Comparing Ontological Alternatives to Model Engineering Systems and Components

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

[1] ISTC-CNR Laboratory for Applied Ontology, via alla cascata 56/C, 38123, Povo, Italy
[2] Adige S.P.A, via per Barco, 11, Levico Terme, 38056, Italy

Abstract

In this paper, we study from an ontological viewpoint some modeling challenges related to the design, realization, and maintenance of engineering systems such as what it means that an assembly complies to a specification and how it persists through time when some of its components are missing or replaced. These challenges intertwine with *the missing component* and *the replacement* problems, which are by themselves fundamental from an engineering stance and are thus explicitly addressed in the paper. We describe these topics and for each one we compare different modeling approaches dealing with them, considering also the differences between the three- and four-dimensionalist views. In conclusion, we summarize the comparison of the approaches by highlighting their advantages and shortcomings from an engineering perspective.

Keywords

Engineering systems, Maintenance, Assembly modeling, Engineering design, Ontology, 3D, 4D

1. Introduction

This paper continues a line of research started in [1]. The goal is to systematically assess a number of ontological models which focus on the same engineering scenario. This goal is achieved by formalizing the given scenario according to different ontological viewpoints, and by technically comparing the obtained formalizations. In this way, one acquires information on the formal and conceptual consequences of adopting an ontological perspective as well as on the technical aspects in which these perspectives differ. Beside highlighting advantages and disadvantages of each ontological model, this analysis can help knowledge engineers and practitioners to choose the (foundational, top-level) ontology most suited to their interest and purposes.

More specifically, our research line concentrates on the design, realization, and maintenance of engineering *systems*, intended here as mereologically compound devices, i.e., assemblies of inter-related components. The paper considers only general modeling approaches which are well-known and widely used in ontology research. This allows us to make general assessments which hold in general and shed light on commonalities across models that adopt the same ontological perspective. Even though the analysis we present remains general for the given purposes, our

FOMI 2021: 11th International Workshop on Formal Ontologies meet Industry, held at JOWO 2021: Episode VII The Bolzano Summer of Knowledge, September 11–18 , 2021, Bolzano, Italy
francesco.compagno@loa.istc.cnr.it (F. Compagno); stefano.borgo@cnr.it (S. Borgo); claudio.masolo@cnr.it (C. Masolo); emilio.sanfilippo@cnr.it (E. M. Sanfilippo)

work is driven by real industrial scenarios that we analyze in collaboration with Adige S.P.A (of the BLM Group[1]), a company specialized in the manufacturing of laser cutting machines.

Our analysis applies to systems which are characterized by a *specification*. Such specification is the result of a design process and identifies a set of conditions that the system must satisfy.

A first challenge concerns the formal representation of these specifications and, related to it, the clarification of what it means that a system must satisfy the conditions to be *compliant* with the specification. A second challenge concerns the manner in which systems persist through time. In industrial scenarios, there may be periods of time in which a system is not operative, e.g., because it undergoes maintenance operations, which could have been scheduled earlier or due to unexpected malfunctioning of some of its components. Maintenance can also end up with the replacement of some components. Therefore, the modeler has to state whether during these periods of maintenance the physical system still complies (perhaps in a weaker way) with its specification, and to which kinds of components replacements it can survive. From an ontological perspective, this aspect is intimately related to the spatio-temporal nature of systems that one adopts. For instance, the standard ISO 15926 [2] embraces a four-dimensionalism (4D) view. According to this view, systems persist through time by accumulating temporal parts. In a three-dimensionalism (3D) view, one assumes that systems persist by being wholly present at each time in which they exist. The 4D and 3D views are basic ontological distinctions that have practical consequences on the models one obtain. We will analyze how these differences impact the representation of systems' compliance through time.

Starting from this general stand, we will pay particular attention to two more specific problems in engineering modeling considered in the literature, e.g. [3, 4, 5]. The first problem, i.e., the *replacement problem*, concerns components replacement. In experts' terms, it is common to claim that a system's component can be replaced *n*-times, as if a component were something that could be entirely substituted while keeping its identity. Understanding what components are in this view becomes therefore crucial to make sense of experts' talks. The second problem, called *missing component problem*, concerns reference to systems or components when they are not physically present or even realized. This can happen at design time, e.g., when experts talk about the machine that they are designing claiming that, e.g., it has given dimensions (within certain tolerances). A similar situation occurs during maintenance tasks, when components are removed from their hosting systems, e.g., to be controlled, cleaned or repaired. During these periods, or when a component has never been installed like in a system under construction, experts may address the component as if it were present in its position; e.g., they may assemble the system while referring to how the missing component should be connected to other systems' parts. Again, we wonder how to make sense of experts' talks from an ontological perspective; e.g., whether it is necessary to introduce a class for engineering items which exist without being physically present. Of course, one can always take a revisionist approach, i.e., claiming that engineers' expressions, when taken ontologically, should be opportunely rephrased.Yet, before considering this option, one should take engineers' talk at face value since this is the information actually exchanged and needed in engineering.

The paper is structured as follows. Section 2 introduces the notation and key notions that we use in the formal discussion like feature, component, and specification. Section 3 presents

[1] https://www.blmgroup.com/

alternative ways to formalize engineering notions and their usage in particular from the 3D vs. 4D ontological viewpoints. Section 4 draws some conclusions.

2. System's Specifications

We introduce in this section the general view and notation that we will use in section 3 to discuss systems from different ontological perspectives.

By engineering systems we mean engineering assemblies, i.e., physical objects based on an explicit design and composed of several interrelated components. To exemplify the presentation, we represent the structure of systems-*types* as (connected) graphs like the one depicted in Fig. 1. Each node represents a systems' component characterized in terms of (*complex*) *features*, labeled F*i*. A feature F*i* is a Boolean combination of (non-relational) properties that may include types (e.g., *pump*, *hole*) and/or properties describing shape, size, color, etc.[2] For instance, we could have that feature F1 characterizes physical objects with weigh 2kg $\pm$ 0.1kg, made of metal or plastic but not wood, with a rectangular slot, etc. Usually, a feature does not identify a unique object; this is necessarily true when the feature is associated with a replaceable component. Arcs between nodes denote *relations*, labeled R*i*, holding between the corresponding components.[3] For simplicity, we consider only binary relations, however, the framework is general and applies to *n*-ary relations as well. Thus, Fig. 1 shows a specification of systems with three components (one for each node) characterized by features F1, F2 and F3, respectively, and satisfying relations R1 and R2 as shown.

Figure 1: A system-type with three components.

To avoid ambiguities, engineers tend to use tags to securely identify each component in a system. These tags are useful for maintenance purposes, e.g., to exactly identify the component upon which a technician needs to operate. Fig. 2 shows a graph similar to the one in Fig. 1 but now enriched with tags $\mathbb{C}1$, $\mathbb{C}2$, and $\mathbb{C}3$. The tags are assumed to be unique identifiers of components in the system. Note that in some graphs components' features cannot explicitly play an identification role. To see this, consider the diagram in Fig. 3. Here, the components $\mathbb{C}2$ and $\mathbb{C}4$ are characterized by the same feature and are distinguished only because the relation R3 is asymmetric.

The components $\mathbb{C}i$ (like the features F_i) could be recursively specified and further decomposed. For simplicity, however, we will work with $\mathbb{C}i$ components that do not have proper parts. It follows that the components of a system do not have common parts, i.e., they are mutually disjoint.

[2] Note that we use the term 'feature' in a more general sense in comparison to the engineering literature, where it commonly refers to things such as holes, slots, bumps, etc. described from different viewpoints [6].

[3] Things like wires, pipes, or sets of bolts and nuts can be explicitly taken into account by designers, hence they can be treated as components linked to other components via specific relations. An example of such a relation is the contact between a pin and its socket. Alternatively, designers can (partially) encapsulate them in the R*i*-relations.

Figure 2: The graph specification in Fig. 1 with tags ℂ1, ℂ2, ℂ3.

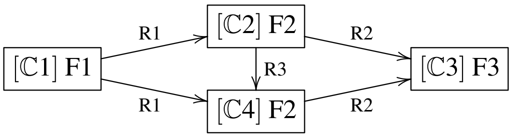

Figure 3: A graph specification where the components ℂ2 and ℂ4 have the same feature F2.

Also, we will refer in the following sections to the example in Fig. 3, although the analysis applies to arbitrary graphs of components/features and relations between them.

3. Modeling Options for Engineering

The purpose of this section is to make explicit the underlying assumptions that different ontological approaches require to model the problems raised in the introduction.

3.1. 3D vs. 4D Approaches

The way the 3D and the 4D approaches understand and model systems is definitely different. The difference is evident when we look at how systems and components relate to time. Let us represent the existence of objects *in time* with the primitive $\mathtt{EX}_t x$ standing for "at time t, the object x is present".[4] We focus here only on objects that exist at some time, i.e., we assume formula (f1) stating that for every object x there exists a time t at which it exists.[5]

f1 $\exists t(\mathtt{EX}_t x)$

According to 3D, objects persist in time by enduring, i.e., by being wholly present at every time at which they exist. To account for change, 3D qualifies properties and relations on objects with respect to time. For parthood we introduce $x \leq_t y$ ("at time t, the object x is part of the object y") characterized by (f2)-(f5), where the *overlap* relation $\between$ is defined in (d1) and holds between two objects at the times at which they have a common part. At time t, two objects *coincide* if, at t, they have the same parts (d2).[6] Parthood *simpliciter* can be defined as in (d3) which allows the introduction of the mereological sum of a plurality of objects—we write $x_1 + \ldots + x_n$ for the mereological sum of n objects—and the sum (called fusion) of all the objects satisfying some property (in the language)—we write $\sigma x(\phi(x))$ for the fusion of the objects satisfying property ϕ.

[4]We write $\mathtt{P}_t x$ instead of $\mathtt{P}xt$ to highlight the time-argument t.

[5]Since it is clear from the language when a variable is quantified over time or over objects, we leave implicit these quantification constraints.

[6]Neither coincidence at t nor coincidence during the whole lives of objects imply their identity.

A mereological system is called *closed* when it assumes the existence (and uniqueness) of the sum of any (finite) plurality of objects and of any fusion based on properties in the language [7].

d1 $x \between_t y \triangleq \exists z(z \leq_t x \wedge z \leq_t y)$

d2 $x \equiv_t y \triangleq \forall z(z \leq_t x \leftrightarrow z \leq_t y)$

d3 $x \leq y \triangleq \forall t(\mathtt{EX}_t x \rightarrow x \leq_t y)$

f2 $x \leq_t y \rightarrow \mathtt{EX}_t x \wedge \mathtt{EX}_t y$

f3 $\mathtt{EX}_t x \rightarrow x \leq_t x$

f4 $x \leq_t y \wedge y \leq_t z \rightarrow x \leq_t z$

f5 $\mathtt{EX}_t x \wedge \mathtt{EX}_t y \wedge \neg x \leq_t y \rightarrow \exists z(z \leq_t x \wedge \neg z \between_t y)$

Regarding 4D within the engineering literature, we refer to West's work [3],[7] which follows closely the standard ISO 15926 [2]. More generally, 4D is the thesis for which objects consist of both spatial and temporal parts, i.e., they persist in time as they extend in space. This hypothesis allows 4D models to take a classical (atemporal) extensional mereology axiomatized as in (f6)-(f10) with overlap defined as in (d4). According to Sider [9], 4D can be characterized by introducing the notion of *temporal part* (*temporal slice*) as in (d5)—here $\mathtt{TP}_t xy$ stands for "at time t, the object x is a temporal part of the object y"—and by assuming that at every time at which an object exists it has a temporal part, as stated in (f11). From (f1) and (f6)-(f11) it is possible to prove that the temporal part of an object at a given time is unique. To simplify the notation, when x exists at t, we write $x_{@t}$ to indicate the unique object y such that $\mathtt{TP}_t yx$. In this view, objects consist of the mereological sum of their temporal parts. For instance, if x exists only at t and $\bar{t}$, it has two different temporal parts at those times, i.e., respectively, $x_{@t}$ and $x_{@\bar{t}}$, and x itself is equal to their sum: $x = x_{@t} + x_{@\bar{t}}$.

4D allows for the definition of temporary parthood as in (d6), i.e., $x_{@t} \leq y_{@t}$. This strategy can be replicated for all temporally qualified predicates, i.e., $\mathtt{R}_t x_1 \ldots x_n$ reduces to $\mathtt{R} x_{1@t} \ldots x_{n@t}$. Furthermore, the temporal coincidence $x \equiv_t y$ reduces to identity of the corresponding temporal parts, i.e., $x_{@t} = y_{@t}$.

Notice that, starting from (f1) and (f6)-(f11), all the axioms considered for the temporally qualified parthood can be proven. The vice versa does not hold: 3D does not commit to temporal parts. In this sense 4D makes a heavier ontological commitment with respect to 3D (see [10]).

d4 $x \between y \triangleq \exists z(z \leq x \wedge z \leq y)$

d5 $\mathtt{TP}_t xy \triangleq \mathtt{EX}_t x \wedge \mathtt{EX}_t y \wedge \neg\exists\bar{t}(\mathtt{EX}_{\bar{t}} x \wedge t \neq \bar{t}) \wedge x \leq y \wedge \forall z(z \leq y \wedge \mathtt{EX}_t z \rightarrow z \between x)$

d6 $x \leq_t y \triangleq \exists uv(\mathtt{TP}_t ux \wedge \mathtt{TP}_t vy \wedge u \leq v)$

f6 $x \leq y \wedge \mathtt{EX}_t x \rightarrow \mathtt{EX}_t y$

f7 $x \leq x$

f8 $x \leq y \wedge y \leq x \rightarrow x = y$

f9 $x \leq y \wedge y \leq z \rightarrow x \leq z$

[7] The reader can refer to [8] for readings on 4D.

f10 $\neg x \leq y \rightarrow \exists z(z \leq x \wedge \neg(z \between y))$

f11 $\mathtt{EX}_t x \rightarrow \exists y(\mathtt{TP}_t yx)$

3.2. Specification and Compliance

We now consider how to logically represent the graph specifications introduced in Sect. 2 and the compliance of individual systems with them.

A common approach consists in embracing 3D and seeing the specifications as (complex) relations represented in first-order logic (FOL) via predicates. Let us assume that Fig. 3 depicts the specification of a target system-type K. To state that four specific components, say x, y, z and v, at a certain time t, satisfy the constraints in the specification we can use a single complex relation $\mathtt{S}$, see (f12).[8] The component types can be characterized by introducing a specific $\mathtt{S}$-relational role; see, e.g., (f13) for $\mathbb{C}1$. Following standard practice, one can introduce necessary and sufficient conditions for *being a K* (at a given time) as in (f14) where $\mathtt{K}_t x$ is read as "at time t, the physical object x is an instance of the type K". Since we adopted a closed mereology, (f13) and (f14) guarantee that an entity x is a $\mathbb{C}1$-instance only if there exists a K-system which has x among its components, i.e., a $\mathbb{C}1$-component implies the existence of a K-system. In other words, $\mathbb{C}1$-components are existentially dependent on K-systems.

Introducing formula (f13) for $\mathbb{C}2$-$\mathbb{C}4$, too, one can trivially prove (f15), i.e., K-systems are the mereological sums of the $\mathbb{C}i$-components considered in the specification. However the converse of (f15) does not hold in general. For instance, assume that $\mathtt{C1}_t a$ holds for $\mathtt{S}_t a\bar{b}cd$ only, while $\mathtt{C1}_t b$ holds for $\mathtt{S}_t \bar{a}bcd$ only (with $a \neq \bar{a}$ and $b \neq \bar{b}$), then there is no suitable 4-tuple satisfying $\mathtt{S}_t$ to infer the converse of (f15).

f12 $\mathtt{S}_t xyzv \leftrightarrow \mathtt{F1}_t x \wedge \mathtt{F2}_t y \wedge \mathtt{F3}_t z \wedge \mathtt{F2}_t v \wedge \mathtt{R1}_t xy \wedge \mathtt{R1}_t xv \wedge \mathtt{R2}_t yz \wedge \mathtt{R2}_t vz \wedge \mathtt{R3}_t yv \wedge$
$\neg(x \between_t y) \wedge \neg(x \between_t z) \wedge \neg(x \between_t v) \wedge \neg(y \between_t z) \wedge \neg(y \between_t v) \wedge \neg(z \between_t v)$

f13 $\mathtt{C1}_t x \leftrightarrow \exists bcd(\mathtt{S}_t xbcd)$

f14 $\mathtt{K}_t x \leftrightarrow \exists abcd(\mathtt{S}_t abcd \wedge x \equiv_t a + b + c + d)$

f15 $\mathtt{K}_t x \rightarrow \exists abcd(\mathtt{C1}_t a \wedge \mathtt{C2}_t b \wedge \mathtt{C3}_t c \wedge \mathtt{C4}_t d \wedge x \equiv_t a + b + c + d)$

In a 4D setting, the previous formulas can be rewritten with atemporal predicates following what done for parthood in (d6); e.g., (f14) can be rewritten as in (f16). Given this observation, we will consider in the remaining of the paper only temporally qualified predicates and discuss 4D only for some specific problems mainly concerning components (e.g., in Sect. 3.3).

f16 $\mathtt{K}_t x \leftrightarrow \exists abcd(\mathtt{S}a_{@t}b_{@t}c_{@t}d_{@t} \wedge x_{@t} = (a+b+c+d)_{@t})$

Another approach in conceptual modeling [11], even though less common than the ones just presented, considers specifications as complex relations represented by individual constants. In this view, the predicate $\mathtt{S}_t xyzu$ is replaced by $xyzu ::_t \mathtt{s}$ where $\mathtt{s}$ represents the complex relation S and $::$ is a sort of *instantiation* relation: here $xyzu ::_t \mathtt{s}$ stands for "at time t, objects x, y, z, u satisfy the relation S". For simplicity, the different instantiation primitives corresponding to the arities of

[8] We add the non-overalapping conditions for generality. Since by assumption our components do not have proper parts, strictly speaking these constraints are not needed.

predicates are noted in the same way. We can then follow the previous discussion by observing that the formulas (f12), (f13), (f14), and (f15) can be easily rewritten in this new framework; e.g., (f14) now becomes (f17). We will explicitly consider this option only in Sect.3.4 to manage the problem of the missing component.

f17 $x ::_t \mathtt{k} \leftrightarrow \exists abcd(abcd ::_t \mathtt{s} \wedge x \equiv_t a+b+c+d)$

In this approach the main difficulty consists in characterizing the instantiation relation. However, the introduction of system-types in the domain of quantification allows taking into account their *intensional* and *intentional* dimensions. Different types could have the same instances and type differences may be grounded on contextual information; e.g., a design feature (or the entire specification), since designed by a company, may have copyright, etc.

A further interesting point of this approach concerns the possibility to align the mereological structure of a system-type with that of its instances. For instance, since K-systems have four components, individuated by the $\mathbb{C}i$-tags, so the type K is composed by four components-types, i.e., $\mathtt{k} = \mathtt{c1} + \mathtt{c2} + \mathtt{c3} + \mathtt{c4}$. Thus, K can be seen as a *structural* property composed by the $\mathbb{C}i$-properties. The fact that the $\mathbb{C}i$-properties uniquely identify components rules out the cases where the same property is part of the structural one several time as, for instance, H in H_2O (see [12] for the debate on structural universals). The mereological alignment is partially guaranteed by the rewriting of (f13)-(f15).

3.3. The Components of a Given System and the Replacement Problem

The notion of component in (f13) can be relativized to a system s as in (f18). Note that now (f19) holds. The left to right arrow is trivial. To see that the opposite direction holds, assume $\mathtt{C1}_t ax$. By (f18) there exist b, c, d such that $\mathtt{S}_t abcd$ and $x \equiv_t a+b+c+d$ that, by (f14), imply $\mathtt{K}_t x$.

f18 $\mathtt{C1}_t xs \leftrightarrow \exists bcd(\mathtt{S}_t xbcd \wedge s \equiv_t x+b+c+d)$

f19 $\mathtt{K}_t x \leftrightarrow \exists abcd(\mathtt{C1}_t ax \wedge \mathtt{C2}_t bx \wedge \mathtt{C3}_t cx \wedge \mathtt{C4}_t dx)$

In the context of a specific system, each tag $\mathbb{C}i$ should have an *identification role*, i.e., the relational constraints in the specification S should grant the fact that all the $\mathbb{C}i$s are unambiguous identifiers of the specific components composing the system, so that, e.g., (f20) would hold for $\mathbb{C}1$. For instance, in Fig. 3, $\mathbb{C}2$-components can be distinguished from $\mathbb{C}4$-components only when the relation R3 is asymmetric.[9] A specification S should indeed ensure (f21) from which (f20) follows. Similarly for the other $\mathbb{C}i$s.

f20 $\mathtt{C1}_t xs \wedge \mathtt{C1}_t ys \rightarrow x = y$

f21 $\mathtt{S}_t xbcd \wedge \mathtt{S}_t ybcd \rightarrow x = y$

Concerning the replacement problem, the replacement of the $\mathbb{C}1$-component for system s can be represented by formula $\mathtt{C1}_{t_0} xs \wedge \mathtt{C1}_{t_1} ys \wedge \mathtt{C1}_{t_2} zs \wedge x \neq y \neq z$, which shows a case where the role of 'being the $\mathbb{C}1$-component of s' is played, at different times, by three different objects.

---

[9] Features themselves may not be enough because often systems have several components of the same type, e.g., the same kind of valve, electric engine or fastener can be present in several places.

According to 4D, there exists three temporal slices: $x_{@t_0}$, $y_{@t_1}$, and $z_{@t_2}$. One can then consider the mereological sum of all the $\mathbb{C}1$-slices of s, see (f22), as the referent of the expression 'the $\mathbb{C}i$-component of s'. As claimed by West [3], systems' components are existentially dependent on the whole system and they are *non-ordinary* objects since they are submitted to discontinuous changes, i.e., the installed component can be (instantaneously) replaced by a new one.

f22 $\mathtt{C1}xs \leftrightarrow x = \sigma y(\exists tz(y = z_{@t} \wedge y \leq s \wedge \mathtt{C1}y))$

Vice versa, in a 3D-framework, to have, for a given system s, a single $\mathbb{C}i$-component and, at the same time, allowing its replacement, one needs to add a type of *individuals* in the domain of quantification. Call these new individuals *stable components* or simply s-components. Stable components abstract from the actual objects installed in the system during its lifetime. Intuitively they are *essentially* characterized in terms of the patterns of features and relations considered in the specification (see (f18) for $\mathbb{C}1$). This move is similar to the classical one where the statue is distinguished from the clay, see [13]. To characterize s-components, (f20) is strengthened as in (f23), i.e., the unicity is now guaranteed also diachronically. To state that at every time at which a K-system exists it always has the same four s-components, we can rewrite (f19) as in (f24). Note that in (f23) and (f24) the first argument of the $\mathtt{C}i$ predicates is now an s-component. To account for the replacement process, one could assume that the s-components may change *constituents*. This requires to introduce a relation $x \lhd_t y$ for "at time t, x constitutes y". The identification role of the $\mathbb{C}i$s is preserved by requiring the unicity (at a given time) of the constituent of an s-constitutent (f25). In this approach an s-component c that has been replaced twice will satisfy formula $\mathtt{C1}_{t_0}cs \wedge \mathtt{C1}_{t_1}cs \wedge \mathtt{C1}_{t_2}cs \wedge x \lhd_{t_0} c \wedge y \lhd_{t_1} c \wedge z \lhd_{t_2} c \wedge x \neq y \neq z$. Note that according to 4D constitution amounts to coincidence, i.e., $x \lhd_t y$ reduces to $x \equiv_t y$ which reduces to $x_{@t} = y_{@t}$.

f23 $\mathtt{C1}_t xs \wedge \mathtt{C1}_{\bar{t}} ys \rightarrow x = y$

f24 $\exists t(\mathtt{K}_t x) \rightarrow \exists abcd(\forall t(\mathtt{K}_t x \leftrightarrow \mathtt{C1}_t ax \wedge \mathtt{C2}_t bx \wedge \mathtt{C3}_t cx \wedge \mathtt{C4}_t dx))$

f25 $x \lhd_t c \wedge y \lhd_t c \rightarrow x = y$

3.4. Missing Components and Partial Compliance

According to the approaches discussed in the previous sections, both systems and components can exist *intermittently*, i.e., their temporal extension may not be a temporal interval. Yet, when the system exists, all its components must be physically present. Hence, to deal with the missing component problem, i.e., the possibility that some components of a system are not physically realized, one needs to conceive both notions of system and compliance in a more flexible way.

A way to realize this is to allow for *partial compliance*, i.e., systems are classified under a type even though they do not fully match the corresponding specification. One can intend partial compliance as a sort of specification-weakening; e.g., for K-systems, in (f13) and (f14), $\mathtt{S}$ can be substituted with $\check{\mathtt{S}}$ where $\mathtt{S}_t xyzv \rightarrow \check{\mathtt{S}}_t xyzv$ but not vice versa,[10] i.e., some constraints in the specification $\mathtt{S}$ are relaxed or ruled out. K-systems and $\mathbb{C}i$ components could then lack some (optional) characteristics indicated in the original specification. Interestingly, one could consider different conditions at different times. For instance, immediately after the completion of the

[10] As done for $\mathtt{S}$, we assume that components of $\check{\mathtt{S}}$ do not have proper parts.

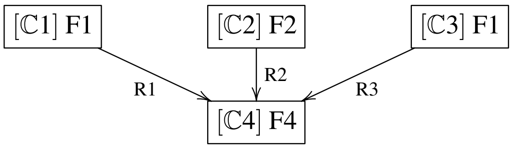

Figure 4: A graph specification with the "reference" component ℂ4.

production process, a system could be required to fully comply with the specification, i.e., it must satisfy S. However, through time, one can tolerate the loss of some characteristics of the system, so that it is enough if it satisfies Š, especially when this loss does not compromise its basic functionalities and is not attributable to production defects (e.g., cars can be scratched).

This approach allows representing optional characteristics of systems but it still requires the presence of all the components. A second way to intend partial compliance is to consider optional components (the two possibilities can clearly be combined). Suppose, for instance that K-systems can lack ℂ4-components, i.e., consider $\bar{\mathtt{S}}$ in (f26) and modify (f14) as in (f27) allowing K-systems to lose or acquire components during their lifecycle.[11] Unfortunately, by adopting (f26), i.e., by tolerating that some components are missing, some ℂi-tags could lose their effectiveness because one can have troubles in re-identifying them through time: the second disjuncts in the definitions (f28) and (f29) of, respectively, ℂ2 and ℂ4, are identical.

f26 $\bar{\mathtt{S}}_t xyz \leftrightarrow \mathtt{F1}_t x \wedge \mathtt{F2}_t y \wedge \mathtt{F3}_t z \wedge \mathtt{R1}_t xy \wedge \mathtt{R2}_t yz \wedge \neg(x \between_t y) \wedge \neg(x \between_t z) \wedge \neg(y \between_t z)$

f27 $\mathtt{K}_t x \leftrightarrow (\exists abcd(\mathtt{S}_t abcd \wedge x \equiv_t a+b+c+d) \vee \exists abc(\bar{\mathtt{S}}_t abc \wedge x \equiv_t a+b+c))$

f28 $\mathtt{C2}_t xs \leftrightarrow (\exists acd(\mathtt{S}_t axcd \wedge s \equiv_t a+x+c+d) \vee \exists ac(\bar{\mathtt{S}}_t axc \wedge s \equiv_t a+x+c))$

f29 $\mathtt{C4}_t xs \leftrightarrow (\exists abc(\mathtt{S}_t abcx \wedge s \equiv_t a+b+c+x) \vee \exists ac(\bar{\mathtt{S}}_t axc \wedge s \equiv_t a+x+c))$

This problem is prevented in the case of systems having one "reference" component c, i.e., all the other components can be identified just by taking into account the relation they have with c. For instance, consider the specification in Fig. 4 where the reference component ℂ4 is characterized only by F4, i.e., it does not have a relational characterization, and where ℂ1 is characterized as in (f30) (similarly for ℂ2 and ℂ3). A typical example of a reference component is a frame: the "position" in the frame is enough to identify all the other components. One can then assume that the system can persist the replacement and/or the loss of all the components except the frame. By relaxing (f30) as in (f31) one could take into account some cases of ℂ1-components that are malfunctioning because they lack some features required by the specification.

f30 $\mathtt{C1}_t xs \leftrightarrow \mathtt{F1}_t x \wedge \exists a(\mathtt{F4}_t a \wedge \mathtt{R1}_t xa \wedge x+a \leq_t s)$

f31 $\mathtt{C1}_t xs \leftrightarrow \exists a(\mathtt{F4}_t a \wedge \mathtt{R1}_t xa \wedge x+a \leq_t s)$

Let us now consider the interaction between partial compliance and the approach based on s-components previously discussed. A first possibility is to assume that, during its whole lifetime, a system (of a given kind) maintains its essential s-components, but can in principle lose (for

[11]Note that we cannot just write $\mathtt{K}_t x \leftrightarrow \exists abc(\bar{\mathtt{S}}_t abc \wedge a+b+c \leq_t x)$ otherwise it would be possible to have K-systems with five or more components or with four components that however do not satisfy S.

some periods) optional s-components. In this case we need to modify (f24) taking into account only essential s-components. Optional components are then, in general, *intermittent* entities, a view embraced by West in a 4D-setting.

A second possibility is to allow for "empty" s-components, which we will argue for in the following. When experts talk about components or systems, they seem to refer sometimes to physical individuals and some other times to conceptual or abstract ones. In a sense, components and systems seem to have both a *physical* and a *conceptual* dimension, a sort of "dual nature". To capture this duality, one could allow s-components to be "empty", i.e., to lack a physical realization: when an s-component is physically realized, experts refer to this realization while when it is empty, they refer to the s-component itself. Systems would be therefore static with respect to their s-components while optional components would reduce to possibly empty s-components. However, this notion of s-component does not match the one discussed at the end of Sect. 3.3 where s-components are intended as physical entities that always have a constituent, i.e., given an s-component c, the constitution relation $\lhd$ usually satisfies $\mathtt{EX}_t c \rightarrow \exists x(x \lhd_t c)$.

We sketch a possible way to capture this duality where s-components have a conceptual nature. Consider the approach based on the reification of properties discussed in Sect. 3.2 but now consider the reifications of the $\mathbb{C}i$-properties specialized to a given K-system $\mathtt{s}$: $\mathtt{c}i^{\mathtt{s}}$ is the reification of $\mathtt{C}i^{\mathtt{s}}_t x \leftrightarrow \mathtt{C}i_t x\mathtt{s}$, i.e., $x ::_t \mathtt{c}i^{\mathtt{s}} \leftrightarrow x\mathtt{s} ::_t \mathtt{c}i$ (these properties are sort of saturated roles as introduced in [14]). By assuming that the instances at time t of a concept all exist at t (i.e., $x_1 \ldots x_n ::_t c \rightarrow \mathtt{EX}_t x_1 \wedge \ldots \wedge \mathtt{EX}_t x_n$), then $\exists x(x ::_t \mathtt{c}i^{\mathtt{s}}) \rightarrow \mathtt{EX}_t \mathtt{s}$ holds for all the components while the vice versa does not hold for optional components, i.e., one can have empty $\mathtt{c}i^{\mathtt{s}}$. Furthermore, when replacement is allowed, $x ::_t \mathtt{c}i^{\mathtt{s}} \wedge y ::_{\bar{t}} \mathtt{c}i^{\mathtt{s}} \rightarrow x = y$ holds in general only when $t = \bar{t}$. However, differently from the previous s-components, the $\mathtt{c}i^{\mathtt{s}}$ are not part of the system $\mathtt{s}$. Our idea is to introduce the conceptual counterpart $\mathtt{d}^{\mathtt{s}}$ of the system $\mathtt{s}$ by defining it as $\mathtt{d}^{\mathtt{s}} = \mathtt{c1}^{\mathtt{s}} + \mathtt{c2}^{\mathtt{s}} + \mathtt{c3}^{\mathtt{s}} + \mathtt{c4}^{\mathtt{s}}$. By guaranteeing that both $\mathtt{EX}_t \mathtt{s} \rightarrow \mathtt{s} ::_t \mathtt{d}^{\mathtt{s}}$ and $x ::_t \mathtt{d}^{\mathtt{s}} \rightarrow x \equiv_t \mathtt{s}$ hold, $\mathtt{d}^{\mathtt{s}}$ becomes a sort of *definite description* of $\mathtt{s}$, i.e., it is a concept that individuates a single system. In this way, both the conceptual (via $\mathtt{d}^{\mathtt{s}}$ and the $\mathtt{c}i^{\mathtt{s}}$) and the physical (via $\mathtt{s}$ and, when present, the instances of the $\mathtt{c}i^{\mathtt{s}}$) perspectives on a system can be represented. Note however that, following the discussion about (f28) and (f29) and assuming that some $\mathbb{C}i$-components are optional for $\mathtt{s}$, this strategy can be pursued only when the definition of the other $\mathbb{C}j$-components (or at least some of them) can be satisfied also when the $\mathbb{C}i$-components are not present.[12] Otherwise the loss of (some of) the $\mathbb{C}i$-components would imply the loss of all the other components.

Let us consider now the problem of the missing component discussed in Sect.1. One can approach the problem at the conceptual level. The specification and the explicit conditions required to (fully or partially) comply with it allow talking about the characteristics of any realization of a system-kind. From the characterization of K-systems (f14) and of their $\mathbb{C}i$-components (e.g., (f13)), together with additional (background) knowledge on the world shared by engineers and (if the system is at least partially realized) the information about the physical properties of the actual components, some properties of K-systems and $\mathbb{C}i$-components can be inferred. For instance, consider again the example of the frame, assuming it is an essential component of the whole system. In this case, we can use the information about its spatial

[12] As discussed, this could be problematic when the absence of $\mathbb{C}1$-components causes the loss of the individuation role of other components, and could be prevented, e.g., by using a component as "reference".

localization or its specific temperature to infer additional information concerning the specific system of which the physical frame is a component. Note however, that only the approach based on saturated roles allows having an individual in the domain of quantification that can be used to refer to the component even though it is missing.

4. Discussion and Conclusion

As we have seen in the previous section, the identified engineering modeling problems can be tackled from different ontological perspectives. The result of our analysis is that none of these views is clearly more advantageous than the others. Therefore, there is no definite preferable approach. We gained however some insights that can hopefully help knowledge engineers to choose an approach to develop their models on the basis of formal implications along with, for those interested, philosophical principles (e.g., 3D vs 4D).

To sum up, in all cases we have seen, engineering physical systems are discussed in relationship with design specifications; the match between the two establishes the degree of systems compliance. In particular, design specifications represent the *properties* (in terms of features F*i* and relations R*i*) of physical systems. In a standard FOL setting, properties correspond to predicates and FOL predication models the ontological notion of instantiation. This formal mechanism can be tuned to both 3D and 4D views (see Sect. 3.2). In addition, there is the option to reify properties to model them as FOL individual constants (see (f17)). This move, which is coherent with both the 3D and 4D, complicates the formal representation, since new predicates are needed to characterize the new constants (see, e.g., (f14) and (f17)).

The notion of compliance can be treated in different ways. A physical system is compliant with a specification when it satisfies all properties established by the latter (see, e.g., (f14) for 3D, (f16) for 4D, and (f17) for the reified variant of 3D). However, there are cases where a weaker sense of compliance, which we called partial compliance, is preferable. To capture this notion, one can introduce the distinction between essential and optional systems' properties such that the lack of the latter does not compromise systems' identity (see Sect. 3.4).

For the representation of components' replacement, in West's 4D-framework [3], a whole system's component is a 4D worm formed by all temporal slices that at different times have been part of it. In this sense, having a system's component replaced at a certain time implies that, from that time, the whole worm has as part a temporal slice of a different object. This view requires however the specification of some rules, possibly motivated from an engineering perspective, to guarantee the continuity of the 4D worm. Otherwise, it is not clear how (possibly) disconnected temporal slices come to form a unique whole. In a standard 3D approach, we showed how to overcome the identity problem of an object whose components are all replaced by relying on the notion of stable components, namely, objects that – at different times – can change constituents while remaining the same entities (see (f23)–(f25)). Hence, in this perspective, to replace a system's component means to change its constituent.

Finally, we have seen that engineers talk of systems and their components referring sometimes to specific physical entities and some other times to their conceptual counterparts. In both 3D and 4D approaches, reference to entities that have not been physically realized or are simply absent at some time can be made meaningful by referring to the reification of the tags in the specification standing for possibly empty components, i.e., the things engineers refer to as if they were fully-fledged physical entities. Overall, the approach based on reification makes the modeling framework logically more complex but it allows making sense of engineers talks about the missing component problem.

Further work is necessary to complete the analysis we presented here. First, other approaches, (e.g., see [1] for a preliminary proposal based on *possible* objects), should be explored and added to the comparison. Second, one should assess the identified ontological views in more specific and varied engineering requirements, a larger set of requirements may help to better identify the dis-/advantages of the views in more articulated scenarios.

References

[1] S. Borgo, F. Compagno, N. Guarino, C. Masolo, E. Sanfilippo, An overview of some ontological challenges in engineering maintenance, in: Workshop on Domain Ontologies for Research Data Management in Industry Commons of Materials and Manufacturing (DORIC-MM) at ESWC 2021. Available at: https://openreview.net/forum?id=CIqto9pUgqm, 2021.

[2] ISO 15926: Industrial automation systems and integration–Integration of life-cycle data for process plants including oil and gas production facilities, Technical Report, ISO, 2004.

[3] M. West, Developing high quality data models, Elsevier, 2011.

[4] N. Guarino, Artefactual systems, missing components and replaceability, in: Artefact Kinds, Springer, 2014, pp. 191–206.

[5] P. Galle, Candidate worldviews for design theory, Design Studies 29 (2008) 267–303.

[6] E. M. Sanfilippo, S. Borgo, What are features? an ontology-based review of the literature, Computer-Aided Design 80 (2016) 9–18.

[7] R. Casati, A. C. Varzi, et al., Parts and places: The structures of spatial representation, MIT Press, 1999.

[8] K. Hawley, Temporal Parts, in: E. N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, summer 2020 ed., Metaphysics Research Lab, Stanford University, 2020.

[9] T. Sider, et al., Four-dimensionalism: An ontology of persistence and time, Oxford University Press on Demand, 2001.

[10] C. Masolo, Parthood simpliciter vs. temporary parthood, in: Proceedings of the Ninth International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense 2009), 2009, pp. 97–102.

[11] E. M. Sanfilippo, C. Masolo, D. Porello, Design knowledge representation: An ontological perspective., in: Proceedings of the 1st Workshop on Artificial Intelligence and Design (at AI*IA 2015), CEUR workshop proceedings vol. 1473, 2015, pp. 41–54.

[12] D. Lewis, Against structural universals, Australasian Journal of Philosophy 64 (1986) 25–46.

[13] J. J. Thomson, The statue and the clay, Nous 32 (1998) 149–173.

[14] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, N. Guarino, Social roles and their descriptions, in: D. Dubois, C. Welty, M. A. Williams (Eds.), Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR 04), Whistler Canada, 2004, pp. 267–277.