An overview of some ontological challenges in engineering maintenance

Stefano Borgo[1], Francesco Compagno[1,2], Nicola Guarino[1], Claudio Masolo[1], and Emilio M. Sanfilippo[1]

[1] ISTC-CNR Laboratory for Applied Ontology (LOA),
via alla cascata 56/C, Povo, Trento, 38123, Italy
{name.surname}@cnr.it
[2] Adige S.P.A, via per Barco, 11, Levico Terme, 38056, Italy
francesco.compagno@istc.cnr.it

Abstract. Maintenance is an important technical aspect that must be considered in engineering practices. In this paper we present a preliminary ontological investigation of questions such as "What is a component of an engineering system?" and "What happens when a component is replaced after a malfunctioning?", which are both fundamental from a maintenance modeling stance. We focus in particular on two inter-related problems, which we call *the missing component* and *the replacement* problems. We describe different approaches dealing with them. First, we start representing *kinds* of components and systems as temporally qualified first-order logic predicates, eventually reified. We then consider a four-dimensionalist (4D) perspective, mainly based on the ISO 15926. Lastly, we briefly mention a novel point of view based on possible worlds. By the end of the paper, we shortly compare the approaches by discussing their advantages and shortcomings.

Keywords: Maintenance · Ontology .

1 Introduction

In this preliminary research work, we address two inter-related problems relative to the ontological conceptualization of experts' knowledge with respect to engineering maintenance. We call these challenges *the missing component problem* and *the replacement problem*. From the perspective taken in the paper, both challenges regard engineering systems, hereby simply understood as products composed of various inter-related components. Both problems are documented in the literature [2,3,8,11] and emerge from real-world engineering practices. The analysis we propose is part of a study that we are carrying out in collaboration with Adige S.P.A (BLM Group), a company specialized in the manufacturing of laser cutting machines (Fig. 1), being developed in the context of the European H2020 project *OntoCommons: Ontology-driven data for industry commons*.[3] The purpose of the collaboration is to support expert decision making about maintenance procedures through the development of an ontology-based maintenance information system. The examples we shall discuss come from this scenario.

[3] Website: https://ontocommons.eu, last accessed April 2021.

Some approaches to deal with the aforementioned problems have been already proposed. However, each of them assumes its own ontological principles and formal system, so that it is not clear to which extent they (dis-)agree, nor what their (dis-) advantages are. This paper aims therefore at comparing them to support knowledge engineers in selecting one approach over the others.

Fig. 1. An Adige LT8 laser machine. These machines cut metal tubes using a cutting head moving along three axes, together with the translatory and rotatory motion of the tube.

The paper is structured as follows. Section 2 briefly describes and discusses the aforementioned problems. Section 3 compares four different approaches for modeling engineering systems and their components, discussing for each of them their relation with the missing component problem and the replacement problem. Finally, Section 4 concludes and summarizes our contribution.

2 The two problems

The missing component problem. The problem of the missing component has been documented in the literature in various places (see for instance [2,3,11]). Here we present it as specifically concerning maintenance scenarios. In these contexts, indeed, components may be physically removed from their hosting systems, e.g., to be controlled, cleaned or repaired (and sometimes even replaced). However, even when a component is not installed in the system, technicians may refer to it as if it were already in its right place; e.g., they can say *"This cable leads to the laser head"*, "*The laser head is placed in this position*", "*The laser head of this machine has not been installed yet*". This talk seems to presuppose a sort of augmented reality scenario, which in certain situations would indeed be useful (so that it is often artificially created nowadays): for instance, in the context of an assembly task, it may be useful to visualize a component in its expected position, even though it is not physically present. Of course, the technical problem is how to clarify the semantics of the statements above, and, in particular,

how to make sense of the fact that engineers talk and reason about an entity that is not physically present (see Fig. 2 for an example of laser cutting head).

Fig. 2. The LT8 laser cutting head in action.

The replacement problem. Let us consider a particular laser cutting machine, like the one in Fig. 1, that has at time t a protective window[4] installed in the appropriate position within the cutting head, we call it *protective-window*$_1$. Assume that, due to malfunctioning, *protective-window*$_1$ is replaced at time $t+1$ with a new one, *protective-window*$_2$. Technicians who replace the protective window are therefore primarily concerned with two specific physical objects: the defective one and the new one. In some cases, however, the same technicians seem to refer to something else. For instance, when claiming *"The protective window of this cutting head has been replaced 3 times in the last year"*, they clearly don't refer with the expression 'the protective window of this cutting head' to an ordinary physical object (since no physical object was replaced three times), but to some other entity whose ontological nature is unclear.

The reader should notice the strong connection between the two problems just mentioned. In both cases, engineers talk about entities whose nature is different from that of ordinary physical objects; first, by referring to them even when they are not physically present, second, by thinking of them as items that keep their identity even when replaced. One has therefore to bite the bullet and make sense of these entities from an ontological stance. In particular, the two problems above are intimately related to the possibility for a system to lack some component or have a component replaced. This involves subtle notions concerning the existence conditions of engineering systems and their persistence through time.

In engineering practice, the design process commonly ends up with the production of a technical specification (a.k.a. design model). Hence, to understand when a material

[4] The protective window is a particular glass that protects the laser beam optics from, e.g., the metal sparks flying around during a cutting process.

realization of a certain specification physically exists, it is crucial to clarify the conditions by which a physical system *complies* with the specification. For instance, suppose to have an object a that has only some of the components required by the specification. There are at least two possibilities here: (i) no system (of the given kind) exists because the object a does not fully comply with the specification; (ii) a is a system (of a certain kind) because it *sufficiently* complies with the specification. To define what 'sufficiently' means, one may refer to the *relevance* of the components, e.g., it may be necessary for a to have only the *fundamental* components. This approach would however require to characterize fundamental vs. non-fundamental, or even optional, components while being sure that this manner of framing compliance matches well with engineering practices. The possibility of replacing the components of a system makes the overall situation more complex, because (at least in principle) fundamental components can be replaced and maintained, too. During maintenance operations, therefore, a system may lack some of its fundamental components and, thus, stop existing, at least in principle.[5]

3 Overview of approaches

We provide in this section an overview of some ontological approaches that address the problems previously mentioned. We will assume that an engineering system of a certain *kind* (e.g., a mechanical assembly) can be described by means of technical specifications in terms of the *features* of its components and the *relations* holding among them. To depict a specification, we use a graph like the one in Fig. 3, where nodes stand for components (with their characterizing features), and arcs for relations between components. For instance, the graph in Fig. 3 describes a generic system with four components, denoted $\mathbb{C}i$, each characterized by some (complex) feature[6], F_i, and five relationships R_{ij} holding between them.

Generally speaking, the $\mathbb{C}i$s could be recursively specified and further decomposed into components. However, we assume that the $\mathbb{C}i$s do not have proper parts. The main designing activity regards therefore the choice of both the components and the way in which they are structurally related. For the sake of the discussion, we will often refer to the example in Fig. 3, although the analysis is generalizable to arbitrary specifications given in terms of features and relations between components.

[5] A practical choice to avoid this problem (not discussed here) might be to admit that, in a service and maintenance context, the identity of the system to be repaired or maintained is assumed by convention, independently of the status of its components. For instance, we may say that, as long as a certain machine maintains its serial number, and a service contract concerning a machine with that serial number is still operational, the machine to be serviced exists (although possibly in a nonfunctional state), independently of the presence and functional state of its components

[6] We assume that a complex feature is a boolean combination of a property that characterizes the component's kind (say, *pump* or just *physical object*) plus one or more qualitative properties describing a particular shape, size, color, etc. For instance, F_1 could mean that $\mathbb{C}_1$ is a physical object that weighs 2kg $\pm$ 0.1kg, is made of metal or plastic but not wood etc.

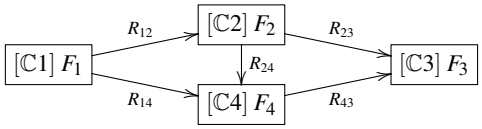

Fig. 3. An example of graph-based technical specification

3.1 Approach 1: system kinds as predicates

In this first approach, the specification of a certain system is conceived as a specification of its most specific kind (i.e., the most specific property it shares with all its duplicates). In turn, the latter consists in stating the necessary and sufficient conditions for being, at a certain time, an instance of that kind. Formally, a temporally indexed monadic predicate $\mathtt{K}_t$ will be used to represent the property of being an instance of kind K at time t. Let us assume that Fig. 3 specifies a system of kind K. The corresponding logical specification would be expressed by (f1), where $o_1 \oplus \ldots \oplus o_n$ denotes the mereological sum of n mutually disjoint objects, and $o_1 \equiv_t o_2$ means that, at time t, the objects o_1 and o_2 have the same parts. Formula (f1) ensures therefore the *full compliance* of all K-systems with respect to the specification in Fig. 3.

f1 $\mathtt{K}_t x \leftrightarrow \exists abcd(x \equiv_t a \oplus b \oplus c \oplus d \wedge \mathtt{F1}_t a \wedge \mathtt{F2}_t b \wedge \mathtt{F3}_t c \wedge \mathtt{F4}_t d \wedge \mathtt{R12}_t ab \wedge \mathtt{R14}_t ad \wedge \mathtt{R23}_t bc \wedge \mathtt{R43}_t dc \wedge \mathtt{R24}_t bd)$

To address the replacement problem, we need now to suitably characterize the tags $\mathbb{C}i$, which denote the unique role played by each component in the system. In general, this role depends on all the relations holding among components, so, for example, $\mathbb{C}1$ would be defined as follows:[7]

f2 $\mathtt{C1}_t x \leftrightarrow \mathtt{F1}_t x \wedge \exists sbcd(s \equiv_t x \oplus b \oplus c \oplus d \wedge \mathtt{F2}_t b \wedge \mathtt{F3}_t c \wedge \mathtt{F4}_t d \wedge \mathtt{R12}_t xb \wedge \mathtt{R14}_t xd \wedge \mathtt{R23}_t bc \wedge \mathtt{R43}_t dc \wedge \mathtt{R24}_t bd)$

However, in some cases it is possible to uniquely characterize a certain $\mathbb{C}i$ just in terms of the relations holding between the i-th component and its neighbor components, as in formula (f3), where $o_1 \leq_t o_2$ stands for 'at time t, the object o_1 is part of the object o_2'.

f3 $\mathtt{C1}_t x \leftrightarrow \mathtt{F1}_t x \wedge \exists sbd(\mathtt{K}_t s \wedge x+b+d \leq_t s \wedge \mathtt{F2}_t b \wedge \mathtt{F4}_t d \wedge \mathtt{R12}_t xb \wedge \mathtt{R14}_t xd)$

Note that, in this particular case, the necessary and sufficient conditions in (f3) may be further weakened out, since $\mathbb{C}1$ is the only component with feature $F1$ in a K-system, but, in general, relational constraints are needed when systems have several components with the same features (which is typically the case). For instance, should we only consider constraints on features, $\mathbb{C}2$ and $\mathbb{C}4$ would turn out to be indistinguishable if $F2 = F4$, the relations they have with the other components are the same (i.e., $R_{12} = R_{14}$ and $R_{23} = R_{43}$), and R_{24} is symmetric.

[7] If a K system is defined by (f1), then (f2) implies that x can be a $\mathbb{C}1$-instance only if there exists a K-system s which has x as a part, i.e., $\mathbb{C}1$-components are existentially dependent on K-systems.

Note also that the lack of a single $\mathbb{C}i$-component could induce the lack of all other kinds of components. In the example in Fig. 3 this happens when a $\mathbb{C}2$- or $\mathbb{C}4$-component is missing. The mutual existential dependencies between components need therefore to be carefully taken into account.

Let us show now how we can address the replacement problem with this approach. Observe first that component definitions can be *relativized* to a specific system by adding a system-argument in the antecedent of the formula and discarding the existential quantification on systems in the consequent; see (f4) representing x as $\mathbb{C}1$-component of the specific system s. The replacement of the $\mathbb{C}1$-component of s can be then represented as in (f5).

f4 $\mathtt{C1}_t xs \leftrightarrow \mathtt{F1}_t x \wedge \exists bd(\mathtt{K}_t s \wedge x{+}b{+}d \leq_t s \wedge \mathtt{F2}_t b \wedge \mathtt{F2}_t d \wedge \mathtt{R12}_t xb \wedge \mathtt{R14}_t xd)$
f5 $\mathtt{C1}_{t_0} xs \wedge \mathtt{C1}_{t_1} ys \wedge \mathtt{C1}_{t_2} zs \wedge x \neq y \neq z$

What (f5) says is that the role of 'being the $\mathbb{C}1$-component of s' is played by three different objects at different times. Note that no single individual corresponding to the expression 'the $\mathbb{C}1$-component of s' is present in the domain of quantification.

However, this approach does not allow dealing with the missing component problem in the strict sense, since only the actual physical components are included in the domain of discourse. On the other hand, however, having isolated the properties of system-kinds and their components, one can talk of systems and their components in general without pointing to their physical realizations. According to the defenders of this approach, this view seems close to design practices where experts commonly talk about and reason over design specifications and their background knowledge without necessarily pointing to physical items.

3.2 Approach 2: system-kinds as individual constants

The approach considered in this section is conceptually similar to the previous one, the main difference being technical. System-kinds are indeed not represented via predicates but by means of individual constants: $\mathtt{K}_t x$ is now replaced by $x ::_t \mathtt{k}$ where $\mathtt{k}$ represents the kind K and :: stands for the *instantiation* relation so that $x ::_t \mathtt{k}$ reads 'x is an instance of kind K.' Similarly for relations, e.g., $\mathtt{R}_t xy$ is replaced by $xy ::_t \mathtt{r}$ where the *new* instantiation relation has four arguments (i.e., x, y, t and $\mathtt{r}$).[8] We can follow the discussion in Sect.3.1 by observing that the previous formulas can be rewritten in this new framework; e.g., (f1) can be rewritten as in (f6).

f6 $x ::_t \mathtt{k} \leftrightarrow \exists abcd(x \equiv_t a \oplus b \oplus c \oplus d \wedge a ::_t \mathtt{f1} \wedge b ::_t \mathtt{f2} \wedge c ::_t \mathtt{f3} \wedge d ::_t \mathtt{f2} \wedge$
$ab ::_t \mathtt{r12} \wedge ad ::_t \mathtt{r12} \wedge bc ::_t \mathtt{r23} \wedge dc ::_t \mathtt{r23} \wedge bd ::_t \mathtt{r24})$

The introduction of system-kinds in the domain of quantification allows taking into account their *intensional* and *intentional* dimensions. Different kinds could have the same instances and their difference can be grounded on some meta-information, e.g., that a design feature (or an entire specification) has been designed by an engineer working in the company, therefore copyrights apply to it. Furthermore, rather than relying on formulas like (f6) or on the counterpart of (f5), system-kinds can be considered as

[8] For simplicity we write in the same way all the instantiation primitives.

composed by their component-kinds (e.g., by writing $\mathtt{k} = \mathtt{c1} + \mathtt{c2} + \mathtt{c3} + \mathtt{c4}$), which is an approach similar to what done in formal ontology with respect to the debate on structural universals [9]. In this view the mereological structure of a system-kind is aligned with the one of its instances; see (f7)-(f8).

f7 $x ::_t k \wedge c \leq k \rightarrow \exists y(y ::_t c \wedge y \leq_t x)$
f8 $x ::_t k \wedge y ::_t c \wedge y \leq_t x \rightarrow c \leq k$

This idea can be pushed further towards the introduction of component-kinds *relativized* to a specific system. These component-kinds—that indeed, at a given time, can have only one instance —can be intended as the subject of replacements, i.e., they can have different realizations (instances) through time. Furthermore, they can be deployed also to address the missing component problem because their existence is independent from the one of their material instances, i.e., they can be 'empty' (i.e., not instantiated) at some times. However, in order to exist they require the whole system to exist, therefore one needs to accept partial compliance with the drawbacks discussed before.

3.3 Approach 3: systems as four-dimensionalism objects

Four-dimensionalism (4D) is the philosophical perspective according to which the objects of everyday experience have both spatial and temporal parts. To understand this, let us shortly comment on the opposite position, called three-dimensionalism (3D). According to 3D, objects have only spatial parts; e.g., a laser cutting machine has a cutting head, a laser source, a water chiller, etc. The machine can lose and acquire parts, but whenever it is present in time, it consists of all parts that it has at that time. Differently, for four-dimensionalists an object x exists at a time t if and only if there exists its temporal slice $x_{@t}$ (which is present only at t). That is, if my laser cutting machine c exists at both t and t', it has two different temporal slices at those times, i.e., $c_{@t}$ and $c_{@t'}$, respectively. A whole object consists therefore in the mereological sum of all its temporal parts.[9]

In the landscape of applied ontology, West [11] has developed a 4D-approach for engineering which has eventually led to the standard ISO 15926 [5]. Similarly to Approach 1 (see Section 3.1), according to this theory, system-kinds are intended in an extensional way but both physical systems and their components are now conceived as 4D-objects in the sense just introduced. Following the 4D approach, a proposition $\mathtt{P}_t x$, which reads 'x satisfies P at t', can be reduced to $\mathtt{P}x_{@t}$, i.e., 'the temporal slice of x existing at and only at t satisfies P'. In the same line, (f1) can be rewritten as in (f9).

f9 $\mathtt{K}_t x \leftrightarrow \exists abcd(x_{@t} = a_{@t} \oplus b_{@t} \oplus c_{@t} \oplus d_{@t} \wedge \mathtt{F1}a_{@t} \wedge \mathtt{F2}b_{@t} \wedge \mathtt{F3}c_{@t} \wedge \mathtt{F2}d_{@t} \wedge$
$\mathtt{R12}a_{@t}b_{@t} \wedge \mathtt{R12}a_{@t}d_{@t} \wedge \mathtt{R23}b_{@t}c_{@t} \wedge \mathtt{R23}d_{@t}c_{@t} \wedge \mathtt{R24}b_{@t}d_{@t})$

Looking at West's position more deeply, he argues that systems' components (1) are existentially dependent on the systems they are part of, and (2) are non-ordinary physical objects which can undergo moments of non-existence. Recall the example of a particular LT8 cutting machine and consider its cutting head. According to West's

[9] Admittedly, this is a simple way to look at the distinction between 3D and 4D. The reader can refer to [4] for more information.

first claim, the cutting head exists as a system component of the whole machine only when the latter exists. In this sense, *being a system component* is a sort of *role property* which an object might satisfy at some time. For the second claim, the idea is that a component of a certain system s is the mereological sum of all the temporal slices of the ordinary objects which have played the corresponding component-role in relation to s. For instance, the $\mathbb{C}1$-component of a system s can be defined as in (f10) (note that the formula makes explicit the existential dependence of components on the system s).

f10 $\mathtt{C1}xs \leftrightarrow x = \sigma y(\exists tz(y = z_{@t} \wedge y \leq s \wedge \mathtt{C1}y))$

By (f10), x is a single individual that represents the succession of all the (temporal slices of the) ordinary objects playing the $\mathbb{C}1$-role. This is an interesting move to deal with the replacement problem. According to West, indeed, the 4D-component of a system (the object x in (f10)) is the sum of the temporal slices of the different 'ordinary' objects (the slices $z_{@t}$ in (f10)) that are placed in the system during its existence. It follows, as noted by West, that components' existence can be intermittent, i.e., when no ordinary object plays the $\mathbb{C}1$-role in the system s, the $\mathbb{C}1$-component of s does not exist. Hence, in principle, also the whole system s can intermittently exist, e.g., when one of its (fundamental) components is missing (a fact which is reflected in (f9)). However, it should be clear that in order to talk about missing components of a system s, one needs to include a notion of partial compliance into the framework in such a way to guarantee the existence of the system s even when it lacks components. For, a component can be said to be missing at some time t only if the system of which it is a component exists at t notwithstanding it lacks parts.

3.4 Approach 4: Embracing possibilism

Let us now briefly introduce a fourth approach based on some former work [3], which is however still a research hypothesis that deserves a proper formalization. Differently from Approach 1, we assume here that the C_i do not denote properties, but specific parts of a certain system. The tags are therefore *part names*, as usual when describing assemblies, and not property names. We also assume that a system s is an individual that exists even at design time, when the system has not been physically realized yet. Of course, we need to characterize the ontological nature of these entities, which turns out to be definitely non-standard.

We shall consider *design objects* as *possible* (although non-actual) physical objects. Under this view, designing an object means *choosing* a possible physical object among many others, and *describing* it. For example, the design object satisfying the specification of Fig. 3 would be simply described as follows(where s, c_1, c_2, c_3, c_4 are constants denoting distinct elements of the domain of discourse);

f11 $s = c_1 \oplus c_2 \oplus c_3 \oplus c_4 \wedge \mathtt{F1}c_1 \wedge \mathtt{F2}c_2 \wedge \mathtt{F3}c_3 \wedge \mathtt{F4}c_4 \wedge$
$\mathtt{R}_{12}c_1c_2 \wedge \mathtt{R}_{14}c_1c_4 \wedge \mathtt{R}_{23}c_2c_3 \wedge \mathtt{R}_{43}c_4c_3 \wedge \mathtt{R}_{24}c_2c_4$

Differently from the previous cases, where the formulas fully specify the *conditions* for an object to be of a particular kind K, corresponding to a certain product model, this formula simply presupposes that a certain object s exists in the mind of the designer,

and provides a description of its nature in terms of the properties (P_i) of its components and the relations (R_{ij}) among them. So, under this view, the content of a design activity (i.e., *what* is designed) is an individual and not a kind. The corresponding kind K would be defined as the most specific property shared by all *duplicates* of s, i.e. (forgetting tolerances), by all individuals which share s's intrinsic properties.

To account for the peculiar way design objects exist, we shall adopt Lewis' possibilist realism [7,12], according to which a possible (physical) object is simply a part of a maximally spatiotemporally related whole, i.e., a whole among whose parts spatiotemporally relations hold. More exactly, forgetting temporal relations for the sake of simplicity, we shall assume that a possible physical object is an object such that a certain *spatial distance* relation holds among its parts (this means that all the relations occurring in Fig. 3 imply some spatial relation between their arguments). Under this view, objects are *physical* not because they are located in a region of space, but simply because they are *extended in space*, in the sense that spatial relations hold between their parts. Of course, they may also have a location, but only relative to some other physical objects.

Following this line of thought, we shall think of a design object as a maximally spatially related whole. Forgetting temporality, this means that a design object is just a possible world, in Lewis' terms. So, we can think that formula (f11) holds in a world (different from the actual one) where only an instance of K exists. For example, a laser cutting machine that has been designed but has not been built yet is something like a possible object floating in the void, i.e., a world of which it is the only inhabitant.

We shall assume that the *actual physical world* is the largest spatially related whole that includes us. For a physical object to exist means to be part of some maximal spatially related whole, and to actually exist means to be part of the actual world. So, we can have a physical object that exists, but it is not actual. If, in addition, this object is the content of an intentional design act (whose author is an agent who inhabits the actual world), then it is a design object: design objects exist but they are not actual.

We have therefore a solution to a variant of the missing component problem: when engineers talk of a certain component within a system (possibly even before the system is realized), they may talk of it in a generic way, as in the statement 'the protective window is a part of the cutting head of the LT8 machine.' In this case we shall say that they are talking of a design object, i.e., of a physical object that exists in a non-actual world and is the content of an intentional design act.

In most cases, however, the missing component problem concerns a specific, actually existing system, one of whose components has been removed. To address this case we must discuss the relationship between design objects and actual objects. We shall say that an object x is a *realization* of a design object y iff x is a duplicate of y (forgetting tolerance) and x exists in the actual world, that is, the spatial distance of its parts from the other objects belonging to the actual world is defined. We shall assume that, when engineers talk of a missing component as if it was there, they talk of a *virtual* entity, which is a sort of *projection* of a design system's component on the actual world: it is a duplicate of that component which is assumed to have a location (and therefore exist) in the actual world, although it is not *real*, since it is not capable of causal interactions with the actual world. The actual world can host therefore both *real* and *virtual* entities. The

latter are intentional entities (like design objects), which are existentially dependent on one or more agents. In conclusion, when technicians talk of a missing component as if it were there (e.g., using the expression *'This frame supports the cutting head's protective window'* they refer to a virtual component, which is a projection of a design object on the actual world. Such a virtual component may be visualized by using augmented reality techniques.

Finally, let us briefly mention how the replacement problem may be addressed in this approach. The problem is the reference of a certain identifier *i*, which in our example is *'the protective window of this cutting head'*, used to talk of something related to a real object existing in the actual world. Assuming that *i* denotes a virtual object does not work, since when the protective window is installed we expect *i* to refer to the real window. On the other hand, one cannot assume that the *i* denotes a real physical object, since the reference link would break as soon as the object is removed.

A solution is to assume that the identifier refers to a *variable object*, i.e., following Fine [1], an object that is a *variable embodiment* of other objects. Suppose to have a function f that, at each time when an object exists, tells us what its parts are. Such object is called the *variable embodiment of f*, and the sum of its parts at t is called its *manifestation* at t. In our case, we can assume that *'the protective window of the cutting head'* denotes a variable embodiment such that, when a physical protective window is installed, its manifestation is just that physical window. When the physical window is removed, its manifestation is the virtual window.

4 Discussion and conclusions

The paper presented an initial analysis of ontological problems in the maintenance domain. In particular, it addressed what happens in the world when system components are missing or replaced. These problems are inextricably intertwined with compliance conditions, as one must consider how a system can be compliant with its specification when some of its components are not in place, and how it survives the replacement of some components.

We have presented four different modeling approaches with the purpose of analyzing their ontological commitments and modeling (dis-)advantages. The first two (Sect. 3.1 and Sect. 3.2) are very similar from a conceptual perspective in that system kinds and their characterizing features (all represented through design specifications) stand for *properties* that particular physical systems satisfy (if compliant at some suitable degree). In both approaches, dealing with the missing component and the replacement problems means looking at the physical items that, at different times, do physically exist. As we have seen, a sentence like *'the protective window of this cutting head has been replaced twice'* needs to be rephrased in something like *'this cutting head had three different protective windows at different times.'* As said, the only difference between the two approaches is that in the second one properties are reified. Among other advantages, we have noticed that this choice allows one to deal with the missing component problem, because one can (at least) point to the reified properties that the missing component is meant to satisfy, although, again, there is no way to account for the superficial semantics of sentences like *This cable leads to the laser head*.

The third approach (see Sect. 3.3), based on West [11], relies on a 4D ontological framework. Similarly to Approach 1, system-kinds, their components, and features are (first-order) properties that particular entities satisfy. Hence, although West does not explicitly address the representation of compliance conditions, the considerations done in Sect. 3.1 and Sect. 3.2 could be tuned to this approach by adopting a 4D representational system. For the replacement problem, this approach offers an interesting solution. Indeed, a whole 4D-component of a certain system can well correspond to the mereological sum of the temporal slices of different physical objects, which (at different times) are linked to that system. Similarly to the previous cases, the missing component problem remains problematic: if at a certain time there is no temporal slice in the world to which one can point to, there is no 4D-component altogether at that time.

Finally, the paper (preliminarily) introduced a fourth approach (Sect. 3.4), which relies on the introduction of a new type of objects, namely design objects, which depend on the intention of an agent, have a spatial extension, and yet are not actual (i.e., they do not physically exist in the common-sense reality). Differently from the previous approaches where formulas represent system kinds (as represented in specifications), this one is mainly focused on design objects, logically treated as possible individuals that may be *realized* in the actual world, originating real individuals, or just 'projected' onto the actual world, originating virtual individuals. From this perspective, when a real component is missing from a larger system, one can still refer to its virtual counterpart in the physical world. The replacement problem remains more challenging and we discussed a way to deal with it based on Fine's [1] theory on variable embodiments.

To conclude, the results of the paper are still preliminary and further work is necessary at both the conceptual and representational levels. All approaches have their (dis-) advantages. For instance, the 4D perspective, adopted in the third approach, is not very popular in applied ontology. The fourth approach is appealing but it commits to modal realism, whose assumptions are hotly debated in analytic ontology [12]. In addition, since design objects are individual entities, this approach may face some problems when dealing with design tolerances and variants, which are however crucial in engineering design, since a design specification is typically underdetermined. From this perspective, the first two approaches are more flexible; in addition, they both rely on a 3D ontological framework that is standardly used in engineering (see, e.g., [6,10] and the references thereby quoted). Detailing the engineering use case may help to set a suitable benchmark for comparing the approaches and their consequences on related notions.

Acknowledgments: the research presented in the paper is partially supported by the CSA European project H2020 *OntoCommons: Ontology-driven data for industry commons* (GA 958371).

References

1. Fine, K.: Things and their parts. Midwest Studies in Philosophy **23**(1), 61–74 (1999)
2. Galle, P.: Candidate worldviews for design theory. Design Studies **29**(3), 267–303 (2008)
3. Guarino, N.: Artefactual systems, missing components and replaceability. In: Artefact Kinds, pp. 191–206. Springer (2014)
4. Hawley, K.: Temporal Parts. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University (2020)
5. ISO, I.: 15926: Industrial automation systems and integration–integration of life-cycle data for process plants including oil and gas production facilities. Tech. rep., ISO (2004)
6. Karray, M.H., Ameri, F., Hodkiewicz, M., Louge, T.: Romain: Towards a BFO compliant reference ontology for industrial maintenance. Applied Ontology **14**(2), 155–177 (2019)
7. Lewis, D.: On the plurality of worlds, vol. 322. Oxford Blackwell (1986)
8. Masolo, C., Sanfilippo, E.M.: Technical artefact theories: A comparative study and a new empirical approach. Review of Philosophy and Psychology **11**(4), 831–858 (2020)
9. Masolo, C., Vieu, L.: Graph-based approaches to structural universals and complex states of affairs. In: Formal ontology in information systems. Proceedings of the 10th International conference (FOIS). pp. 69–82. IOS press (2018)
10. Sanfilippo, E., Kitamura, Y., Young, R.I.: Formal ontologies in manufacturing. Applied Ontology **14**(2), 119–125 (2019)
11. West, M.: Developing high quality data models. Elsevier (2011)
12. Yagisawa, T.: Possible Objects. In: Zalta, E.N. (ed.) The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University (2018)