Features and Components in Product Models

Emilio M. SANFILIPPO [a,b,1], Claudio MASOLO [a] Stefano BORGO [a]
Daniele PORELLO [a]

[a] *Laboratory for Applied Ontology (ISTC-CNR)*
[b] *Department of Information Engineering and Computer Science, University of Trento*

Abstract. Product structures are represented in engineering models by depicting and linking *components*, *features* and *assemblies*. Their understanding requires knowledge of both design and manufacturing practices, and yet further contextual reasoning is needed to read them correctly. Since these representations are essential to the engineering activities, the lack of a clear and explicit semantics of these models hampers the use of information systems for their assessment and exploitation. We study this problem by identifying different interpretations of structure representations, and then discuss the formal properties that a suitable language needs for representing components, features and combinations of these. We show that the representation of components and features require a non-standard mereology.

Keywords. design, component, feature, conceptual space, mereology

1. Introduction: the Design of Product Structure

Product development is a knowledge intensive activity consisting in a variety of tasks, from product ideation to detailed product specification, manufacturing realisation and dispatching [1]. The success of Computer Aided (CAx) technologies pushes to formally expand the representation of product knowledge. In particular, engineering design research is seeking suitable and expressive formal theories to support the development of product models, to facilitate their integration and to allow for their sharing among the stakeholders. Also, current research efforts look towards advanced CAx systems whose modeling elements are not just focused on geometry, but rather resemble experts' thinking and require the specification of complex qualitative knowledge about the design intents. Ontologies are currently exploited in this context for various tasks (e.g. [2,3]).

We focus hereby on the (representation of) products' structure, which plays a fundamental role in product design, since it establishes the physical layout of the product at stake and it also allows evaluating the best choices for manufacturing. A product's structure is usually represented by means of models that describe how *components* are assembled into the whole product. The components are either *constructional parts* (simple parts) or *assemblies*. The former are the lowest level of the structural hierarchy in that they "[...] are not normally capable of further disassembly" [4, p.307] without being

[1] Corresponding Author: via alla cascata 56/c, Trento, Italy, 38123; E-mail: sanfilippo@loa.istc.cnr.it

destroyed. Examples include screws and resistances. Assemblies, as the name suggests, result from the assembly of components, which, in their turn, can be constructional parts or (simpler) assemblies. Products like washing machines and laptops are thus assemblies. Product models represent the decomposition of assemblies, typically down its constructional parts, but they include also information about e.g. holes, shapes and colors, since these are crucial to the fulfillment of customers' requirements [5]. For instance, holes may be needed for assembling and to ensure functionalities. These types of entities are called *features* and products are often said to be the *aggregation* of features [6,7].

From an ontological perspective, making sense of product models that combine components and features is challenging. On the one side constructional parts, which are structurally indecomposable, may be *locally* characterised via features. Take a screw. Engineers distinguish between the screw's head and body on the basis of local features: the body has a fillet, the head does not. This shows that the models have to integrate different principles for composition, i.e., those for structural parts and those for feature-based parts. However, even if one considers only the composition of constructional parts into assemblies, a choice is needed among different possible ways of assembly models. These may indeed indicate the list of components in an arbitrary order (similarly to an ingredient list), or indicate the specific order in which they have to be assembled (like in a manufacturing bill of materials). Although engineering terminology can capture these differences [1], experts sometimes rely on informal conventions and other times consider the ambiguity beneficial, since it allows experts to choose how to interpret the model depending on the context and application scenario. However, when assembly models are handled by computer systems, it becomes relevant to specify their semantics, so to discriminate between different types of information.

On the other side, making sense of product models means to deal with feature's representation, a challenging task in itself [8]. Firstly, the term 'feature' is used to refer to different things including holes and bumps, but also colors and shapes. Secondly, holes and the like are sometimes treated as entities on their own on the par of screws and cars, while sometimes as qualifying properties that products carry. As a consequence, the same hole in a (physical) product may be differently described in two models. This calls for a systematic treatment of feature knowledge to be coherently processed by computer systems and used by experts.

Regarding the assemblies, there exist some approaches that rely on mereological notions to represent them [2,9]. However, classical mereology lacks the necessary expressivity in that, as shown in Sect. 4, it cannot represent structural ordering. Also, when holes are conceived as 'first-class citizens', the sum of material objects and holes behaves strangely from the mereological viewpoint; a non-standard treatment of mereological sum and spatial inclusion seem necessary.

Thus, to model and share product data, advanced CAx systems require product knowledge to be richly axiomatised in a way that reflects design intents, is transparent to human users and is accessible to software agents [2,9]. Currently, most CAx systems are based on geometry, while relevant qualitative knowledge concerning, e.g., assembly constraints is only annotated via text expressions, which cannot be computed and are hardly shareable across communities with the risk of loosing relevant information.

This paper is a first step towards an ontology for product knowledge with a target on products' structures. We analyse what are the formal and conceptual challenging aspects behind the notions of assembly and feature highlighting the main problems and indicating some possible solutions. However, we do not develop here a complete theory that formalizes the proposed solutions; more than a ready-to-use ontology, we provide a rigorous analysis of some notions crucial in the design of material products.

The paper is structured as follows. Sect. 2 provides the state of the art as relevant to our discussion. Sect. 3 presents a theory for the representation of design properties. Sect. 4 and Sect. 5 analyse different interpretations of assembly models, and compare different formal representations discussing their ontological commitments. Sect. 6 concludes the paper by addressing future work to be done.

2. Review of the literature

The increasing application of knowledge-based techniques in product knowledge management has led to ontologies covering various aspects of the product lifecycle. We consider some of the previous works dealing with the notions of component and feature, and the relationships among them.

Ontologies maintain a cut-off distinction between (constructional) parts and assemblies, generally called components. As mentioned in the introduction, a part is a product's component at the lowest level of a structural hierarchy, whereas an assembly is composed of at least two parts [3,10]. However, the terminology can vary across and within communities, sometimes because of compliance with standards.

Parthood and connection relations are mostly used to relate components. These are usually only weakly characterised in terms of domain and range constraints. In [11], for instance, the relationship *subArtifactOf* holds between an assembly and its parts; it is however unspecified whether the relation is, e.g., reflexive or transitive. The same applies for the relation of *connection* [3], which is meant in the topological sense.

The application of mereotopologies for assembly has been proposed by [2,9]. These works focus on the formal definition of different kinds of assembly joints, e.g., fusion welding, adhesive bonding, brazing. However, first, these approaches suffer from some formal flaws; second, they characterize the difference between assembly joints in topological terms, ignoring structural, functional, or manufacturing aspects that are essential for design purposes.

Along with parts and assemblies, the notion of feature is recurrently used for product representation. Although it is used with various meanings across the literature [8], 'feature' typically refers to entities like holes and fillets [3,10], but also to properties like dimension and color [12]. On the other hand, there is not a theory for feature-based modeling that distinguishes between the different entities that are modeled via features. This ambiguous treatment has contributed to hamper the application of feature-based approaches to handle qualitative information in CAx systems. Furthermore, there is no agreement on how to relate features to products, or components. When a dedicated relationship is introduced, its semantics is typically not characterised [7]. It is e.g. unclear whether the region of a component includes the region of the holes it hosts.

Generally speaking, the analysis of previous works reveals that current approaches have been developed without a suitable investigation of the formal properties of the employed notions, nor have been based on a careful analysis of the ontological theories that better support the representational tasks at hand. As a result, the exploitation of ontologies for product knowledge management is threaten by hidden-assumptions in their formalisations, as well as the underspecification of the employed terms. If computer systems are required to handle the semantics of terms used for product development purposes, as well as the representation of qualitative product knowledge, the analysis of concepts like feature and component is necessary to provide a formal theory that is coherent with experts' modeling practices and expressive enough to capture the desired interpretations.

3. Holistic Design Properties and Conceptual Spaces

We proposed to model design properties as *concepts* [13]. The starting observation was that complex concepts (associated with complex properties) are *characterized* in terms of basic (and simpler) concepts that are fixed (perhaps just within a context) and are *shared* by the stakeholders in design. These basic concepts are part of the common background of experts and have a conventional nature.

Following DOLCE-CORE [14], concepts are reified into the domain of quantification. Formally, $\mathrm{CN}(x)$ means "x is a concept". Reification allows to quantify on concepts in a first-order setting, but since reified concepts cannot have instances, a *classification* primitive is used to relate concepts to objects; $\mathtt{CF}(x,y,t)$ stands for "the concept x classifies the entity y, *as it is at time t*". The temporal argument qualifies when y is considered (measured, perceived, etc.) and not when the classification is done, i.e., $\mathtt{CF}(x,y,t)$ implies the existence of y at t and says nothing about the temporal existence of the concept x (axiom (a1), $\mathtt{EX}(y,t)$ stands for "the entity y exists at time t").

The shared basic concepts, written BCN, are modeled as regions of *quality spaces* [14], a variation of *conceptual spaces* [15]. Qualities spaces are introduced to classify and compare objects' properties and capabilities like weight, shape and resistance. There is a finite number N of basic properties, to which correspond N quality spaces (noted SP_i). Complex concepts, written CCN, are *characterized* in terms of basic ones: $\mathtt{CH}(x,y)$ stands for "the complex concept x is characterized by the basic concept y". For instance, the complex concept *being a popcorn popper* may be characterized by a set of basic concepts like *being 1kg heavy*, *being red*, *having the capability to convert electricity to thermal energy*, etc. Basic and complex concepts form disjoint subclasses of CN where complex concepts are logical conjunctions of two or more basic concepts, see (a2) and (a3).

a1 $\mathtt{CF}(x,y,t) \rightarrow \mathrm{CN}(x) \wedge \mathtt{EX}(y,t)$
a2 $\mathrm{CCN}(x) \rightarrow \exists yz(\mathtt{CH}(x,y) \wedge \mathtt{CH}(x,z) \wedge \bigvee_{i\in\{1,\dots,N\}}(\mathrm{SP}_i(y) \wedge \neg \mathrm{SP}_i(z)))$
a3 $\mathrm{CCN}(x) \rightarrow (\mathtt{CF}(x,y,t) \leftrightarrow \forall z(\mathtt{CH}(x,z) \rightarrow \mathtt{CF}(z,y,t)))$
d1 $x \sqsubseteq_{\mathsf{c}} y \triangleq \mathrm{CCN}(x) \wedge \mathrm{CCN}(y) \wedge \forall z(\mathtt{CH}(x,z) \rightarrow \exists z'(\mathtt{CH}(y,z') \wedge z \sqsubseteq z'))$

Engineering standards, like the Industry Foundation Classes[2], use a notion of *product type* that can be understood as a complex concept. Given a product type x, $\mathtt{CF}(x,y,t)$ says that a physical object y at t satisfies the design properties listed by x. Here it is important to notice that relation $\mathtt{CH}$ gives an *intensional* characterization of product types: complex concepts characterized by the very same basic properties may not be identical. Indeed, definition (d1) uses the subsumption relation between concepts $\sqsubseteq$, introduced in [13], which is not antisymmetric; thus, so is $\sqsubseteq_{\mathsf{c}}$ between complex concepts. This

[2]http://www.buildingsmart-tech.org/ifc/IFC2x4/rc4/html/

choice is motivated by the fact that in general the characterization of product types in terms of basic properties is only *partial*, due to the fact that the designers, especially in the first phases of the design process, do not specify all the basic properties of the product. Furthermore, one may want to distinguish two product types even though they are formed by exactly the same basic properties when quality spaces lack sufficient expressiveness. In [13], it was observed that the basic properties are always 'holistic', they apply to the whole product. The reduction of distributed features (e.g., the distribution of red-stripes on the surface of an otherwise white product) to holistic properties is problematic (see [16] for a proposal), and the framework in [13], focused on structural information about composition relationships between the parts of a product, is not rich enough to deal with this kind of information. We shall first dig into the analysis of composition in the next section and will come back to distributed properties later in the paper.

4. Assemblies

Assembly models like the ones in Fig. 1 represent whole assemblies and how they are composed. Notice that, in the engineering practice, the link between the assembly and its components is sometimes left implicit [11,1]. Vice versa, some models contain additional explicit information, e.g., concerning the way the components are connected, such as exploded views of a product [3]. Here we focus on models that explicitly take into account the *composition* relation (that is graphically represented by a line from the component to the assembly) but lack any information on the topological or geometrical structure of the components. Our intent is to discuss several ways to understand the composition relation and to provide some hints on how these different interpretations can be captured in a formal way.

Fig. 1 shows a cross-section of a product which is informally described as a hollow sphere (the hole section has a rectangular shape) partially filled with two assembled C-shaped plates that, in the middle, compress a spring. We start by assuming that each diagram in Fig. 1 represents the composition of a specific physical *individual* with compositional parts $c_1,\ldots,c_5$, i.e., the diagrams depict wholes obtained by assembling the same atomic parts.[3] Towards the end of the section, we will discuss the interpretation of the different diagrams in terms of product types. Following Fine [17], our primitive relation of composition must be intended in an *operationalist* way, i.e., composition is not a binary relation like parthood; it is rather an operation that takes some objects (the components) into a unique new object (the assembly). We write $\Sigma(x_1,\ldots,x_n)$ for the whole obtained by composing the objects $x_1,\ldots,x_n$. Using this notation, the diagrams (a)-(d) correspond to $a^{\mathrm{a}} = \Sigma(c_1,c_2,c_3,c_4,c_5)$, $a^{\mathrm{b}} = \Sigma(c_3,c_4,c_2,c_5,c_1)$, $a^{\mathrm{c}} = \Sigma(c_1,\Sigma(c_2,c_3,c_4),c_5)$, $a^{\mathrm{d}} = \Sigma(\Sigma(c_1,c_5),\Sigma(c_2,c_3,c_4))$, respectively. Assuming the depicted constitutional parts always represent the same individuals, the properties of operator Σ characterize whether some or all the assemblies a^{a}, a^{b}, a^{c}, and a^{d} also represent the same individual or not. For this reason we added a superscript to the notation of wholes, e.g., a^{a} represents the assembly that is obtained by composing $c_1,\ldots,c_5$ as indicated in Fig. 1(a). As observed by Fine, one can say that 'x is a component of y' if y is obtained by applying Σ to x (and other objects).

[3]The reader can identify the corresponding constructional parts in all the diagrams due to their shapes *and* orientation (parts with similar shape but different orientation are assumed to be different).

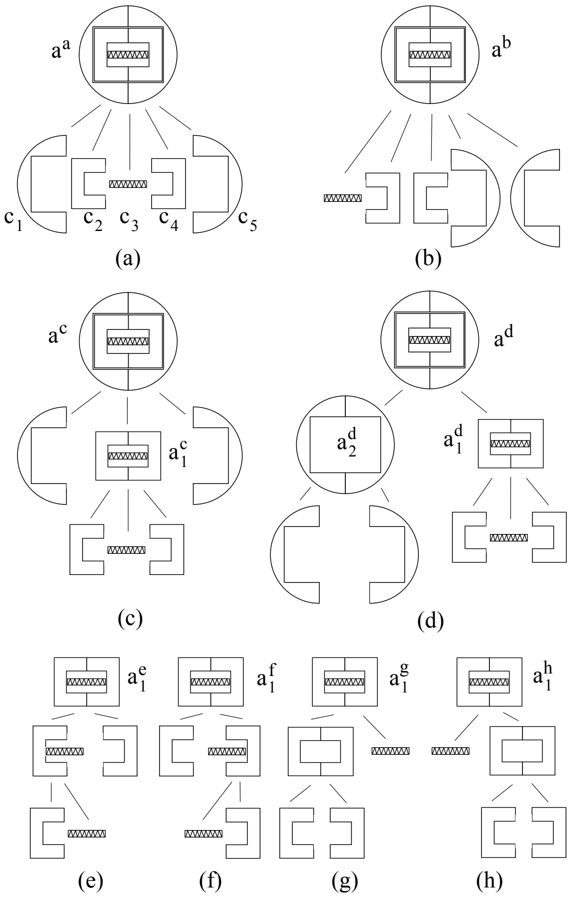

Figure 1. Composition models of assemblies

Fine considers four possible properties for Σ:

(C) $\Sigma(x) = x$ (*collapse*)

(L) $\Sigma(\dots,\Sigma(\dots,x,y,z,\dots),\dots,\Sigma(\dots,u,v,w,\dots),\dots) = \Sigma(\dots,x,y,z,\dots,u,v,w,\dots)$ (*leveling*)

(A) $\Sigma(\dots,x,x,\dots,y,y,\dots) = \Sigma(\dots,x,\dots,y,\dots)$ (*absorption*)

(P) $\Sigma(\dots,x,y,z\dots) = \Sigma(\dots,y,z,x,\dots)$ (similarly for the other permutations). (*permutation*)

Different composition operators are characterized by assuming or negating the properties (C), (L), (A), and (P). Let us write $\mathrm{C\overline{L}A\overline{P}}$, for example, for a form of composition that satisfies (C) and (A) but not (L) and (P), and see which interpretations of the diagrams in Fig. 1 are modeled by which combination of properties.

A first interpretation of (a)-(d) is obtained by considering all the tree-shaped diagrams as representing only the *list* of the *compositional parts* of the assembly. This is the reading, e.g., in [11,3,5]. Accordingly, (a)-(d) must represent identical wholes since the compositional parts ($c_1,\dots,c_5$) are the same. The composition relation captures neither the order of the components, nor the composition steps that refer to intermediate assemblies. *Classical extensional mereology* (CEM) [18] is a formal theory adequate to model this interpretation. It is enough to assume a CLAP composition operator noted Σ_+: $\Sigma_+(x_1,\ldots,x_n) = x_1 + \ldots + x_n$, where '$+$' is the mereological sum defined in CEM (a commutative and associative operator): here assemblies reduce to sums of components. In the examples (a)-(d), we have $a^{\mathrm{a}} = c_1 + c_2 + c_3 + c_4 + c_5 = a^{\mathrm{b}} = c_3 + c_4 + c_2 + c_5 + c_1 = a^{\mathrm{c}} = c_1 + (c_2 + c_3 + c_4) + c_5 = a^{\mathrm{d}} = (c_1 + c_5) + (c_2 + c_3 + c_4)$, i.e., these figures represent the same whole that is determined only by its compositional (atomic) parts. Also, the diagrams in Fig. 1(c)-(d) are redundant.

In a different interpretation, the intermediate compositional steps and assemblies are used to discriminate among products. For instance, in the example in Fig. 1(c), first c_2, c_3, c_4 are assembled into a_1^{c} and then a_1^{c} is assembled with c_1 and c_5 to give rise to the final whole a^{c}. Vice versa, the example in Fig. 1(d) relies on two intermediate assemblies: a_1^{d}, composed by c_2, c_3, c_4 and a_2^{d} composed by c_1 and c_5. The whole a^{d} is then obtained by composing a_1^{d} with a_2^{d}. As we have seen, the sum operator of CEM satisfies the leveling (L), therefore it cannot capture the levels of composition. Vice versa, standard set-theory can express this distinction via the set-construct, i.e., one can adopt a $\overline{\text{CL}}$AP composition $\Sigma_\in$: $\Sigma_\in(x_1,\ldots,x_n) = \{x_1,\ldots,x_n\}$. The case in Fig. 1(c) would then be represented by $a^{\mathrm{c}} = \{c_1, \{c_2, c_3, c_4\}, c_5\}$ (where $a_1^{\mathrm{c}} = \{c_2, c_3, c_4\}$), while the one in Fig. 1(d) by the set $a^{\mathrm{d}} = \{\{c_1, c_5\}, \{c_2, c_3, c_4\}\}$ (where $a_1^{\mathrm{d}} = \{c_2, c_3, c_4\}$ and $a_2^{\mathrm{d}} = \{c_1, c_5\}$). The two sets are different, i.e., Fig. 1(c) and Fig. 1(d) represent different assemblies. The rationale behind this reading of the diagrams is that the *identity criteria* for a product considers the intermediate assemblies that are associated with the product. Note that in this interpretation we have $a^{\mathrm{a}} = \{c_1, c_2, c_3, c_4, c_5\} = a^{\mathrm{b}} = \{c_3, c_4, c_2, c_5, c_1\}$, which are different from both a^{c} and a^{d}, i.e., Fig. 1(a)-(d) represent three different assemblies.

From an engineering point of view, the distinction is useful. For instance, one can (i) individuate unfeasible assemblies—e.g., the case in Fig. 1(d) might not be realizable, because it might not be possible to place the C-plates in the hole, once the two half-spheres are mounted—and (ii) evaluate assemblies according to some goal—e.g., to find the most adequate with respect to the available resources. Fig. 1(e)-(h) illustrate alternative compositions of the (intermediate) assembly $a_1^{\mathrm{c}} = a_1^{\mathrm{d}}$ to be placed in the hole. The case in Fig. 1(g) could be preferred to the one in Fig. 1(d), if it results easier to insert the spring between the C-plates, rather than assembly the three components in one step.

Set-theory, as used above, does not distinguish the cases in Fig. 1(g)-(h) since $\{\{c_2, c_4\}, c_3\} = \{c_3, \{c_2, c_4\}\}$. For instance, the case in Fig. 1(g) could be intended as "the spring is introduced into the assembled C-plates (perhaps by first compressing it)", while the one in Fig. 1(h) as "the assembled C-plates are put around the spring (which is compressed during the assembly step)". To capture this difference one needs to represent the order, i.e., we need at least to reject permutation (P). For instance, one could consider *ordered lists*, i.e., assume a $\overline{\text{CLAP}}$ composition noted Σ_{s}: e.g., $\Sigma_{\mathrm{s}} = \langle x_1,\ldots,x_n\rangle$. The assemblies in Fig. 1(g),(h) would then be different because $a_1^{\mathrm{g}} = \langle\langle c_2, c_4\rangle, c_3\rangle \neq a_1^{\mathrm{h}} = \langle c_3, \langle c_2, c_4\rangle\rangle$. This formalization takes track of the order relation between the product's components. The identity criteria of complex products depend now on both the components and the order in which these are assembled at any step. An hybrid approach that combines sets with tuples would allow to make explicit the ordering only when needed. For instance, the previous examples in Fig. 1(g),(h) could be represented, respectively, by $a_1^{\mathrm{g}} = \langle\{c_2, c_4\}, c_3\rangle$ and $a_1^{\mathrm{h}} = \langle c_3, \{c_2, c_4\}\rangle$, i.e., by using tuples that make explicit the irrelevance of the order of the components c_2 and c_4. However, the previous principles for Σ are not enough to represent this hybrid case.

Finally, note that even when the assemblies in Fig. 1 result to be all different, this difference is based only on the composition constraints, that is, no *physical* discriminating property is used, as the products have the same weight, diameter, colors, etc.

The situation is particularly interesting and challenging when features, in particular holes, are treated as first-class citizens and included among the components [6,7,19].[4] In philosophy a similar position has been investigated in [20]. In this perspective, a hole is an entity located in space and not constituted by material. The hole cannot be confused with the region of space where it is located, as the hole, but not the region, can move. Fig. 2 shows a model where among the compositional parts there are two holes labeled c_7 and c_8, respectively.

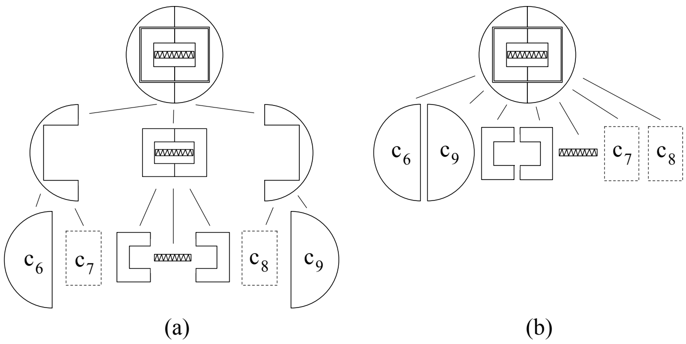

Figure 2. Composition models with hole components

Beside the properties C, L, A, P discussed earlier, Σ could be characterized in terms of the link between the spatio-temporal location of the whole and those of the components. When holes are compositional parts, this link becomes quite 'atypical'. First, the location of c_1 (written $l(c_1)$) does not include the location c_7, thus $l(c_1)$ is properly included in $l(c_6)$ even though c_6 is a part of c_1 [21]. Second, when an assembly is seen as the composition of a constructional part and a hole, the former ceases to exist in the assembly. Assuming that the introduction of c_1 coincides with the "use" of c_6, when c_1 exists, then necessarily c_6 does not.

Moreover, in philosophy, holes are 'parasitic' upon their hosts in that they cannot exist if detached from the material objects to which they are related. This does not seem true in design engineering where holes can be designed and considered in isolation. Engineers not interested in modeling the assembly steps substitute the model in Fig. 2(a) with the one in Fig. 2(b) where the link between the hole and its host is lost.

These observations show that the compositional diagram in Fig. 2(a) could be intended in an *operational*, or at least a *temporal*, perspective, i.e., it includes information on the steps to obtain the assembly. In Fig. 2(a), c_1 is obtained by "holing" c_6, i.e., by "adding" a hole to it and not vice versa. This suggests that both the order in which the parts occur and the steps of the composition are relevant. One could then use the approach

[4] We concentrate on holes since their use in engineering is more frequent than that of other features like bumps and edges [19].

based on tuples (or the hybrid one) to represent this information paying attention that the usual constraints on the spatio-temporal locations apply only to *material* components.[5]

Let us now assume that the diagrams in the previous figures describe a *product type* instead of a specific product individual. This reading is quite common, for instance, in models for mass production. Now the parts $c_1,\ldots,c_5$ and the (intermediate) assemblies are understood as product types and recall from Sect. 3 that in our approach these have a conceptual nature.

The interpretations discussed above for the composition relation apply to product types as well. According to the importance attributed to the structure and the order of the assembled components, one can embrace a mereological, set-theoretical, list-oriented, or hybrid interpretation for product types. However, the composition of types introduces an additional problem. In Fig. 1(a), for instance, one presumably has that the product types c_1 and c_5 are identical, the same for c_2 and c_4. The intended interpretation is that the whole product requires, among its components, *two* instances of a hollow semi-sphere and *two* instances of a C-shaped plate, i.e., every physical realization of this product type has two physical realizations of the hollow semi-sphere type and two physical realizations of the C-shaped plate type. At the level of types, which is what is represented in the diagram, there is only one hollow semi-sphere type and only one C-shaped plate type. Since both Σ_+ and $\Sigma_\in$ satisfy absorption (A), i.e., $a+a+b=a+b$ and $\{a,a,b\}=\{a,b\}$, only the formalization in terms of ordered lists can be used since $\langle a,a,b\rangle \neq \langle a,b\rangle$. If parts' ordering is not relevant, one can still use an extension of set-theory via *multi-sets*, i.e., sets where the number of occurrences of a member in the set is relevant. In this case $\{a,b,a\}=\{a,a,b\}\neq\{a,b\}$, while clearly $\langle a,b,a\rangle \neq \langle a,a,b\rangle$ (see [22] for a formalization of multi-sets in a framework similar to the one of Fine).[6] Finally, the introduction of holes as independent entities in the diagram does not complicate the framework more than what seen before. Hole types are very similar to product types, they are designed and specified in terms of basic properties even though those relative to materiality do not apply in this case.

Finally, assume that the previous diagrams refer to *classes* of objects, i.e., in logical terms, a^{a}, c_1, etc. are the graphical correspondent of unary predicates. Then, the meaning of Fig.1(a) is captured by axiom (a4) (alternative axioms can be considered). Similarly for the other diagrams. The expressivity of the logic language allows to distinguish the different Σ operators with the expected consequences on the classification of a given physical assembly under the types a^{a}, a^{b}, etc.

a4 $a^{\mathrm{a}}(x) \rightarrow \exists x_1,\ldots,x_5(c_1(x_1)\wedge\ldots\wedge c_5(x_5)\wedge x=\Sigma(x_1,\ldots,x_5))$

5. Features of Constructional Parts

As seen in the introduction, a constructional part is an undecomposable object. Fig. 3 depicts a single constructional part with some local features: slots, sectors, dimensions, etc.

[5]Pushing this idea further, all engineering features, e.g., colours, slots, bends, bumps etc., could be modeled as components. For instance, red objects are obtained by adding colour-features to objects (that do not have then), that is, a red-object is a whole resulting from 'applying' a red-feature to a 'bare' particular.

[6]If the order is irrelevant, in the case of product types, but not in the one of physical objects, the wholes in Fig.1(e),(f) are identical because $c_2=c_4$, i.e., $\{\{c_2,c_3\},c_4\}=\{\{c_2,c_3\},c_2\}=\{c_2,\{c_3,c_2\}\}=\{c_2,\{c_3,c_4\}\}$.

Similarly, the drawing of a chessboard would show black and white boxes even though they do not correspond to detachable parts. Despite not being considered as components of the constructional part, these features are essential for the product. To make sense of this, one has to consider properties that are located only in sublocations of an atomic object, namely, *local* properties. These properties can be represented neither using the framework presented in Sect. 3, nor using its composition-based extension illustrated in Sect. 4. To fill the gap, we explore three possible ways, based on different ontological commitments, to answer this problem.

The options we consider are the spatial counterparts of approaches to represent *temporally* qualified properties for change through time. Consider an object a that exists at different times t and t', while being red at t and blue at t'. Four-dimensionalists [23] would represent this situation with $\texttt{Red}(a@t) \wedge \texttt{Blue}(a@t')$, i.e., they would commit to the temporal slices of a at t and at t' (denoted by $a@t$ and $a@t'$, respectively) that are the actual instances of the properties *being red* and *being blue*. Tropists instead would rely on *trope*-substitution [24]. There exist two different tropes (individual properties), namely, *the red of a* and *the blue of a*, that *inhere* in a and are substituted through time. Formally, $\exists rb(\texttt{Inheres}(r,a) \wedge \texttt{Inheres}(b,a) \wedge \texttt{EX}(r,t) \wedge \texttt{EX}(b,t') \wedge \texttt{Red}(r) \wedge \texttt{Blue}(b))$. The last option, usually embraced by endurantists, considers time as an additional parameter of the properties, i.e., Red and Blue become binary relations between objects and times: $\texttt{Red}(a,t) \wedge \texttt{Blue}(a,t')$. The formalization of these alternatives in terms of space is quite direct: it is enough to consider regions of spaces s, instead of instances of times t. However, as we shall see, the use of a space parameter introduces some complications.

Fiat parts. The first option to represent local properties comes from four-dimensionalism. The situation where an object is red at space s but blue at s' is represented by $\texttt{Red}(a@s) \wedge \texttt{Blue}(a@s')$, i.e., the properties apply to two different *slices*, $a@s$ and $a@s'$ where, for the sake of simplicity, we consider only spatial slices instead of spatio-temporal ones. (Furthermore, we do not specify what counts as spatial slice. It suffices to assume that any feature is associated with a part of the object that qualifies as a spatial slice.) Note that, for a constructional part a with local features, the spatial slices $a@s$ and $a@s'$ corresponding to the given features do not identify detachable components, thus there is no compositional diagram to represent.

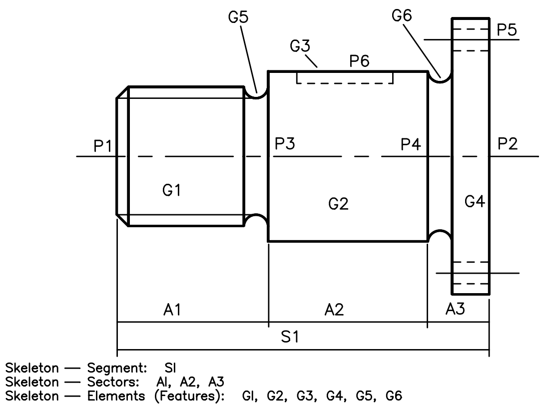

Figure 3. Constructional part and its characterising features (from [4])

Four-dimensionalism assumes that temporal slices are prior to objects, i.e., any object is a mereological sum of instantaneous slices. Analogously, we should claim that products are the sum of spatial slices. However, from an operational perspective, the constructional part is not built by assembling spatial slices, but rather by 'adding' certain local features to a whole. For instance, to add a color or a hole, one has to assume the existence of the constructional part. It seems then that spatial-parts require a cognitive mechanism which contrasts that of four-dimensionalism: spatial-parts, rather than objects, are the result of a cognitive construction.

Following engineering practice, features are usually introduced to identify properties of the product (e.g., color, texture, etc.) that are relevant for the fulfillment of some requirements. In this perspective, features are coupled with physical discontinuities—e.g., the transition between a smooth and a rough surface—that, however, do not result in detachability. One can then think that a feature (or the associated discontinuity) identifies a spatial location in the constructional part a on the basis of which a is cut into spatial parts. However, at design- or manufacturing-level, one needs to identify *where* a feature has to be placed before adding it to the constructional part a. For instance, to specify that the top of a is smooth, one needs first to isolate a spatial part of a, the top of a, and then attribute to it the property *being smooth*. The top of a existentially depends on a but it is not determined by the physical discontinuity associated to the feature.[7] Furthermore, the space s used to identify the spatial part (e.g., the top of a) is object-centric, i.e., the constructional part itself provides the spatial reference system. More precisely, s is not an abstract spatial region in an absolute reference system, instead it behaves as a *relative place* [25]. Therefore, spatial parts survive the relocation of the constructional part (or its assembly in a product) as relative places do. These observations provide additional support to the idea that constructional parts precede spatial parts: the first are needed as reference systems for introducing relative places.

Following this approach, the spatial parts $a@s$, where s is now a relative place, become quite similar to the *fiat parts* introduced in [26].[8] Note that *fiat parts* are necessarily not marked by any physical discontinuity. Vice versa, we have seen that at design-level features have a double nature. On one hand, they are introduced referring to a place and not to a discontinuity. On the other hand, they are intended to capture a discontinuity. For instance, by placing a smooth-feature in s, one presuppose that s determines the boundary between a smooth and a rough part of the constructional part. Therefore, to model the way features are used during the design phase, the approach presented in [26] has to be modified to account for this double nature.

Finally, note that features like holes, slots, etc. could undermine this view if taken as objects on their own that are not spatially included in their host. To address this problem one has to see holes as properties of a fiat part of (the surface of) the object. For instance, the slot G3 in Fig. 3 can be represented by attributing a parallelepiped shape to a given part of the constructional part.

Tropes. The second option to formally understand local properties follows the trope theory approach. An object a red at s and blue at s' is represented by $\exists rb(\mathtt{Inheres}(r,a) \wedge \mathtt{Inheres}(b,a) \wedge \mathtt{Red}(r) \wedge \mathtt{Blue}(b) \wedge \mathtt{Located}(r,s) \wedge \mathtt{Located}(b,s'))$, i.e., the properties apply to different *tropes*, b and r, both inhering in a but with different spatial locations.

[7] In addition, different features could be located into the top of a.
[8] Even though *fiat parts* do not necessarily refer to relative places.

Tropes are, by definition, non-detachable from the constitutional part a and existentially dependent on a. Again, this motivates the lack of composition diagrams for constructional parts.[9] This option requires the framework in Sect. 3 to be extended with tropes (similar to the individual qualities of DOLCE) and their spatial locations.

Intuitively, tropes represent specific aspects of objects. Given a (fully determinate) property, e.g., *being scarlet*, a *unique* scarlet-trope can inhere in an object. The case of a scarlet-trope that inheres in a and is located at a subregion of a offers the possibility to represent local properties. If a is red at two different places, one can assume that the red-trope is located at the sum of the places (possibly a 'disconnected' place). Assume now that slots are represented by shape-tropes—i.e., an object has a slot at a given place if and only if there is a slot-shape-trope that is located there—and that a has two slots with identical shape. The slot-shape-trope cannot be located at the sum of the places because, at the sum location the object has not a slot shape (colors, but not shapes, are spatially summative). In this case one needs to assume the existence of two different slot-shape-tropes both inhering in a. More generally, given a (fully determinate) property P, at a given time, several P-tropes can inhere in a single object but at different or disjoint places. The fact that tropes have a spatial location has been criticized [27], the existence of several fully specified P-tropes inhering in the same object is still more problematic. More strongly, assume that the compositional part a has a slot with a given depth (say $1cm$). Following the previous approach, one would say that the shape-trope and the depth-trope are colocated. Instead, designers think of them as complex entities: it is the slot that has depth $1cm$. To preserve this view, one should assume that the 1cm-deep-trope inheres in the slot-trope, but that view requires to revise trope theory to accept tropes of tropes.

Spatially qualified predicates. In the third option to model local properties, one uses binary relations to represent the fact that an object is red at space s but blue at s', i.e., $\mathtt{Red}(a,s) \wedge \mathtt{Blue}(a,s')$. Clearly, no additional entity is introduced in the domain of discourse. To integrate this solution in our framework it suffices to add a spatial argument to the classification relation. Let $\mathtt{CF}(x,y,t,s)$ stand for "the concept x classifies y, *as it is at time t and space s*". (Again one could consider relative places instead of absolute spaces.) To classify the whole object y it is enough to take s to be the location of y.

As with tropes, we cannot have features of features: to say that a compositional part a has a $1cm$ deep slot one can only write that a is classified by two different properties (*being slot-shaped* and *being* $1cm$ *deep*) at the same time and space. Moreover, to represent the fact that a has two slots with identical shape, we can just write that, at a given time, a is classified by the same property (*being slot-shaped*) at two different spaces but not at the sum of these spaces. In this way we can 'count' the slots. Vice versa, for summative features like colors, on can assume that the *being red* property is (also) located at the sum of the spaces. These examples are then less problematic in this framework than in the one based on tropes. Furthermore, one could assume that assemblies are classified only holistically (s is the location of x in all the $\mathtt{CF}(x,y,t,s)$ statements) while constructional parts can have local properties (there are $\mathtt{CF}(x,y,t,s)$ statements for x a constructional part where s is a proper part of the location of x). This means to reduce the local properties of assemblies to the local or holistic properties of their constructional parts.

[9] Recall that a trope is a quality characterizing a specific object that is not sharable among different particulars at the same time, and cannot migrate across particulars at different times.

6. Conclusion

The exploitation of advanced CAx systems requires formal theories representing technical notions like feature and component in a way that is coherent with experts' conceptualisations, expressive enough to deal with complex knowledge, and accessible to software agents in a meaningful manner. Furthermore, the development of ontologies to support engineering modeling tasks requires to understand the implicit assumptions underlying product models and to identify suitable formal languages. We focused on two issues related to the specification of product knowledge, namely the interpretation and representation of (*i*) assembly models and (*ii*) features.

So far, the first issue has not been carefully analysed by experts and stakeholders in the practice of product development. This paper provides a first analytical step. We studied various assembly relations and the needed logical properties. Classical mereology is well suited to represent assemblies as lists of components, while non-standard mereology is required to deal with ordered and layered composition. One possible way to formally address this (quite complex) task is to follow [22] to precisely model the different compositional relations sketched in Sect. 4. Alternatively, following [28], one could supplement a standard mereology with a theory of levels.

Concerning the second issue, features can be treated either as compositional parts or as qualities. The former reading is well suited for hole-like features (slots, pockets, etc), which are commonly treated as physical objects on their own, as we have seen throughout the paper. This suggests to consider products as composed of both material and immaterial entities, a consequence that, although may appear puzzling at first glance from an engineering modeling stance, results from current design approaches. At the level of product models, these features are not necessarily related to material objects and their dependence on the host is lost. In this perspective, a hole can exist even if not attached to any other product's element. The interpretation of features as qualities is better suited for features like color, weight and dimension, but it also applies to hole-like entities provided these are treated as shapes. Both approaches are currently found in the literature, even if not properly characterised, nor distinguished.

Finally, we have seen that features as qualities are localised in constructional parts. This calls for products' parts that are not composable from an assembly perspective, but are rather identified as bearers of features. We discussed various ontological approaches by which this can be represented. In our understanding, the method based on the fiat parts is better suited, as it facilitates both direct reference to parts carrying features and the co-localisation of multiple features. If this analysis is confirmed, it suggests to include the class of fiat parts in ontologies for product knowledge representation – along with constructional part and assembly – so to increase the conceptual expressivity of the system as well as to ensure a coherent representation of the needed entities.

The analysis presented in these pages is only a first step towards the characterization of a formal theory for material products and their corresponding engineering models. Further work is necessary to properly define the mereological (and mereo-topological) operators, their properties for assembly and a theory for engineering features.

References

[1] G. Pahl and W. Beitz. *Engineering Design: a Systematic Approach*. Springer Science & Business Media, 2013.

[2] K.-Y. Kim, H. Yang, and D.-W. Kim. Mereotopological Assembly Joint Information Representation for Collaborative Product Design. *Robotics and Computer-Integrated Manufacturing*, 24(6):744–754, 2008.

[3] S. Rachuri, Y.-H. Han, S. Foufou, S. C. Feng, U. Roy, F. Wang, R. D. Sriram, and K. W. Lyons. A Model for Capturing Product Assembly Information. *Journal of Computing and Information Science in Engineering*, 6(1):11–21, 2006.

[4] W. E. Eder and S. Hosnedl. *Design Engineering: a Manual for Enhanced Creativity*. CRC Press, 2008.

[5] M. Imran and R. I. Young. Reference Ontologies for Interoperability Across Multiple Assembly Systems. *International Journal of Production Research*, pages 1–23, 2015.

[6] J.Shah and M.Mäntylä. *Parametric and Feature Based CAD/CAM. Concepts, Techniques, Applications*. John Wiley and Sons, 1995.

[7] Z. Usman, R. I. Young, N. Chungoora, C. Palmer, K. Case, and J. Harding. Towards a Formal Manufacturing Reference Ontology. *International Journal of Production Research*, 51(22):6553–6572, 2013.

[8] E. M. Sanfilippo and S. Borgo. Feature-Based Modelling and Information Systems for Engineering. In *AIIA 2015, Advances in Artificial Intelligence*, pages 151–163. Springer, 2015.

[9] F. Demoly, A. Matsokis, and D. Kiritsis. A Mereotopological Product Relationship Description Approach for Assembly oriented Design. *Robotics and Computer-Integrated Manufacturing*, 28(6):681–693, 2012.

[10] K.-Y. Kim, D. G. Manley, and H. Yang. Ontology-based Assembly Design and Information Sharing for Collaborative Product Development. *Computer-Aided Design*, 38(12):1233–1250, 2006.

[11] S. J. Fenves, S. Foufou, C. Bock, and R. D. Sriram. CPM2: a Core Model for Product Data. *Journal of Computing and Information Science in Engineering*, 8(1):014501, 2008.

[12] S. Pourtalebi and I. Horvath. Towards a Methodology of System Manifestation Features-based Pre-Embodiment Design. *Journal of Engineering Design*, pages 1–37, 2016.

[13] E. M. Sanfilippo, C. Masolo, and D. Porello. Design Knowldege Representation: An Ontological Perspective. In F. A. Lisi and S. Borgo, editors, *Proceedings of the 1st Workshop on Artificial Intelligence and Design (AIDE)*. CEUR vol-1473, 2015.

[14] S. Borgo and C. Masolo. Foundational Choices in DOLCE. In S. Staab and R. Studer, editors, *Handbook on Ontologies*. Springer Verlag, 2009.

[15] P. Gärdenfors. *Conceptual Spaces: The Geometry of Thought*. MIT press, 2004.

[16] S. R. Fiorini, P. Gärdenfors, and M. Abel. Representing Part–Whole Relations in Conceptual Spaces. *Cognitive processing*, 15(2):127–142, 2014.

[17] K. Fine. Towards a Theory of Part. *The Journal of Philosophy*, 107(11):559–589, 2010.

[18] A. Varzi and R. Casati. *Parts and Places: The Structures of Spatial Representation*. MIT Press, 1999.

[19] N. Chungoora and R. I. Young. Semantic Reconciliation across Design and Manufacturing Knowledge Models: A Logic-based Approach. *Applied Ontology*, 6(4):295–315, 2011.

[20] R. Casati and A. C. Varzi. *Holes and other Superficialities*. MIT press, 1994.

[21] T. Hahmann and B. Brodaric. The Void in Hydro Ontology. In M. B. Donnelly and G. Guizzardi, editors, *Formal Ontology and Information Systems (FOIS)*, pages 45–58. IOS Press, 2012.

[22] A. Cotnoir. Abelian Mereology. *Logic and Logical Philosophy*, 24(4):429–447, 2015.

[23] T. Sider. *Four Dimensionalism. An Ontology of Persistence and Time*. Oxford University Press, 2001.

[24] C. Daly. Tropes. In D. Mellor and A. Oliver, editors, *Properties*, pages 140–159. Oxford University Press, Oxford, 1997.

[25] M. Donnelly. Relative places. *Applied Ontology*, 1(1):55–75, 2005.

[26] B. Smith. Fiat objects. *Topoi*, 20(2):131–148, 2001.

[27] F. Moltmann. *Abstract Objects and the Semantics of Natural Language*. OUP Oxford, 2013.

[28] C. Masolo. Understanding ontological levels. In F. Lin and U. Sattler, editors, *Proceedings of the Twelfth International Conference on the Principles of Knowledge Representation and Reasoning (KR 2010)*, pages 258–268. AAAI Press, 2010.