

# What is Conceptual Engineering and What Should It Be?\*

David J. Chalmers

*Abstract:* Conceptual engineering is the design, implementation, and evaluation of concepts. Conceptual engineering includes or should include de novo conceptual engineering (designing a new concept) as well as conceptual re-engineering (fixing an old concept). It should also include heteronymous (different-word) as well as homonymous (same-word) conceptual engineering. I discuss the importance and the difficulty of these sorts of conceptual engineering in philosophy and elsewhere.

I was going to call this article “Conceptually Engineering Conceptual Engineering” but that title was already taken by an excellent article by Manuel Gustavo Isaac. Instead I’ve chosen a title paying tribute to the Sally Haslanger mode of conceptual engineering (“Gender and Race: What are they? What do we want them to be?”). My original thought was to argue for a number of theses about conceptual engineering, where the main case study is engineering the concept of conceptual engineering. That got too vertiginous, but there will still be some of that. I’ll start by talking about the concept of conceptual engineering. I’ll distinguish different varieties of conceptual engineering, and I’ll argue that conceptual engineering should be understood broadly rather than narrowly. I’ll go on to apply this framework to issues about conceptual pluralism and about the importance and the difficulty of conceptual engineering.

## **What is conceptual engineering?**

What is conceptual engineering? We can start by trying to generate a definition. I’m officially committed to definitions being rare in philosophy, and more generally to the view that “What

---

<sup>0</sup>Published in *Inquiry*, 2020. <https://doi.org/10.1080/0020174X.2020.1817141>. This article is an edited transcript of a talk given at the NYU workshop on “Foundations of Conceptual Engineering” on September 14, 2018. Thanks to Banafsheh Beizaei for the transcription, to Vera Flocke for organizing the conference, to Manuel Gustavo Isaac and Steffen Koch for comments, and to the audience for discussion.

is X?” questions are less important than “What should X be?” questions. But I’ll start with definitions and the “is” question—conceptually analyzing conceptual engineering. These provide a useful first pass and a way into the “should” question—conceptually engineering conceptual engineering.

Here’s one obvious way to generate a potential definition of conceptual engineering. We can look up the definition of engineering, and then appeal to compositionality, applying the definition to concepts wherever possible. There are many definitions of engineering, often put forward by different engineering associations such as the mechanical engineers, the civil engineers, and the electronic engineers. But these definitions have a common core. There’s a fairly common simple definition, which a lot of the more complicated definitions are variations on.

On this definition, engineering is the process of utilizing knowledge and principles to design, build, and analyze objects. The key thing there is design, build, and analyze. You’ll find variations on those three things in most definitions of engineering. Some of them extend the list to about twenty different things. There’s to operate, to maintain, to repair, to forecast, to evaluate, and so on. But they mostly fit within this rubric of designing, building, and analyzing.

Invoking compositionality, conceptual engineering will be something like the process of designing, building, and analyzing concepts. That’s not a bad definition, except that ‘analyze concepts’ already has a meaning which is not totally apropos in this context. Maybe ‘evaluate concepts’ is better. And maybe ‘implementing’ is better than ‘building’ where concepts are concerned. With these tweaks, we get the following definition: conceptual engineering is the process of designing, implementing, and evaluating concepts. That is not a bad first pass at what conceptual engineering is all about.

This definition gives us different broad stages or types of conceptual engineering. There’s the design stage, where we design concepts. There are various ways to do that. One classic way would be to give a definition, or maybe an inferential role, or some paradigm cases, or something like that. Next is the implementation stage, like the stage where you actually build the bridge. In the implementation stage you actually use a concept, and perhaps try to get others to use it too. This is what Herman Cappelen calls conceptual activism. And then there’s the evaluation stage, which plays a central role in the conceptual ethics work by people like Alexis Burgess and David Plunkett. Here what’s key is the evaluation of how good these concepts are in themselves and for certain purposes, to see how well they play key roles.

You can see all three of these things playing a role, say, in bridge engineering. You design a bridge, you implement a bridge, you evaluate the bridge to see how well it’s doing. If the

evaluation isn't positive, you design some repairs and you implement the repairs. And so on. You also see something like this in software engineering. You design a program, implement the program, evaluate the program, and so on in a continuing circle. Of course in practice the stages are intertwined. For example, you can't design without some evaluation, and implementation typically requires some redesign.

Incidentally, I like the software engineering analogy because it addresses a worry about how conceptual engineering can work if concepts are abstract objects. It's not obvious that you can engineer a new resident of the third realm. Exactly the same issues arise for software engineering. Programs are usually thought of as abstract objects, but we have no problem saying that programs are engineered. There are complicated things but fairly obvious things you can say about how this is possible in both cases. If programs are pre-existing abstract objects, then what's relevant to designing and implementing a new program is specifying that program and realizing it by having a computer run the program. Similarly, what's relevant to designing and implementing a new concept is specifying the concept and realizing it by having people coming to grasp the concept. Anyway, I won't make heavy weather of abstracta versus concreta here.

A more important issue here is a distinction between creating versus fixing as kinds of engineering. In creating a bridge, we design and implement a new bridge from scratch, perhaps after a period of evaluating what is needed. In fixing a bridge, we evaluate an old bridge and then design and implement repairs. Both projects are central in ordinary engineering, central in software engineering, and ought to be central in conceptual engineering. This distinction will be what I'm focusing on to a considerable extent. Some theorists have suggested that conceptual engineering is roughly the project of fixing concepts. I'm going to argue that it should be understood more broadly to include the project of creating concepts: that is, designing, evaluating, and implementing concepts from scratch.

More generally, I'll suggest that conceptual engineering should be understood as the project of designing, evaluating, and implementing concepts. I won't try to give precise necessary and sufficient conditions. For example, are all three required for conceptual engineering, or does just one out of three suffice? I'm inclined to think that each of the three taken alone is at least a distinctive and important mode of conceptual engineering, and that conceptual engineering is most powerful when all three modes are combined. The projects of fixing concepts and of creating concepts both involve all three modes, and I think both should clearly count as conceptual engineering.

## Examples of conceptual engineering

First I want to go through a bunch of examples of what I think of as paradigmatic conceptual engineering, in both the creating and the fixing mode. These examples are all within philosophy. That's not because philosophy is the only locus of conceptual engineering. I think it's absolutely everywhere. But philosophy is the locus that I know best and that many in my audience will have particular expertise on. So I might as well talk about what we know.

Take something from metaphysics. For me, the concept of supervenience is a paradigm example of conceptual engineering. Someone once said supervenience has the smell of something that was thought up in the metaphysics lab. One class of properties supervene on another class if when you duplicate properties in one class, you duplicate properties in the other. This concept was engineered over the twentieth century. Moore had it without the name, Hare introduced the name, Davidson and Kim and others made much of it. To me that's paradigmatic conceptual engineering. And indeed the notion of supervenience was once thought to be one that could do a lot of philosophical work that previous concepts like identity might have been hoped to do. Then later on people thought that supervenience doesn't do that work so well, so they introduced concepts like grounding. This involves a bit more conceptual engineering or at least conceptual abstraction of ideas which are in the air, under a new label. So these are examples in the mode of creation. In the mode of fixing I think of Amie Thomasson's work on existence in metaphysics, which takes a pragmatic approach to which notion of existence or of object will serve us best.

I think conceptual engineering in the mode of creation is everywhere throughout the philosophy of language. Semantic values and notions of meaning are engineered all the time. Take Carnap's notion of intension (in *Meaning and Necessity*), Frege's notion of sense, Grice's notion of implicature, Kripke's notion of rigid designator. I think all of those can be seen as engineering fruitful new concepts in the philosophy of language. In the fixing mode, there's various work on truth. Actually Carnap's work on explication (in *Logical Foundations of Probability*) took truth as one of his central examples, and he saw Tarski's explication as a kind of conceptual engineering. More recently Kevin Scharp has diagnosed truth as an inconsistent concept, proposing ascending and descending truth as replacements.

In the philosophy of mind, Ned Block is a paradigmatic conceptual engineer. His 1995 paper on the function of consciousness says that consciousness is a mongrel concept that's problematic in various ways. He teases out more precise and interesting concepts, such as the concept of access consciousness. Herman Cappelen in his book on conceptual engineering puts forward my

paper with Andy Clark on the extended mind as a paradigm case of conceptual engineering, in engineering the concept of belief. My attitude towards that is slightly complicated, as I'll discuss shortly. Belief has actually been one of the central candidates for conceptual engineering since the 1940s and 50s, with Carnap and others giving it various explications. More recently Tamar Gendler has been engineering a new concept in the vicinity of belief with her concept of alief.

In social philosophy you find this kind of thing all the time. People use engineered concepts to point to phenomena that may have been overlooked, or to draw distinctive concepts out of strands of discussion. Miranda Fricker's work on epistemic injustice and its varieties like testimonial and hermeneutic injustice, would be a paradigmatic example here of drawing out a fruitful concept. Sally Haslanger's work on gender and race is another. A paradigm example would be her work towards the analysis of the concept of woman in terms of oppression. What Haslanger calls ameliorative analysis is conceptual engineering in the revisionary mode to serve various ends, including the ends of social justice. This ameliorative strand of conceptual engineering has been picked up by many other people in recent social philosophy. Kate Manne's revisionary analysis of misogyny is an example.

In metaphilosophy, which is roughly the area of this article, you can find a few paradigmatic examples. I think Carnap's discussion of explication was itself a wonderful example of conceptual engineering. Explication was a new and useful concept. And of course the very concept of conceptual engineering is a marvelous piece of conceptual engineering. Incidentally, in the recent literature on conceptual engineering, the credit for the term 'conceptual engineering' is often given to Simon Blackburn in 1999. The credit should really go to Richard Creath in 1990. Creath made a big deal of Carnap as a conceptual engineer. It first plays a role in his 1990 book *Dear Carnap, Dear Van*, and many other papers. Conceptual engineering has been all over the Carnap literature for decades. PhD theses have been written on it. As Manuel Gustavo Isaac pointed out to me, Carnap himself actually talks about linguistic engineering in a couple of places, but never uses the term 'conceptual engineering'. In any case the concept of conceptual engineering is itself a model of useful conceptual engineering.

### **De novo conceptual engineering vs. conceptual re-engineering**

With those examples in place, let's think about the whole fixing versus creating distinction. Herman Cappelen's definition of conceptual engineering ties it very close to the fixing project. The title of his book is *Fixing Language*. The more detailed definition (p. 3) is: assessing and improv-

ing our deficient representational devices. I'd say that's part of conceptual engineering. Suppose we said that civil engineering, say bridge engineering, was the project of fixing and improving our deficient bridges, or that software engineering was fixing and improving our deficient programs. That's part of bridge engineering and software engineering. But it's not the only part, and it's probably not the most important or most exciting part. There's also the whole project of building new bridges and building new software. I'm inclined to say that this is the paradigm case of bridge engineering and software engineering. It would be odd to omit projects like this from conceptual engineering. Cappelen in his book acknowledges the possibility of this project, and goes on to say something like, that's just not the project I'm interested in here. That's fine, but I think the project is worthy of our attention.

So I encourage making a distinction between what I call *de novo* engineering and re-engineering. *De novo* engineering is building a new bridge, program, concept, or whatever. Re-engineering is fixing or replacing an old bridge, program, concept, or whatever. It's not totally straightforward to draw the distinction. There are some hard cases. Take the Tappan Zee Bridge, just up the Hudson River from New York City. The old Tappan Zee bridge is still there for now. They're building a new bridge in the same location as the old bridge, in order to replace the old bridge. Is that *de novo* engineering because it's a new bridge, or is it re-engineering because it's a replacement? For my purposes I'm going to count that kind of thing as re-engineering, because the new bridge is being used to fix an old bridge. Likewise in conceptual re-engineering, the point is to fix a certain concept. We can argue about how to draw the lines.

Many or most of the standard examples in the recent conceptual engineering literature are cases of conceptual re-engineering. Certainly the Carnapian explication literature is very much a literature on re-engineering. Georg Brun has a nice paper on explication as conceptual re-engineering, using that exact phrase. The conceptual engineering of belief has largely been re-engineering. There's also been re-engineering with the concept of truth. More recently there has been a lot of interest in re-engineering social concepts such as the concept of woman and the concept of race.

Many of the examples I gave, on the other hand, look more like *de novo* conceptual engineering. Take the cases of epistemic injustice, supervenience, rigid designation, and, indeed, conceptual engineering. These weren't particularly trying to fix or replace other concepts. If you squint really hard, you might say that supervenience is intended as a replacement for identity. But that's not quite right. The concept of identity is doing fine. It's just that there's a job people were using identity for, in some reductive projects, that people then tried to use supervenience to do. Maybe you could argue that the concept of conceptual engineering is a replacement for the concept of

explication. Again I don't think that's the most productive way to think about it. The spirit is de novo, building rather than fixing. Conceptual engineering is perhaps somewhat more general than explication, but there's no need to get rid of the former. Both are some useful, fruitful concepts that we can use to do some interesting philosophy with, and that may have useful consequences.

My proposal is that conceptual engineering both includes and should include both de novo conceptual engineering and conceptual re-engineering. I'm more confident of the normative "should" claim. If the linguistic facts about current usage turn out to be that 'conceptual engineering' doesn't already cover both, then this will be a proposal for conceptually re-engineering conceptual engineering. That's fine. Conceptual re-engineering is important too, and this would be an example of it. But my own view is that 'conceptual engineering' already covers both. This is partly in virtue of compositionality. It would be very odd for conceptual engineering to work so differently from other kinds of engineering.

I may be wrong. Maybe the recent use of "conceptual engineering" by philosophers focusing on re-engineering has given us some semantic glue that makes it stick to that. So I don't want to put too much weight on the descriptive semantic claim. That's really a verbal dispute about conceptual engineering. But the normative claim has an underlying nonverbal point: conceptual engineering should cover de novo conceptual engineering, because of unity with conceptual re-engineering and because de novo conceptual engineering is at least as important in philosophy and elsewhere as conceptual re-engineering.

This leads to my sideline on conceptually engineering belief. The attitudes I just expressed about the conceptual engineering of conceptual engineering were pretty much Andy Clark's and my attitude about the conceptual engineering of belief back in "The Extended Mind". Cappelen quotes us saying: "We don't intend to debate what is standard usage; our broader point is that the notion of belief ought to be used so that Otto qualifies as having the belief in question." Here Otto is the guy who carries around the notebook which his memories are stored in. I'm happy to be taken as a paradigm conceptual engineer, but I'm not sure that we saw this as conceptual engineering. Our own view was that these extended cases of beliefs were literally beliefs. So the word 'belief' already covers them, perhaps in virtue of unity and the explanatory fecundity of the category. We were just saying, if someone wants to argue with that, here is the underlying more important claim.

My metaphilosophical ideology, expressed in my article "Verbal Disputes", is that you shouldn't get too hung up on words. On the other hand, words matter at least for practical purposes. This was roughly our attitude in "The Extended Mind". Andy and I could have introduced a new term,

‘e-believe’, to cover all these extended cases, and we could have only made claims about how unified e-belief is with the ordinary cases of believing and how e-belief plays the most important role. We could have done that, but what fun would that have been? We thought e-belief plays the most important roles associated with ‘belief’, and that as a result ‘belief’ picks out e-belief. Saying that belief is e-belief helps to communicate those claims. Furthermore, in practice the word ‘belief’ plays a central role in both social and scientific explanation. So attaching the word ‘belief’ to e-belief helps to create a mindset where e-belief can actually play those roles.

I’m inclined to say something like this about about ‘conceptual engineering’. I think it does cover de novo conceptual engineering. If you think that’s not the actual meaning of the term, then I’ll argue that it *should* be used to cover de novo conceptual engineering. Conceptual engineering is a hot topic, people have conferences on it. So using that name has what Cappelen calls a lexical effect (p. 122). Using only another name, like ‘conceptual coinage’ or something, is not going to work in the same way. So pragmatically it makes sense to try and attach this thing you’re interested in to this word. That’s not just a selfish benefit. I think de novo conceptual engineering plays really central roles throughout philosophy, and it has a great deal in common with conceptual re-engineering. If people understand conceptual engineering in the broader way, then they will benefit, by cutting the world at its joints. I think that as with belief, understanding conceptual engineering this way yields a deeper, richer, and more unified subject matter.

Of course there is a legitimate narrower subject matter of conceptual re-engineering with some distinctive concerns. At the same time, words have power, and “conceptual engineering” is a powerful expression. If people had just talked of “concept repair” or “concept re-engineering” from the start, I don’t think the interest would have been so great. The power comes in part because of the suggestion of something that goes beyond mere fixing. I think we should use the term in a way that vindicates that power.

## **Homonymous and Heteronomous Conceptual Engineering**

This immediately leads us to delicate questions about the connection between conceptual engineering and linguistic engineering. Sometimes these phrases are used almost interchangeably. Certainly it’s the case that wherever you find conceptual engineering, you almost always find some linguistic engineering. Whenever there’s a proposal about a new concept, there’s also a proposal about a word for it to be attached to. This is a version of what Peirce calls “the ethics of terminology”, which Stephen Yablo introduced me to years ago. Much of this really comes



down to the ethics of terminology, or the ethics of language—how should we be using new and old words to express new and old concepts?

In the linguistic mode, there's a related distinction between homonymous and heteronymous conceptual engineering. Homonymous is same-word linguistic engineering—fixing a new meaning for an old expression. Cases like 'woman' or 'misogyny' and 'truth' are homonymous, at least insofar as they involve conceptual engineering at all. The central cases of homonymous conceptual engineering are cases of conceptual re-engineering: fixing the old concept expressed by the word. There will be some non-central cases where you can use an old word for something totally new. The word 'supervenience' already existed with a different meaning, but I don't think the new meaning is usefully regarded as fixing the old meaning of the word. In that case it's more or less coincidental that the same word is being used.

Heteronymous conceptual engineering is different-word linguistic engineering. For example, Ned Block re-engineered the concept consciousness into multiple strands such as access consciousness as phenomenal consciousness that he took to be present in our thought and talk about consciousness. Instead of saying, let's use the old word 'consciousness' to express access consciousness—that would be homonymous—he said let's use the term with a prefix, 'access consciousness', to express the new concept. Using a prefix is a very common strategy in philosophy. The prefix makes it heteronymous, but with a degree of continuity. In other cases there's a new concept and a new word: 'supervenience' and, indeed, 'conceptual engineering' may be cases of this.

The existing literature on conceptual engineering focuses almost entirely on the homonymous case, but I think the heteronymous case is just as interesting. One thing that makes it interesting in this context is that it's at least somewhat less vulnerable to challenges such as Strawson's challenge to Carnapian explication: aren't you just changing the subject? For heteronymous conceptual engineering, the answer is often yes, I'm changing the subject—here's an interesting and important subject. It also helps with the worry about control of meaning, where external factors make it hard to change meanings. It's not that these worries don't arise in coining a new word. But in the use of an old word, they arise in a big way, because there's a whole social community of users of the word to defer to, making it all the harder to effect conceptual change.

You might ask why the homonymous/heteronymous distinction matters. I'm on record as saying that the words we use for things don't matter all that much in philosophy. So why does it matter whether we use a new word or an old word to make our claim? What's the point of homonymous conceptual engineering or indeed of heteronymous conceptual engineering? Here

we should distinguish between broadly theoretical and broadly practical projects. For theoretical purposes, I think in principle anything you can say with homonymous conceptual engineering can also be said with heteronymous conceptual engineering, and vice-versa. You can use an old word, you can use a new word. For theoretical purposes it's just a verbal difference between those two. And indeed, as I say in "Verbal Disputes", for an ideal reasoner this difference wouldn't matter. But we aren't ideal reasoners. We have strong associations with the use of certain words. So it makes a big practical difference whether one uses a new word or an old one.

Of course projects in heteronymous conceptual engineering could be carried out homonymously, and vice versa. Instead of introducing the new word 'supervenience', we could say, here is what 'reduction' should mean. And then give a definition of reduction in terms of supervenience. Or instead of using the word 'sense', we could appropriate a word like 'meaning'. The same goes vice-versa for various homonymous projects, like the revisionary analysis of truth. Kevin Scharp ends up using different phrases, 'ascending truth' and 'descending truth'. For Sally Haslanger's ameliorative analysis of 'woman', there's certainly going to be a related heteronymous project that introduces a new word to express the new concept. But the practical effects will be different. The upsides and downsides here are mainly practical rather than theoretical, although the lines can blur in some cases. As Ari Koslow discusses in her paper on these issues, sometimes we can achieve our purposes better with a new word, and sometimes with an old word.

Here are some upsides and downsides. One downside of the heteronymous project is that new words are expensive and harder to get people to use. A corresponding upside of the homonymous project is that old words are entrenched and often have associated prestige. 'Conceptual engineering' is prestigious—there are special issues of journals with that label. So if you want to get your concept adopted, attaching it to that powerful engine can help. All this applies especially to roles. An old word has certain fixed roles. Homonymous conceptual engineering can allow a new concept to very easily become associated with those roles. The concept of marriage, for example, plays a huge role in society. If you have revisionary views about what should play those roles and want to implement them, tying them to the old word is a powerful way to do it. This ties closely to projects in social justice. Given fixed social roles for terms like 'woman' or 'marriage', allowing a new concept to play those social roles can in many cases make for a more just world. One could introduce a new term like 'civil union' to play some of the roles of marriage for same-sex couples, but this term doesn't have the built in power, associations, and legal roles of 'marriage'. Homonymous engineering is much more powerful in this case. Those are some of the upsides of homonymous engineering.

There are downsides too. Homonymous conceptual engineering can lead to confusion, with all the multiple meanings floating around. This applies especially in theoretical contexts. It's often theoretically cleaner to have distinct terms for distinct phenomena. As Cappelen's discussion brings out, homonymous engineering can also be extremely difficult to implement, unless one is very powerful or very lucky, or in a small community. Perhaps there are only fifty people in the world working on conceptual engineering, so a speaker at a conference where many of them are present has a chance of changing the meaning or usage of 'conceptual engineering'. But even then it's not a big chance. Heteronymous conceptual engineering, with a new word or a new prefix like 'de novo', is perhaps somewhat easier to implement, though it's still not easy to get it adopted. Those are upsides and downsides.

### **Conceptual engineering and thesis engineering**

Let me say something about the importance of conceptual engineering. Some people say it's all of philosophy. Is it important? Yes, some of the most important advances in philosophy have quite clearly involved conceptual engineering. That goes especially for some of the cases I gave of de novo conceptual engineering—rigid designation, implicature, epistemic injustice, and so on. These new concepts capture important phenomena and can do a lot of work.

Is conceptual re-engineering important in philosophy? It's certainly practically important, especially for practical roles in the wider world, such as achieving social justice. Philosophers have an important role to play in that project—both the activist part and the more theoretical part of figuring out which concepts can best play these roles. Aside from these roles in the wider world, conceptual re-engineering can also be theoretically useful within the philosophical community. For example, it can clean up a concept to make it more natural and powerful, modify a concept so it can better play explanatory roles, and distinguish different concepts that play different roles. Maybe the different notions of probability and consciousness provide examples of that. I think most of the time this can be done just as well in heteronymous or de novo mode. We'll come back to the reasons for that shortly.

Is conceptual engineering everything in philosophy? This is clearly false. Lots of important philosophy involves arguing for theses using old language, and is none the worse for that. Think of cases like Jackson's knowledge argument or Parfit's repugnant conclusion. As far as I can tell, there's no real conceptual engineering in these cases. You can use the old concepts still for very interesting phenomena and theses.

Even when philosophy involves new concepts, I think it typically also involves new theses involving those concepts. I think the role of theses here is crucial. Why is supervenience interesting? You don't just look at supervenience and say, that's a cool concept. Rather, it serves some potential roles for you that can connect to certain theses. Many people think, for example, that physicalism requires supervenience. Frank Jackson or David Lewis may have held more strongly that if you have supervenience, then you have physicalism. On this view, supervenience suffices for various reductive projects. Then Kit Fine comes along and argues with that, and we can assess those theses.

Or take implicatures. The concept of implicature is interesting, but really the key work it does is through theses about where implicature happens, or about which maxims govern implicatures. Implicatures play these roles in discursive practice, and so on.

In general theses are where the rubber meets the road. Theses have a certain primacy in philosophy, at least the sort of philosophy aimed at discovering truths. In this sort of philosophy, conceptual engineering is largely driven by thesis engineering. The debate over grounding and supervenience in metaphysics, for example, really comes down to what they are useful for, which comes down to which theses about them are correct. For example, does supervenience on the physical suffice for a physicalist ontology, or is grounding in the physical required? Likewise, for assessing the status of various concepts of meaning, we look at the various properties of language that these concepts can explain, which comes down to assessing various theses they are involved in.

This is not saying that concepts are not important in truth-aiming philosophy. It's to say that the importance of concepts derives from the importance of theses that they're involved in. This is something that Carnap tacitly recognized in his discussion of explication (Carnap 1950, p. 7), where he talked about the fruitfulness of a concept as being on one of the dimensions of assessment. The fruitfulness of a concept comes down to the number of interesting and useful and explanatory theses it might be involved in. So on this view the importance of concepts derives from the importance of theses, and certainly formulating theses and arguing for them goes well beyond conceptual engineering.

It's true that philosophy has many aims distinct from discovering truths. Philosophy also aims to raise questions, to help us understand, to help us see the world differently, to live better lives, to improve the world, and so on. New and improved concepts certainly can help with those things. For example, once you have the concept of epistemic injustice, you see all sorts of old situations in a new way, and this can help achieve more just outcomes.

Even in these cases I'm not sure that a new and improved concept has non-derivative importance. What's important is the role it plays, in this case in helping us see the world in a new way and in improving the world. I think that very often these roles are tied to theses—such-and-such cases are cases of epistemic injustice, epistemic justice is bad in a certain distinctive way, and so on. Where practical projects are concerned, normative and evaluative theses will often be central to the philosophical grounding of these projects. But perhaps aiming for true theses doesn't capture all that's going on. So perhaps more generally, one should speak of the primacy of roles rather than the primacy of theses, and understand the fruitfulness of a concept as tied to the roles it can play. Still, at least one very common mode of analytic philosophy is mainly aimed at discovering truths, and at least in this mode thesis engineering has priority over conceptual engineering.

### **Conceptual engineering and concept pluralism**

This leads to the metaphilosophical view of these things which I favor, concept pluralism, which Cappelen cites in his book as a kind of conceptual engineering view. On this view, in the vicinity of most philosophical words, there are many concepts playing different roles. We ought to articulate those roles and find the concepts that play them best. We're evaluating concepts in many cases, finding relatively new concepts that can play them. That's a kind of conceptual engineering. For the most part I don't care whether it's homonymous or heteronymous, except for some practical purposes. But again, what matters here is not just the concept. What matters is theses, such as theses of the form 'x plays role y'. You get an important concept when you get some important theses.

For a concept pluralist, de novo conceptual engineering is often better than re-engineering. This is the strongest thing I'm going to say in favor of de novo conceptual engineering. My paradigm of conceptual re-engineering is that you somehow fix or replace the old concept, so the old concept is no longer around. As a concept pluralist, I often want to ask, why not have both concepts? Even, say, for the case of Haslanger's analysis of woman. Even if you think that that the old biologically-based concept of woman was an unjust concept for various social purposes, somebody might still think that nonetheless it's a useful concept to have around, say for certain medical purposes. Just because a concept is useless for some roles, doesn't mean it's useless for all roles.

In some cases, the old concept is so defective—maybe an inconsistent concept, an imprecise concept, an immoral concept—that it's better gone. My own view is concepts which are that

defective are actually fairly rare, and it's not the the typical case. There are certainly words that are so morally odious, or so imprecise or so ambiguous, that we would do better to get rid of them entirely. Even in these cases, I don't think it's obvious that there's always an underlying concept that shares the problem. But this probably depends on one's view of concepts. Anyway, there are plenty of cases where the old concept is not so defective and can still serve some purposes here and there. In those cases, the concept pluralist says, why not keep them both around for different purposes? Then we have a fruitful conceptual apparatus.

### **The difficulty of conceptual engineering**

I should say something about the difficulty of conceptual engineering. Cappelen argues that as a corollary of semantic externalism, we have very little control over the meaning of our words. My view is that conceptual engineering is clearly possible and clearly difficult. It's a very difficult social project. I think that breaking it into stages helps. Our three stages are (i) designing a concept, such as proposing a meaning for a word, (ii) evaluating a concept, such as seeing how well this meaning can play certain roles, and (iii) implementing the concept, say ensuring the word is used to express this concept within a community. The design project is not that hard. You propose a definition or inferential role or a set of paradigm cases. Evaluation is nontrivial. People can argue about what concepts best play what roles. Still, both of these are things that philosophers do all the time, and it's not clear that externalism or lack of control is a major obstacle. The hardest part is implementation. First, you have to use an expression in the relevant way yourself. Even that much is nontrivial. Second, you have to get others to use it that way. That's really hard. And then third, even after an expression is used the right way, you have to ensure that the meaning, the reference, is right.

Externalism and metasemantic issues mostly affect this final step, the step from use (which we can construe narrowly in terms of internal processes and behavior) to meaning. The externalist will say that narrow use doesn't fix meaning. I think there's various things one can say here. My main reaction (related points have been made by Steffen Koch, Mark Pinder, and Jared Riggs) is that the most important work of the implementation project of conceptual engineering can be done just by changing use widely enough. Usually doing that will be enough to change the meaning. If it turns out that it doesn't change meaning, it's not clear how much that matters. Changing use is what we need for many practical purposes. If everyone uses 'marriage' as if it applies to same-sex marriage, then even if historical external links say that 'marriage' still refers only to unions between men and

women, this will matter very little for practical purposes. If everyone uses ‘supervenience’ as if it expresses the modal concept, it won’t matter if it actually means something else. So I’m not sure that the externalist gap between use and meaning is the biggest obstacle to conceptual activism.

A bigger obstacle is the project of changing use. The hardest part is the social project of getting others to use a word your way. As a philosopher I’ve managed to get people to use a word or phrase my way occasionally over the years, most often with relatively technical terms in small academic communities. But attempts like this fail much more often than they succeed. I wish I knew the secret of success so I could bottle it. As it is, one just tries and hopes that one can break through social inertia.

Stepping back, I think once you break down conceptual engineering into design, implementation, and evaluation, it becomes clear that concept design and concept evaluation are relatively straightforward, or at least don’t face the same sort of social obstacles. They suffice for many theoretical purposes. Concept implementation requires changing use, which is a difficult social project, but it’s possible.

## **Conclusions**

Some conclusions. In my view, conceptual engineering is the design, implementation, and evaluation of concepts. At least, that’s how it should be understood. Conceptual engineering includes, or at least should include, de novo conceptual engineering as well as conceptual re-engineering, and heteronymous as well as homonymous conceptual engineering. In my view de novo conceptual engineering is often the most fruitful, especially for theoretical purposes. Homonymous conceptual engineering is also very important, especially for practical purposes that can go well beyond philosophy. Conceptual engineering certainly doesn’t exhaust philosophy. In much of philosophy theses have a degree of priority over concepts, and in these domains the importance of conceptual engineering derives largely from its role in thesis engineering. Concept design and concept evaluation are relatively achievable, but widespread concept implementation is a difficult social project. As a result, conceptual engineering on a community-wide scale is difficult, but it is possible.

## **References**

Blackburn, S. 1999. *Think: A Compelling Introduction to Philosophy*. Oxford University Press.

- Block, N. 1995. On a confusion about a function of consciousness. *Brain and Behavioral Sciences* 18:227-247.
- Brun, G. 2016. Explication as a method of conceptual re-engineering. *Erkenntnis* 81:1211-1241.
- Burgess, A. & Plunkett, D. 2013. Conceptual ethics I. *Philosophy Compass* 8 (12):1091-1101.
- Cappelen, H. 2018. *Fixing Language. An Essay in Conceptual Engineering*. Oxford: Oxford University Press.
- Carnap, R. 1947. *Meaning and Necessity: A Study in Semantics and Modal Logic*. University of Chicago Press.
- Carnap, R. 1950. *Logical Foundations of Probability*. University of Chicago Press.
- Chalmers, D.J. 2011. Verbal disputes. *Philosophical Review*.
- Clark, Andy & Chalmers, David J. (1998). The extended mind. *Analysis* 58 (1):7-19.
- Creath, Richard (ed.) 1990. *Dear Carnap, Dear Van: The Quine-Carnap Correspondence and Related Work*. University of California Press.
- Davidson, D. 1970. Mental events. In L. Foster & J. W. Swanson (eds.) *Experience and Theory*. Clarendon Press.
- Fine, K. 2001. The question of realism. *Philosophers' Imprint* 1: 130.
- Frege, G. 1892. ber Sinn und Bedeutung. *Zeitschrift fr Philosophie Und Philosophische Kritik* 100:25-50.
- Fricker, M. 2009. *Epistemic Injustice*. Oxford University Press.
- Gendler, T. 2008. Alief and belief. *Journal of Philosophy* 105:634-663.
- Grice, H.P. 1961. The causal theory of perception. *Aristotelian Society Supplementary Volume* 35:121-168.
- Hare, R.M. 1952. *The Language of Morals*. Oxford University Press.
- Haslanger, S. 2000. Gender and race: (What) are they? (What) do we want them to be? *Noûs* 34:31-55.
- Isaac, M.G. 2020. How to conceptually engineer conceptual engineering?, *Inquiry*. DOI: 10.1080/0020174X.2020.1719881
- Jackson, F. 1982. Epiphenomenal qualia. *Philosophical Quarterly*.
- Kim, J. 1993. *Supervenience and Mind: Selected Philosophical Essays*. Cambridge University Press.
- Koch, S. forthcoming. The externalist challenge to conceptual engineering. *Synthese*.



- Koslow, A. forthcoming. Conceptual bridge building. Presented at “Foundations of Conceptual Engineering”, NYU, September 2018. [<https://www.youtube.com/watch?v=cDmpdC6np3E>]
- Kripke, S. 1980. *Naming and Necessity*. Harvard University Press.
- Manne, K. 2017. *Down Girl: The Logic of Misogyny*. Oxford University Press.
- Moore, G.E. 1922. *Philosophical Studies*. Routledge.
- Parfit, D. 1984. *Reasons and Persons*. Oxford University Press.
- Pinder, M. forthcoming. Conceptual engineering, metasemantic externalism, and speaker-meaning. *Mind*.
- Riggs, J. forthcoming. Conceptual engineers shouldn't worry about semantic externalism. *Inquiry* 1-22.
- Scharp, K. 2013. *Replacing Truth*. Oxford University Press.
- Strawson, P.F. 1963. Carnap's views on conceptual systems versus natural languages in analytic philosophy. In Paul Arthur Schilpp (ed.), *The Philosophy of Rudolf Carnap*. Open Court: La Salle 503-518.
- Thomasson, A. 2015. *Ontology Made Easy*. Oxford University Press.