Design knowledge representation: An ontological perspective

Emilio M.Sanfilippo[1,2], Claudio Masolo[1], and Daniele Porello[1]

[1] Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy
[2] Ph.D. School in ICT, University of Trento, Trento, Italy

Abstract. We present a preliminary high-level formal theory, grounded on knowledge representation techniques and foundational ontologies, for the uniform and integrated representation of the different kinds of (qualitative and quantitative) knowledge involved in the designing process. We discuss the conceptual nature of engineering design by individuating and analyzing the involved notions. These notions are then formally characterized by extending the DOLCE foundational ontology. Our ultimate purpose is twofold: (*i*) to contribute to foundational issues of design; and (*ii*) to support the development of advanced modelling systems for (qualitative and quantitative) representation of design knowledge.

Keywords: Design model, integration, conceptual space, DOLCE

1 Engineering design: from requirements to models

Despite the variety of definitions and theories for engineering design, it is common to understand it as the activity of producing a full description of the technical product to be realised that satisfies some (market) requirements [1] (cf. Figure 1). In this sense designing is a creative activity of *conceptualization* and *modelling*: designers have to think out (possibly innovative) practical solutions to satisfy customer needs and to release technical specifications[1] for the production of physical goods [2]. The designing process comprises different tasks, from the elicitation of customers' requirements to conceptual embodiment and detail design [3]. During each phase cross-functional expert teams work on different modelling aspects of the same design project. Consequently, various models are firstly obtained with respect to the required level of detail and content, and secondly integrated into an all-comprising representation of the product [4].

The specification of the requirement list about the properties that the future product has to satisfy plays a fundamental role, since it represents the document on which both the overall designing process is firstly based, and secondly evaluated [1]. According to Pahl and colleagues [3], (product's) properties in a requirement list can be meant either as *demands*, if they have to be necessarily met, or as *wishes*, if they should be taken into account whenever possible.

---

[1] 'Technical specification', 'product model' and 'design model' will be interchangeably used throughout the paper.

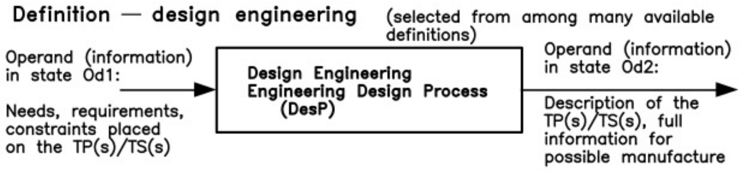

Fig. 1. Design engineering defintion [1]. TP and TS are interchangeably used for "technical product" and "technical system".

After the requirement list has been completed, experts start specifying the design solution by means of various technical specifications. Examples include:

- *Functional model*: the main functionality of the product under design is represented and decomposed into sub-functionalities (e.g. [3, 169-181]);
- *Component model*: (also called *part model*) describes what constructional parts are required [5, 6]. In some theories of design, *organ models* are used to represent structural elements carrying a functionality [7];
- *Assembly model*: specifies spatial relationships holding among components [8]. The most common assembly relationship is *part-of* holding between components; *connection* is also used to represent physical connection [9];
- *Material model*: represents the type of material used for each component in a product. It can also be used to specify material's properties like stress resistance, or malleability [7];
- *Geometry model*: describes the product shape, typically together with its nominal dimensions and tolerances [5].

These are however only a few cases; at the current state of the art there is no complete, nor standardised list of models used in engineering design [6].

The importance of technical specifications is due to the fact that they determine the final properties that particular objects have to satisfy to be considered as products of a certain type. If a particular object satisfies the properties specified by its corresponding model, it is said *to conform to* the model [4].

This brief introduction suggests that the main core of designing is the development of (design) *concepts*, rather than their codified *description*, as suggested by Figure 1. It also shows that various quantitative but also qualitative knowledge aspects have to be considered. For instance, the functional aspects—a sort of teleological information about the product—have often a qualitative nature. Furthermore, at the early phases of the development of a product, the requirements and the characteristics of the product, including the geometrical ones, are not precise. The designer still has a quite general idea of the definitive form of the product and vague tolerances are accepted. The need to enrich quantitative product models with qualitative specifications about the design intents has been advocated for more than 20 years now [2]. However, computer-based modelling systems are mainly focused on quantitative knowledge, whereas qualitative aspects are mainly expressed by text annotation for human reading. As a consequence, design relevant models are not (computationally) represented [2].

The ultimate purpose of this work is twofold: (*i*) to contribute to foundational issues of engineering design; and (*ii*) to support the development of advanced modelling systems for design knowledge. However, in order to achieve these goals, it is needed beforehand a clear understanding of what a product model is and how it relates to its corresponding physical products. We thus present in this paper an high level approach, based on foundational ontologies and knowledge representation techniques, concerned with the individuation and analysis of the general notions needed for an integrated model of design. Our proposal offers just a conceptual base that needs to be specialized and instantiated to be useful for the practical purposes of knowledge representation in design.

The paper is structured as follows: in Section 2 we introduce the proposed formal model and we discuss some shortcomings and problematic issues related to design knowledge representation. In section 3 we briefly compare our approach with similar research initiatives.

2 Representing design product knowledge

We sketch a *general and high-level* theory that is capable of representing in a uniform and integrated way both the qualitative and the quantitative aspects of design. Manufacturing aspects are excluded from this analysis, even though our model is compatible with a future extension that addresses these aspects. First-order logic (FOL) is employed as representational language. This has clearly some impacts on the technics, in particular *reification* technics, used to overcome some limitations in the expressive power.

2.1 The conceptual nature of design

We want to distinguish the design of a product from both its specifications, such as a Computer-Aided Design (CAD) file or a printed drawing on a piece of paper, and from the set of objects that are possibly produced. For this purpose, it is worth presenting our point by means of the *semiotic triangle* developed in semantics. For instance, the word ‘red’ in its predicative sense is related to the concept of *being red* (the *intension*) and to the class to things that are red (the *extension*). In the case of a design, we distinguish three aspects of it (cf. Figure 2): its *specifications* as the physical supports of the design, the *product type* as its intension, and the class of *physical products*, or simply products, as the extension of the product type. For example, John’s car (particular physical product) satisfies the technical specification of Ferrari Testa Rossa 512 TR (product type).

Firstly, the product type is distinguished from the specification, as the same product type may be represented by different supports, e.g., a CAD model and a pencil-made sketch. The specification is useful to represent and communicate aspects of the product type, but the design cannot be reduced to it.

Secondly, the product type is not identifiable with the class of products that can be realized by means of the design. This view copes with the fact that the

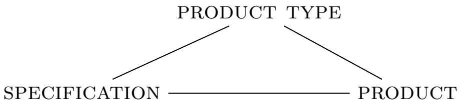

Fig. 2. Semiotic triangle for design

design may be about not yet realized products. If the design reduces to a class of products, then, before their realizations, one could only talk about the design in terms of future or possible objects. Although this view may be formalized by means of modal logic, we believe that it does not capture the nature of design. In principle, a design may exist even if the objects that are its realizations are never produced. This does not mean that a design may be about impossible objects: the point is that a design still exists, we can understand it, and practitioners can interact about it, even in the case that no concrete product is ever going to be realized. By focusing on the product type, we indeed construe the design as a *type* rather than a set of *tokens* (physical objects). Accordingly, we shall treat the product type as a *concept* in the sense made precise by the axioms we shall introduce in the next sections. From this perspective, our interpretation of the designing process departs from the one depicted in Figure 1. The outcome of the process there is understood as the "description of the technical product (TP(s))" plus "full information for possible manufacture" of a class of objects. By contrast, even though we recognize the importance of the specification (cf. Figure 2), we stress here the conceptual nature of the design. In the following sections, we do not discuss specification languages for engineering purposes, nor we will approach the analysis of manufacturing. We will rather formalize the conceptual nature of a design.

For this purpose, we rely on the DOLCE ontology [10]. More specifically, we consider an extension of DOLCE for representing roles [11] and the evolution of its core, called DOLCE-CORE [12]. These extensions are significant to our aim because they explicitly address the problem of representing the *intension* of concepts or properties. However, these theories are not enough to represent some important aspects of the design; hence, in what follows, we modify and extend them. In a first-order (and non-modal) setting, properties (and concepts) are usually represented by means of *predicates*. The *method of temporal arguments* [13] is a standard technique to account for (*i*) the dependence on time of the classification and (*ii*) the representation of change of objects through time. It consists of the introduction of time as an additional argument of the predicates and functions in the vocabulary that depend on time. For instance, $\mathtt{Red}(x)$ becomes $\mathtt{Red}(x,t)$. However, as noted in [12], this technique is not adequate to represent neither the contextual, social, or constructive aspects of concepts, nor their intensions.

To address these crucial aspects in FOL, we *reify* concepts into the domain of quantification, i.e., we introduce a new kind of entities: $\textsc{cn}(x)$ stands for "x is a *concept*" (or more generally a property). In this way, we can predicate on concepts, but we loose the possibility of representing classification via predication. It is then necessary to introduce a new primitive to relate the concepts to the entities they classify, a sort of (possibly intensional) 'instance-of' relation, that here we call *classification*: $\mathtt{CF}(x, y, t)$ stands for "the concept x classifies the entity y *as it is at time* t". From a design perspective $\mathtt{CF}(x, y, t)$ can be interpreted in a more restricted way as "the *product* y, as it is at time t, conforms to (the specification of) the *product type* x", or, more lengthily, "at time t, the *physical product* y has all the properties required to satisfy (the specification of) the *product type* x". The classification relation is temporally qualified: the entity y may change through time, thus the classification under a certain concept is, in general, contingent. For instance, a product y that at time t conforms with the (specifications of the) type x, could not conform anymore with x at time t' because of the loss of some properties necessary to be classified under x. That means that t does not individuate the time at which the classification is done but the time at which y is considered (measured, perceived, etc.). Consequently, the classification at time t does not necessarily implies the existence at t of the concept, see (a1) where $\mathtt{EX}(y, t)$ stands for "the entity y exists at time t". However, concepts are in time (a2), they can be created or destroyed (for instance, when no specification, including mental ones, exists any longer). Concepts created at a given time t can then classify, according to past information, entities that do not exist anymore at t.[2]

A similar modelling approach, even though (change through) time is not considered, is employed in the Industry Foundation Classes[3] (IFC)—a data modelling standard supported by most of the major CAD vendors—in which classification is called `IfcRelDefinesByType`. This relationship is used to represent the fact that the instances of a `IfcTypeProduct` (a product type) satisfy the properties defined by a `IfcProduct` (a physical product).

Standard extensional relations between concepts can be introduced relying on `CF`, e.g., the *extensional subsumption* relation (d1). However, concepts have an intensional nature, they are not reducible to their extensions, different concepts may classify the same entities: the extensional subsumption is not anti-symmetric, i.e., $x \subseteq y \land y \subseteq x \rightarrow x = y$ does not hold in our theory.

a1 $\mathtt{CF}(x, y, t) \rightarrow \textsc{cn}(x) \land \mathtt{EX}(y, t)$
a2 $\textsc{cn}(x) \rightarrow \exists t(\mathtt{EX}(x, t))$
d1 $x \subseteq y \triangleq \forall zt(\mathtt{CF}(x, z, t) \rightarrow \mathtt{CF}(y, z, t))$

2.2 The intensional dimension of design

For the sake of example, assume that in our domain all the round entities are also red and vice versa. In our framework, this does not imply the identity of the concepts (properties) *being red* and *being round.* The distinction between the extensional and the intensional level of design is also used in design models and

---

[2] For our goal, the time at which the classification is done is not relevant. However it is easy to add a second temporal argument to `CF` to account for that.

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

data modelling standards. In the IFC, for example, `IfcTypeProduct` is understood as the information common to all instances of `IfcProduct`. Borgo and colleagues [14] propose to understand the former in an intensional sense, i.e., as the properties characterising the instances of the latter.

After Carnap, the intensionality of properties is traditionally approached in logic from a modal perspective: the co-extensionality of *being red* and *being round* is contingent, it holds in the *actual world* but there are other *possible worlds* where *being red* and *being round* have different extensions. In our theory, the reference to possible worlds is not necessary although, indeed, one can still have an informal modal understanding of intensionality. Here, the intension is captured in a different way that is especially effective for products types.

The idea is that concepts cannot be characterized "in isolation", they always refer to other concepts. For instance, product types are usually characterized in terms of simpler concepts that are typically shared by the designers involved in a given phase, the common background of designers. This idea is quite similar to the one followed in [11], where the *identity criteria* for concepts are based on their *definitions*. However, in [11] the definitions of concepts are a sort of "placeholders" of their intensions, they are not structured and they are very weakly interlinked (by the notion 'used'). Here we use the DOLCE-CORE *quality spaces*—a formal variation of *conceptual spaces* [15]. The idea is to make (partially) explicit what characterizes a concept in intensional terms by linking it to some quality spaces.

Objects can be compared and characterized in terms of a variety of aspects such as weight, shape, size, color, function, etc. Such aspects are represented via (quality) spaces composed by *basic properties* (called *regions* in DOLCE-CORE).[4] For instance, the color space contains several basic properties, e.g., *being red*, *being orange*, *being scarlet*, etc. Basic properties in the same space can be organized in taxonomies or in more sophisticated ways: from ordering (weight, size) to complex topological or geometrical relations (color splinter).

We assume a finite number N of spaces SP_i that partition the basic properties BCN (a3)–(a5).[5] The idea is that basic properties—that still have an intensional nature—are used, but not created, during the design process, i.e., they represent the conceptual knowledge in the background, the conceptual knowledge that allows the product type to be characterized throughout the design phases.[6] Consequently, the intensional subsumption relation $\sqsubseteq$, assumed to be a classical extensional mereology closed under sum [16], is also local to spaces (a6). Ba-

---

[4] Quality spaces correspond to the *dimensions* of conceptual spaces in [15].

[5] This implies that basic properties are local, private, to spaces. This choice can be debated if one assumes that colors can be organized in different ways, or that spaces are associated to instruments with different resolution. In this case, it seems reasonable to assume that spaces share basic properties. Here we prefer to duplicate the basic properties, given their quite clear conceptual nature in these cases, and add *correspondence* links between them.

[6] As in the case of correspondence links, one can think that there are other intensional (logical) links among these basic properties (in the same or in different spaces). This is a very interesting aspect that, for reasons of space, we do not consider here.

sic properties are concepts, therefore they classify entities; (a7) establishes the link between the intension and the extension of concepts (the vice versa does not hold). Firstly, note that, differently from DOLCE-CORE, the *individual qualities*—a sort of tropes, e.g., the redness of my car—are not in the domain of quantification anymore.[7] Secondly, and more importantly, in DOLCE-CORE when an object has a quality, then this quality is completely determined.[8] For instance, a *uniformly* colored object is always mapped to an *atomic* basic property, a maximally specified property, the maximal information one disposes of (according to the resolution assumed in the space). A *multi-colored* object is mapped to a non-atomic property. However, this property is just the sum of the colors of all its uniformly colored parts. Differently, to account for underspecification and tolerances, a uniformly colored object could here be mapped to a *non*-atomic basic property; its color is just one of the atomic parts of the non-atomic property. This *disjunctive* reading is more useful in the design process, especially during the first phases when one has only a qualitative and rough characterization of the product type. This means that the basic properties need to be interpreted as the properties of the whole object under classification. The properties of the parts and of the structural aspects of a product type are quite problematic and will be briefly discussed in Section 2.3.

As we saw, the product types are complex concepts cCN (a8) characterized in terms of basic properties. We need then to link complex concepts to their basic properties: $\mathtt{CH}(x, y)$ stands for "the complex concept x is characterized by the basic property y" (a9). For example, Ferrari Testarossa 512 TR (product type) is characterised by the basic properties *being red*, *being 1500kg heavy*, among others. Complex concepts are characterized at least by two, but usually several, basic properties, i.e., they have a multi-dimensional nature (a10).[9] This is similar to composition of spaces in more complex ones, at least if the geometry of the complex space can be defined in terms of the geometries of its components.

a3 $\mathrm{bCN}(x) \to \mathrm{CN}(x)$
a4 $\mathrm{bCN}(x) \leftrightarrow \bigvee_{i\in\{1,\dots,\mathrm{N}\}} \mathrm{SP}_i(x)$
a5 $\bigwedge_{i\neq j\in\{1,\dots,\mathrm{N}\}}(\mathrm{SP}_i(x) \to \neg\mathrm{SP}_j(x))$
a6 $x \sqsubseteq y \to \bigwedge_{i\in\{1,\dots,\mathrm{N}\}}(\mathrm{SP}_i(x) \leftrightarrow \mathrm{SP}_i(y))$
a7 $x \sqsubseteq y \to x \subseteq y$
a8 $\mathrm{cCN}(x) \to \mathrm{CN}(x) \wedge \neg\mathrm{bCN}(x)$
a9 $\mathtt{CH}(x, y) \to \mathrm{cCN}(x) \wedge \mathrm{bCN}(y)$
a10 $\mathrm{cCN}(x) \to \exists yz(\mathtt{CH}(x, y) \wedge \mathtt{CH}(x, z) \wedge \bigvee_{i\in\{1,\dots,\mathrm{N}\}}(\mathrm{SP}_i(y) \wedge \neg\mathrm{SP}_i(z)))$
a11 $\mathrm{cCN}(x) \to (\mathtt{CF}(x, y, t) \leftrightarrow \forall z(\mathtt{CH}(x, z) \to \mathtt{CF}(z, y, t)))$

---

[7] This option has already been considered in [17].

[8] In [15] objects are points in multi-dimensional spaces. Objects are then fully characterized with respect to all the possible qualities.

[9] Actually this is also the case of some basic properties, e.g., the color space has three dimensions: hue, saturation, and brightness. One could also think that colors (or, better, shapes) may be *designed*, i.e., there exist some original or proprietary color-properties. We do not consider these aspects that, however, could be modeled by extending CH or by assuming hue-, saturation-, and brightness-properties as basic.

Firstly, we need to guarantee that the classification under a complex concept x reduces to the classification under all the basic properties that characterize x (a11), i.e., the extension of x is the intersection of the extensions of all the basic properties that characterize x. This seems to authorize to interpret CH as a sort of intensional subsumption. However, the antisymmetry of $\sqsubseteq$ would imply the identity of complex concepts with the same characterizing basic properties. This is acceptable only if we assume that the characterization of complex concepts is always complete. For the moment, we prefer a weaker approach that allows also for partial characterizations of concepts, i.e., two different concepts can have the same partial characterization. Note that these partial characterizations are particularly interesting for hiding very specific or practical properties.

Secondly, while it seems quite reasonable to have a static view on the background knowledge, the product type under design could be intended as an evolving concept. This can be modeled by adding a temporal parameter to both CH and CF, i.e., what characterizes a concept can vary through time, thus the classification depends also on the time at which the concept is considered: $\mathtt{CF}(x, t, y, t')$ stands for "the complex concept x, *as it is at* t, classifies the object y, *as it is at* t'" (a12). The double temporal qualification allows to reclassify an object as it is at t' across the evolution through time of a given concept.[10] (a11) needs then to be substituted by (a13) where we assume the basic properties to be static. Consequently, the extensional subsumption relations between complex concepts may be temporally qualified as in (d2). The identity of concepts can be intended in terms of the 'trajectory' across time of its characterizing properties, i.e., $(\mathtt{CH}(x, z, t) \leftrightarrow \mathtt{CH}(x', z, t)) \rightarrow x = x'$, but weaker options that consider the designers and/or the design process can be considered. For this reason, we do not commit to this identity criterion.

a12 $\mathtt{CF}(x, t, y, t') \rightarrow \mathrm{cCN}(x) \wedge \mathtt{EX}(x, t) \wedge \mathtt{EX}(y, t')$
a13 $\mathrm{cCN}(x) \rightarrow (\mathtt{CF}(x, t, y, t') \leftrightarrow \forall z(\mathtt{CH}(x, z, t) \rightarrow \mathtt{CF}(z, y, t')))$
d2 $x \,{}^{t_1}\!\subseteq^{t_2} y \triangleq \mathrm{cCN}(x) \wedge \mathrm{cCN}(y) \wedge \forall zt(\mathtt{CF}(x, t_1, z, t) \rightarrow \mathtt{CF}(y, t_2, z, t))$

At this point we can also address the notion of requirement that is usually defined as the "[p]roperty that is required to be fulfilled during the origination phase of the object to satisfy the [customer] requirements" [1, p.317]. Customer requirements can then be seen as the (maybe rough) idea of product that the customers, as opposed to designers, have. Note that also the requirements, i.e., the customer concept, can evolve in time during the design process, maybe because of market change, or maybe because of the interaction with engineers that discovered some unrealizable constraints. We have then two complex concepts characterized in terms of intensions (their characterizing basic properties) and extensions (the objects that they classify). This allows for both an inten-

---

[10] Firstly, note that t is not the time at which the classification is done, it just 'freezes' the concept (while t' freezes the object). Secondly, to avoid evolving concepts, one could follow [11] and introduce a 'revision' relation between static concepts. Formally the two approaches are equivalent, we preferred the first approach because it seems more adequate for capturing the design practice.

sional and an extensional comparison. One can say that, at t, the requirements, represented by the concept c_r, are satisfied by a product type c_p if ${c_p}^t \subseteq^t c_r$. Requirements are then a sort of necessary properties of the product. However, this could mean that the product type is too specific, i.e., the matching exists only when both ${c_p}^t \subseteq^t c_r$ and ${c_r}^t \subseteq^t c_p$ hold. In any case, the way in which we realized the products can be very different from what the customer requirements report. An intensional matching at t may be defined in terms of $\mathtt{CH}$ as $\mathtt{CH}(c_r, x, t) \rightarrow \exists y (y \sqsubseteq x \wedge \mathtt{CH}(c_p, y, t))$, i.e., the basic properties that characterize c_p are intensionally subsumed by the ones that characterize c_r. Again we can strengthen this notion to a perfect matching. Finally, note that one could distinguish two $\mathtt{CH}$ relations, one for necessary and one for optional properties of product. This would allow for distinguishing, at the level of requirements, *demands* from *wishes*. In addition, because intensions are expressed in terms of basic properties and any space may have a metric, the level of (mis)matching between requirements and design can be measured.

As an illustrative example, let us consider a customer asking for a product to prepare Italian coffee. In addition to the function, her requirements regard the height (between $16cm$ and $20cm$) and the material (aluminium) of the product. Assuming to dispose of the function, height, and material spaces, the required product type can be represented by a complex concept $\mathtt{rpt}$ that, at the starting time $\mathtt{t_0}$ is characterized by three basic properties: $\mathtt{CH}(\mathtt{rpt}, \mathtt{prep\_it\_coffee}, \mathtt{t_0})$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{16\text{-}20cm}, \mathtt{t_0})$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{alu}, \mathtt{t_0})$. Assume that $\mathtt{16\text{-}20cm}$ is a non-atomic basic property while $\mathtt{alu}$ an atomic one, even though, because properties apply to the whole product, this does not necessarily imply the product to be exclusively made of aluminium. More controversial is the case of $\mathtt{prep\_it\_coffee}$ because it is unclear, for instance, whether the kind of energy used to prepare the coffee is part of the function or pertains a separate space. For our illustrative purposes we assume the first hypothesis and that there are at least two (intensional) subfunctions ($\sqsubseteq$) of $\mathtt{prep\_it\_coffee}$, namely $\mathtt{prep\_it\_coffee\_methane}$ and $\mathtt{prep\_it\_coffee\_elect}$. The designers start to consider the requirements (specified by the customer in some way) by assuming a designed product type $\mathtt{dpt}$ (a complex concept) such that $\mathtt{CH}(\mathtt{dpt}, \mathtt{prep\_it\_coffee\_elect}, \mathtt{t_0})$, $\mathtt{CH}(\mathtt{dpt}, \mathtt{16\text{-}20cm}, \mathtt{t_0})$, $\mathtt{CH}(\mathtt{dpt}, \mathtt{steel}, \mathtt{t_0})$. This choice is partially based on the expertise of the company in developing products for induction hobs. Note that the designers know that (at $\mathtt{t_0}$) $\mathtt{dpt}$ does not match $\mathtt{rpt}$ because $\mathtt{steel}$ is not a subconcept of $\mathtt{alu}$. Therefore they interact with the customer to explain the strategical importance of having a product to prepare italian coffee to work with induction hobs, and this constraints the material to steel. The customer agrees on that but change the size-constraints, in this case she wants a very small, less than 12cm high, product. The designers agree on that and at time $\mathtt{t_1}$ the requirements are matched: $\mathtt{CH}(\mathtt{rpt}, \mathtt{prep\_it\_coffee}, \mathtt{t_1})$, $\mathtt{CH}(\mathtt{rpt}, \leq\mathtt{12cm}, \mathtt{t_1})$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{steel}, \mathtt{t_1})$ and $\mathtt{CH}(\mathtt{dpt}, \mathtt{prep\_it\_coffee\_elect}, \mathtt{t_1})$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{10\text{-}11cm}, \mathtt{t_1})$, $\mathtt{CH}(\mathtt{rpt}, \mathtt{steel}, \mathtt{t_1})$.[11]

---

[11] Here we are totally liberal with respect to how concepts can change through time. However, constraints on the way concepts "move" inside the spaces can be added.

2.3 Advanced and problematic aspects

Dependencies between spaces One interesting aspect of spaces is that it is quite easy to add dependence links between basic properties, i.e., to shape the composed space. This is the case of the color splinter: not all the combinations of hue, saturation, and brightness correspond to a color, a given hue constraints the possible values of brightness and saturation. In this case the consistency of the different characteristics of a model can be guaranteed from the beginning. The proposed framework can be extended to take into account these dependencies. From the designing perspective, this is interesting because it makes possible to represent the dependencies between different designing phases and design specifications by means of the dependencies between the spaces that are used at these phases. For instance, functional modeling is particularly relevant during the first stages of the design process, where more emphasis is given on *what* the product is designed to do, rather than on *how* this purpose will be achieved. Vice versa, at the design embodiment stage, designers focus more on the physical properties of the product, i.e., on how the functionality is realized. Dependencies between functional and physical properties can help the designer in (*i*) understanding how a required function constrains the product's physical layout—a top-down perspective: from the abstract functional view to the concrete physical one; (*ii*) verifying whether the chosen physical layout can, at least in principle, fulfill the required functionality (a bottom-up perspective); (*iii*) making explicit possible *accidental functions*, as opposed to *proper functions*, that could result from (improper) usages of the product. For instance, a screwdriver has the proper function of, it has been designed for, driving screws, while it has the accidental function of opening cans. According to de Vries [18], this analysis is very important to individuate dangerous improper uses of products to be avoided.

Structural knowledge We have already observed the importance of the component and assembly models. Together they allow to represent the *structure* of the product, i.e., how it can be decomposed into simpler products and which assembly constraints hold among these components. The "recursive" decomposition usually stops at standard (at least for a given company) units that are reused in several products. Structural knowledge is then really central to any modern design of complex products, e.g., cars or planes.[12] This structure is usually represented by means of a parthood relation between product types, i.e., by a set of necessary (and sufficient) conditions about the parts of an object. Both UML and ER languages have primitives to represent different kinds of part-whole relations and, in knowledge representation, mereology has now a quite long tradition (see

---

[12] One could say that the design itself reduces to reuse components already designed. However, as already said some properties like colors or shapes can be designed (actually, structural information seems fundamental to design new shapes). In these cases, it is not clear to us if the designed properties are reducible to original ways of putting together more basic properties or if an *extension* of a space is needed. It would be interesting to analyze creativity in terms of spaces.

[16, 19]). However, in this way, the structure is not intended as a space of properties of the product. Consequently, one lacks a structural similarity or distance that would offer quite precious information to guide the product development, test, and refinement. Recently, there has been a number of proposals to build a conceptual, and cognitively based, space for parthood [20, 21]. These approaches are quite complex and do not really represent the assembly knowledge. More promising, at least in our view, is the idea of representing structures as *patterns* or, more precisely, as *structural graphs*, i.e., labelled and directed graphs which nodes stand for product types and arcs for assembly relations. The construction of these graphs is complex, but it allows to fully capture the structural knowledge and to import or adapt standard measures of similarity between graphs to introduce a metric in the space of structures. In addition, similar graphs could also be useful to represent *relational* properties, i.e., properties that hold because the product has some relations with the external world, e.g., ergonomic properties or affordances. This would be quite relevant for *user-centered* design.

Without structure-spaces, to minimally represent structural constraints, we extend our framework with a temporally qualified classical extensional mereology defined on objects: $\mathtt{P}(x, y, t)$ stands for "the object x is part of the object y at time t". Consider the previous example of the Italian coffee and assume that at time $\mathtt{t}_2$ the designers subdivide the main functionality *prepare Italian coffee* in sub-functionalities *to boil water*, *to store powdered coffee*, *to filter coffee by boiled water*, *to store brewed coffee* and *to serve coffee.* Amongst the various solutions for the embodiment of these functionalities, they decide to develop a moka pot product type (`moka`) with, among the others, three components (that are themselves product types): boiler pot (`pot`), coffee container (`container`), and filter (`filter`) that are characterized (in terms of `CH`) by the first three functions discussed above (plus additional heigh and material properties). In the proposed theory, this structural constraint is represented by (f1), but because a structure space does not exist, it is not possible to understand if this constraint matches the original one about the function of the whole object.

$$\mathbf{f1}\ \ \mathtt{CF}(\mathtt{moka}, \mathtt{t}_2, x, t) \rightarrow \exists yzw(\mathtt{CF}(\mathtt{pot}, \mathtt{t}_2, y, t) \wedge \mathtt{CF}(\mathtt{container}, \mathtt{t}_2, z, t) \wedge \\ \mathtt{CF}(\mathtt{filter}, \mathtt{t}_2, w, t) \wedge \mathtt{P}(y, x, t) \wedge \mathtt{P}(z, x, t) \wedge \mathtt{P}(w, x, t))$$

3 Discussion and related work

We presented a high-level modelling approach based on conceptual spaces and ontology engineering methods for the formal representation and integration of (qualitative and quantitative) aspects of design. Foundational issues related to engineering design and the integration of different aspects of design knowledge have been discussed in various research areas, from knowledge representation approaches, to theories of design and philosophy of technology.

The conceptual view proposed in this paper is mostly based on the engineering literature. According to Pahl and colleagues, the goal of designing is "the mental creation of a new product" [3, p.1]. Along the same lines, Lindemann [6] distinguishes between the *content* of a design model (e.g. geometric content) and its specification by means of a representational language (2D geometry). Despite the emphasis on the "mental" side of design given in [3], a design does not have to be confused with its mental representation in our theory. Kroes stresses the difficulty to capture this conceptual nature of design: "When referring to a car design [model], for instance, what is meant is not usually its production plan but something that has more to do with the properties of the car itself [...]. It is not easy to grasp what this 'something' is" [22, p.146]. In section 2.1 concepts are intensionally understood as properties that their corresponding instances have to satisfy. In this sense, a (complex) concept (product type) is a bundle of basic properties defined by designers to satisfy customers' requirements.

In the philosophical analysis of engineering design proposed by Vermaas and colleagues [23], designing tasks are mainly aimed at providing a description for a technical product: "[...] the *core* of technical designing lies in finding a description D_S of an artefact with a physical structure S that is able to effectively and efficiently fulfil a certain function F" [23, p.28] (emphasis ours). In this perspective, D_S is a document specified in a modelling language suited for engineering purposes. The authors thus focus on specification as the core dimension of design, while we have rather focused on its conceptual nature. The two approaches, however, are not contrasting but rather orthogonal, since concepts have to be specified for representational and communication purposes.

Knowledge representation has been primarily focused on the physical makeup of products, while little (if any) attention has been given to design models. For instance, the Core Product Model (CPM) [24] associates the class `artifact` to (requirement) `specification`, meaning that the former has to satisfy the latter's properties. However, neither the difference between requirements properties and design specifications properties is discussed, nor it is clear whether `specification` refers to an encoded description, or its content.

In ontology engineering, the Information Artifact Ontology (IAO) [25] has been explicitly developed to represent information entities, that is, the content of encoded descriptions. The IAO thus shares in some degree the same purpose of the work hereby presented. There are however some relevant distinctions to be noted: (*i*) the IAO directly links a representation to the physical object that it is about. In our approach, concepts mediate this relationship (cf. Figure 2), since they cannot be reduced to their extensions, as argued in Section 2.1. (*ii*) `information content entity` (`concept` in our terminology) is only weakly characterised as an entity that existentially depends on some representation and is about something else. In our theory, there is no such a dependence, since a product type is not necessarily specified in a physical medium. Additionally, we have provided a more detailed characterisation of concepts by taking into account the theories of quality spaces in DOLCE-CORE.

Acknowledgements: This work was partially funded by the VISCOSO project financed by the Autonomous Province of Trento through the "Team 2011" funding programme, and the FourByThree project funded by the European Horizon 2020 program (grant agreement 637095).

References

1. W.E. Eder and S. Hosnedl, *Design Engineering. A Manual for Enhanced Creativity.* CRC Press, 2008.
2. S.K. Chandrasegaran, K. Ramani, R.D. Sriram, I. Horvath, A. Bernard, R.F. Harik, and W. Gao, "The evolution, challenges and future of knowledge representation in product design systems," *Computer-Aided Designd*, vol. 45, pp. 204–228, 2013.
3. G. Pahl, W. Beitz, J. Feldhusen, and K.H. Grote, *Engineering Design. A Systematic Approach.* Springer-Verlag Berlin Heidelberg, 2007.
4. C. Bock, X. Zha, H. Suh, and J. Lee, "Ontological product modeling for collaborative design," *Advanced Engineering Information*, vol. 24, pp. 510–524, 2010.
5. F.L. Krause, F. Kimura, T. Kjelberg, and S.C.Y. Lu, "Product modelling," *Annals of the CIRP*, vol. 42, no. 2, pp. 695–706, 1993.
6. U. Lindemann, "Models of design," in *An Anthology of Theories and Models of Design. Philosophy, Approaches and Empirical Explorations* (A. Chakrabarti and L.T.M. Blessing, Eds.), ch. 6, pp. 100–121, Springer-Verlag London, 2014.
7. M.S. Erden, H. Komoto, T. Beek, V. D'Amelio, E. Echavarria, and T. Tomiyama, "A review of function modeling: Approaches and applications," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, vol. 22, pp. 147–169, 2008.
8. D. Deneux, "Introduction to assembly features: an illustred synthesis methodology," *Journal of Intelligent Manufacuring*, vol. 10, pp. 29–39, 1999.
9. 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*, vol. 6, no. 11, pp. 12–21, 2006.
10. C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari, "Wonderweb deliverable D18," tech. rep. CNR, 2003.
11. C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, and N. Guarino, "Social roles and their descriptions," in *Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning (KR 04)* (D. Dubois, C. Welty, and M. A. Williams, Eds.), (Whistler Canada), pp. 267–277, 2004.
12. S. Borgo and C. Masolo, "Foundational choices in DOLCE," in *Handbook on Ontologies* (S. Staab and R. Studer, Eds.), pp. 361–381, Springer Verlag, 2nd ed., 2009.
13. L. Vila and H. Reichgelt, "The token reification approach to temporal reasoning," *Artifical Intelligence*, vol. 83, pp. 59–74, 1996.
14. S. Borgo, E.M. Sanfilippo, A. Sojic, and W. Terkaj, "Ontological analysis and engineering standards: An initial study of ifc," in *Ontology Modeling in Physical Asset Integrity Management*, Springer Verlag Berlin Heidelberg, 2015.
15. P. Gärdenfors, *Conceptual Spaces: the Geometry of Thought.* Cambridge, Massachussetts: MIT Press, 2000.
16. R. Casati and A. Varzi, *Parts and Places. The Structure of Spatial Representation.* Cambridge, MA: MIT Press, 1999.
17. C. Masolo and S. Borgo, "Qualities in formal ontology," in *Foundational Aspects of Ontologies (FOnt 2005) Workshop at KI 2005* (P. Hitzler, C. Lutz, and G. Stumme, Eds.), (Koblenz, Germany), pp. 2–16, 2005.
18. M. Vries, "Translating customer requirements into technical specifications," in *Philosophy of Technology and Engineering Sciences. Handbook of the Philosophy of Science.*, pp. 488–512, Elsevier, 2009.

19. P. Simons, *Parts: a Study in Ontology.* Oxford: Clarendon Press, 1987.
20. S. Rama Fiorini and M. Abel, “Part-whole relations as products of metric spaces,” in *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pp. 55–62, 2013.
21. S. Rama Fiorini, P. Gärdenfors, and M. Abel, “Representing part-whole relations in conceptual spaces,” *Cognitive Processing*, vol. 15, no. 2, pp. 127–142, 2014.
22. P.Kroes, *Technical Artefacts: Creations of Mind and Matter. A Philosophy of Engineering Design.* Springer, 2012.
23. P. Vermaas, P.Kroes, I. de Poel, and M.Franseen, *A Philosophy of Technology. From Technical Artefacts to Sociotechnical Systems.* Morgan and Claypool Publishers, 2011.
24. S. Fenves., S. Foufou, C. Bock, and R.D. Sriram, “CPM: a core model for product data,” *Journal of Computing and Information Science in Engineering*, vol. 8, 2008.
25. B. Smith and W. Ceusters, “Aboutness: Towards foundations for the information artifact ontology,” in *Proceedings of the Sixth International Conference on Biomedical Ontology (ICBO)*, 2015.