

Methods as a form of engineering knowledge

Martin Stacey¹, Claudia Eckert², Zachary Gallagher Pirtle³,
Michael Poznic⁴, Beth-Anne Schuelke-Leech⁵ and Loretta von der Tann⁶

¹Faculty of Computing, Engineering and Media, De Montfort University, Leicester, UK

²Department of Design and Innovation, The Open University, Milton Keynes, UK

³Independent Scholar, Washington DC, USA

⁴Institute for Technology Assessment and Systems Analysis, Karlsruhe Institute of Technology, Karlsruhe, Germany

⁵Department of Mechanical, Automotive and Materials Engineering, University of Windsor, Windsor ON, Canada

⁶CIRIA, London, UK

Abstract

Methods comprise a significant part of the knowledge engineers are taught and that they use in professional practice. However, methods have been largely neglected in discussions of the nature of engineering knowledge. In particular, methods prove to be hard to track down in the best-known and most influential typology of engineering knowledge, put forward by Walter G. Vincenti in his book *What Engineers Know and How They Know It*. This article discusses contemporary views of what engineering methods are and what they contain, how methods (fail to) fit into Vincenti's analysis, and some characteristics of method knowledge. It argues that methods should be seen as a distinct type of engineering knowledge. While characterizing the knowledge that methods include can be done in different ways for different purposes, the core of method knowledge that does not fit into other categories is explicit 'how-to' knowledge of procedures, that draw on other types of knowledge.

Keywords: Design method, Design process, Engineering knowledge, Epistemology, Philosophy of technology

Received 29 July 2024
Revised 09 April 2025
Accepted 10 April 2025

Corresponding author
Michael Poznic
michael.poznic@kit.edu

© The Author(s), 2025. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

Des. Sci., vol. 11, e12
journals.cambridge.org/dsj
DOI: 10.1017/dsj.2025.9



1. Introduction: understanding what engineers need to know

To create the technology of the modern world, engineers need to generate and deploy an immense amount of engineering knowledge. Engineering is very seldom an individual activity: to work together, and to meet future challenges, engineers need to exchange technical knowledge and train students and junior colleagues. They also need to organize and coordinate very complex large-scale projects so they run efficiently and effectively and produce products that meet the requirements of their users and society. To accomplish this, it is not enough to possess knowledge of mathematics and physics and chemistry, and about what machines and their elements exist and are possible, what they are for, how they work and how they perform. Engineers also need to possess and share knowledge of how to use all their engineering knowledge, at different levels of generality and abstraction, to know how to solve different kinds of engineering problems. Engineers also need to

understand the processes and procedures their organisation follows or that are demanded by legislation or regulation. What does this knowledge of how to do engineering look like, and how does it fit into a broader understanding of the range of engineering knowledge?

Much of this knowledge is embedded in *methods* for carrying out particular engineering tasks: explicit, articulated accounts of how to carry out activities or achieve results. Plato and Aristotle understood that craft knowledge of how to do technical things – *techné* – involved a combination of craft skill and theoretical understanding (see Parry 2020), while the importance of engineering of practical rules of thumb drawn from experience has been recognised (see Norström 2011). But explicit communicable strategies for carrying out complex tasks go beyond ad hoc rules and practical skills. They also go beyond deterministic procedures for computing results, to include ways to do tasks that *might* work, or that provide guidance for producing results that depend on the knowledge and skill of the engineers using them, as well as ways to organize complex processes. While the nature and distinctive characteristics of engineering knowledge have been debated by philosophers (see Kant & Kerr 2018, for a review), methods have attracted remarkably little attention from epistemologists. This article argues that methods are an important component of engineering knowledge and that they constitute a distinct type of engineering knowledge with its own characteristics.

While methods have been neglected in the discussion of engineering knowledge, there has been renewed interest in the engineering design community. Methods are discussed in terms of what constitutes a method (e.g. Gericke, Eckert & Stacey 2017; Daalhuizen & Cash 2021) and how methods for particular applications can be validated (e.g. Gericke et al., 2022; Eisenmann *et al.* 2021), that is, how we can be sure that a method constitutes knowledge. Engineering educators devote significant parts of engineering education to teaching methods, and working engineers do a lot of their work by applying methods whether they conceptualize their activities in terms of methods or not. We can say the same for computer science and other technological and design professions. A significant part of design research is devoted to developing new methods, while in industry methods are also employed as explicit means of sharing how to approach problems across organisations or over time.

Understanding what engineering knowledge is – what engineers know and how they know it – is important for understanding what is going on in engineering design and other engineering activities, and finding ways to make the work of engineers easier and more effective. The best-known and most influential attempt to map the space of engineering knowledge was put forward by Walter G. Vincenti in his book *What Engineers Know and How They Know It* (Vincenti 1990). By focusing on the content of the knowledge rather than its form, Vincenti presents a view of what engineers know and the knowledge-creating activities that add to what they know, that fit how engineers themselves see the world, and that communicate the types of the subject matter of engineering knowledge to non-engineers. We regard Vincenti's enterprise – examining the content of engineering knowledge, and explaining engineering to non-engineers – as valid and important, while his work remains a foundation stone for explorations of engineering epistemology. Therefore, we consider assessing how well Vincenti's framework fits contemporary engineering, and developing and updating it if necessary, as a valuable contribution.

Knowledge of methods turns out to be remarkably hard to track down in Vincenti's analysis. We argue in this article that method knowledge does not fit any of Vincenti's categories of engineering knowledge – *fundamental design concepts, criteria and specifications, theoretical tools, quantitative data, practical considerations*, and *design instrumentalities* – and that a new and distinct province needs to be added to his map.

The article begins with a discussion of what methods in engineering are, and why characterizing what methods are and describing them is not straightforward, in Section 2. In Section 3, the article explores what engineering knowledge is knowledge of, focusing on the views of Walter G. Vincenti, and whether methods fit into Vincenti's analysis. Section 4 looks at what method knowledge is, and what its characteristics are. The article concludes in Section 5 with brief remarks on how a better understanding of method knowledge can contribute to epistemology, to the development of methods, to the training of engineers and engineering practice.

2. On methods

What a method is and what it includes – and what it should include – is a problematic issue for engineers, software developers and design researchers. We all think that we know what a method is. But the word 'method' means different things to different people, and the picture is muddled further by the use of 'methodology' both as a pretentious word for 'method' and as a term for a way to structure and organise a development process that can encompass a number of more local methods (see Gericke et al. 2017; Gericke, Eckert & Stacey 2022 for a discussion of these varying meanings). We start from the view that a method is a specification of how a specified result is to be achieved (That is, what a person or team should do to achieve a result. Solution principles for how components of designs might work to meet human and technical needs are sometimes labelled methods, but are something else, and a well-recognised part of engineering knowledge; producing knowledge about solution principles is the goal of a lot of engineering work. See Stacey & Eckert 2022, for some of our thoughts on how design elements described at different levels of abstraction function as knowledge). However, depending on the method, a satisfactory result might be a fully worked out design solution, or an exact numerical analysis finding, or just a promising idea or some insight into a problem. And what constitutes a satisfactory result can depend on need and context as well as on the method itself.

Specifying *how* a result is to be achieved involves defining a procedure to be followed – a sequence of steps, specified as activities or as results to be achieved at each stage, that the users of the method 'ought' to perform to achieve the result. As this is guidance to be followed, the statements of the procedure are prescriptive; knowledge of methods involves knowing what the prescriptions are, so includes prescriptive knowledge (see Houkes 2013; Zwart 2021). Depending on the method, what the guidance on how to achieve the result adds to the user's understanding of the problem and how to solve it can focus on the set of subgoals and intermediate results that it asks for, on the sequence of activities or how to construct it, the form that the results should take, the reasoning processes to employ, or how to formulate a well-structured soluble problem whose solution is likely to be a satisfactory solution to the original problem.

Different fields in engineering use different methods in their work. For instance, a Failure Modes and Effects Analysis (FMEA) is a method for evaluating the likelihood, severity, and detection methods for design and process failures. Stress analysis and Finite Element Analysis (FEA) are methods for studying the effects of forces on structures and materials. A fishbone diagram is a method for identifying the cause-and-effects interactions in a process. Methods differ in how much they focus on specifying the organization of the activities to be carried out, versus specifying the reasoning processes or calculations to be performed and the form of the results to be produced. Methods also differ in the size of the subset of the total engineering challenge they apply to; for example, Quality Function Deployment (QFD) is a complex method for integrating quality management into product design and production; it covers large parts of a product development process and specifies a set of sub-methods for parts of the process. By contrast, doing a card sort plays a specific limited role in figuring out how to structure information presentation with a software interface. Most methods, though by no means all, produce information as their outputs. As a rough approximation, we can divide methods into three main categories: (i) deterministic methods through which outputs can be computed from inputs in a reliable and repeatable way, such as stress-strain diagrams; (ii) heuristic methods that specify how to generate new information and the form it should take, where the results depend on the knowledge and skills of the practitioner, such as QFD and FMEA; and (iii) methods for structuring processes.

In engineering design and software development, methods for structuring product development processes, such as Scrum and Rapid Application Development (RAD), play an important role, different from that of methods for using particular techniques to generate information and solve local problems.

We can view the guidance that methods provide more abstractly as adding information to the original problem to make it (some combination of) more structured, more detailed, or more concrete, so that formulating the problem to be solved in terms of the additional information should lead the user of the method to reach a solution of the original problem more easily, or more efficiently, or to reach a better solution.

2.1. Methods and processes

What methods are for is to guide problem-solving behaviour. They are thus concerned with specifying processes that individuals, teams or organisations should go through, though the activities can vary enormously in scale and the specifications in the level of detail. ‘Process’ is another problematic word. In engineering, it has two distinct and equally well-established meanings: a specified set of actions or activities to be carried out; and the actual sequence of actions that are carried out. In industrial practice, the relationship between the two can get complicated.

Processes can be planned, and large-scale processes in engineering usually are. Process models that are constructed as specifications of how to carry out individual tasks can constitute process plans. Methods are more general: they are specifications for how to carry out a *class* of activities; they need to be *instantiated* by fitting the details of the problem to the elements of the method, sometimes through an explicit process of customization.

Processes get modelled at different scales for a variety of purposes (see Wynn & Clarkson 2018). Many process models are constructed as specifications of how to carry out tasks; these are frequently instantiations of standard methods. Frequently, process models are constructed on the assumption that particular methods will be used for particular steps. Sometimes processes need to be planned for idiosyncratic, one-off problems with their own structure, possibly by drawing on elements of existing methods. Engineers and researchers also sometimes produce objective factual accounts or rational reconstructions or idealisations of procedures that are followed in practice by experts, and are thus the processes they *do* use, rather than *should* use. But when accounts of actual practice get produced, it is usually with the aim to pass on good practice. The distinction between description and prescription can get slippery when descriptions idealize away from messy reality, and descriptive accounts get used as guides (Eckert & Stacey 2010). They can thus become, de facto, elements of methods for tackling particular types of problems when not created as such. Moreover, the roles and epistemological status of a process model can change in the course of the development of a product (see Stacey, Eckert & Hillerbrand 2020).

2.2. Method content

But there is more to a method than naming a sequence of actions. Using a method involves making use of particular ideas, and often particular notations, tools and theoretical equipment; but what of these belong to a method and what do not is perhaps a matter of perspective. The content of methods for engineering design has been characterized in different ways by different design researchers.

Gericke *et al.* (2017, 2022) argue that a method contains multiple elements (see Figure 1): all methods possess a *core idea*, which encompasses the basic activity or transformation that can be achieved through the method; and a *procedure*, a sequence of the activities that need to be carried out in order to achieve a result. Many methods specify what *representations* of particular types of information should be used with the method. Additionally, many methods make use of *tools*; in engineering these are often software systems that automate parts of the activity or manage information. For Gericke *et al.*, the core idea includes the set of concepts

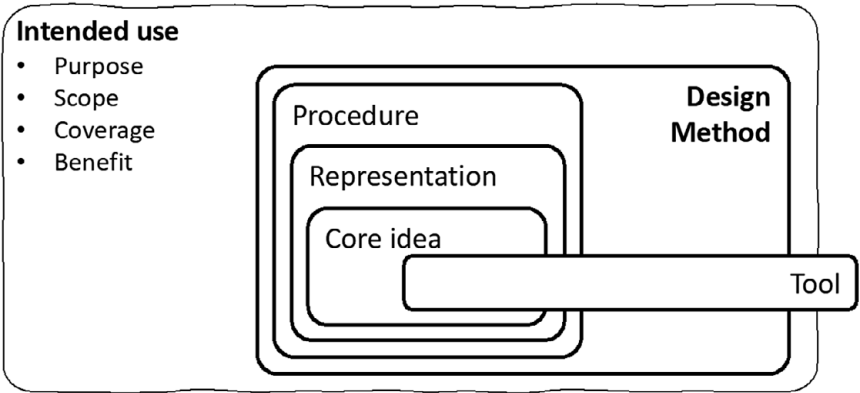


Figure 1. Elements of a method (figure reproduced from Gericke *et al.* 2017).

that people need to think with when using the method. For design synthesis or analysis methods in engineering, these concepts include design elements and their structure and behaviour and the functions they can be used to achieve, as well as the ways these should be conceptualised, plus the physical theory and mathematics that are used in the method. Extrinsic to the method, but essential to it, are its scope and its intended use: *what* it is supposed to achieve, and for what kinds of problems in which circumstances. Gericke et al. argue that all these aspects of methods need to be explained in an adequate description of a method.

Daalhuizen & Cash (2021) start with a slightly more structured but congruent definition of a design method: “a formalised representation of a design activity, which functions as a mental tool to support designers in achieving a goal, in relation to the circumstances and resources available” (Daalhuizen et al. 2019). They put forward a *Method Content Theory*, which divides the content of a method into *method goal*, *method procedure*, *method rationale*, *method framing* and *method mindset* (Figure 2). Daalhuizen & Cash (2021, section 2.1.5) draw on Andreasen (2003) for the idea of a method mindset, which they define as “the set of described values, principles, underlying beliefs, and logic that inform a method and its use.” This covers the concepts and ways of thinking about the problem that Gericke et al. (2017, 2022) includes in the core idea. Daalhuizen and Cash recognise the importance of using and generating information artefacts to many methods, but they do not discuss representations and do not treat what form the information artefacts take as a distinct part of (many) methods. Conversely, Gericke et al. do not treat method rationale (why should the method be used in this context?) as central to the content of a method, but rather as extrinsic to the method and belonging to the description of its intended use (Figure 1).

2.3. The scope of the method

In Daalhuizen & Cash’s (2021) Method Content Theory, method framing is defined as “the context of use described in the method and its implications and prerequisites for method use.” Gericke et al. (2022) file this under the scope of the method, and thus as extrinsic to the method itself. Gericke et al. argue that method descriptions should include a clear statement of the range of uses their creators intend them for, or that they have been found to be useful for, and that this is important for making methods useful. However, the generators of a method have no control over the deployment of methods. For example, in the early 2000s, Six Sigma was a very fashionable set of methods for improving processes. It had arisen from quality control in manufacturing processes, where companies monitored the deviations from the production norm and aimed to narrow the variation, so that a product six standard deviations away from nominal would still work. As anything beyond six standard deviations is rare or an outlier it improved quality standards by design. However, companies then started to apply Six Sigma methodology to product development processes including aspects that are not easily measurable such as communication. For example, one of the authors observed a company measuring the distance between desks to get a sense of who was or could be talking to whom, from which they claimed to have gained useful insights. Similarly, DSMs (Design Structure Matrices) are widely used to map dependencies between components of systems (Eppinger & Browning 2012); but some applications just use a basic DSM as a convenient representation of relationships, whereas others make

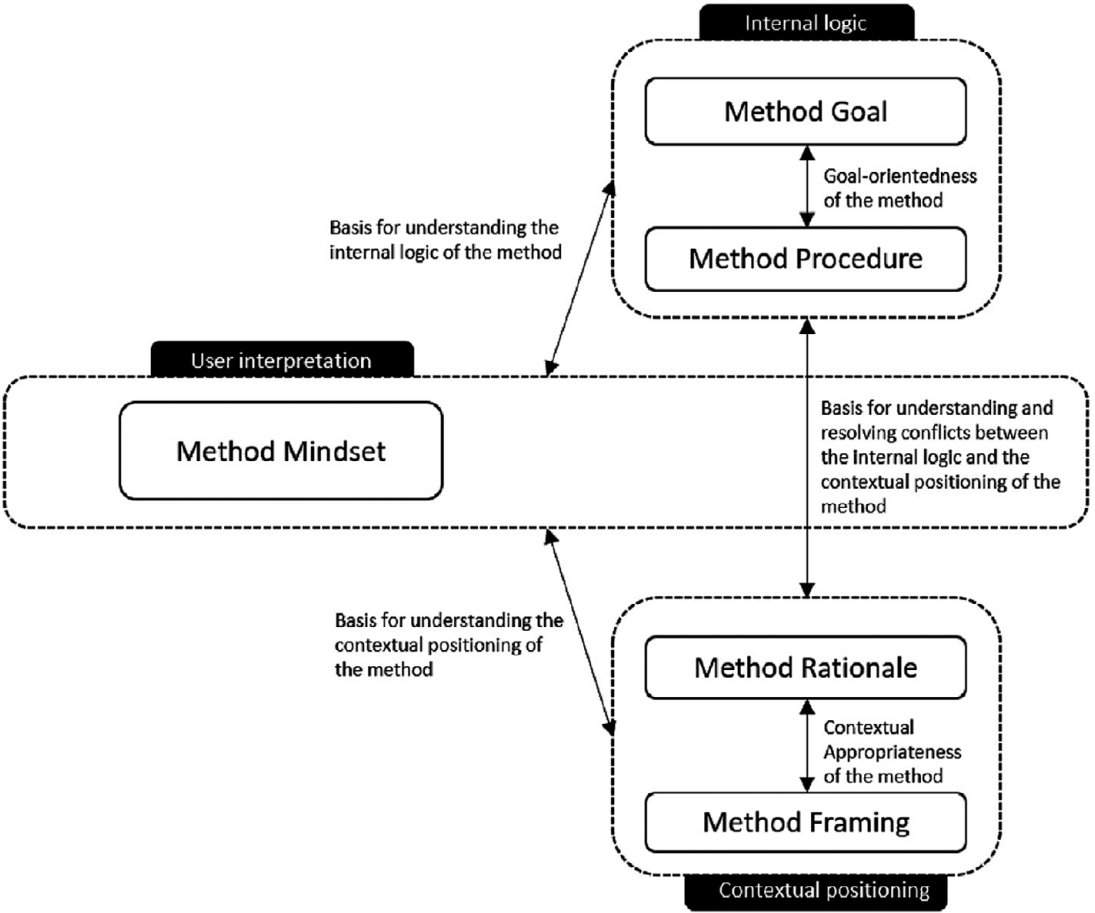


Figure 2. Method content theory (figure reproduced from Daalhuizen & Cash 2021).

use of algorithms to analyse and modify the relationships between the elements, or use dedicated DSM tools.

2.4. Method descriptions are artefacts; methods in use are sociotechnical constructs

Gericke *et al.* (2022) make the point that a description of a method is an artefact, but a design method in use is a sociotechnical construct enacted by the participants in the process, who employ what they know about the design problem and its context as well as their skills and background knowledge. Daalhuizen & Cash (2021) also draw on previous authors including Dorst (2008) and others (Badke-Schaub & Frankenberger 1999; Daalhuizen, Person & Gattol 2014; Gericke, Kramer & Roschuni 2016) to emphasise the importance of a situated conceptualisation of a method connected to the abilities of its users, their goals, and the context of use. Taking a more individual-centric ethnomethodological stance drawn from the work of Garfinkel (1967), Jensen & Andreasen (2010) point out

that in industrial practice, methods are treated flexibly and more as resources than prescriptions, but this depends on how individuals understand the methods – sometimes methods are applied more rigidly than their creators intended. They report that their students working in Danish engineering companies have found that only experienced managers appear to dare to use the flexibility available in the methods they employ.

Figure 3 shows a method in its sociotechnical context. While a comprehensive discussion of the elements of this sociotechnical context is beyond the scope of this article (see De Weck, Roos & Magee 2011), Figure 3 indicates the importance of the experience and expertise of the users of the method and their roles in the teams in which they are operating, as well as the times when a method is used – for design methods, at which points in a development process. This sociotechnical context also comes with a set of terminology and representations that the users bring to the application of the method. Methods are also situated in the context of a sequence of applications. Methods are typically abstractions from example processes, created with the aim of the transferring best practice to similar cases within and outside an organisation. Notable methods are one of the ways in which knowledge is shared between different and even competing organisations – often mediated by researchers.

Participants in development processes need to negotiate a shared understanding of their activities as well as the state of the artefact being designed, in order to work together effectively, and use rhetorical strategies to do so (Minneman 1991; see Minneman & Leifer 1993). The usefulness of a method or the likelihood of benefitting from using a method depends on the context the method operates in, in particular the other methods on whose results it depends; Gericke *et al.* (2020) stress the importance of having an ecosystem of interlocking methods and related representations that share concepts and assumptions. One problem that often limits the adoption of new methods in industry is lack of understanding (coming from lack of explanation in descriptions of the methods) of where the methods should get their inputs from.

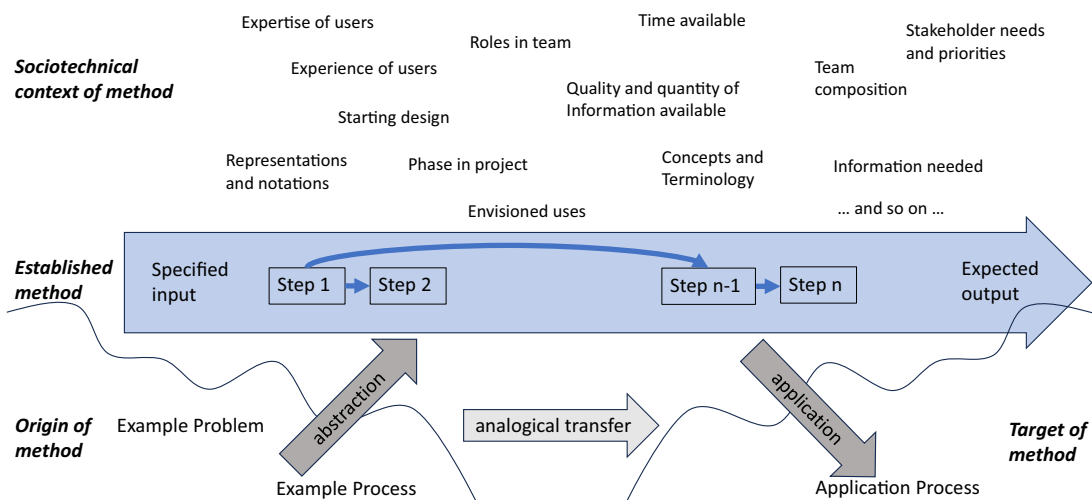


Figure 3. Methods in a sociotechnical and application context.

2.5. Drawing a boundary around a method

Our view is that how one draws a boundary between what a method includes and what is extrinsic to a method depends on one's purpose. Both Gericke *et al.* (2022) and Daalhuizen & Cash (2021) are chiefly concerned with practical purposes in engineering design; they aim to explicate how a method needs to be described so it can be used and evaluated. They take relatively inclusive positions including the concepts that the method involves thinking with, though they take different views on what is or can be intrinsic to a method. Methods serve to organize concepts and knowledge that are part of the method but are not special to it. Identifying and describing all the knowledge that engineers require for applying engineering methods, to decide where the boundary around them really goes, would be infeasible in real cases; Gericke *et al.*, Daalhuizen and Cash, and others concerned with using methods in practice are not concerned with this, rather with what is central enough to need description.

But the rather different aim of separating method knowledge from other forms of knowledge possessed by designers or engineers, and trying to characterise it, suggests taking a much narrower view of what being a method consists of. This is a different question. What is irreducibly central to the 'methodness' of a method is an explicit prescriptive statement of a sequence of steps to be followed to achieve a result. But even this can vary in form and status: the prescribed steps may be characterized in terms of activities to perform or results to be produced, and methods differ in how completely or exactly the prescription needs to be followed.

3. On the subject matter of engineering knowledge

What engineering knowledge is, and whether and how it is distinct from scientific knowledge and from the non-technical knowledge used by everyone, has been considered from a variety of perspectives (see Kant & Kerr 2018, for a review; see also Faulkner 1994; Houkes 2009; Houkes & Meijers 2022). We are not going to survey the field; instead, we will focus on placing methods in the context of engineering knowledge.

Discussion of what engineering knowledge is goes back, of course, to Plato and Aristotle, who considered what different kinds of knowledge there might be. The Greeks distinguished between *episteme* (roughly, scientific knowledge) and *techne* (roughly, craft knowledge), as well as *phronesis* (roughly, practical wisdom), and recognised the importance of the practice aspect of *techne* – knowing how to do things – being grounded in an 'account' – theoretical understanding (see Parry 2020). More recent accounts have focused on whether or not engineering or technological knowledge is distinct from scientific knowledge, and how much of engineering knowledge is grounded in scientific knowledge but is purpose-directed and has content of little interest to scientists, or is distinctively different (see Kant & Kerr 2018; Houkes & Meijers 2022). Most concur with Vincenti (1990) in regarding much of the scientifically grounded factual and theoretical foundations of engineering as belonging to engineering, and rejecting the view that engineering is just applied science. The slogan "technology as applied science" was used by Mario Bunge for the title of an often-cited paper in *Technology and Culture* (Bunge 1966). There, Bunge defended the claim that technological knowledge is divided into applied science and craft knowledge of how to achieve technological results.

Houkes & Meijers (2022) argue for seeing engineers and scientists as engaging in *epistemic activities* to generate knowledge, many of which are shared or only subtly different. The how-to-do-it side of engineering knowledge has attracted relatively little attention from philosophers (but see Houkes 2006; Norström 2011).

Several authors have set out typologies of either engineering knowledge or design knowledge for engineering. These typologies have been influenced both by the authors' purposes and by the branches of engineering they have looked at most closely. The detailed description by Vincenti (1990) comes from an engineer who became an engineering educator. His interest is the knowledge that engineers have or can have; he acknowledges that multiple engineers come together with different knowledge to create a product, but this is not his focus. The knowledge an individual engineer possesses was also looked at by de Vries (2003), but from a philosopher's understanding of very general knowledge classes. Meijers & Kroes (2013) also presented a typology from a philosophical perspective, to consider a specific question: what parts of engineering knowledge fit a justified-true-belief-based view of what constitutes knowledge, and what to do with the parts that don't. Kornwachs (2012) presented a typology of engineering knowledge organized according to the logical properties of different kinds of statements that one can do different kinds of logical operations. Hubka & Eder (1996) came from the engineering design community and were interested in the engineering design knowledge that is required in an engineering design project (see Eder 2008, for an account of the development of Hubka and Eder's thought over a sequence of publications). Faulkner (1994) came from a science and technology studies perspective and saw engineering knowledge in a wider social context including users and research and development; she aimed to extend Vincenti's account to cover research and development in technical fields. The technical system perspective of Ropohl (1997) looks at the knowledge that is required to understand, describe and create a product.

Surveying these typologies, or locating method knowledge within them, is beyond the scope of the present article. In our view, none is entirely satisfactory, and none is a superior alternative to Vincenti for Vincenti's purposes or ours, though they offer interesting insights. Methods are less salient than 21st-century readers might expect, raising the question of whether they fit into these typologies, and if so, how. Only Hubka & Eder (1996) give an explicit account of the place of methods in engineering knowledge (see Eder 2008).

3.1. Vincenti categories

In his 1990 book *What Engineers Know and How They Know It*, the aeronautical engineer and historian of technology Walter G. Vincenti set out to explain engineering to non-engineers in terms of what it is that engineers know *about*, in terms that fit how engineers themselves think about what they know, drawing on examples from mid-20th century aircraft design, that is before the widespread use of computer tools. In Chapter 7, he proposed a typology of engineering knowledge, with six types of knowledge: *fundamental design concepts*, *criteria and specifications*, *theoretical tools*, *quantitative data*, *practical considerations*, and *design instrumentalities*. When explaining what these types are, he mentioned a number of subtypes without highlighting them as types or claiming that they are exhaustive; we list these in Table 1, but this might be giving them more salience than they

Table 1. Vincenti’s (1990) Typology of engineering design knowledge

- **Fundamental design concepts**
 - Operational principles
 - Normal configurations
- **Criteria and specifications**
 - Criteria
 - Specifications
- **Theoretical tools**
 - Mathematical methods and theories
 - Mathematical tools without physical content
 - Mathematically structured physical knowledge
 - Device-specific physical theories
 - Phenomenological theories with ad hoc assumptions
 - Qualitative assumptions used for expedience (which may be known to be wrong but useful)
 - Intellectual concepts (too diverse to describe completely)
 - General
 - Device specific
- **Quantitative data**
 - Descriptive knowledge from measurements
 - Prescriptive knowledge of how things should be to attain a desired end
- **Practical considerations**
 - What follows from how something will be manufactured
 - What follows from how something will be used or maintained
 - Heuristics for design induced by experience
- **Design instrumentalities**
 - Structured procedures
 - Ways of thinking
 - Judgemental skills

deserve. Vincenti also put forward an accompanying typology of engineering knowledge-generating activities, with seven types of activities: *Transfer from science*; *Invention*; *Theoretical engineering research*; *Experimental engineering research*; *Design practice*; *Production*; *Direct trial*.

While Vincenti was philosophically informed, engaging with philosophers’ debates about what knowledge *is* or how different kinds of knowledge have different forms was not the focus of his project, although he *was* concerned with establishing the autonomy of engineering knowledge. Instead, his focus was on the subject matter of engineers’ knowledge, and how it extends a long way beyond the content of scientific knowledge. Of course, engineers also possess and use a lot of scientific knowledge, but Vincenti makes the point that this includes a lot that is peripheral to scientists’ purposes and has been created to serve engineers’ needs. One of Vincenti’s most valuable contributions was to highlight the central importance of knowledge generation by engineers to the engineering enterprise and the amount of effort that goes into it, as well as outline the different knowledge-generating activities engineers engage in. Vincenti acknowledged that his account was biased by his background in aeronautical engineering (from which he takes his examples) and his focus on design to the exclusion of production and maintenance.

3.2. Where are methods in Vincenti's categories?

Engineering practice has changed since the late 1980s. Aeronautical engineering made very extensive use of physical and theoretical models, computer simulations and mathematical calculations, but engineering design is increasingly shaped by standardised computer tools and progressively more sophisticated computational models. More industry best practices and internal company practices and policies are encapsulated in formally defined methods. Particularly in the aerospace industry, regulators mandate the application of particular methods as a way to ensure appropriate quality standards in safety-critical systems. Practice and theoretical concerns also differ between cultures. The development of methods as an important part of engineering research, practice and teaching has a long history in Germany, but Vincenti makes no reference to any work in the German method development tradition. Gerhard Pahl and Wolfgang Beitz' enormously influential textbook *Konstruktionslehre* first appeared in 1977 (Pahl & Beitz 1977); while Ken Wallace's first English translation arrived in 1984, it was little known in the English-speaking world before Nigel Cross published his widely known book *Engineering Design Methods* in 1989 (Cross 1989).

So where are methods in Vincenti's map of what engineers know? As methods form such a large part of the knowledge engineers apply, this question should have an answer, but methods prove surprisingly elusive in Vincenti's scheme. Vincenti (1990, p. 198) discusses how *explicit knowledge* comprises *descriptive knowledge* and *prescriptive knowledge*, and that it thus overlaps with *procedural knowledge* of how to do things, which he divides into *prescriptive knowledge* and *tacit knowledge*. However, prescriptive knowledge does not have a straightforward place in his typology, except as criteria and desirable quantitative values. Criteria and specifications and quantitative data are clearly extrinsic to the methods that generate or make use of them; so too are fundamental design concepts even if some might form an essential part of the concepts a method requires its user to think with. Vincenti's other three categories look obvious places where methods might be found; we argue here that methods do not fit comfortably into any of them.

Theoretical tools. These are the mathematical tools and the physical theories and concepts, for example. Laplace transformation, that engineers use to solve problems. Calculating results by employing physical theories and models and bits of mathematics is an essential part of many methods, but the methods that employ them comprise guidance and structuring of the problem-solving that calls on the theoretical tools – which is extrinsic to the tools themselves. Both Gericke *et al.* (2017, 2022) and Daalhuizen & Cash (2021) stress the centrality to methods of the sets of intellectual concepts (in Vincenti's terminology) that people need to think with to apply the method, but in their views, the problem-structuring aspects of the methods are extrinsic to the concepts needed for reasoning about problems and solutions.

Vincenti appears to assume that the aeronautical engineers he is familiar with use these theoretical tools without needing an explicitly articulated procedure to follow. Given an understanding of the mathematics and physics embodied in the theoretical tools, they are able to apply what they know about solving mathematical problems to reach the results they need. If an explicitly articulated procedure for using the theoretical tool is needed, the procedure would naturally belong to the theoretical tool, so that the combination of conceptual equipment and procedure

would constitute a method. However, this doesn't fit the way the theoretical tools discussed by Vincenti are conceptualized or used. Vincenti is concerned here with techniques for computing exact results as part of analysis or the deterministic, cashing-out-of-decisions part of synthesis. There is a lot more to the use of methods in contemporary engineering.

Practical considerations. For Vincenti, these are considerations of how to design something, that are drawn from experience without being grounded in physical theory or systematic measurements of a phenomenon (which would qualify them as quantitative data). Notable subcategories mentioned by Vincenti include knowing how something will be used or maintained, and what can and cannot be done with the available manufacturing equipment.

Design instrumentalities. This is the least well-defined of the Vincenti categories, which Vincenti (1990) includes “more for the sake of balance than any attempt at the complete coverage.” It is a catch-all category for what engineers know about how to design. But Vincenti doesn't tell us much about what this is. His discussion is strongly focused on cognitive processes involved in thinking about designs and their behaviour. As well as the use of analogies, and visuospatial reasoning, he mentions more-or-less well-known procedures for solving design problems, but the ones mentioned are extremely general cognitive strategies – what cognitive scientists of the 1980s called weak methods, for solving problems when we lack domain-specific problem-solving knowledge (see Laird & Newell 1983). What engineers get told about how to design is missing from Vincenti's account, leaving readers to debate if it's implicitly there or it isn't. Of course, engineers think about how to solve design problems and other types of engineering problems, and employ strategies that they think are likely to produce good results; how they do this has been studied using empirical psychological techniques (see for example Koen 2003; Daly *et al.* 2012; Johnson-Glauch & Herman, 2019). These strategies will vary along two dimensions: from general-purpose and knowledge-free to specific and knowledge-rich; and from situated perceptual recognition of what to do to explicit and articulated rules for how to proceed. At what point do these become methods? In our view, when they are explicit, articulated procedures for how to solve problems, that can be passed on or taught.

Vincenti includes rules of thumb learned from experience under practical considerations: what cognitive scientists call heuristics – in situation A it's a good idea to do B; in situation C aim for parameter value D. Per Norström (2011) argues that “technological know-how from rules of thumb” is an important form of technological knowledge and of a type distinct from scientifically grounded theoretical knowledge or skills. But what Vincenti discusses are heuristics for what the design should be like, not what the development process should be like. One can certainly make the case that methods are grounded in design experience rather than theory, and constitute knowledge of the form ‘in situation E organize the design process into a sequence of stages F’, and ‘in situation G use theoretical tool H with values I’, and are thus codifications of practical experience. Some of what Norström terms rules of thumb are un mistakeable methods; Norström (2011) begins with a good example, about how to adjust a PID controller (proportional, integrating, and derivative controller). However, including methods for how to design, or arrive at particular results, or generate certain kinds of information, under Vincenti's heading of practical considerations seems awkward and misleading. What methods have in common with practical considerations for what designs

should be like is provenance, not what the knowledge is about. We wish to argue here that making sense of what methods are and how they contribute to engineering knowledge requires splitting out a different component of engineering knowledge along a different axis.

From this, we conclude that Vincenti's map of engineering knowledge is incomplete, and needs another province. Methods do not fit into any existing category and require their own.

3.3. Method knowledge as a distinct category of engineering knowledge

So far, we have argued that methods constitute an important part of what engineers are taught and can employ to solve engineering problems, and an important part of the capabilities engineering organizations possess and that can be transferred to other organizations. And we have argued that Vincenti's (1990) map of what of engineering knowledge comprises does not deal adequately with methods. Vincenti put forward an account of the subject matter of engineering knowledge that appears to leave methods out – one can try to stuff methods into his categorization but not in a way that does justice to the characteristics of engineering methods or the roles they play in engineering and design.

Vincenti's (1990) typology remains the most useful for the purpose of explaining what engineers know and how they know it, in terms that make sense to working engineers. However, methods do not fit any of the categories in his typology, so it needs the addition of a separate knowledge category for methods (we agree with Ropohl's (1997) argument that sociotechnical knowledge is also an important type of engineering knowledge that constitutes a distinct category missing from Vincenti's account, that should be added to it, but this is beyond the scope of the present article).

In proposing treating method knowledge as a category in a typology of engineering knowledge, we are skating over the significant distinction between knowledge about how to design (or make, maintain or dispose of) artefacts themselves, and knowledge about how to organize processes. Our view is that both the essential 'methodness' of prescriptive, transferable statements of sequences of steps, and the fuller structure of methods (as discussed by Gericke *et al.* 2017, 2022; Daalhuizen & Cash 2021) are similar in these two broad subcategories of methods. A fuller, multi-dimensional analysis of the space of types of engineering knowledge of the sort put forward by Faulkner (1994) should include this distinction. Of the accounts of the space of engineering knowledge competing with Vincenti's, only Hubka & Eder (1996); see Eder (2008) pay any attention to methods as being part of what engineers know. In Hubka and Eder's more hierarchical account, product versus process is treated as the primary distinction while methods are mentioned in both. Our view is that while this is appropriate to Hubka and Eder's purposes, it tends to obscure the distinctive character of method knowledge.

While we argue for method knowledge as a distinct type of engineering knowledge, and most engineering methods are within the realm of knowledge that is solely the preserve of engineers, we do not claim that method knowledge in engineering is different from method knowledge in other fields. In particular, methods for a wide range of scientific activities have similar characteristics to

methods in engineering, for example how to design different types of controlled experiments in experimental psychology, or how to use various techniques for DNA sequencing. However, to avoid terminological confusion, scientists and philosophers of science will need to be careful to distinguish method knowledge from knowledge of ‘the scientific method.’

4. On method knowledge

Using a method requires the user to know the prescriptive guidance the method provides on how to carry out tasks – method procedure – and the set of concepts required to think in the terms expected by the method, sometimes including notations for representing information, elements of mathematics and physics, and so on. The guidance is both central to the method and inseparable from it, while the other elements of knowledge are separable from the method and often have a long history predating the method. In this section, we will look more closely at the procedure elements of method knowledge.

While we argue that method knowledge is a category of engineering knowledge in its own right, the use of a method depends on the participants involved in enacting the method applying the knowledge needed to understand both the problem and the method, as illustrated in Figure 4. The engineering knowledge that participants bring to the method covers a range of different subject matter, as described by Vincenti (1990), as well as sociotechnical knowledge (as discussed for instance by Ropohl 1997), and knowledge of this and other methods. While psychologists disagree about how many different types of memories and mental representations humans possess, that can encode knowledge, mental representations can combine conceptual/propositional information, visuospatial information (which is inherently viewpoint-specific), and episodic information (which is inherently time-dependent). Applying this knowledge requires thinking skills (know-how) and the strategies and heuristics that Vincenti puts under design instrumentalities.

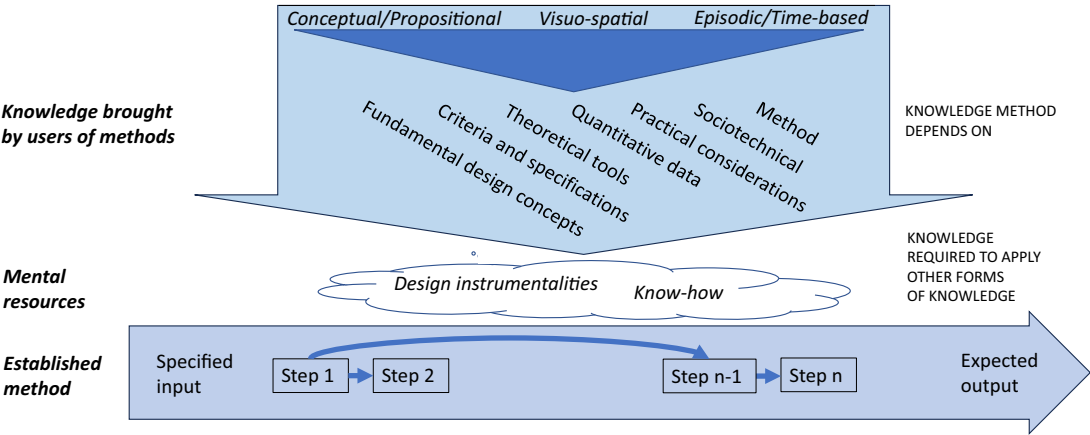


Figure 4. Knowledge brought to the use of methods. This figure adopts Vincenti’s typology of engineering knowledge with Ropohl’s (1997) addition of sociotechnical knowledge and our own addition of method knowledge.

4.1. What method of knowledge might be

We can look at how method knowledge constitutes knowledge in different ways:

- **Knowledge of the method:** A method can be viewed as the possession of explicit beliefs about the content of the method. Most centrally, knowledge of the content of explicit prescriptive specifications of sequences of activities (or, in some cases, the results that should be achieved by activities). However, using such knowledge of sequences of activities requires knowledge of the concepts and theoretical tools that the method requires its user to think with, and in some cases the representations it requires them to read or to employ to express information. This conceptual knowledge is required to understand what a specification of a method means, as well as to be able to use it. It is part of the content of the method but does not inherently belong to the method. At the low extreme, an engineer's knowledge of a method might consist of knowing that it exists and has a particular purpose, so that the engineer is able to find it when it becomes relevant. Of course, using a method requires action skills as well as the possession of information. The limitation of this explicit belief view of method knowledge is that it excludes these action skills. This is something we can accept as long as we are concerned with descriptions of methods and how these are employed, but it is inadequate when we are concerned with the performance of processes using methods.
- **The capacity to perform actions and achieve results:** Applying a method can give an agent the ability to do tasks that would be impossible without the method, or would be harder or less efficient or less reliable. So we can see knowledge of the method as being what confers this ability. Separating out the contribution to this of the explicit knowledge of a procedure to follow from the action skills involved in applying the procedure or from the knowledge of concepts and tools the method employs looks infeasible in real cases.
- **Knowledge that is embedded in a method:** The creators of a method embed knowledge in it of how to achieve certain results; this is embodied in descriptions of a method. This is harder to characterise, but we can look at it in terms of what actions it enables an actor to perform, which would either be impossible without it, or would require additional knowledge or additional effort, or would be done less well or less reliably. Exploring this is beyond the scope of this article. Descriptions of methods, as well as computer tools for helping people follow the methods, may enable their users to achieve results without possessing the knowledge required to do so in any brain-internal cognitive sense.

4.2. Know-how, know-that

Engineers' knowledge of how to create and use technology is often referred to as 'know-how'. Engineering methods are descriptions of how to create and use technology (and analyse its properties, maintain it, and so on). But 'know-how' is another term that causes terminological confusion.

Much modern philosophical discussion of knowing how to do things has been shaped by the work of Ryle (1946, 1949), who pointed out that our ability to operate in the world depends at least as much on our learned skills for carrying out actions as on our knowledge of information. Ryle distinguished between 'know-how', meaning learned skills for performing mental and physical actions, and 'know-that', meaning knowledge of information that can be recalled and articulated. Ryle

argued that know-how cannot be reduced to know-that, and that both are within the province of epistemology (see also Löwenstein 2017).

What Ryle and followers would term ‘know-that’ is very different from the ability to perform actions. The possession of factual information about states of affairs (whether or not it qualifies as knowledge) seems to be clearly distinct from knowing how to carry out actions to achieve results. The possession of explicitly described information is distinct from the mental ability to recognise situations, reason about them and perform actions. Knowledge that one is consciously aware of and can describe is different from the experientially learned perceptual pattern recognition and motor skills that are customarily termed tacit knowledge. These dichotomies are not quite the same, so we need to examine the nature of the prescriptive procedure knowledge at the heart of engineering methods. Most philosophers view know-that as comprising beliefs; many adopt some (more sophisticated) version of the view that knowledge is justified true belief. However, according to Kremer (2016), Ryle himself saw ‘know-that’ as something different from belief, and had a unified view of know-how and know-that as constituting a ‘capacity to get things right’.

Many everyday uses of the phrase ‘know-how’ (in engineering and other contexts) take it to mean what we know about how to solve problems and achieve results, especially using technology. This covers explicit, articulatable propositional knowledge of good ways to do things, drawn from observations of one’s own experience or from others’ testimony. It also includes factual knowledge of assertions about how to perform tasks, including knowledge of the procedures specified by methods. In this article, we avoid this usage. We adopt Rylean terminology, and use the term ‘know-how’ to refer to learned mental skills for recognising things and situations, reasoning, performing actions and so on.

Houkes (2006, p. 46) makes a useful distinction between ‘procedural knowledge’ that a sequence of actions leads to the realization of a goal, and ‘operational knowledge’ – the skills needed to take these actions. He terms both ‘use know-how’ (Note that this use of ‘procedural’ corresponds to normal English usage but not with the use of ‘procedural knowledge’ in cognitive science, to refer to knowledge forming or embedded in a person or intelligent agent’s reasoning processes, in contrast to ‘declarative knowledge’ which is independent of situation-specific reasoning mechanisms). The core of a method is explicit, stated procedural knowledge, in Houkes’ terminology (Whether it counts as know-how depends on how the term is defined; for us, it is know-that, so not know-how). Using a method involves applying a lot of know-how with reference to know-that of the procedures specified by the method and other explicitly articulated information. We prefer the term ‘how-to knowledge’ for this explicit knowledge of how to achieve a result.

One complicating factor in making sense of method knowledge is that what is in a method description, what is explicit, articulated how-to knowledge, and what is situated, tacit awareness of how to proceed is fluid and subject to change. As how-to knowledge is used, it becomes proceduralized into learned action rules that are triggered in appropriate situations, often with an increasingly loose and hazy association with the explicit knowledge they are grounded in. The mechanisms through which this happens and their consequences for human behaviour have been extensively studied by cognitive psychologists (see Anderson 1983, 1993; Clancey 1997). Studies of working engineers have found them applying elements of

methods they have been taught without thinking they were using a specific method (Gericke *et al.* 2016).

Polanyi (1966) and many writers since have stressed that a lot of knowledge is ‘tacit’ – our abilities to perceptually recognise objects and situations, reason, and perform actions are learned perceptually from our experiences, and we only have a very limited and partial understanding of what our abilities are and how they work. Houkes (2009, pp. 331–333) points out that ‘tacit knowledge’ is another tricky term that is a source of ambiguity and confusion in discussions of technological knowledge. It is used in three distinct senses to make three different distinctions. We prefer to reserve the term ‘tacit knowledge’ for the first of these meanings. (1) A psychological distinction between knowledge about things that we have accurate mental representations of that we are aware of, versus ‘tacit’ knowledge that we can use without fully understanding what we know or how we know it, and that may only be available to us in particular situations. (2) Ryle’s distinction between skills and information. (3) A social-epistemic distinction between knowledge that can be transferred verbally, versus ‘tacit’ knowledge that needs to be acquired through experience, which Houkes prefers to term ‘verbal’ versus ‘non-verbal’.

The first of these is important for understanding the nature of engineering problem solving and design, thus how methods are used in practice, and what elements of problem-solving we can document directly and turn into methods versus what goes beyond the practitioners’ own ability to analyse it. The second is important for understanding the relationship between know-that and action. The third is critical for understanding what verbal knowledge can be written or needs to be written in descriptions of methods, and what non-verbal knowledge needs to be passed on in a different way.

4.3. Method knowledge is prescriptive

A lot of what engineers know is about what artefacts (that may not yet exist) *should* be like or what they should do, as opposed to what existing artefacts are actually like. This is well-recognised by engineers. For instance, Vincenti (1990) includes information about what values for design parameters or material or performance properties should be in the category of quantitative data (see Section 4.1).

The specifications of procedures at the heart of methods are necessarily prescriptive. Meijers & Kroes (2013) consider which aspects of engineering knowledge are amenable to a justified-true-belief-based view of what constitutes knowledge, and argue that prescriptions of the sort we are concerned with here are epistemologically different from knowledge of facts, and need to be *accepted* by the user rather than *believed*, though they neglect the importance of factual knowledge of what the prescriptions are.

Houkes (2009) points out that a lot of technological knowledge that engineers take as guidance can be expressed in descriptive forms (using parameter values A and B will have consequence X) or prescriptive forms (to achieve result X, use parameter values A and B), with very little effort required to make the mapping between the two, and little additional conceptual equipment required to take knowledge expressed in descriptive form as statements of how things are and use it for prescriptive purposes.

While methods necessarily have prescriptive elements, the boundary between the prescriptive specification of a procedure and the conceptual and theoretical

tools it employs might be hard to find. As we noted in Section 3.2, engineers often use theoretical equipment provided by mathematics and physics, when obtaining the results they need involves following sequences of steps; but the steps are just assumed to follow from the desired endpoint, and the structure of the mathematics and the user's competence in using it. Vincenti treats mathematical and physical models and conceptual equipment for arriving at results as a distinct category of knowledge he terms 'theoretical tools', but skips over the procedural knowledge and skills involved in using them. Is there a method, if the method can be constructed on the fly or is implicit in the expert's actions, and can be made explicit as and when needed to teach novices? Our answer is no: we are concerned with the nature of the explicit, transmissible how-to knowledge that is codified and exchanged in engineering practice.

4.4. Distinct characteristics of method knowledge

This section focuses on the knowledge captured in engineering design methods as outlined in Section 2, which are generated by industry or academic experts to share successful ways to address particular problems with colleagues, other practitioners or students. This knowledge encompasses more than just the statement of a procedure.

While engineering methods are too diverse to admit an exact characterization of method knowledge, we observe that method knowledge typically has a number of characteristics that make it distinct from other classes of knowledge.

Method knowledge is relational. The prescriptive, activity-sequence-specifying components of method knowledge connect other forms of knowledge, the elements of what Gericke *et al.* (2022) term the core idea of the method (see Figure 4). Methods comprise knowledge of inferences to be made from other knowledge and information. In some cases, they have specific factual knowledge built into the method; more typically they have, implicitly, slots to be filled in with particular types of knowledge and information that is dependent on the individual problem. This knowledge and information may in turn be generated by other methods. What using a method should achieve is that a result is generated that has a particular form, and has particular properties. In most cases, the properties required from the outputs of engineering methods are that they contain certain kinds of information, and often the information should be expressed in particular representations and meet certain criteria to do with its relationships to other information. This is often needed to ensure that the information needed by later activities appears and meets the required quality standards. While such methods span the range from detailed specification of reasoning steps to assertions that particular activities should produce information of a particular kind without saying what form it should take, they have this in common. One way to think about knowledge of methods for creating information is that it is a form of information transformation knowledge, comprising mappings from inputs to outputs, as the steps of the methods usually require particular types of information as inputs and produce particular types of information as outputs.

Method knowledge is abstract. Method knowledge is generated through a process of abstraction from one or more processes. It is generated through reflection, as steps are made explicit and the specificity of the context is stripped away. Steps that might be tacit in a practical process need to be spelled out and

made explicit. As methods are abstractions from instances or encapsulations of best practice, there is often an implicit scope of the method. How-to knowledge encoded in methods can be transferred to new problems at different levels of abstraction and to varying extents. The elements of the problem need to be matched to concepts used in the method by fitting specific instances into general categories. While, in principle, this is a recognition of category membership, it may be a recognition of similarity or analogy. If the analogies are too distant, the elements of the method can be matched to elements of problems that are outside the range of the category concepts. Whether this is ‘wrong’ can only be judged by whether the method produces useful results. If a method is applied in a similar context to the one originally intended, then most of the method knowledge contained in the method can be deployed, and with relatively close mappings between the problem and the concepts of the method. However in other situations, only a small fraction might be useful, as in the case of the company applying Six Sigma to communication where only the fact that something is a measurement is related to the original Six Sigma method. However, what methods do provide is a way of thinking about problems. They introduce a vocabulary and propose a way of chunking up knowledge that might be applicable in different contexts, as well as a set of concepts. This way engineers can import what Daalhuizen & Cash (2021) call the method mind set, and Gericke *et al.* (2017, 2022) see as the core idea of the method. This is evident when engineers trained at different universities talk about problems with the terminology used in the methods developed at their home institutions; this can occasionally cause problems around the understanding of key concepts (Vermaas and Eckert, 2013; Eckert 2013; Vermaas 2013).

Methods provide reusable chunks of how-to knowledge. As well as providing ways to perform engineering activities, methods serve to index and organize knowledge about how to perform them, that engineers can use to reason about how to fit these activities into larger-scale processes as well as how to customize them to meet the needs of particular problems. Referring to methods or elements of methods can serve as a shorthand for complicated clusters of activities or reasoning steps.

Method knowledge is (partly) sociotechnical. A method in use is a socio-technical construct, that involves the people applying it and their social context, as well as the problem they are working on, the tools they use, and the environments they are working in Gericke *et al.* (2022). Method knowledge depends on its application both on the individual enacting the process and the context in which it operates. While many methods are for single-person activities and focus entirely on using technical information to generate technical results, others require collaboration or are there to coordinate large-scale group activities. If multiple people use a method together, they need to negotiate an understanding of what the key concepts mean and how the steps need to be interpreted. At the same time, method knowledge sometimes includes knowledge of the social roles of the participants in processes they specify or support, and the skills they require, as well as how they should interact (cf. Jensen & Andreasen 2010).

Method knowledge is interpersonal. While many philosophers tend to think of knowledge as cognitive, and as individual, this is not how engineers and especially their managers see the world. What matters is the knowledge that the team or the company can deploy (see Stacey & Eckert 2022, for some thoughts on how engineering knowledge needs to be seen as collective for some purposes and

individual for others). Methods exist to transfer good practice between individuals or organizations, and some are used to coordinate the work of groups. Methods thus exist in a public space of shared, interpersonal knowledge. In many situations, group members having a sufficient shared understanding of a method to coordinate their activities is crucial to its success; the members of the group having a shared understanding may be more important than whether that understanding conforms to any external criterion of correctness. (Of course, the group needs to have a method that works; trying to adopt a method whose basic principles you haven't understood will lead to failure). The questions of how groups coordinate what their members believe, and what it is for a group to have a belief have, of course, been explored in the field of social epistemology (see O'Connor, Goldberg & Goldman 2024). Which models of group belief best-fit method knowledge (and when) is a subject for further investigation. Gilbert's (1989) view of group belief involving a shared commitment is one way forward; Bird (2014, 2022) argues that this is one of a number of legitimate models of group belief, pointing out that complex tasks involve the coordination of information possessed by different members of a team who do not possess all the beliefs of the group.

5. Why method knowledge matters

In this article, we have pointed out what modern engineers are well aware of – that methods for how to do a wide range of activities, from generating ideas to performing exact analyses of the properties of designs to organising design processes, are an important part of what engineering students are taught and a vitally important part of the knowledge engineers use in their professional work. Despite this, methods – explicit, articulated accounts of how to tackle particular types of problems – are often neglected in discussions of what it is that engineers know.

Vincenti's (1990) book *What Engineers Know and How They Know It* is a foundational work in engineering epistemology and is still in our view the best starting point for doing what it sets out to do: explaining, especially to non-engineers, what engineers know and how they know it, in terms that make sense to working engineers. Understanding engineering knowledge is important for explaining the enterprise of engineering to people who don't have, or don't yet have, technical education. This is increasingly important in a society where people live lives that are completely shaped by the work of engineers but have little understanding of what that work is and what they can expect from it, often failing to understand the difference between engineering and science.

The book presents a typology of engineering knowledge, with six categories: *fundamental design concepts, criteria and specifications, theoretical tools, quantitative data, practical considerations, and design instrumentalities*. It presents an accompanying typology of seven categories of knowledge-generating activity: *Transfer from science; Invention; Theoretical engineering research; Experimental engineering research; Design practice; Production and Direct trial*. Meijers & Kroes (2013) describe Vincenti's approach well: "Vincenti's taxonomy is set up typically from an engineer's perspective. He describes the knowledge toolbox available to engineers when they try to solve design problems. It is thus a more process-oriented taxonomy. He is not so much interested in the nature of the instruments

in the engineer's knowledge toolbox as in what instrument to use for solving what kind of engineering problem when designing and making technical artefacts."

Where methods fit into Vincenti's scheme is not obvious; we conclude that they don't. We conclude further that Vincenti's typology needs to be extended to give methods their own category as a distinct and important type of engineering knowledge.

The methods used in engineering and the kinds of results they produce are diverse. They include ways to use mathematics and physics (in what Vincenti terms 'theoretical tools') for computing exact or more approximate results. But methods also include explicit statements of heuristic strategies for generating ideas and producing other kinds of results that depend on the knowledge and skill of the user as well as on the situation, and that are not guaranteed to be correct and useful. This distinction is underemphasised in many discussions of methods. Another key distinction is between methods for generating ideas, obtaining results or making decisions about the artefacts themselves, and methods for analysing or managing human activities (one made by Hubka & Eder 1996).

While the content of method knowledge might be fluid, making chunks of how-to knowledge explicit and open to examination and criticism is an important part of making experience reusable by others in a way that is not achieved by capturing and recording other knowledge types. Methods therefore provide a degree of separation between the engineers and the results of the engineering process. They deemphasise the skills of the individual in favour of highlighting the systematic and rational nature of engineering. This goes contrary to a prevailing narrative of individual creativity, which wants to single out the genius creator. The likes of Nicola Tesla are not the norm, but the exceptions that camouflage the hard and systematic work of hundreds of engineers.

Methods provide predictability. Applying a mathematical procedure or physical model (a theoretical tool in Vincenti's terminology) to compute an exact result should guarantee a result that is entirely determined by the input information. By contrast, most methods produce results that depend on the knowledge, skills and immediate concerns of the engineers using them, as well as on decisions about how much of the method to use or adapt. In practice, engineers differ in how they interpret methods (see Jensen & Andreasen 2010). Method knowledge does not guarantee a particular answer or even any usable answer. Nonetheless, the application of a method can be used to try to ensure that results of a particular form get produced, and that particular sources of information are considered and certain reasoning steps are followed. This is highly important for planning and managing complex product development processes, both where considering certain issues is required, sometimes by standards and regulations, and where choosing an appropriate way to structure and organize activities is needed to achieve an efficient process.

Methods are a way of sharing how-to knowledge across groups of people and also across time. While companies study each other's products once released in the market and carefully guard their own evolving designs, they are often much more willing to share methodological understanding, especially with companies they do not compete with. Creating methods can be advantageous to companies as they can enhance their reputations and attract skilled workers, as has been the case with Toyota.

Methods and design approaches are subject to fashions, which periodically change. As these fashions change, companies retain useful parts of methods but collectively forget much of the additional apparatus associated with methods. What is useful to them is not necessarily the core idea of a method or the entire procedure it specifies, but can be an incidental element of a method.

The knowledge of how a particular process was carried out, and what the lessons learned from it were, is very often lost, as it depends on the memories of individuals with only a small fraction being captured in documentation. As teams reconfigure and individuals move on, much of this how-to knowledge of complex engineering processes is lost. So it is often down to researchers to capture some of the best practice in the form of methods to share them. This is illustrated by the huge impact the studies of Harvard academics on lean engineering in the Japanese car industry had in the 1990s.

Method knowledge is an important part of engineering education, as young engineers need to learn how to apply factual engineering knowledge in concrete problems. Engineering educators need to teach their students how to frame problems so that they can apply their knowledge to these problems. Methods provide the terminology and more or less elaborate indications of the sequence of activities. In the course of a degree, many have to learn in the first place that the discipline of systematic working in engineering is what delivers reliable results. Much of this will not be presented to them as explicit methods but constitute a form of method knowledge. How well engineers and educators recognise methods as significant varies. However, wider recognition of the nature of method knowledge should foster consideration of how best to include method knowledge in education and on-the-job training, and how to develop and use method knowledge in industrial practice.

The study of engineering knowledge is a small but vigorous branch of epistemology (see Kant & Kerr 2018, for a review). Meanwhile, design studies is an autonomous academic discipline that not only studies design processes, including in engineering, but aims to improve them. The development of methods, both for computing analytical results and for generating ideas and design solutions as well as for organizing development processes, is an important part of this. So far these developments have seldom been connected (an exception is the work of Hubka & Eder 1996; see Eder 2008). Consideration of the nature of method knowledge and its relationship to other forms of engineering knowledge should help studies of engineering epistemology adopt a fuller view of what engineering knowledge involves that can draw on a large body of empirical research looking at the problems designers face in design processes as well as at what designers actually do.

At the same time, more informed consideration of the nature of method knowledge should improve both the development and the introduction and teaching of methods and tools to support designers and other engineers to develop, manufacture, maintain and dispose of products in a more effective, efficient and increasingly more ethically informed way.

Acknowledgements

Discussions leading to this article began at the 2020 Forum on Philosophy, Engineering and Technology (fPET), which led to the authors creating an engineering knowledge tagup group. We appreciate past member contributions from

Nina Jirouskova and Rémi Gandoin. For Pirtle, he notes that opinions expressed in this article belong to the authors and do not necessarily represent his employer. Eckert and Stacey have gained enormously from their discussions of engineering methods with Prof Kilian Gericke of the Universität Rostock.

References

- Anderson, J. R.** 1983 *The Architecture of Cognition*. Harvard University Press.
- Anderson, J. R.** 1993 *Rules of the Mind*. Lawrence Erlbaum Associates.
- Andreasen, M. M.** 2003 Improving design methods usability by a mindset approach. In *Human Behaviour in Design* (ed. U. Lindemann), pp. 209–218. Springer.
- Badke-Schaub, P. & Frankenberger, E.** 1999 Analysis of design projects. *Design Studies* **20**, 465–480. doi:[10.1016/S0142-694X\(99\)00017-4](https://doi.org/10.1016/S0142-694X(99)00017-4)
- Bird, A.** 2014 When is there a group that knows? In *Essays in Collective Epistemology* (ed. J. Lackey), pp. 42–63. Oxford University Press. doi:[10.1093/acprof:oso/9780199665792.003.0003](https://doi.org/10.1093/acprof:oso/9780199665792.003.0003)
- Bird, A.** 2022 *Knowing Science*. Oxford University Press.
- Bunge, M.** 1966 Technology as applied science. *Technology and Culture* **7**, 329–347.
- Clancey, W. J.** 1997 *Situated Cognition*. Cambridge University Press.
- Cross, N. G.** 1989 *Engineering Design Methods*. Wiley. (Current Edition: Cross, N. G. 2021 *Engineering Design Methods*, 5th edn. Wiley).
- Daalhuizen, J. & Cash, P.** 2021 Method content theory: towards a new understanding of methods in design. *Design Studies* **75**, 101018. doi:[10.1016/j.destud.2021.101018](https://doi.org/10.1016/j.destud.2021.101018)
- Daalhuizen, J., Person, O. & Gattol, V.** 2014 A personal matter? An investigation of students' design process experiences when using a heuristic or a systematic method. *Design Studies* **35**, 133–159. doi:[10.1016/j.destud.2013.10.004](https://doi.org/10.1016/j.destud.2013.10.004)
- Daalhuizen, J., Timmer, R., Van Der Welie, M. & Gardien, P.** 2019 An architecture of design doing: a framework for capturing the ever-evolving practice of design to drive organizational learning. *International Journal of Design* **13**(1), 37–52. <http://www.ijdesign.org/index.php/IJDesign/article/view/2814/847>
- Daly, S. R., Yilmaz, S., Christian, J. L., Seifert, C. M. & Gonzalez, R.** 2012 Design heuristics in engineering concept generation. *Journal of Engineering Education* **101**, 601–629.
- De Weck, O. L., Roos, D. & Magee, C. L.** 2011 *Engineering Systems: Meeting Human Needs in a Complex Technological World*. MIT Press.
- Dorst, K.** 2008 Design research: A revolution-waiting-to-happen. *Design Studies* **29**(1), 4–11. doi:[10.1016/j.destud.2007.12.001](https://doi.org/10.1016/j.destud.2007.12.001).
- Eckert C. M.** 2013 That which is not form: The practical challenges in using functional concepts in design. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **27**(3), 217–231. doi:[10.1017/S089006041300022X](https://doi.org/10.1017/S089006041300022X)
- Eckert, C. M. & Stacey, M. K.** 2010 What is a process model? Reflections on the epistemology of design process models. In *Modelling and Management of Engineering Processes* (ed. P. Heisig, P. J. Clarkson & S. Vajna), pp 3–14. Springer.
- Eder, W. E.** 2008 Theory of technical systems and engineering design science – legacy of Vladimir Hubka. In *Proceedings of Design 2008*, pp. 19–30. Design Society.
- Eisenmann, M., Graubeger, P., Üreten, S., Krause, D. & Matthiesen, S.** 2021 Design method validation—an investigation of the current practice in design research. *Journal of Engineering Design* **32**(11), 621–645.

- Eppinger, S. D. & Browning, T. R.** 2012 *Design Structure Matrix Methods and Applications*. MIT Press.
- Faulkner, W.** 1994 Conceptualising knowledge used in innovation: a second look at the science-technology distinction. *Science, Technology, & Human Values* **19**(4), 425–458.
- Garfinkel, H.** 1967 *Studies in Ethnomethodology*. Prentice-Hall.
- Gericke, K., Eckert, C. M., Campean, F., Clarkson, P. J., Flening, E., Isaksson, O., Kipouros, T., Kokkolaras, M., Köhler, C., Panarotto, M. & Wilmsen, M.** 2020 Supporting designers: moving from method menagerie to method ecosystem. *Design Science* **6**, e21.
- Gericke, K., Eckert, C. M. & Stacey, M. K.** 2017 What do we need to say about a design method? In *Proceedings of the 21st International Conference on Engineering Design, ICED 17*. The Design Society.
- Gericke, K., Eckert, C. M. & Stacey, M. K.** 2022 Elements of a design method – a basis for describing and evaluating design methods. *Design Science* **8**, e29.
- Gericke, K., Kramer, J. & Roschuni, C.** 2016 An exploratory study of the discovery and selection of design methods in practice. *Journal of Mechanical Design* **138** (10), 101109. doi:[10.1115/1.4034088](https://doi.org/10.1115/1.4034088)
- Gilbert, M.** 1989 *On Social Facts*. Routledge.
- Houkes, W. N.** 2006 Knowledge of artefact functions. *Studies in History and Philosophy of Science Part A* **37**, 102–113.
- Houkes, W. N.** 2009 The nature of technological knowledge. In *Philosophy of Technology and Engineering Sciences* (ed. A. Meijers), pp. 309–350. Elsevier. doi:[10.1016/B978-0444-51667-1.50016-1](https://doi.org/10.1016/B978-0444-51667-1.50016-1)
- Houkes, W. N.** 2013 Rules, plans and the normativity of technological knowledge. In *Norms in Technology* (ed. M. J. de Vries, S. O. Hansson & A. W. M. Meijers), pp. 35–54. Springer Netherlands. doi:[10.1007/978-94-007-5243-6_3](https://doi.org/10.1007/978-94-007-5243-6_3)
- Houkes, W.N. & Meijers, A.W.M.** 2022 Engineering Knowledge. In *The Oxford Handbook of the Philosophy of Technology* (ed. S. Vallor), pp. 128–151. Oxford University Press. doi:[10.1093/oxfordhb/9780190851187.013.10](https://doi.org/10.1093/oxfordhb/9780190851187.013.10)
- Hubka, V. & Eder, W. E.** 1996 *Design Science: Introduction to the Needs, Scope and Organization of Engineering Design Knowledge*. Springer.
- Jensen, T. E. & Andreassen, M. M.** 2010 Design methods in practice – beyond the ‘systematic’ approach of Pahl & Beitz. In *Proceedings of DESIGN 2010*. Design Society.
- Johnson-Glauch, N. & Herman, G. L.** 2019 Engineering representations guide student problem-solving in statics. *Journal of Engineering Education* **108**, 220–247.
- Kant, V. & Kerr, E.** 2018 Taking stock of engineering epistemology: multidisciplinary perspectives. *Philosophy & Technology* **32** (4), 685–726. doi:[10.1007/s13347-018-0331-5](https://doi.org/10.1007/s13347-018-0331-5).
- Koen, B.V.** 2003 *Discussion of the Method: Conducting the Engineer’s Approach to Problem Solving*. Oxford University Press.
- Kornwachs, K.** 2012 *Strukturen Technologischen Wissens*. Edition Sigma.
- Kremer, M. J.** 2016 A capacity to get things right: gilbert ryle on knowledge. *European Journal of Philosophy* **25**, 25–46. doi:[10.1111/ejop.12150](https://doi.org/10.1111/ejop.12150)
- Laird, J. E. & Newell, A.** 1983. A universal weak method: summary of results. In *Proceedings of the Eighth international Joint Conference on Artificial Intelligence – Volume 2 (IJCAI’83)*, pp. 771–773. Morgan Kaufmann
- Löwenstein, D.** 2017 *Know-how as Competence*. Klostermann.

- Meijers, A. W. M. & Kroes, P. A. 2013 extending the scope of the theory of knowledge. In *Norms in Technology* (ed. M. J. de Vries, S. O. Hansson & A. W. M. Meijers). Springer. doi:[10.1007/978-94-007-5243-6_2](https://doi.org/10.1007/978-94-007-5243-6_2)
- Minneman, S. L. 1991 *The Social Construction of a Technical Reality: Empirical Studies of Group Engineering Design Practice*. PhD Thesis, Department of Mechanical Engineering, Stanford University, Stanford, CA. Xerox Palo Alto Research Center Report SSL-91-22.
- Minneman, S. L. & Leifer, L. 1993 Group engineering design practice: the social construction of a technical reality. In *Proceedings of the 9th International Conference on Engineering Design, The Hague*, pp. 301–310. Heurista.
- Norström, P. 2011 Technological know-how from rules of thumb. *Techné* **15**, 96–109.
- O'Connor, C., Goldberg, S. & Goldman, A. 2024 Social epistemology. In *The Stanford Encyclopedia of Philosophy* (Summer 2024 edn.) (ed. E. N. Zalta & U. Nodelman). Stanford University <https://plato.stanford.edu/archives/sum2024/entries/epistemology-social>
- Pahl, G. & Beitz, W. 1977 *Konstruktionslehre*. Springer. (Current edition: Bender, B. & Gericke, K., eds., *Pahl/Beitz Konstruktionslehre*, 9th edn. Berlin: Springer, 2021. English translation: Pahl, G. & Beitz, W., *Engineering Design*. K. Wallace, ed., London: Design Council, 1984. Most recent English edition: Pahl, G., Beitz, W., Feldhusen, J. & Grote, K.-H. 2007. *Engineering Design*, 3rd edn., K. Wallace & L.T.M. Blessing, eds., London: Springer London. doi:[10.1007/978-1-84628-319-2](https://doi.org/10.1007/978-1-84628-319-2).)
- Parry, R. 2020 Episteme and Techne. In *The Stanford Encyclopedia of Philosophy* (Spring 2024 edn.) (ed. E. N. Zalta & U. Nodelman). <https://plato.stanford.edu/entries/episteme-techne/>
- Polanyi, M. 1966 *The Tacit Dimension*. Doubleday.
- Ropohl, G. 1997 Knowledge types in technology. *International Journal of Technology and Design Education* **7**, 65–72. doi:[10.1023/A:1008865104461](https://doi.org/10.1023/A:1008865104461)
- Ryle, G. 1946 Knowing how and knowing that. *Proceedings of the Aristotelian Society* **56**, 212–225.
- Ryle, G. 1949 *The Concept of Mind*. Hutchinson.
- Stacey, M. K. & Eckert, C. M. 2022 Objects as carriers of engineering knowledge. *Engineering Studies* **14**(2), 87–108. doi:[10.1080/19378629.2022.2121213](https://doi.org/10.1080/19378629.2022.2121213)
- Stacey, M. K., Eckert, C. M. & Hillerbrand, R. 2020 Process models: plans, predictions, proclamations or prophecies? *Research in Engineering Design* **31**, 83–102.
- Vermaas, P. E. 2013 The coexistence of engineering meanings of function: Four responses and their methodological implications. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **27**(3), 191–202. doi:[10.1017/S0890060413000206](https://doi.org/10.1017/S0890060413000206).
- Vermaas, P. E. & Eckert, C. M. 2013 My functional description is better!. *Artificial Intelligence for Engineering Design Analysis and Manufacturing* **27**(3), 187–190. doi:[10.1017/S089006041300019X](https://doi.org/10.1017/S089006041300019X)
- Vincenti, W. G. 1990 *What Engineers Know and How They Know It*. Johns Hopkins University Press.
- Vries, M. J. de. 2003 The nature of technological knowledge: extending empirically informed studies into what engineers know, *Techné* **6**(3), 117–130.
- Wynn, D. C. & Clarkson, P. J. 2018 Process models in design and development. *Research in Engineering Design* **29**, 161–202. doi:[10.1007/s00163-017-0262-7](https://doi.org/10.1007/s00163-017-0262-7)
- Zwart, S. 2021 Prescriptive engineering knowledge. In *The Routledge Handbook of the Philosophy of Engineering* (ed. D. P. Michelfelder & N. Doorne), pp. 111–126. Routledge.