

The Logic of Design as a Conceptual Logic of Information

Luciano Floridi^{1,2}

Abstract In this article, I outline a logic of design of a system as a specific kind of conceptual logic of the design of the model of a system, that is, the blueprint that provides information about the system to be created. In section two, I introduce the method of levels of abstraction as a modelling tool borrowed from computer science. In section three, I use this method to clarify two main conceptual logics of information (i.e., modelling systems) inherited from modernity: Kant’s *transcendental logic of conditions of possibility* of a system, and Hegel’s *dialectical logic of conditions of in/stability* of a system. Both conceptual logics of information analyse structural properties of given systems. Strictly speaking, neither is a conceptual logic of information about (or modelling) the *conditions of feasibility* of a system, that is, neither is a logic of information as a logic of design. So, in section four, I outline this third conceptual logic of information and then interpret the conceptual logic of design as a logic of *requirements*, by introducing the relation of “sufficiency”. In the conclusion, I argue that the logic of requirements is exactly what we need in order to make sense of, and buttress, a constructionist (poietic) approach to knowledge.

Keywords Conceptual logic · Design · Dialectical logic · Hegel · Kant · Requirements · Transcendental logic

✉ Luciano Floridi
luciano.floridi@oii.ox.ac.uk

¹ Oxford Internet Institute, University of Oxford, 1 St Giles, Oxford OX1 3JS, UK

² The Alan Turing Institute, 96 Euston Road, London NW1 2DB, UK

1 Introduction

This article is about understanding the *conceptual logic* of design¹ of a system. Philosophy is often concerned with the study of structural properties of systems—whether empirical, like political organisations, or conceptual, like scientific theories—and their dynamics.² Before the advent of modern mathematical logic, such structural studies might have been considered to be part of logic broadly understood.³ Today, this would be very misleading. But so would be the use of labels such as “informal logic” (see Sinnott-Armstrong and Fogelin 2015; Walton and Brinton 2016) and “philosophical logic” (see Burgess 2009; Sainsbury 2001), already appropriated by other branches of philosophy. So, for lack of a better expression, I shall refer to such structural studies as “conceptual logic”. A conceptual logic is what we have in mind when we ask questions about the abstract properties, principles, mechanisms, or dynamics characterising or underpinning systems like international relations, a bureaucracy, or the internal arrangement of elements in a device to perform a specific task. A conceptual logic focuses on formal features that are independent of specific implementations or idiosyncratic contingencies, on types rather than tokens, on invariants and their mutual relations, and on state transitions that are generalisable. This much is obvious. What remains unclear is exactly what further qualifies as a conceptual logic of a system.

Later in the article, I shall use a more precise vocabulary and talk about the “model” (semantic information about) of a “system” (the referent “abouted” by some semantic information). Anticipating, this means that the question I intend to address can already be phrased more precisely by asking: what is the conceptual logic of modelling (i.e., generating a description of some structural properties of) a system? Note that, once unpacked, this question is not ontological but epistemological: it is not about the intrinsic logic of a system in itself, but rather about the logic of the model of a system. If this is unclear, compare the conceptual logic of a watch with the conceptual logic of the design of a watch. The focus is really about a conceptual logic of design of a system understood as a specific kind of conceptual logic of information about (the design of the model of) a system; and not any kind of information, but that particular kind we call *semantic* or *factual* information, i.e. information about something. For some philosophers it may be obvious that the logic of a model of a system is the logic of the system. I am not so sure, and I often doubt it but, most importantly, this is not an issue that needs to be resolved here. We can all refrain from stepping into any ontological discussion, and phrase anything that we need to discuss epistemologically, and therefore more safely in terms of commitment. If, as a bonus, some people may believe that we pay one and get two,

¹ For a philosophical analysis of the debate on the concept of design Parsons (2016) is a great source. The book is also an essential starting point to frame the proposal made in this article. It dedicates quite some space to the debate on modernism. It is a must read for anyone interested in understanding the conceptual nature of design together with the excellent Dunne and Raby (2013). These works, together with Barthes (1981) and Flusser (1999), have deeply influenced my views on the topic.

² See Alexander (1964), p. 8.

³ For an enlightening reconstruction of the late modern developments of logic between Kant and Frege see Heis (2012).

as it were, that is, if anyone is convinced that, by developing a conceptual logic of the model we shall also have obtained a conceptual logic of the system, I shall not complain, but neither shall I join them.

Formulating all this more carefully and clearly, and providing a reasonable answer will require several steps. The first thing to do, in section two, will be to introduce some standard terminology and the methodology of levels of abstraction. This is a modelling tool borrowed from computer science. I will need it in the rest of the article to make sense of the nature of conceptual logic. In section three, I shall clarify that we have inherited from modernity two main conceptual logics of information understood as logics of modelling systems: Kant's and Hegel's. Both are useful to analyse the structural properties of systems. So both are implicitly used by many disciplines. Yet I shall argue that neither is by nature a conceptual logic of modelling as *designing*, even though they have been repurposed to fulfil such a task, partly because they easily lend themselves to this, and partly because they were the only conceptual logics available. A logic of design is a third conceptual logic that is still missing and needs to be developed.⁴ Thus, this article can also be read as a call to action for, and a first stepping-stone in the direction of, the full articulation of a conceptual logic of design in general, so that we may go beyond practitioners' rules of thumb and recipes for best practices.⁵ In section four, I shall outline the conceptual logic of design as the logic of requirements. In the conclusion, I shall argue that the logic of requirements is exactly what we need to make sense of, and buttress, a constructionist approach to how we model systems informationally, that is, a shift from a representationalist (*mimetic*) to a constructionist (*poietic*) understanding of knowledge.

2 On the Very Idea of Levels of Abstraction

In order to introduce the method of abstraction and the concept of “level of abstraction” (LoA), let us begin with a few examples.⁶ Imagine Alice asks Bob the price of an item, let's say a second-hand car. Bob gives her the following answer: 5000. The question concerned a *variable*, namely the price x of the car in question, and Alice received an exact numerical value for the variable, yet something is missing. She still has no idea about the price because she does not know the *type* of the variable: is it British pounds, US dollars, euros...? Of course, the context usually

⁴ As excellent, indeed now classic examples, see Munari (1971), Arup (2012), Norman (2013). For more recent case-based studies see Buurman and Rölli (2016).

⁵ This is not a new invitation. In Simon (1996), pp. 121–125, Herbert Simon discusses “The Logic of Design”. Simon acknowledges that First Order Logic (FOL) does not deliver what is needed. However, his approach moves towards a logic of design as heuristics, i.e. a logic of searching for “candidates” for a good solution, within a space of possible worlds in which a variety of solutions for a problem is available. As the reader will see, in this article I share the same view about FOL, but I take a different direction, in favour of a logic of design in terms of a logic of requirements. The two approaches are not incompatible and probably complementary. To put it simply, in a personal discussion with Simon at Carnegie Mellon, we agreed that his interest was in a logic that would help to identify the right answers, whereas mine was and still is in a logic that would help to design the right questions.

⁶ The reader interested in a more technical treatment and further applications may wish to consult Hoare (1972) and Floridi (2008, 2010, 2011, 2013a). I am grateful to Hoare for pointing out the relevance of that text.

helps. If Alice is in England and Bob is a local car dealer, her question should be understood as concerning the price in British pounds and so should the answer. This is trivial, you may think. Grice's conversational rules obviously apply (Grice 1989). It is, and they do. But this is also a crucial assumption, easily forgotten. In November 1999, NASA lost the \$ 125 m Mars Climate Orbiter (MCO) because the Lockheed Martin engineering team used British (also known as Imperial) units of measurement, while the agency's team used the metric system for a key spacecraft operation. As a result, the MCO crashed into Mars.⁷ Assuming that contexts will always disambiguate the types of your variables paves the way to costly mistakes. This is why the idea of a "level of abstraction" plays an absolutely crucial role in how we handle any information process carefully, and so in how we negotiate our interactions with the world, and therefore in how we develop our philosophy and ethics of information. This is so even when a specific LoA is wrong or left implicit, as in the MCO example. So what is a LoA exactly?

The method of abstraction comes from modelling in science, where the variables in the model correspond to observables in reality, all others being abstracted. The terminology has been influenced by an area of computer science, called Formal Methods (Boca et al. 2010), in which discrete mathematics is used to specify and analyse the behaviour of information systems. Despite that heritage, the idea is not at all technical and, for the purposes of this article, no mathematics is required.⁸ Let us consider another everyday example.

Suppose we join Alice, Bob, and Carol at a party. They are in the middle of a conversation. Alice is a collector and potential buyer; Bob tinkers in his spare time; and Carol is an economist. We do not know the subject of their conversation, but we are able to hear this much:

- Alice observes that it (whatever the reference of "it" is) has an anti-theft device installed, is kept garaged when not in use, and has had only a single owner;
- Bob observes that its engine is not the original one, that its body has been recently re-painted, but that all leather parts are very worn;
- Carol observes that the old engine consumed too much, that it has a stable market value, but that its spare parts are expensive.

The participants view the "it" according to their own interests, which orient the choice of their conceptual interfaces or, more precisely, of their own levels of abstraction. We may guess that they are probably talking about the second-hand car we already met at the beginning of this section, but, if you check carefully, it could be a motorcycle, or even an airplane, since any of these three artefacts would satisfy the descriptions provided above by A, B, and C. Whatever the reference is, it provides the source of information under discussion. We shall call this "it" the *system*. A LoA consists of a collection of observables, each with a well-defined

⁷ "Mars Climate Orbiter Mishap Investigation Board Phase I Report" (Press release). NASA: ftp://hq.nasa.gov/pub/pao/reports/1999/MCO_report.pdf.

⁸ For a recent, mathematical implementation of the method of abstraction see in particular Le Goc and Vilar (2017).

possible set of values or outcomes. For the sake of simplicity, let us assume that Alice's LoA matches that of an owner, Bob's that of a mechanic, and Carol's that of an insurer. Each LoA (imagine a computer interface) makes possible a determinate analysis of the system. We shall call the result or output of such analysis a *model* of the system. Evidently, a system may be described at a range of LoAs and so one can have a range of models. We are now ready for a more formal definition.

We saw that a "typed variable" is understood as a variable qualified to hold only a declared kind of data. For example, if Bob asks Alice for her telephone number, whatever the latter is (variables), he expects natural numbers to be the type of the variables she will provide. Since the system investigated may be entirely abstract or fictional the term "observable" should not be confused here with "empirically perceivable". Historically, it might be an unfortunate terminological choice, but, theoretically, an *observable* is just an *interpreted typed variable*; that is, a typed variable together with a statement of what feature of the system under consideration it represents. For example, a set of data could have natural numbers as a type and *telephone number* as a feature of the system. A level of abstraction (LoA) is a finite but non-empty set of observables, which are expected to be the building blocks in a theory characterized by their very choice. An "interface" (called a "gradient of abstractions") consists of a collection of LoAs and is used in analysing a system from varying points of view or, to be more precise, at varying LoAs.

Models are the outcome of the analysis of a system developed at some LoA(s). The *method of abstraction* consists in formalizing the model. This often happens implicitly, and only in *qualitative* rather than *quantitative* terms, by using the concepts just introduced (and others relating to system behaviour which we do not need here). In the previous example, Alice's LoA might consist of observables for SECURITY, METHOD OF STORAGE, and OWNER HISTORY; Bob's might consist of observables for ENGINE CONDITION, EXTERNAL BODY CONDITION, and INTERNAL CONDITION; and Carol's might consist of observables for RUNNING COST, MARKET VALUE, and MAINTENANCE COST. For the purposes of discussion, the interface might consist of the set of all three LoAs. In this case, the LoAs happen to be *disjoint*, but in general they do not have to be. LoAs can be *nested*, *disjoint*, or *overlapping* and may, but do not have to be, *hierarchically related*, or ordered in some scale of priority, or support some *syntactic compositionality*. A particularly important case is that in which one LoA includes another. Suppose, for example, that Dave joins the discussion and analyses the system using a LoA that includes those of Alice and Bob. Dave's LoA might match that of a buyer. Then Dave's LoA is said to be more concrete, finely grained, or lower than Alice's or Bob's, each of which is said to be more abstract, more coarsely grained, or higher; for both Alice's and Bob's LoAs abstract some observables that are available at Dave's. Basically, Dave can obtain all the information about the system that Alice and Bob might have, for example the name of the previous owner, and that it is rather expensive to maintain, and so he can obtain some information that is, in principle, unavailable to one or the other of them, since Alice does not know about running costs and Bob has no clue about the ownership history.

A LoA qualifies the level at which a system is considered. One should rely on the method of abstraction in order to refer to the LoA at which the properties of the

system under analysis can sensibly be discussed. In general, it seems that many uninteresting disagreements might be clarified, if the various interlocutors could make their LoAs explicit and precise. Yet a crucial clarification is in order. It must be stressed that a clear indication of the LoA at which a system is being analysed allows *pluralism* without endorsing *relativism*. One may refer to this middle-ground position as *relationism*. When one criticizes a position as *relativistic*, or when one objects to *relativism*, one should not equate such positions to non-absolutist, as if there were only two alternatives, e.g. as if either moral values were absolute or relative, or truths were either absolute or relative, objective or subjective, observer-independent or observer-dependent. The method of abstraction enables one to avoid exactly such false dichotomies, by showing that a subjectivist position, for example, need not be relativistic, but only relational. To use a simple example: Alice may be tall when compared to Bob, but not when compared to someone in the basketball team. It does not mean that her height changes, but only that she is or is not tall depending on the frame of reference, that is, on the LoA. Relativism is really the equivalent of an “anything goes” position. Now, it is a mistake to think that “anything goes” as long as one makes explicit the LoA, because LoAs are mutually comparable and assessable. Consider again the example of Alice’s telephone number. There might be some significant differences in the way in which Alice communicates it to Bob. She might add a plus and the relevant country code at the beginning, thus modifying the overall type of the information provided. She might omit the plus, the country code, and the city code, if it is a local number. So there is quite a lot of ‘relationism’ (‘it depends on...’) but no ‘relativism’: it would be silly to conclude that any LoA would do. A string of letters would not work, nor would a mix of letters and numbers, or numbers and non-alphanumeric symbols, or an endless string. Using a different example, when we are asked to provide the number of our credit card, the type is (a finite number of) of natural numbers. This is why an interface can easily constrain the sort of input required. In general, only some LoAs are possible and, among those, some are better than others. Relativism, as a position about the interchangeability of LoAs at no epistemic cost, is simply mistaken because, crucially, the assessment and corresponding preference is usually dictated by the *purpose* driving the original request for information. Deciding whether Alice is tall depends on the purpose for which we are asking the question in the first place. The implicit LoA may easily be the average population height. She may be a tall child. Or it may be an old army parameter according to which someone below a specific height cannot be recruited. This is clearly not relativism, because one may challenge the appropriateness of the LoAs in question on perfectly reasonable and justifiable grounds.

Introducing an explicit reference to the LoA clarifies that:

- (a) the model of a system is a function of the available observables;
- (b) different interfaces may be correctly ranked depending on how well they satisfy modelling specifications (e.g. informativeness, coherence, elegance, explanatory power, consistency with the data, etc.) and the *purpose* orienting the choice of the LoA (LoAs are teleologically oriented);

- (c) different analyses can be correctly compared provided that they share the same LoA.

Let us now agree that a system is characterized at a given LoA by the properties it satisfies at that LoA. Systems are often dynamic: they change, and this means that some of those properties change value. The evolution of a changing system is captured at a given LoA and at any instant by the values of its observables (the attributes of the system). Thus, a system can be thought of as having states, determined by the value of the properties that hold at any instant of its evolution, for then any change in the system corresponds to a state change and vice versa. Generalizing, this enables one to view any system as having states and transitions. The lower the LoA, the more detailed the observed changes and the greater the number of state components required to capture the change. Each change corresponds to a transition from one state to another. A transition may be non-deterministic. Indeed, it will typically be the case that the LoA under consideration abstracts the observables required to make the transition deterministic. As a result, the transition might lead from a given initial state to one of several possible subsequent states.

We have now moved from a static to a dynamic observation of a system, analysed as a transition system. The notion of a “transition system” provides a convenient means to support the identification of the necessary and sufficient criteria for agency, being general enough to embrace the usual notions like automaton and process.⁹

A *transition system* comprises a (non-empty) set S of states and a family of operations, called the *transitions* on S . Each transition may take input and may yield output, but, at any rate, it takes the system from one state to another and in that way forms a relation on S . If the transition does take input or yields output, then it models an interaction between the system and its environment and so is called an *external* transition; otherwise the transition lies beyond the influence of the environment (at the given LoA) and is called *internal*. It is to be emphasized that inputs and outputs are, like states, observed at a given LoA. Thus, the transitions that model a system are dependent on the chosen LoA. At a lower LoA, an internal transition may appear external; at a higher LoA an external transition may appear internal.

Returning to our example, the system being discussed by Alice might be further qualified by state components for location, whether in-use, whether turned-on, whether the anti-theft device is engaged, history of owners, and energy output. The operation of garaging the system might take as input a driver, have the effect of placing the system in the garage with the engine off and the anti-theft device engaged, leaving the history of owners unchanged, and outputting a specific amount of energy. The ‘in-use’ state component could non-deterministically take either value, depending on the particular instantiation of the transition. Perhaps the system is not in use, being garaged for the night; or perhaps the driver is listening to a

⁹ For a formal treatment of much more than is required in this context, the reader might wish to consult Arnold and Plaiice (1994).

programme broadcast on its radio in the quiet solitude of the garage. The precise definition depends on the LoA. Alternatively, if speed were observed but time, accelerator position, and petrol consumption abstracted, then accelerating to 60 miles per hour would appear as an internal transition.

We can now use the method of abstraction and the concept of LoA to make explicit the ontological commitment of a theory, in the following way.

A theory comprises at least a LoA and a model. The LoA allows the theory to analyse a given system and to elaborate a model that identifies some properties of the system at the chosen LoA (see Fig. 1).

The ontological commitment of a theory—the grey triangle—can be clearly understood by distinguishing between a *committing* and a *committed* component within the scheme. On the one hand, a theory commits itself ontologically by opting for a specific LoA. Compare this to the case in which one has chosen a specific kind of car (say a Volkswagen Polo) but has not yet bought one. On the other hand, a theory is ontologically committed in full by its model, which is therefore the bearer of the specific commitment. The analogy here is with the specific car one has actually bought (that blue, four-wheeled, etc. specific vehicle in the car park that one owns). To summarize, by adopting a LoA, a theory commits itself to the existence of some specific types of observables characterizing the system and constituting the LoA (by deciding to buy a Volkswagen Polo one shows one’s commitment to the existence of that kind of car), while, by adopting the ensuing models, the theory commits itself to the corresponding tokens (by buying that particular vehicle, which is a physical token of the type Volkswagen Polo, one commits oneself to that token, e.g. one has to insure it). Figure 2 summarizes this distinction.

By making explicit the ontological commitment of a theory, it is clear that the method of abstraction plays an absolutely crucial role in philosophy, especially if the latter is understood as conceptual design. Quine once wrote that:

The very notion of an object at all, concrete or abstract, is a human contribution, a feature of our inherited apparatus for organizing the amorphous welter of neural input. [...] Science ventures its tentative answers in man-made concepts, perforce, couched in man-made language, but we can ask no

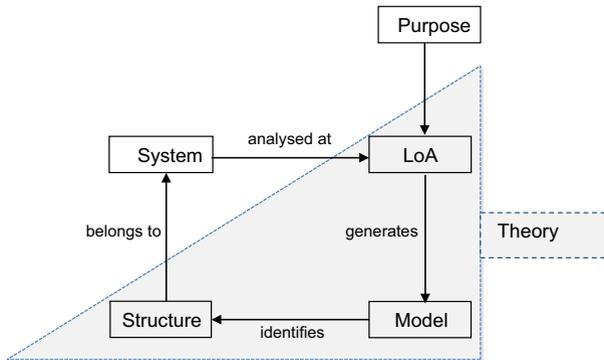


Fig. 1 The scheme of a theory

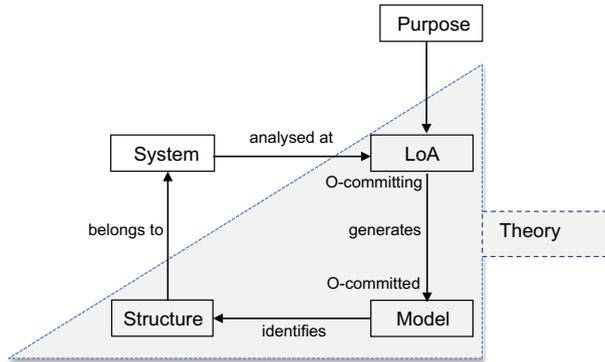


Fig. 2 The SLMS (System-LoA-Model-Structure) scheme with ontological commitment

better. The very notion of object, or of one and many, is indeed as parochially human as the parts of speech; to ask what reality is really like, however, apart from human categories, is self-stultifying. It is like asking how long the Nile really is, apart from parochial matters of miles or meters.¹⁰

This section may be read as an attempt to clarify what ‘human categories’ in the first quotation above means, in terms of a less parochial method of LoA. We can now use the method to clarify what a conceptual logic is, in the next section.

3 Two Modern Conceptual Logics of Information

What kind of structural LoA could one focus on, when analysing a system? A simple taxonomy may be given in terms of a past-oriented, present-oriented and, as we shall see, a future-oriented approach (let me warn the reader that what follows contains significant simplifications, some probably controversial).

Imagine that one is interested in understanding the genesis of a system, how it came to be, or what brought it about. This is the purpose orienting the choice of LoAs. One then focuses on modelling the *conditions of possibility* of a system under investigation (Fig. 3). This is clearly a past-oriented approach, consistent with causal, genetic, or genealogical forms of reasoning that lead to a particular modelling or conceptualisation of reality, with the identification of necessary and (perhaps) jointly sufficient conditions, with investigations about what must have been the case for something else to be the case. The approach goes hand in hand with an interest in the natural sciences, or any discipline that tries to mimic their logic. One looks at a system and tries to understand what brought it about. As the reader may have already guessed, this conceptual logic finds its roots in Kant’s transcendental logic.

It is tempting and not impossible to use Kant’s transcendental logic as a logic of construction as design. I mentioned above the compatibility between the natural sciences and Kant’s transcendental logic, but the latter has also been employed in

¹⁰ Quine (1992), p. 6 and p. 9.

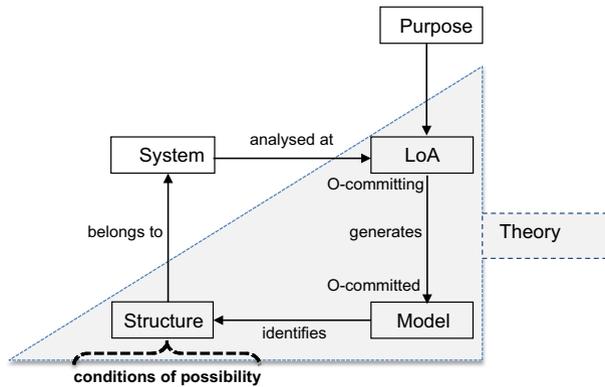


Fig. 3 Transcendental logic

jurisprudence, for example, in order to look at normative systems in terms of ideal or hypothetical conditions of construction of a system. This is not an incorrect repurposing of it, but it does stretch its nature. It is true that the reasoning that takes one from a system to its conditions of possibility can also be adopted to move from the conditions of possibility (now to be understood as requirements of feasibility) to the system itself. But while the former move is univocal, the latter is not, unless one of the two following conditions is satisfied, each of which is problematic. Either one already knows what system one is going to obtain anyway, and so it is actually begging the question. Or the system is entirely constrained by its conditions of possibility, which now become both necessary and jointly sufficient to deliver it. In this case, the assumption is that the transcendental conditions constrain the issuing system univocally. This is unlikely to say the least, because it hardly ever happens. Just think of the requirements prompting the design of a house. I shall come back to this limit case of design below, when I shall define it as “degenerate” in the mathematical sense of the word. For the time being, let me stress that such a degenerate approach seems implicitly adopted by Kant and neo-Kantians of various orientations. The issue may be explained with an Italian analogy. Even in the apparently constrained case of a pizza, you can backtrack an actual pizza in front of you uniquely to the ingredients used and a general backing recipe followed, but if you have the ingredients and the general baking recipe they may lead to a variety of pizzas (deep pan? filled crusts? Chicago style? More Neapolitan? Preferably Roman?), because there can be quite a lot of room for variations, and the pizza you started from is unlikely to be the only one feasible and may not even be the best, or the most interesting, in terms of full exploitation of the ingredients and the backing recipe itself. In Kant, it is the univocal relation between conditions of possibility of a system and its requirements of feasibility that supports the whole transcendental project. And this is so because Kant is mainly concerned about guaranteeing the objectivity of knowledge even when knowledge is no longer representational. Kant cannot ground the univocal nature of knowledge (“this is what the world looks like”) on the univocal nature of the reality it refers to (“this is the way the world is”) because this is a strategy open to representationalism but not to constructivism. So

he replaces the univocal nature of a represented reality in itself with the univocal genesis and relationship knowledge enjoys in its encounter with reality in itself. In other words, realism is guaranteed by the fact that the set of ingredients plus the recipe lead to only one pizza and that pizza leads back to only one recipe and set of ingredients. Likewise, in Kelsen's *Pure Theory of Law* Kelsen (1967), for example, the crucial concept of *basic norm* (*Grundnorm*) seems to play a twofold role. It is used to refer to the basic order or rule that forms an underlying basis for a legal system understood as a hierarchical relationship of norms, akin to a first principle or axiom. It therefore works as a point of origin or source for all law, which obtains legitimacy from it. Yet it remains inevitably ambiguous whether the *Grundnorm* and the issuing construction is an *as-if* model of how the law that exists may be reconstructed, in closer similarity to a transcendental analysis, or a hypothetical blueprint of how the law that may, but does not yet, exist should be constructed. The tension is unresolvable because the two interpretations look at the same road, from system to model and from model (now blueprint) to system. So, from a "degenerate" analysis, the answer is that the *Grundnorm* is both a blueprint and a model. But this solution is challenged by the fact that, historically, there is a recursive relation between the blueprint for the design of a system that is then analysed by a model that turns into a blueprint for a new system, and so forth. I shall return to the recursive nature of any design process in section four. For now, what I am suggesting here is that we should take the univocity of the transcendental as problematic, indeed untenable in most cases, and probably in all interesting cases. Certainly the American pragmatists' view of the a priori and hence of the transcendental conditions of possibility of knowledge is already significantly not univocal.¹¹ In particular, I share C. I. Lewis's neo-Kantian "conceptualistic pragmatism",¹² according to which empirical knowledge is the outcome of the exercise of the constructive activities of the mind on the data (Lewis' sensuous 'give') offered by the world, but the set of a priori concepts that are used in such a process are not only biologically determined, but they are also the historical product of the knower's social and cognitive circumstances as well; pragmatically and not absolutely, contrary to Kant. In the vocabulary of this article, they are historical LoAs, which admit of alternatives and preferential choice, where the choice of the right LoAs (the right set of a priori concepts) depends on purpose-oriented, pragmatic considerations, and it is modifiable and hence improvable, depending on the success achieved. This means that, in Lewis, there is no equation of univocity. Once you socialise and historicise the transcendental—as we should here, given that the method of abstraction is driven by a contextually justified choice of purposes—it becomes very difficult to accept the Kantian equation:

¹¹ See Rosenthal (1987).

¹² See Lewis (1929). As it is clear from Floridi (2017), I am utterly unconvinced by Lewis' later naturalism as presented in Lewis (1946) because it seems to me to exchange Kant's logical univocity for a biological, impoverished version that is even more controversial. Having freed the transcendental from its univocal relationship with the construct, Lewis seems to put it back inside a smaller cage of univocal relation with successful behaviour and planning. The point is that we should be more, not less Kantian than Kant, so to speak.

[conditions of possibility = requirements of feasibility] ↔ [designed system]

in which one set of conditions works both as the set of conditions of possibility and as the set of conditions of feasibility to which only one designed system corresponds, and vice versa. The truth is that, in most design cases, the identity and the double implication are hardly ever justified. Fortunately, instead of resolving the tension, it may be preferable to bypass it by sticking to the natural vocation of transcendental logic as a logic of re-construction. This becomes easier once a proper logic of design as construction becomes available, as I intend to show below. But before this, we need to look at another alternative.

One may wish to understand not the past genesis but the current (present-oriented) state of dynamics of a system. If this is the purpose orienting the choice of the LoA, then one does not look at what brought about the system, but focuses instead on modelling the *conditions of in/stability* of a (transition) system under investigation (Fig. 4). This approach is consistent with polarised and procedural ways of reasoning, with the identification of contrasts and their resolutions (or lack thereof), with present-oriented analyses of processes (evolutions, developments), and mutual interactions. The approach goes hand in hand with an interest in the social and human sciences, especially economics, history, law, and political science, although the natural sciences have also been approached from this perspective, if with remarkable lack of persuasiveness (consider Marxist–Leninist analyses of scientific theories such as quantum mechanics). One looks at a system and tries to understand what dynamic equilibria keep it stable or unstable. In this case too, the reader may have already guessed that this conceptual logic finds its roots in Hegel’s dialectics. As with transcendental logic, dialectics has been stretched to repurpose it as a logic of design, most famously by Marx. The outcome is unconvincing insofar as the state transitions of the system identified by the logic, e.g. a specific political organisation of a society, are at the same time necessary and yet open to steering or influence. As in Kant, in Hegel too we deal with a logic of necessity: what must have been the case for the system to be what it is, or what will have to be the case

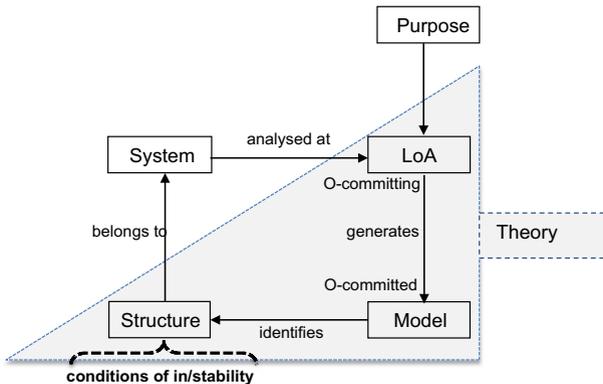


Fig. 4 Dialectical logic

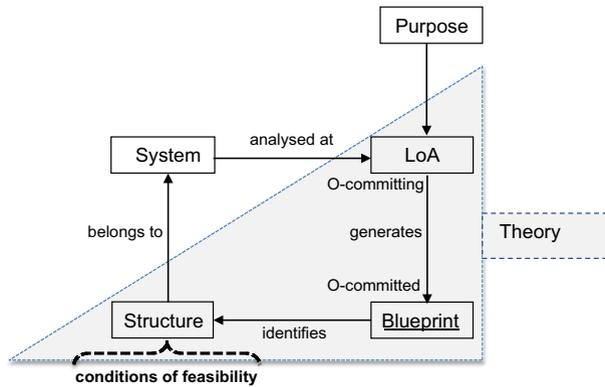


Fig. 5 The logic of design

given that the system is what it is. In fact, design needs a logic of “sufficiency”, as I shall explain presently.

From the previous sketch it follows that modernity has been dominated by two main conceptual logics of information: Kant’s and Hegel’s. Both can be seen at work, methodologically, in philosophers such as Marx (Hegel) and Husserl (Kant). Yet they are not incompatible and can also easily be found interacting, e.g. in game theory. Indeed, part of the difficulty in understanding Foucault, for example, is due to the fact that his genealogical approach often switches between a transcendental and a dialectical conceptual logic of information.

Both conceptual logics model systems as something given, whether in the natural universe, as more significantly in Kant, or in human history, as more influentially in Hegel. Thus both move from the system to the model, broadly understood as a description of (information about) it, looking for generalities and patterns. Both are constructionist in a broad sense of the term: they look at how a system came to be and its dynamics. One is then reminded that German idealism is among the most constructionist-friendly approaches in the history of philosophy. Yet strictly speaking neither is a conceptual logic of *construction as design*, which moves from the model—now understood as a *prescription* or *blueprint*—to its implementation, as a realised system that does not precede the model. That is, neither is in itself primarily a *logic of design*, although both have been stretched and repurposed to fulfil this role. In this case, the goal is to understand the future-oriented possibility of construction of the system. One then focuses on modelling the *conditions of feasibility* of the system under investigation (Fig. 5). As I have argued elsewhere Floridi (2016), one important consequence is that, asking whether a logic of design is a priori or a posteriori, means not having grasped the shift from *mimesis* to *poiesis*.¹³ A designer’s knowledge is a very significant case of what is known as the maker’s knowledge. This is the sort of knowledge that Alice enjoys when she holds the information (true content) that Bob’s coffee is sweetened because she just put

¹³ The classic on *mimesis* is of course Auerbach (2003). I have introduced the idea of *homo poieticus* in Floridi (1999) and further discussed it in Floridi and Sanders (2005).

two spoons of sugar in it herself, i.e., because she designed the system that way. Her knowledge is *contingent, synthetic, weakly a priori (ab anteriori)*, and *uninformative*, in the technical sense that she cannot be informed about the sugar in the coffee by hearing a public announcement stating that there is sugar in the coffee. This is why I agree with Ammon (2017) that designing is not an empirical kind of experimenting (contrary to widespread methodological claims) but more an independent epistemic praxis through which one can acquire genuine *ab anteriori* knowledge.

The lack of a logic of design is not just a scholarly gap, or a source of over-interpretations of transcendental logic and dialectical logic in our philosophical reasoning. It is a particularly pressing problem because of a significant shift in how we prioritise forms of knowledge and sciences in particular.

For a long time, and at least since Greek philosophy, our epistemic culture has privileged a *mimetic* view of knowledge and hence science, as a description of reality, whether geometrical, physical, biological, chemical, or human, for example. Astronomy provided a kind of regulative ideal. For this reason, still nowadays, our philosophy of science tends to focus more on physics than on any other discipline. However, as our knowledge of reality has progressed and become more refined, as our need to investigate reality has increasingly relied on technology, experiments, and artefacts, and as our ability to create, control and modify aspects of the world has grown, our epistemic outlook has become increasingly constructionist, that is, *poietic*.¹⁴ Today, it would be perfectly reasonable to say that the most influential sciences and scholarly disciplines that shape our world and our implicit, cultural understanding of it are architecture, computer science, economics, engineering, and law. They all do much more than just describe their objects, they actually contribute to construct them, and, in the case of engineering and computer science, empower other sciences to construct their own, through new artefacts, technologies, simulations, algorithms, and data-based research. *Poietic*, as opposed to *mimetic* science, relies on a conceptual logic of construction that does not start from the system to analyse it in terms of a model, but actually starts from the model (the blueprint) to realise the system. In this case understanding is constructing. Or, as Vilém Flusser insightfully put it:

the factory of the future will have to be the place where *homo faber* becomes *homo sapiens sapiens* because he has realised that manufacturing means the same thing as learning—i.e. acquiring, producing, and passing on information.¹⁵

This information manufacturing, this poietic knowledge, is implicitly based on a logic of design. We find its roots in Kant and Hegel, as I tried to show above, but it is under-theorised. Such a conceptual logic of *semantic information modelling as designing* is consistent with creative forms of reasoning, with the identification and exploitation of constraints and affordances, and hence the satisfaction of requirements, with future-oriented analyses of what systems could be built and could work, often recursively, and with engineering sciences broadly understood. It is a logic of

¹⁴ For a defence of constructionism see Floridi (2017).

¹⁵ Flusser (1999) is a gem, full of remarkable insights. The quotation comes from p. 50.

sufficient rather than necessary conditions, in a specific sense that I shall explain in the next section.¹⁶

4 The Logic of Design as a Logic of Requirements

The search for a logic of design is not (and it would be suspicious if it were) entirely new. Carnap, for example, who described philosophy as conceptual engineering, is a great representative of the German tradition, beginning at least with Kant and going all the way at least to Wittgenstein's *Tractatus*, in which philosophical analysis is expected to unearth the deep, conceptual structures of knowledge, language, and semantics. This is design of some sort:

The programme of rational reconstruction set out in 1922-3 had given great prominence to a notion of 'structure' that remained somewhat enigmatic. And though 'structure' itself was purely logical or analytic, its application instance in the 'constructive' project of the Aufbau sought to capture the component of knowledge that Helmholtz and Poincare had said was all we ultimately could know about the world: 'What science can attain to is not the things themselves but only the relations among the things; apart from these relations there is no knowable reality' (Poincare 1904, p. XIII). Carnap, as we saw, had taken this further, and built his project of rational reconstruction on the idea that 'every science is a science only insofar as the study of structures [Strukturlehre] is contained in it' (above, p. 163). Did this make our knowledge of 'structure' itself a form of synthetic knowledge? And finally, even if 'structure' was purely logical, where did the principles of logic come from? All Carnap's great role models in logic had answered this last question by gesturing toward a special source of knowledge, different from the empirical - the principles of logic were synthetic a priori, or traceable to a 'logical source of knowledge' that gives us our ability to grasp 'thoughts' residing in a 'third realm' distinct from the first realm of physical objects and the second realm of subjective consciousness.¹⁷

However, Carnap, analytic philosophers, and structuralists and deconstructionists of all kinds, seem to forget that structures are not only unveiled, investigated, known, discovered, reconstructed or deconstructed. They are also, and perhaps today above all, *designed*. What does this mean?

Keeping in mind the risk just highlighted, it is obvious that architecture, the design discipline *par excellence*, provides a great heuristics to clarify what a logic of design may be. According to the American Institute of Architects, there are five phases of design.¹⁸ The taxonomy is popular and rather uncontroversial. It provides

¹⁶ On a poetic epistemology see Floridi (2004, 2016, 2017) (Floridi 2016 #147; Floridi, forthcoming #147; Floridi, forthcoming #146; Floridi 2004 #269). Information manufacturing should not be confused with the contemporary movement known as information architecture, which is really about how to manage available information in ways that make it more accessible, see for the example the entertaining Covert and Fenton (2014) or a standard textbook treatment in Morville et al. (2015).

¹⁷ Carus (2007), p. 185.

¹⁸ AIA website: <http://howdesignworks.aia.org/fivephases.asp>.

an excellent starting point. I shall paraphrase and summarise it below, with only minor adaptations and a significant proviso¹⁹:

Phase 1: *Originate* This is the thinking phase in which one realises that something new (in our terminology, a new system) needs to be built in order to satisfy a particular purpose. It is the “needing” moment of the project.

Phase 2: *Focus* This is the phase in which one defines the system’s *requirements* (these are broadly understood as scope, features, purpose, or functionality, more on this presently) that the system must have in order to satisfy the purpose. It is the “vision” moment for the project.

Phase 3: *Design* This is the phase in which one models the system’s requirements. It is the “shaping” moment of the project.

Phase 4: *Build* This is the phase in which one constructs the system. It is the “making” moment of the project.

Phase 5: *Use*²⁰ This is the last phase in which the system is finally available and starts satisfying its purpose. It is the “testing” moment of the project.

The proviso to be added is that the five phases can easily be *recursive*. If any phase, including *Phase 5*, is unsatisfactory, one may go back all the way to *Phase 1*. Indeed, the phases are often recursive in many design contexts in which fast prototyping has become ordinary. 3D printing has had a profound influence on this more cyclical approach. This holds true in architecture as well. Of course, one does not build whole architectural systems to re-do them if unsatisfied. The difference lies in the longer, diachronic time scale. For example, the refinement of the Doric, Ionic, and Corinthian orders of architecture took a long time, but it was no less recursive than the trial and error one may experience in designing a web page for a customer. Rome was not built in a day, but this does not mean that it was not built recursively through the centuries. And we saw above that Kant’s transcendental logic may be in a recursive relation with the logic of design: what is initially a blueprint is translated into a first system, which is then analysed by a model that can become a new blueprint for a revised system, and so forth. All this highlights the fact that design is not limited to the creation/implementation of systems anew, but also applies to the repurposing or improving of existing systems, relying on repeated iteration with, sometimes, small adjustments, in a variety of time-scales.

As it is obvious, the logic of design may cover all five phases. If we ignore *Phase 1*, as the moment when one realises that a new system to fulfil a specific goal is needed, then during *Phases 4–5* one *verifies* and *validates* (in the design, not the philosophical sense of the two terms) the system that is being built, that is, one checks its “fitness for purpose”. During the verification stage, one evaluates whether the system is being developed in the right way, e.g. by checking for consistency. And during the validation stage, one evaluates whether the right system is being

¹⁹ I am very grateful to James Williams for having called my attention to the potentially misleading nature of architecture as an apparently non-recursive area of design.

²⁰ “Occupy” in the original text. I modified the terminology slightly to cover all kinds of designs not just the design of buildings.

developed, e.g. by evaluating the correctness of the final system with respect to users' needs. Both stages are methodological. It is the step from *Phase 2* to *Phase 3* that is significantly *inferential* and can then be defined correctly as belonging to the conceptual logic of design: the satisfaction of a specific purpose leads to the identification of some requirements which then lead to the design of a specific system that implements them in order to satisfy the purpose, with all the necessary loops in recursive refinements. This should sound familiar: it means that architectural design contains within itself an inferential stage—the passage from *Phase 2* to *Phase 3*—that is fully captured by system engineering. This is our next step.

In system engineering,²¹ one provides the specification for a system (S) to be designed by describing an explicit, non-empty, finite, and complete set of requirements $\{R_1, \dots, R_n\}$ to be satisfied by S. Such requirements in $\{R_1, \dots, R_n\}$ can be non-functional or functional, or, to use a more classic terminology, may refer to the *form* or the *function* of the system. Non-functional requirements are also simply known as requirements *tout court*. They define a system's architecture, that is, what the system is *supposed to be*. For example, a refrigerator's main non-functional requirement is that its compartment is thermally insulated and easily accessible in order to store food. Functional requirements are also known as functions. They define a system's behaviour, that is, what the system is *supposed to do*. In our example, a refrigerator main functional requirement is to maintain a constant temperature a few degrees above the freezing point of water, inside its easily accessible, thermally insulated compartment. The distinction is crucial and it has philosophical implications too, so it is important to keep it clear and firm in mind. For example, Kant's formulations of the categorical imperative have often been seen as formal, motivational features of an agent's behaviour, i.e. as functional requirements. However, from a system engineering design perspective, and a more consistent interpretation of Kant's autonomous dictates of reason, they are better understood as non-functional requirements, namely universalizability and non-instrumentalisation, i.e. what a categorical imperative is supposed to be. Finally, there is a third sense in which one may speak of requirements, and that is when resources needed to build a system are under discussion. This is not going to play a role in the logic of design discussed in this article, but it is an obvious constraint, both practically (e.g., funding) and theoretically (e.g. computational power or memory, two resources that translate into time and space, which then translate into complexity issues). Figure 6 summarises the previous analysis and taxonomy.

It turns out that the inferential step in the logic of design from *Phase 2* to *Phase 3* concerns the relation between a set of requirements (in the comprehensive sense of "requirements" just clarified), on the one hand, and a system that implements them (more or less satisfactorily), on the other. In other words, the (conceptual) logic of design turns out to be the (conceptual) logic of requirements. But then, exactly what

²¹ On the philosophy of design in engineering and architecture see the collection of essays in Vermaas et al. (2007).

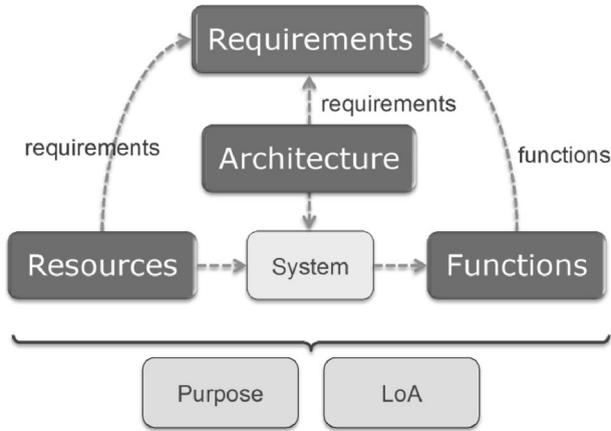


Fig. 6 Elements of system design

kind of designing relation between requirements and system are we discussing? This is not a classic entailment as understood in First Order Logic (FOL).²² In FOL, we are acquainted with the relation of *satisfaction*: a truth-structure A *satisfies* a set of sentences Γ if and only if A *models* Γ , that is: $A \models \Gamma$.²³ By analogy, one may argue that the system S satisfies (models) a set of requirements, that is, we could use the double turnstile to formalise the design relation between a system and the requirements thus:

- $S \models \{R_1, \dots, R_n\}$.

There is, however, a difficulty. Read literally, such a formula would mean that S *necessitates* $\{R_1, \dots, R_n\}$. This is obviously not true in our case. Recall the difficulty of using transcendental or dialectical logic for design purposes. The set of requirements logically precedes and “conduces” to S , and S does not have a relation of necessitation with respect to the requirements it implements: a fridge does not necessitate “preserving food”, it is rather this requirement that makes the fridge a sufficient solution. There is still a relation of satisfaction, but it needs to be re-interpreted. So let me introduce to mean that whatever is on the left “sufficientises”

²² Opting for a modal logic approach only seems to complicate instead of solving the problem. This is not to say that Modal Logic cannot be helpful to study patterns of inference in design reasoning and rationalisation, see for example Galle (1997).

²³ One of the anonymous referees correctly remarked that the notion of satisfaction referred to from FOL and underlying classic entailment is restricting the meta-theory of the language to its semantics. I see this as a non-problematic element because the object of the analysis developed in this article is semantic information. I agree with the reviewer that “the strategy in use to define a more formal approach to the logic of design can very well be constrained to syntactic relations”.

(“makes sufficient”) whatever is on the right as an implementation. This no longer resembles a deduction.²⁴ One may object that

The other feasible strategy is the pattern language proposed by Alexander,²⁵ where a deductive relation, between the design requirements, or as he called it the tendency, and the form [of the system], is constructed.²⁶

However, it is questionable whether Alexander’s work may be so interpreted. For example, in Alexander (1964) he writes:

There is no legitimate sense in which deductive logic can prescribe physical form for us. (p. 8)

And even if there may be some cases in which it may be correct to speak of a deductive relation, this only means that there are some “degenerate” (in the mathematical, technical sense of the word)²⁷ cases of design in which the requirements are so completely constraining as to make possible one and only one system that satisfies all of them and no others. We saw above that such cases are neither representative nor interesting. Note, however, that the inferential relation is not an induction or abduction either.²⁸ One may call it a “conduction”. We can now write:

$$\bullet \{R_1, \dots, R_n\} \models S$$

to formalise the design relation between a set of requirements and what they conduce to in terms of a system that suffices to satisfy them. For example, the set of two requirements that indicate that we should design a system that seats one person “sufficientises” several solutions, without necessitating any of them in particular:

$$\begin{array}{l} \bullet \{ \text{seat, one-person} \} \models \text{Chair} \\ \bullet \{ \text{seat, one-person} \} \models \text{Stool} \\ \bullet \{ \text{seat, one-person} \} \models \text{Pouf} \end{array}$$

How we identify the requirements is a matter of analysis, sometimes of necessity, but sometimes of choice. Recall the rejection of the “Kantian equation”: the same set of requirements may lead to a variety of systems understood as solutions that

²⁴ Note that this is in contrast with the interpretation provided by Zeng and Cheng (1991), who argue that the logic of design is a recursive kind of deductive logic. Unfortunately their analysis confuses different levels of abstraction as well as different stages of development of design, and it seem to resemble some kind of Aristotelian practical syllogism. When the relation between requirements and system is deductive we are dealing with.

²⁵ Alexander et al. (1977) and Alexander (1979). Both references are in the original quotation.

²⁶ Zeng and Cheng (1991), p. 140.

²⁷ In mathematics a “degenerate” case is a special, limit case which is non-representative of the class of objects under discussion, for example a geometrical point is a degenerate circle with radius = 0.

²⁸ The temptation of interpreting the logic of design as abduction in Peircean terms is recurrent. March (1984) defends a logic of design as a logic of abduction, but in a problem-solving context, see also Goel (1988), who seeks to extend March’s approach. Kroll and Koskela (2015) re-proposes the abductive interpretation but refers only to March (1976) not to March (1984).

satisfy them, and the choice of which solution may be preferable, or whether the set needs to be revised, is pragmatic, but not determined, and can be as much linked to biological and social frameworks as to cultural, personal, or even accidental circumstances. In terms of choice, for example, demaking a video game means remaking it for an older platform, that is, it's a self-imposed challenge to require simpler graphics and sound, and simplified gameplay to see what a video game would be like, if it had been conceived and produced for an older hardware or software generation. The same applies to so-called "constrained writing". *Green Eggs and Ham*, the famous children's book, uses only 50 different words because of a \$50 bet by Theodor Geisel, also known as Dr. Seuss, with Bennett Cerf.²⁹

Once the requirements have been identified (and often this is a process of discovery that needs recursive revisions as the design process progresses) then choosing between the different systems "sufficientised" by the requirements, and deciding how to implement the preferred choice, become a matter of policy, that is, the application of a deliberate set of principles in order to guide the decision and achieve the wanted outcome. This is why there is no ultimate designed system in the world that does not encapsulate autonomy and choice, even if implicitly and inadvertently.³⁰

We can now extend this analysis to make sense of four standard cases, in analogy with the double turnstile in mathematical logic.

1. If a consistent but boundless set of requirements is provided, that is, if a system can be designed that can satisfy any non empty set of requirements, then in practice there is no such system, but in theory one may be referring to the set of all systems:

- $\{R_1, \dots\} \models \{S_1, \dots\}$

In FOL, this would be analogous to a boundless set of premises that entails a boundless set of conclusions; more an intellectual curiosity than something useful for any design purpose.

2. If an inconsistent set of requirements is provided, that is, if nothing goes and there is no system that can be designed that can ever satisfy such a set (imagine the requirements for a perpetual motion machine), then there is only the set, not followed by any system (the empty set):

- $\{R_1, \dots, R_n, R_1, \dots, R_n\} \models \{\}$

In FOL, this would be analogous to a contradictory set that entails any conclusion. And in the same way that a contradictory set is false in all possible worlds, likewise a set of requirements that is inconsistent is good for no

²⁹ From a similar perspective, one may argue that *War and Peace* is a transcendental novel: http://www.salon.com/2015/01/11/leo_tolstoy_theory_of_everything/.

³⁰ I am grateful to Mariarosaria Taddeo for having pointed out the need for this further clarification.

purpose, which means, it is useless for any design. One may use this terminology to speak of impossible systems.

3. If no requirements are provided, that is, if anything goes and there are no constraints (think of the design of a drug that works like a panacea), then the empty set of requirements sufficientises S:

- $\{\} \models S$

In FOL, this would be analogous to a tautology. And in the same way as a tautology is true in all possible worlds, likewise a system the design of which is unconstrained by any requirement is a system good for any purpose, which means, it is magical or, more seriously, useless.

4. Finally, if a consistent and finite set of requirements, for example {seat, one-person, back} conduces to a unique system, namely a chair, that is, if only one system can satisfy them, then we have a double relation of sufficientisation:

- $\{R_1, \dots, R_n\} \models S.$

In FOL, this would be analogous to a relation of double entailment. This is the “degenerate” case in which a set of requirements sufficientises only one system which satisfies only one set of requirements.

The previous analysis helps one to see that optimization in design consists in any improvement in the number or nature of the requirements and/or in their implementation according to some pre-established purpose. And efficient design is the implementation of a set of requirements such that it is impossible to satisfy any one of them more without satisfying at least one of them less (Pareto efficiency).

As in FOL, what interesting systems one may design (what interesting theorems one may prove) is a matter of talent, intuition, hard work, opportunity, good fortune, free thinking and imagination, and many other variables that are hard to pin down. As Donald Schön correctly put it, the designer (or indeed the logician and the philosopher and anyone who creatively designs solutions) is

like a chess master who develops a feeling for the constraints and the potentials [affordances in the terminology of this article] of certain configurations of pieces on the board.³¹

So the conceptual logic I have suggested here is more in line with an early analysis provided by William Fawcett who was critical of logico-mathematical approaches to the logic of design:

So what is the distinctive ‘logic of design’? The author prefers the point of view that the identity of design lies in empirical considerations, rather than in

³¹ Schön (1983), p. 104 see also Parsons (2016), p. 49.

modes of inference; and in purely empirical terms it has been confirmed that design is a more difficult task than analysis.³²

There is no Leibnizian “calculus” in the logic of design. There can be very little naturalised expediency as there is a qualitative gap between the design of a nest and the design of NEST (the smart thermostat). Nevertheless, the mechanism lying behind good design and good logic is clear. There are three conceptual logics of information as modelling of a system, and the third one is the logic of design as a logic of requirements.

5 Conclusion

In the previous pages, I have sought to outline a conceptual logic of information understood as (1) a transcendental logic of past conditions of possibility of a system (Kant), or as (2) a dialectical logic of present conditions of in/stability of a system (Hegel), or as (3) a design logic of future conditions of feasibility of a system. I then characterised (3) as a conceptual logic of requirements and introduced the relations of “sufficientisation”. The last comment I wish to make is about why a conceptual logic of information as design is particularly important today. Four reasons seem to be salient.

First, the digital re-ontologises (Floridi 2014) our world by transforming the very nature of our concepts and their related phenomena. It does this by decoupling hitherto coupled elements such as presence and location, law and territoriality, ownership and use, successful agency and intelligence. Such decoupling fluidifies both things and ideas about things, making possible the design and redesign of both. In short, the digital provides more affordances and fewer constraints as new requirements for our creative, design efforts. We need a logic of design as a logic of requirements to make the most of such increased opportunities.

Second, the digital poses new challenges, conceptual and ethical. Such challenges are open, that is, they are problems that may be solved in more than one way by rational, well-informed, and open-minded (i.e., not stubborn) interlocutors. This openness calls for a clear understanding of how we design the solutions, how such solutions can better satisfy the requirements that motivate them, and how the requirements may evolve and be improved in order to fulfil the driving purposes. In short, it calls for a logic of design and sufficientisation driven by the right policies.

Third, and probably as a consequence of the previous two reasons, a conceptual logic of information as design is particularly important today because our knowledge of the world is increasingly constructionist. I already mentioned the fact that some of the most influential sciences and intellectual endeavours of our time, including Architecture, Computer Science, Economics, and Engineering and Jurisprudence do not just study their systems, they mainly build and modify them.

³² Fawcett (1987) p. 85.

The more science shifts from a *mimetic* to a *poietic* approach to the world the more we shall need a logic of design. Kant and Hegel were on the right track.

And finally, because the essence of philosophy is not logic, but design. Logic is only a second-best compromise, in the absence of design. So it is time for philosophy to equip itself not only with a transcendental and a dialectical logic but also with a logic of requirements to devise new answers to the philosophical questions posed by the information revolution. Philosophical questions are ultimate but not absolute questions, which are not answerable empirically or mathematically, but are open to informed and rational disagreement.³³ The best way to address them is by developing philosophy as conceptual design, and this requires its own logic.

Acknowledgements I discussed previous versions of this article as a speaker at the following meetings: UNILOG'2015 5th World Congress and School on Universal Logic, (Istanbul, 21 September 2014); 9th IA Summit 2015 (Bologna, 23–24 October 2015); *Lectio Commandiniana*, Università degli Studi di Urbino (Urbino, 22 April 2016); *Copernicus Lecture*, Università degli Studi di Ferrara (Ferrara, 28 April 2016); Seminario di Logica, Università degli Studi di Milano (Milano, 4 May 2016); keynote, European Information Architecture summit EuroIA 2017 (Stockholm, 28–30 September 2017). I am indebted to Jean-Yves Béziau; Vincenzo Di Maria and Luca Rosati; Pierluigi Graziani and Gino Tarozzi; Matteo V. d'Alfonso; Marcello D'Agostino and Corrado Sinigaglia; and finally Raffaella Angus and Francis Rowland respectively, for their kind invitations, and to the participants in such meetings for the fruitful opportunity to discuss my ideas and to receive much helpful feedback. Massimo Durante, Giuseppe Primiero, Mariarosaria Taddeo, Sandra Wachter, and James Williams provided some very significant feedback that hugely improved what became the penultimate version of this article. David Sutcliffe kindly copyedited the final draft. Finally, I am grateful to the two anonymous reviewers for their constructive feedback.

References

- Alexander, C. (1964). *Notes on the synthesis of form*. Oxford: Oxford University Press.
- Alexander, C. (1979). *The timeless way of building*. New York: Oxford University Press.
- Alexander, C., Ishikawa, S., & Silverstein, M. (1977). *A pattern language: Towns, buildings, construction*. New York: Oxford University Press.
- Ammon, S. (2017). Why designing is not experimenting: Design methods, epistemic praxis and strategies of knowledge acquisition in architecture. *Philosophy & Technology*. doi:10.1007/s13347-017-0256-4.
- Arnold, A., & Plaice, J. (1994). *Finite transition systems: Semantics of communicating systems*. Paris: Prentice Hall.
- Arup, O. (2012). *Philosophy of design: Essays, 1942–1981*. London: Prestel.
- Auerbach, E. (2003). *Mimesis: The representation of reality in western literature* (50th anniversary ed.). Princeton, NJ: Princeton University Press.
- Barthes, R. (1981). *Camera lucida: Reflections on photography*. New York: Hill and Wang.
- Boca, P. P., Bowen, J. P., & Siddiqi, J. I. (2010). *Formal methods: State of the art and new directions*. London: Springer.
- Burgess, J. P. (2009). *Philosophical logic*. Princeton, NJ: Princeton University Press.
- Buurman, G. M., & Rölli, M. (2016). *The intrinsic logic of design*. Zurich: Niggli.

³³ On philosophical questions as open questions see Floridi (2013b).

- Carus, A. W. (2007). *Carnap and twentieth-century thought: Explication as enlightenment*. Cambridge: Cambridge University Press.
- Covert, A., & Fenton, N. (2014). *How to make sense of any mess*. Createspace Independent Pub.
- Dunne, A., & Raby, F. (2013). *Speculative everything: Design, fiction, and social dreaming*. Cambridge, MA: The MIT Press.
- Fawcett, W. (1987). A note on the logic of design. *Design Studies*, 8(2), 82–87.
- Floridi, L. (1999). *Philosophy and computing: An introduction*. London: Routledge.
- Floridi, L. (2004). On the logical unsolvability of the gettier problem. *Synthese*, 142(1), 61–79.
- Floridi, L. (2008). The method of levels of abstraction. *Minds and Machines*, 18(3), 303–329.
- Floridi, L. (2010). *Information—a very short introduction*. Oxford: Oxford University Press.
- Floridi, L. (2011). *The philosophy of information*. Oxford: Oxford University Press.
- Floridi, L. (2013a). *The ethics of information*. Oxford: Oxford University Press.
- Floridi, L. (2013b). What is a philosophical question? *Metaphilosophy*, 44(3), 195–221.
- Floridi, L. (2014). *The fourth revolution—How the infosphere is reshaping human reality*. Oxford: Oxford University Press.
- Floridi, L. (2016). What a maker's knowledge could be. *Synthese*. doi:10.1007/s11229-016-1232-8.
- Floridi, L. (2017). A plea for non-naturalism as constructionism. *Minds and Machines*, 27(2), 269–285. doi:10.1007/s11023-017-9422-9.
- Floridi, L., & Sanders, J. W. (2005). Internet ethics: The constructionist values of homo poieticus. *The impact of the internet on our moral lives*, 195–214.
- Flusser, V. (1999). *The shape of things: A philosophy of design*. London: Reaktion.
- Galle, P. (1997). Towards a formal logic of design rationalization. *Design Studies*, 18(2), 195–219.
- Goel, V. (1988). Complicating the 'logic of design'. *Design Studies*, 9(4), 229–234.
- Grice, H. P. (1989). *Studies in the way of words*. London: Harvard University Press.
- Heis, J. (2012). Attempts to Rethink Logic. In A. W. Wood & S. S. Hahn (Eds.), *The Cambridge history of philosophy in the nineteenth century (1790–1870)* (pp. 95–132). Cambridge: Cambridge University Press.
- Hoare, C. A. R. (1972). Chapter II: Notes on data structuring. In O. J. Dahl, E. W. Dijkstra, & C. A. R. Hoare (Eds.), *Structured programming* (pp. 83–174). Cambridge: Academic Press Ltd.
- Kelsen, H. (1967). *Pure theory of law* (Translation from the second (revised and enlarged) German edition by Max Knight, see also the translation of the first ed.: Introduction to the problems of legal theory. Clarendon Press, 1992. Berkeley: University of California Press.
- Kroll, E., & Koskela, L. (2015). On Abduction in Design. In J. S. Gero & S. Hanna (Eds.), *Design computing and cognition '14* (pp. 327–344). Cham: Springer.
- Le Goc, M., & Vilar, F. (2017). Operationalization of the blending and the levels of abstraction theories with the timed observations theory. In *Proceedings of the 9th international conference on agents and artificial intelligence (ICAART 2017)*, Vol. 2, pp. 364–373.
- Lewis, C. I. (1929). *Mind and the world-order; outline of a theory of knowledge*. New York: C. Scribner's Sons.
- Lewis, C. I. (1946). *An analysis of knowledge and valuation*. La Salle, IL: Open Court.
- March, L. (1976). *The architecture of form*. Cambridge: Cambridge University Press.
- March, L. (1984). The logic of design. In *Developments in design methodology* (pp. 265–276). Chichester: Wiley.
- Morville, P., Rosenfeld, L., & Rosenfeld, L. (2015). *Information architecture for the world wide web* (4th ed.). Beijing: O'Reilly.
- Munari, B. (1971). *Design as art*. Harmondsworth: Penguin.
- Norman, D. A. (2013). *The design of everyday things* (Revised and expanded ed.). Cambridge, MA: MIT Press.
- Parsons, G. (2016). *The philosophy of design*. Cambridge: Polity.
- Quine, W. (1992). Structure and nature. *The Journal of Philosophy*, 89(1), 6–9.
- Rosenthal, S. B. (1987). The pragmatic a priori: Lewis and Dewey. *The Southern Journal of Philosophy*, 25(1), 109–121.
- Sainsbury, R. M. (2001). *Logical forms: An introduction to philosophical logic* (2nd ed.). Oxford: Basil Blackwell.
- Schön, D. A. (1983). *The reflective practitioner: How professionals think in action*. New York: Basic Books.
- Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge, MA: MIT Press.

- Sinnott-Armstrong, W., & Fogelin, R. J. (2015). *Understanding arguments: An introduction to informal logic* (Ninth edition, Concise ed.). Stamford, CT: Cengage Learning.
- Vermaas, P. E., et al. (2007). *Philosophy and design: From engineering to architecture*. Dordrecht: Springer.
- Walton, D., & Brinton, A. (2016). *Historical foundations of informal logic*. London: Routledge.
- Zeng, Y., & Cheng, G. D. (1991). On the logic of design. *Design Studies*, 12(3), 137–141.