a particular, paradigmatic context does not mean that it also performs this function well in a new context. In the constructive phase we basically have three options:

1) we conserve one of our current conceptions (*conservation*)
2) we create a new conception *de novo* (*creation*)
3) we convert an existing conception taken from a different context to apply to a new context (*conversion*)

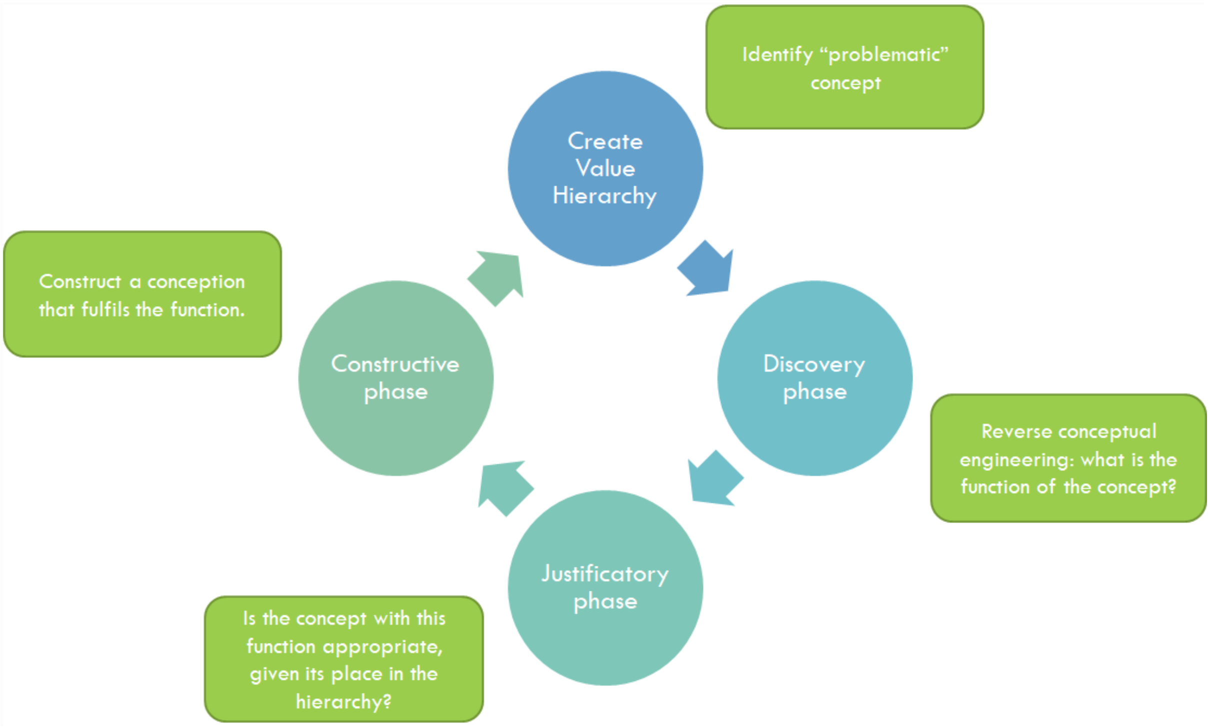
A good place to start when constructing a conception for a given function is to conserve one of the current dominant conception(s) of that concept. If there are multiple conceptions, then it is good practice to list the different inferential roles of these conceptions. These inferential roles can be used to assess whether the conception can fulfill the target function in the target context.

Let us look at an example. Some have argued that the liar paradox shows that our conception of *Truth* is incoherent. This alleged incoherence is taken by some philosophers to be a reason to discard this conception and construct a new one. Kevin Scharp (2013), for example, argues that we should replace *Truth* in certain theoretical contexts, because the inconsistency of *Truth* inhibits its ability to fulfill its function in those contexts. Scharp proposes different conceptions of *Truth* for those contexts. He thinks, however, that in many everyday contexts the inconsistency of *Truth* is unproblematic. He maintains, for this reason, that we should keep using our current conception of *Truth* in those contexts. What this example also makes clear is that it is important to take the context into account in which the conception will be used. Some conceptions can fulfill a function better in novel contexts than other conceptions. This also makes conceptual engineering important when discussing disruptive technologies. The nature of these technologies is such that by definition they create new contexts, new ways of living and experiencing, in which our old conceptions may not function as well as they used to.

If the current conception(s) of a concept turn(s) out to be unable to perform the target function, then *conservation* is not an option. An alternative strategy then is the *creation* of a new conception. Since this is a creative process, it will be difficult to come up with a step-by-step instruction for this purpose. However, some heuristics can be provided. When designing a conception for a target function, it is often also insightful to consider what it means *not* to fulfill the target function (see also (Goertz, 2006; Swedberg, 2017)). Looking at different ways in which we lack control over autonomous systems helps us see what it means to have control over those systems. We can also remove or add some inferential roles of existing conceptions. Doing so increases or decreases the extension of a conception.

We don't always have to create the conception *de novo* (Chalmers, 2020). Sometimes we can convert a conception that has been introduced in another context and use it in the target context (*conversion*). It has, for example, been argued that the conception of freedom as non-interference is conceptually and morally problematic in the context of influential social media companies (Maas, 2022b). Jonne Maas argues that, for these reasons, we should replace this conception with a different one. The conception that she takes to be most appropriate is the neo-republican conception of freedom. This conception was constructed for different contexts. It can, however, also explain much of the intuitions we have about freedom when it is threatened by the influence of Big Tech (see also (Veluwenkamp et al., 2022)).

We want to conclude this section with a note of warning. It is important to recognize which of the three options we are employing in the constructive phase. In the introduction, we pointed out the dangers of combining existing terms to denote a new conception (i.e., colligation). Because the terms used are familiar, it might seem that we are merely using a well-established conception in a new context. However, colligation is often a de novo approach: we introduce a conception (such as e-health or digital trust, blockchain organisation, cyber community) that does not come with a preestablished meaning. This implies that we have to construct the inferential role of the new conception. The process is of course based on the inferential roles of the individual conceptions that make up the new conception, but it is not evident what way of constructing the new conception fulfills the target conception best. So, what seems to be a form of *conversion* is in fact *creation*.



Identify "problematic" concept
Create Value Hierarchy
Construct a conception that fulfils the function.
Constructive phase
Discovery phase
Reverse conceptual engineering: what is the function of the concept?
Justificatory phase
Is the concept with this function appropriate, given its place in the hierarchy?

Figure 5

Use case: Social Media and Freedom

Above we have presented a framework for integrating the Design for Values methodology with a pragmatist approach to conceptual engineering (see Figure 5 for a graphical representation). In the current section we describe how this framework can be applied when designing new technologies.

As an example, we can imagine that we are a designer tasked with the development of a new social media network. Let us say that one works for a non-profit organization that is trying to develop an alternative to Facebook or Twitter. One of the values that is identified during a stakeholder analysis is that the social media network should not limit the freedom of the users. That is, one of the values identified for the design of the social robot is *freedom*. How can the proposed framework help us design for freedom?

Create value hierarchy

The first that we ought to take is to construct a value hierarchy. As freedom is a value, it gets to be put on top of the hierarchy. When we try to translate this value into a context-sensitive norm, we find that there are different conceptions of freedom. Given that there are at least two conceptions of freedom, freedom as the absence of external obstacles (negative freedom) and freedom as self-determination (positive freedom), this constitutes a challenge. For it seems that designing for the absence of external obstacles requires very different types of architectures than designing for self-determination.

Negative Freedom

Agent A is free → there are no external obstacles for A

Positive Freedom

Agent A is free → A is able to control their own destiny in their own interest

We are now faced with the question which conception to design for. This is important, because different conceptions lead to different design requirements. If we think *the avoidance of obstacles* is central to freedom, we will design and develop the network in such a way that it interferes as little as possible in the life of its users, if we think *self-determination* is central to freedom, then the social media network should be developed such that it supports users to achieve those goals that are important to them.

Discovery phase

In the discovery phase we are tasked with finding out what the function of *Freedom* is. So what is the function of indicating that an agent is free? It is beyond the scope of this paper to employ one of the strategies for determining the function of a concept. So, for now, we will rest content with having identified several strategies (the historical and ahistorical strategies as defended in (Dutilh Novaes, 2015), (Queloz, 2020) and (Fricker, 2016)), and stating what we take to be a plausible, albeit simplistic, function of *freedom*. Let us therefore assume that the function of indicating that someone is free if that agent stands in a relation to the powerful that is desirable.

Justificatory phase

In the justificatory phase we aim to justify, or modify the function that was identified in the previous stage. The concept we are reflecting on is itself one of the fundamental values, that is, it is at the top of the value hierarchy. This means that we should set out to determine whether, in the context of social media networks, it is valuable to design in such a way that users stand in a desirable relation to the powerful. Again, this is not the place to argue for this, but it seems plausible that this is exactly what the stakeholders meant when they urged to design for freedom.

Constructive phase

In the constructive phase we first determine whether one of the current dominant conceptions fulfills the function of Freedom well in the context of social media companies. One of the problems with social media companies that is often mentioned, is that these companies are extremely powerful. That is, they are in a position to interfere massively in our daily lives and have the potential to undermine the control we currently have to pursue our own self-interest. However, arguably, they do not do that to a large extent at the moment. Filter bubbles, biases filtering algorithms, etc. notwithstanding, companies such as Facebook, arguably, refrain from interfering and diminishing our ability for self-determination. At least not as much as they could. So if we would design for either the positive or the negative conception of freedom, this would not require us to change this power imbalance between social media companies and user.

The problem, however, is that the power imbalance is taken to be problematic. Even if companies choose not to undermine our positive or negative freedom, the fact that they could make it the case that the relation users have with these companies is undesirable. For these reasons it seems that the dominant conceptions of freedom are unable to fulfill the function of Freedom well.

To remedy this, we can either create a conception *de novo* (creation), or convert an existing conception from a different context (*conversion*). In this case we would like to propose to opt for conversion and adopt a conception that has been proposed in political philosophy: freedom as non-domination. The idea behind this notion of freedom is that it is a kind of *status*. Freedom in this sense is to have certain rights and privileges. The paradigmatic example of someone who is unfree is the slave. Even if the slave is not interfered with by their master, or if the master allows the slave to pursue their own self-interest, this condition depends completely on the master. The fact that the master is in a position to take away these privileges at will is, according to this conception of freedom, exactly what makes the slave unfree (Maas, 2022a, 2022b). We can therefore, tentatively, conclude that in the context of social media companies, freedom as non-domination fulfills the function of freedom better than the dominant ones.

Create value hierarchy

We have gone full circle and can continue creating the value hierarchy. It seems that the centralized nature of many of the current powerful social media networks is responsible for them not embedding the value freedom. So one way of translating freedom is by requiring that the network is set up in a decentralized way. This norm can then further be translated into the requirement that blockchain or some other technique for decentralization is employed. This gives us the value hierarchy that is depicted in Figure 6.

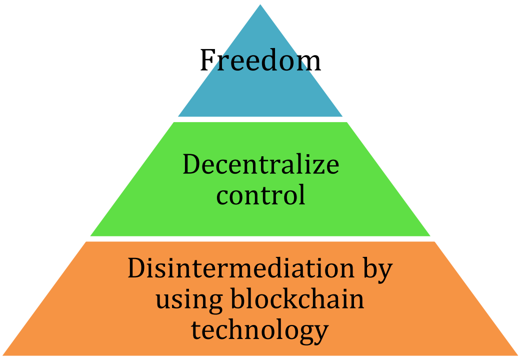


Figure 6

Conclusion

In this paper we have (1) argued that conceptual engineering is crucial for value-based and value sensitive requirement engineering and (2) presented a framework for doing this kind of conceptual work in the context of requirement engineering. We have argued for (1), by first explaining how the Design for Values methodology would be employed when designing for Control. We have shown that different conceptualizations of Control lead to different design requirements, and that we therefore need to decide which conception is best in the context under discussion.

To develop a framework for doing the conceptual work necessary, we have first explained how we understand conceptual engineering. Subsequently, we have developed the pragmatist approach to doing conceptual engineering. This methodology focuses on the function of a concept, and understands the best conception of that concept as that conception that fulfills this function best. To make this pragmatist approach suitable for philosophers of technology, we have suggested how it may be integrated in the Design for Values methodology. Finally, we have examined freedom in the context of social media networks, which both illustrates well the need for conceptual work, and allows us to show how our framework can be employed in practice.

References

Anscombe, G. E. M. (1958). Modern moral philosophy1. *Philosophy*, *33*(124), 1–19.

Cappelen, H. (2018). *Fixing language: An essay on conceptual engineering*. Oxford University Press.

Cavalcante Siebert, L., Lupetti, M. L., Aizenberg, E., Beckers, N., Zgonnikov, A., Veluwenkamp, H., Abbink, D., Giaccardi, E., Houben, G.-J., & Jonker, C. M. (2022). Meaningful human control: Actionable properties for AI system development. *AI and Ethics*, 1–15.

Chalmers, D. J. (2020). What is conceptual engineering and what should it be? *Inquiry*, 1–18.

Diamond, C. (2019). *Reading Wittgenstein with Anscombe, going on to ethics*. Harvard University Press.

Dutilh Novaes, C. (2015). Conceptual genealogy for analytic philosophy. In *Beyond the analytic-continental divide* (pp. 83–116). Routledge.

Eklund, M. (2014). Replacing Truth? In B. Sherman & A. Burgess (Eds.), *Metasemantics*.

Eklund, M. (2015). Intuitions, conceptual engineering, and conceptual fixed points. In *The Palgrave handbook of philosophical methods* (pp. 363–385). Springer.

Eklund, M. (2021). Conceptual Engineering in Philosophy. In J. Khoo & R. Sterken (Eds.), *The Routledge Handbook of Social and Political Philosophy of Language*.

Fricker, M. (2016). What's the Point of Blame? A Paradigm Based Explanation. *Noûs*, *50*(1), 165–183.

Goertz, G. (2006). *Social science concepts: A user's guide*. Princeton University Press.

Haslanger, S. (2012). *Resisting Reality: Social Construction and Social Critique*. Oxford University Press. https://doi.org/10.1093/acprof:oso/9780199892631.001.0001

Himmelreich, J. (2019). Responsibility for Killer Robots. *Ethical Theory and Moral Practice*, *22*(3), 731–747.

Hindriks, F., & Veluwenkamp, H. (forthcoming). The Risks of Autonomous Machines: From Responsibility Gaps to Control Gaps. *Synthese*.

Isaac, M. G. (2021). Post-Truth Conceptual Engineering. *Inquiry: An Interdisciplinary Journal of Philosophy*, 1–16.

Jorem, S. (2021). Conceptual engineering and the implementation problem. *Inquiry: An Interdisciplinary Journal of Philosophy*, *64*(1–2), 186–211.

Jorem, S. (2022). The good, the bad and the insignificant—Assessing concept functions for conceptual engineering. *Synthese*, *200*(2), 106. https://doi.org/10.1007/s11229-022-03548-7

Jorem, S., & Löhr, G. (2022). Inferentialist conceptual engineering. *Inquiry*, 1–22. https://doi.org/10.1080/0020174X.2022.2062045

Klement, K. C. (2002). When is genetic reasoning not fallacious? *Argumentation*, *16*(4), 383–400.

Loeb, P. S. (2008). Suicide, meaning, and redemption. *Nietzsche on Time and History*, 163–190.

Löhr, G. (2021). Commitment engineering: Conceptual engineering without representations. *Synthese*, *199*(5), 13035–13052. https://doi.org/10.1007/s11229-021-03365-4

Löhr, G. (2022). Do socially disruptive technologies really change our concepts or just our conceptions? *Technology in Society*, 102160. https://doi.org/10.1016/j.techsoc.2022.102160

Maas, J. (2022a). Machine learning and power relations. *AI & SOCIETY*. https://doi.org/10.1007/s00146-022-01400-7

Maas, J. (2022b). A Neo-Republican Critique of AI ethics. *Journal of Responsible Technology*, *9*, 100022. https://doi.org/10.1016/j.jrt.2021.100022

Mecacci, G., & Santoni de Sio, F. (2020). Meaningful human control as reason-responsiveness: The case of dual-mode vehicles. *Ethics and Information Technology*, *22*(2), 103–115.

Prinzing, M. (2018). The Revisionist's Rubric: Conceptual Engineering and the Discontinuity Objection. *Inquiry: An Interdisciplinary Journal of Philosophy*, *61*(8), 854–880.

Queloz, M. (2020). From Paradigm-Based Explanation to Pragmatic Genealogy. *Mind*, *129*(515), 683–714.

Queloz, M. (2021). *The practical origins of ideas: Genealogy as conceptual reverse-engineering*. Oxford University Press.

Queloz, M. (forthcoming). Function-Based Conceptual Engineering and the Authority Problem. *Mind*.

Rawls, J. (1999). *A theory of justice*. Belknap Press of Harvard University Press. http://books.google.nl/books?id=ExolAQAAIAAJ

Riggs, J. (2021). Deflating the functional turn in conceptual engineering. *Synthese*, *199*(3), 11555–11586. https://doi.org/10.1007/s11229-021-03302-5

Santoni De Sio, F., Capasso, M., Clancy, R., Dennis, M., Durán, J. M., Ishmaev, G., Kudina, O., Maas, J., Marin, L., Pozzi, G., Sand, M., Hoven, J. van den, & Veluwenkamp, H. (2021). Tech Philosophers Explain The Bigger Issues With Digital Platforms, And Some Ways Forward. *3 Quarks Daily*. https://3quarksdaily.com/3quarksdaily/2021/02/tech-philosophers-explain-the-bigger-issues-with-digital-platforms-and-some-ways-forward.html

Santoni de Sio, F., & Van den Hoven, J. (2018). Meaningful human control over autonomous systems: A philosophical account. *Frontiers in Robotics and AI*, 15.

Scharp, K. (2013). *Replacing Truth*. Oxford University Press UK.

Shapiro, S., & Roberts, C. (2019). Open Texture and Analyticity. In D. Makovec & S. Shapiro (Eds.), *Friedrich Waismann: The Open Texture of Analytic Philosophy* (pp. 189–210). Springer International Publishing. https://doi.org/10.1007/978-3-030-25008-9_9

Simion, M. (2018). The 'should' in conceptual engineering. *Inquiry: An Interdisciplinary Journal of Philosophy*, *61*(8), 914–928.

Simion, M., & Kelp, C. (2020). Conceptual Innovation, Function First. *Noûs*, *54*(4), 985–1002.

Sinclair, N. (2017). Conceptual Role Semantics and the Reference of Moral Concepts. *European Journal of Philosophy*, *1*, 95–121.

Skinner, Q. (2008). Freedom as the absence of arbitrary power. *Republicanism and Political Theory*, *3*(4), 83–101.

Sparrow, R. (2007). Killer robots. *Journal of Applied Philosophy*, *24*(1), 62–77.

Sundell, T. (2020). Changing the subject. *Canadian Journal of Philosophy*, *50*(5), 580–593.

Swedberg, R. (2017). On the heuristic role of concepts in theorizing. In *Theory in Action* (pp. 23–38). Brill.

Teichmann, R. (2021). Conceptual Corruption. In *Cora Diamond on Ethics* (pp. 33–55). Springer.

Thomasson, A. L. (2020). Pragmatic Method for Normative Conceptual Work. In A. Burgess, H. Cappelen, & D. Plunkett (Eds.), *Conceptual Engineering and Conceptual Ethics* (p. 0). Oxford University Press. https://doi.org/10.1093/oso/9780198801856.003.0021

Thomasson, A. L. (2022). How should we think about linguistic function? *Inquiry*, 1–32.

van de Poel, I. (2013). Translating Values into Design Requirements. In D. P. Michelfelder, N. McCarthy, & D. E. Goldberg (Eds.), *Philosophy and Engineering: Reflections on Practice, Principles and Process* (Vol. 15, pp. 253–266). Springer Netherlands. https://doi.org/10.1007/978-94-007-7762-0_20

van de Poel, I. (2020). Embedding Values in Artificial Intelligence (AI) Systems. *Minds and Machines*. https://doi.org/10.1007/s11023-020-09537-4

Van den Hoven, J. (2017). Privacy and the varieties of informational wrongdoing. In *Computer Ethics* (pp. 317–330). Routledge.

van den Hoven, J., Vermaas, P. E., & van de Poel, I. (2015). Design for Values: An Introduction. In J. van den Hoven, P. E. Vermaas, & I. van de Poel (Eds.), *Handbook of Ethics, Values, and Technological Design: Sources, Theory, Values and Application Domains* (pp. 1–7). Springer Netherlands. https://doi.org/10.1007/978-94-007-6970-0_40

Veluwenkamp, H. (2022). Reasons for Meaningful Human Control. *Ethics and Information Technology*, *24*(4), 51. https://doi.org/10.1007/s10676-022-09673-8

Veluwenkamp, H., Capasso, M., Maas, J., & Marin, L. (2022). Technology as Driver for Morally Motivated Conceptual Engineering. *Philosophy & Technology*, *35*(3), 71. https://doi.org/10.1007/s13347-022-00565-9

Vermaas, P. E., Tan, Y.-H., van den Hoven, J., Burgemeestre, B., & Hulstijn, J. (2010). Designing for Trust: A Case of Value-Sensitive Design. *Knowledge, Technology & Policy*, *23*(3), 491–505. https://doi.org/10.1007/s12130-010-9130-8