Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2009), *23*, 3–21. Printed in the USA.
Copyright © 2009 Cambridge University Press 0890-0604/09 $25.00
doi:10.1017/S0890060409000079

A formal ontological perspective on the behaviors and functions of technical artifacts

STEFANO BORGO,[1] MASSIMILIANO CARRARA,[2] PAWEL GARBACZ,[3] AND PIETER E. VERMAAS[4]

[1]Laboratory for Applied Ontology, ISTC-CNR, Trento, Italy
[2]Department of Philosophy, University of Padua, Padua, Italy
[3]Department of Philosophy, John Paul II Catholic University, Lublin, Poland
[4]Department of Philosophy, Delft University of Technology, Delft, The Netherlands

(RECEIVED September 20, 2007; ACCEPTED June 24, 2008)

Abstract

In this paper we present a formal characterization of the engineering concepts of *behavior* and *function* of technical artifacts. We capture the meanings that engineers attach to these concepts by formalizing, within the formal ontology DOLCE, the five meanings of artifact behavior and the two meanings of function that Chandrasekaran and Josephson identified in 2000 within the *functional representation* approach. We begin our formalization by reserving the term "behavior" of a technical artifact as "the specific way in which the artifact occurs in an event." This general notion is characterized formally, and used to provide definitions of actual behaviors of artifacts, and the physically possible and physically impossible behaviors that rational agents believe that artifacts have. We also define several other notions, for example, input and output behaviors of artifacts, and then show that these ontologically characterized concepts give a general framework in which Chandrasekaran and Josephson's meanings of behavior can be explicitly formalized. Finally we show how Chandrasekaran and Josephson's two meanings of artifact functions, namely, device-centric and environment-centric functions, can be captured in DOLCE via the concepts of *behavioral constraint* and *mode of deployment* of an artifact. A more general goal of this work is to show that foundational ontologies are suited to the engineering domain: they can facilitate information sharing and exchange in the various engineering domains by providing concept structures and clarifications that make explicit and precise important engineering notions. The meanings of the terms "behavior" and "function" in domains like designing, redesigning, reverse engineering, product architecture, and engineering knowledge bases are often ambiguous or overloaded. Our results show that foundational ontologies can accommodate the variety of denotations these terms have and can explain their relationships.

Keywords: Behavior; Engineering; Formal Ontology; Function; Technical Artifact

1. INTRODUCTION

In this paper we focus on providing uniform and rigorous characterizations of the engineering concepts of the *behavior* and *function* of technical artifacts in terms of a series of formalized definitions grounded in a foundational ontology. This formalization is motivated by two observations that can be taken as commonplace within the context of this special issue on engineering ontologies. The first is that these two notions are key concepts in engineering; behavior and, especially, function, play pivotal roles in the descriptions of artifacts in designing, redesigning, reverse engineering, product architecture, and engineering knowledge bases. Yet, despite this importance, consensus on what is meant by artifact behavior and function has not yet been achieved in the technological domain (Chittaro & Kumar, 1998; Chandrasekaran & Josephson, 2000; Chandrasekaran, 2005; Kitamura et al., 2005/2006). Engineers may attach different meanings to these key concepts, depending on their different disciplinary backgrounds, or the tasks at hand, and this conceptual freedom hampers the reliable storage, retrieval, and communication of functional and behavioral descriptions of artifacts as needed in, say, redesigning or in the use of engineering knowledge bases.

The second observation that motivates our formalization is that the emerging field of engineering ontology provides an excellent means to overcome this lack of conceptual consensus within the engineering community (Chandrasekaran et al., 1999; Kitamura et al., 2005/2006). Generally speaking, two basic elements set the *quality* and *reliability* of knowledge and information exchange: one, the adoption of a suitable communication language, and two, the existence of a shared conceptual "view" (framework) on reality, that is, an agreement on existing types of objects, properties, and so

Reprint requests to: Stefano Borgo, Laboratory for Applied Ontology (ISTC-CNR), Via alla cascata 56/C, Povo, 38100 Trento, Italy. E-mail: borgo@loa-cnr.it

forth. We elaborate our theory using the language of formal logic because it is at the basis, either explicitly or implicitly, of every scientific language, thus including engineering, and focus our attention on the framework engineers use to understand their domain of study. This framework can be seen as an implicit ontology that engineers work with, one that is only partly explicit. Indeed, the difficulties engineers register in the use of notions like behavior and function show that this implicit ontology should be further analyzed by explicitly recognizing and formalizing existing and new conceptual distinctions. This is, after all, the role of an ontology: to classify types of entities, properties, relationships, and their dependences, and to make explicit the conceptual differences and related interconnections needed to make sense of the domain of interest.

Just spelling out the implicit ontology of engineers will not do the job. When looking into engineering activity one sees that engineers use distinct ontologies depending on the field in which they work, and we must recognize that developing and using engineering ontologies, as such, will not provide conceptual consensus per se. It is common knowledge that different perspectives may lead to different results: what makes sense from one perspective may not have a precise correspondence or even be false in another. Although not recognized as clearly, this also happens for ontologies. Where the layman sees lights and sounds, an engineer sees a variety of wavelengths, more than the layman can imagine within his limited everyday ontology. Engineers use different ontologies depending on the domain in which they work. For example, a civil engineer deals with objects in Newtonian space, whereas an engineer specialized in GPS technology adopts general relativistic space–time; and for a civil engineer a specific material will have different properties than it has for a chemical engineer. The problem of making explicit the ontologies adopted by practitioners is common to many areas and motivated in part the rise of the domain of *applied ontology* (Staab & Stuber, 2004). There are several notions of ontology, ranging from terminological to foundational (Borgo, 2007), and a variety of ways to apply them. Because we aim at a rigorous foundation for engineering terminology, we based our work on the approach called foundational ontology. The crucial characteristics of *foundational ontologies* can be summarized as follows:

- they are *general* in the sense that they are limited to the most reusable and widely applicable concepts leaving to the user to populate the ontology with more specific concepts;
- they are *reliable* because they are logical theories with rich axiomatizations and carefully analyzed formal consequences (theorems); and
- they are *well organized*, because the construction of a foundational ontology is based on explicitly motivated philosophical principles.

Our goal in this paper is to present formalizations of the engineering concepts of *behavior* and *function* of technical artifacts that capture the informal meanings these terms have in engineering practice and that are ontologically motivated. In this formalization, we adopt the meanings provided by domain experts without doubting their adequacy and without trying to reduce or modify them. Once the formalizations are in place, we can see that the relationships among these meanings are clear and can be expressed formally. The analysis was carried out within the framework of the DOLCE ontology (Masolo et al., 2003), the *Descriptive Ontology for Linguistic and Cognitive Engineering*.[1] DOLCE is a *foundational ontology*, that is, a knowledge system developed to characterize *explicitly* a general viewpoint on reality or, at least, a relevant part thereof: the aim is to provide a structure for knowledge of very general concepts and to constrain their intended meaning formally. Foundational ontologies comprise, as said, only concepts and relations that are independent from application domains, and, to be applied to a specific domain, they need to be enriched with the appropriate domain-dependent concepts and relations. Hence, by incorporating the engineering notions of behavior and function in DOLCE, we arrive at an ontological assessment of these notions and at an extension of DOLCE that can be used in the different engineering domains. The current paper builds on previous work (Borgo et al., 2006), in which the focus is limited to the concept of behavior.

Yoshinobu Kitamura and Riichiro Mizoguchi have advanced a similar line of research with their recent contribution (Kitamura et al., 2005/2006). In it they constructed an upper level ontology in which they embed their earlier results of ontological analysis of artifactual functions. In brief, the core model developed in their approach, namely, the functional concept ontology (Kitamura et al., 2002), is based on the ontology of device and function, which in turn, is based on a simple upper level ontology. The whole framework is an extension of the functional modeling language FBRL (Function and Behavior Representation Language).

A (base-)function of a device is defined as a role performed by a certain behavior of this device in a fixed teleological context, whereas a behavior is understood as the objective, that is, independent from designers' intentions, interpretation of the input/output relations that connects the device to its environment. Besides modeling base functions, Kitamura and Mizoguchi define function types and metafunctions, which represent the interdependencies among base functions. Other important notions employed in their theory comprise the notion of operand, conduit and medium. All these entities are categorized within the upper level ontology.

The methodology advanced by Kitamura and Mizoguchi adorns this notion of function with a rich conceptual structure that hinges upon two relationships: "is-a," that is, the relation of subsumption, and "is-part-of," that is, the relation of parthood. Thus, within their approach one is able to classify functions within certain domains and to perform functional decompositions of these functions. The notion of "way of function achievement" they introduce, allows them to

[1] http://www.loa-cnr.it/DOLCE.html

explicate the background knowledge on functional decomposition. This type of knowledge consists of physical principles and theories that explain why a given organization of subfunctions compose the overall function to be decomposed. The whole ontological system of Kitamura and Mizoguchi is available by means of an ontology editor in an environment for building/using ontologies named Hozo (Mizoguchi et al., 2007).

The plan of our paper is as follows. In Section 2 we review how engineers characterize the concepts of artifact behavior and artifact function, focusing on the informal meanings of behavior and function of domain experts. We go on to outline two general approaches to the characterization of behavior and function and present the functional representation approach of Chandrasekaran and Josephson (2000) as the one we aim to formalize. We introduce DOLCE in Section 3, collecting the elements we need for our formalization of behavior and function. These formalizations are then given in Sections 4 and 5, respectively. We conclude, in Section 6, with some observations, including a discussion on the formalization of function in the functional modeling approach, which is the rival to the functional representation approach.

2. ARTIFACT BEHAVIORS AND ARTIFACT FUNCTIONS IN ENGINEERING

The importance of the concepts of *behavior* and *function* in engineering and the lack of consensus on their meaning is vividly illustrated by the existence of a number of design models that already by their names refer to these concepts but give different definitions of behavior and function. Examples of such models are the function–behavior–structure (FBS) model of John S. Gero (Gero, 1990; Rosenman & Gero, 1998; Gero & Kannengiesser, 2004), the function– behavior–state model of Yasushi Umeda and collaborators (Umeda et al., 1996, 2005; Umeda & Tomiyama, 1997), and the structure–behavior–function (SBF) model of Ashok Goel (Goel, 1991; Bhatta et al., 1994; Bhatta & Goel, 2002). These models share an analysis of the design process as one in which the engineers' reasoning can be reconstructed as reasoning from functions, via functional, to physical descriptions of artifacts that are assumed to be able to perform these functions. In Gero's FBS model the functions of technical artifacts are typically taken as the "purposes of the design" but are also on some occasions equated with the "results of the behavior" of the artifact. Behavior is, in turn, characterized by how the structure of an artifact achieves its functions, the artifact's actions or processes in the given circumstances of the natural environment, or as attributes that can be derived from the structure of the artifact (Vermaas & Dorst, 2007). In the FBS model of Umeda and collaborators, function and behavior have more stable meanings: a function is defined as "a description of behavior abstracted by human through recognition of the behavior in order to utilize it," and a behavior is defined as "a transition of states along time," where states consist of "entities, their attributes and their structure" (Umeda et al., 2005). In the work of Goel, the function of an artifact is defined as "a behavioral abstraction" represented by a schema that gives the input and output behavioral states of the function and a reference to the causal mechanism internal to the device that achieves the function, and an (internal causal) behavior is represented by sequences of state transitions (Bhatta et al., 1994). At first sight it may seem that some conceptual consensus can be derived from these proposals. The second and third seem fairly similar, but then one has to ignore Goel's inclusion of the responsible internal causal mechanisms in his definitions of behavior and function. Gero's characterizations may seem consistent with the resulting conceptual unity, but this appearance is mainly due to the fact that Gero's characterizations are divergent. The possibilities for singling out conceptual consensus become even slimmer if one takes into account other models of designing, models that may diverge from the three mentioned by, for instance, denying that the concept of behavior has a role to play in designing. An example of such a model is the functional basis model of designing proposed by Stone and Wood (2000) in which a function is defined as the general input/output relationships of an artifact that have the purpose of performing an overall task, and in which designing is taken as a process in which customer needs are directly mapped to structural design solutions via functional descriptions.

To some extent conceptual consensus also seems not to be a prime value in engineering. Different ways of laying down the meaning of behavior and function exist and coexist, and engineers and design methodologists like Gero, seem to make use of the available conceptual flexibility, say by adopting new definitions when there is reason for that, instead of being concerned about this flexibility. Conceptual uniformity is not a good in itself; effectiveness of design is, and it has been argued that the lack of common meanings is helpful in designing as it provides room for creativity when engineers with different (conceptual) backgrounds collaborate in design processes (Bucciarelli, 1994).

The disadvantage of a lack of conceptual consensus surfaces, however, when functional and behavioral descriptions have to be shared: when, for instance, designing is modeled as a procedure in which, in part, existing knowledge about the relations between the functions, behavior, and physical structure of artifacts is retrieved from knowledge bases,[2] then having a common set of definitions is essential. This sharing of descriptions among engineers forces engineers to come out of their disciplinary and local conceptual "niches," and defines the task of resolving conceptual differences. One way of taking up this task is to continue our discussion of existing models and to arrive at a set of definitions for behavior and

[2] The FBS models of Gero and of Umeda et al., and the functional basis model all incorporate such steps in which knowledge about functions, behaviors, and structure is drawn from knowledge bases. In the functional basis model the retrieved knowledge concerns physical design solutions for functions only, thus ignoring behavioral descriptions (Bryant et al., 2006).

function that integrates the different proposals advanced in these models. This integration may be possible: in Vermaas and Dorst (2007) the different proposals by Gero are merged into one; in Garbacz (2006) and Vermaas (2007) it is argued that the concept of behavior also has a place in Stone and Wood's functional basis model, but it would eventually result in a reconstruction and/or rejection of existing proposals. In this paper our aim is to formalize the meanings that the terms "behavior" and "function" have in engineering practice, accepting, as said, the meanings provided by domain experts without questioning their adequacy or trying to reduce or modify them. The task of resolving the conceptual differences then becomes one of making meanings precise and finding the relationships between them, in this way extending and completing the seminal work of Chandrasekaran and collaborators (e.g., Chandrasekaran et al., 1999; Chandrasekaran & Josephson, 2000; Chandrasekaran, 2005), and Yoshinobu Kitamura and Riichiro Mizoguchi (e.g., Kitamura et al., 2002, 2005/2006).

More specifically, we follow the analyses of Chandrasekaran, first distinguishing between two general approaches toward defining functions of technical artifacts, called the functional representation approach and the functional modeling approach (Chandrasekaran, 2005), and then formalize the different meanings of behavior and function that Chandrasekaran and Josephson (2000) identified within the functional representation approach. In the functional representation approach definitions of artifact functions typically refer to the concept of artifact behavior; the FBS and SBF models, described above, are all instances of the functional representation approach. This is not the case in functional modeling, where the functional basis model of Stone and Wood is a prime instance. Our focus on the functional representation approach implies that we do not formalize all engineering meanings of, in particular, the concept of function. We, however, occasionally also consider the functional modeling approach.

2.1. Behavior and function in the functional representation approach

In their analysis of the concepts of behavior and function as used by engineers, Chandrasekaran and Josephson (2000) isolated five meanings of behavior and two of function. The meanings of behavior are characterized using the primitive notion of *state variable*:

1. Behavior as the value of some state variable of the artifact or a relation between such values at a particular instant.
   Example: the car rattled when the driver hit the curve.
2. Behavior as the value of a property of the artifact or a relation between such values.
   Example: a lintel distributes load to two sides.
3. Behavior as the value of some state variable of the artifact over an interval of time.
   Example: the BHP[3] increased for awhile, but then started to decrease.
4. Behavior as the value of some output state variable of the artifact at a particular instant or over an interval.
   Example: the amplifier is behaving well, the output voltage is constant.
5. Behavior as the values of all the described state variables of the artifact at a particular instant or over an interval. (No example given.)

Note that, for all five meanings, a behavior of a technical artifact is partially objective and partially subjective. Behavior has an objective aspect because it eventually depends on the properties or features of the artifact. Yet, the very same behavior also has a subjective aspect: it depends on the designer(s) and, for the choice of the state variables, indirectly on engineering practice. The underlying intuition of Chandrasekaran and Josephson for this subjective twist is that a state variable of an artifact represents some feature or aspect of this artifact that might be relevant only from a specific point of view (Chandrasekaran, 1994). It is important to emphasize that the behavior of a technical artifact is different from the value of its state variable(s). Behavior is somehow characterized by this value(s) in a sense to be explicated. Thus, if the value of the input voltage of an amplifier is 10 mV at a particular instant, then this value (10 mV) is not identical with any behavior of this amplifier. Rather, part of the behavior of the amplifier consists of the situation that its input voltage is 10 mV.

The two meanings of function that Chandrasekaran and Josephson distinguish are called the *device-centric* and *environment-centric* meanings. Without going into detail (see Section 5), a device-centric function of an artifact is a behavior of the artifact that is selected and intended by some agent. It is a function that is described in terms of the properties and behaviors of the artifact only; an example of a device-centric function is "making sound" in the case of an electrical buzzer. An environment-centric function is, in turn, an effect or an impact of this behavior of the artifact on its environment, provided this effect or impact is selected and intended by some agent. This kind of function is conceptually separated from the artifact that performs or is expected to perform this function; "enabling a visitor to a house to inform the person inside the house that someone is at the door" is an environment-centric function of the buzzer.

2.2. Function in the functional modeling approach

The functional representation approach, as defined by the work of Chandrasekaran and collaborators, covers an important part of engineering modeling of functions of technical artifacts. The rival approach of functional modeling covers another part, and includes the functional basis model by Stone and Wood (2000), which has received a substantial amount of

[3] BHP stands for brake horse power, and it is described as the amount of real horsepower going to the pump.

attention in recent years. In this paper we will not attempt to formalize the informal meanings of function as they occur in this second approach; however, to allow us to sketch how our results apply to these remaining meanings, we now briefly discuss how Stone and Wood define functions.

Stone and Wood model overall product functions of technical artifacts from especially the electromechanical and mechanical domain as sets of connected elementary subfunctions. In line with the design 10 methodology of Pahl and Beitz (1988) an overall product function of an artifact is defined as a general input/output relationship of the artifact having the purpose of performing an overall task, and is represented by a black-boxed operation on flows of materials, energies, and signals. A subfunction performs a part of that overall task and is represented by a well-defined basic operation on well-defined basic flows of materials, energies, and signals, which are laid down in libraries that list all possible basic operations and basic flows. These libraries are called a *functional basis* and can be found in Hirtz et al. (2002).

3. DOLCE

We choose to use the DOLCE ontology to formalize the engineering notions of behavior and function. This ontology is part of the *WonderWeb* project.[4] The vision of this project is to have a library of foundational ontologies reflecting different ontological choices, and the WonderWeb designers' intentions are to make the different alternatives underlying ontological choices as explicit as possible. For this reason they pay extensive attention to carefully isolating the ontological options and their formal relationships: DOLCE's ontology is the first module of this library to be developed.

DOLCE is a foundational ontology of particulars with a clear cognitive bias. Its aim is to capture the ontological categories underlying natural language and human commonsense, and the categories introduced in DOLCE are therefore thought of by its developers as "cognitive artifacts ultimately depending on human perception, cultural imprints and social conventions" (Masolo et al., 2003, p. 13). The categories in DOLCE are based on an analysis of the surface structure of language and cognition. A first consequence of this approach is that DOLCE's categories are placed at the *mesoscopic level* of middle-sized objects that we, as humans, perceive. Middle-sized objects is an expression equivalent to "commonsensical objects," and standard artifacts are examples thereof. A second consequence is that there is not a claim of any special robustness against the state of the art in scientific knowledge: DOLCE's categories are just descriptive notions that assist in making already formed conceptualizations explicit.

3.1. A general introduction to DOLCE

DOLCE's taxonomic structure is pictured in Figure 1. Each node in the graph is a category of the ontology. A category that is a direct subcategory of another is depicted by drawing the latter higher in the graph and linking them with an edge. PARTICULAR is the top category. The set of direct subcategories of a given category forms a partition except where dots are inserted.

As said above, we want to extend DOLCE to capture crucial notions in the area of engineering design to allow us to use this ontological framework to analyze, extend, and possibly improve the work in this area. Here we provide a minimal introduction to the whole ontology, the interested reader can read (Masolo et al., 2003), where the underlying motivations and a throughout discussion of technical aspects of DOLCE are given.

From the graph in Figure 1, it is clear that the DOLCE ontology concentrates on particulars as opposed to universals. Roughly speaking, a *universal* is an entity that is instantiated or concreted by other entities, for example, "human" and "being taller than." A *particular* is an entity that is not instantiated by other entities, as, for example, the Eiffel Tower in Paris. Particulars comprise physical or abstract objects, events, and qualities. It seems to us that the DOLCE ontology provides a good framework for the needs of engineering design: it adopts the distinction between objects (e.g., products) and events (e.g., operations); and it includes a differentiation among individual qualities (such as the weight of a specific material item), quality types (e.g., weight and color), quality spaces (e.g., spaces to classify weights and colors), and quality positions or qualia (e.g., informally, locations in quality spaces). These together with measure spaces, where the quality positions are associated with a measure system and thus to numbers, are important for describing and comparing devices and will be described below. Indeed, an important element among the motivations to use DOLCE was its robustness and flexibility that allows us to capture, in a natural way, the views proper of engineering practice.

The DOLCE ontology category (class) ENDURANT comprises objects, for example, a *hammer*, and amounts of matter, for example, *an amount of water, an amount of gold*,[5] whereas the category PERDURANT comprises events like *making a hole* or *a soccer game*, that is, things that happen in time. The term "object" is used in the ontology to capture a notion of unity as suggested by the partition of the class PHYSICAL ENDURANT into classes AMOUNT OF MATTER, FEATURE, and PHYSICAL OBJECTS (see Fig. 1).

[4] http://wonderweb.semanticweb.org

[5] Amounts of matter are endurants in DOLCE because they have a location in space and time and do not depend on other entities as qualities do. The reason they are distinct from objects, for example, an amount of gold is distinct from the wedding ring it forms, is that these two types of entities have different identity criteria: the wedding ring has necessarily a round shape, whereas the amount of gold has it accidentally (the wedding ring is destroyed by a change of shape, the amount of gold is not), the wedding ring indicates a marital status, whereas the amount of gold does not, and so forth.

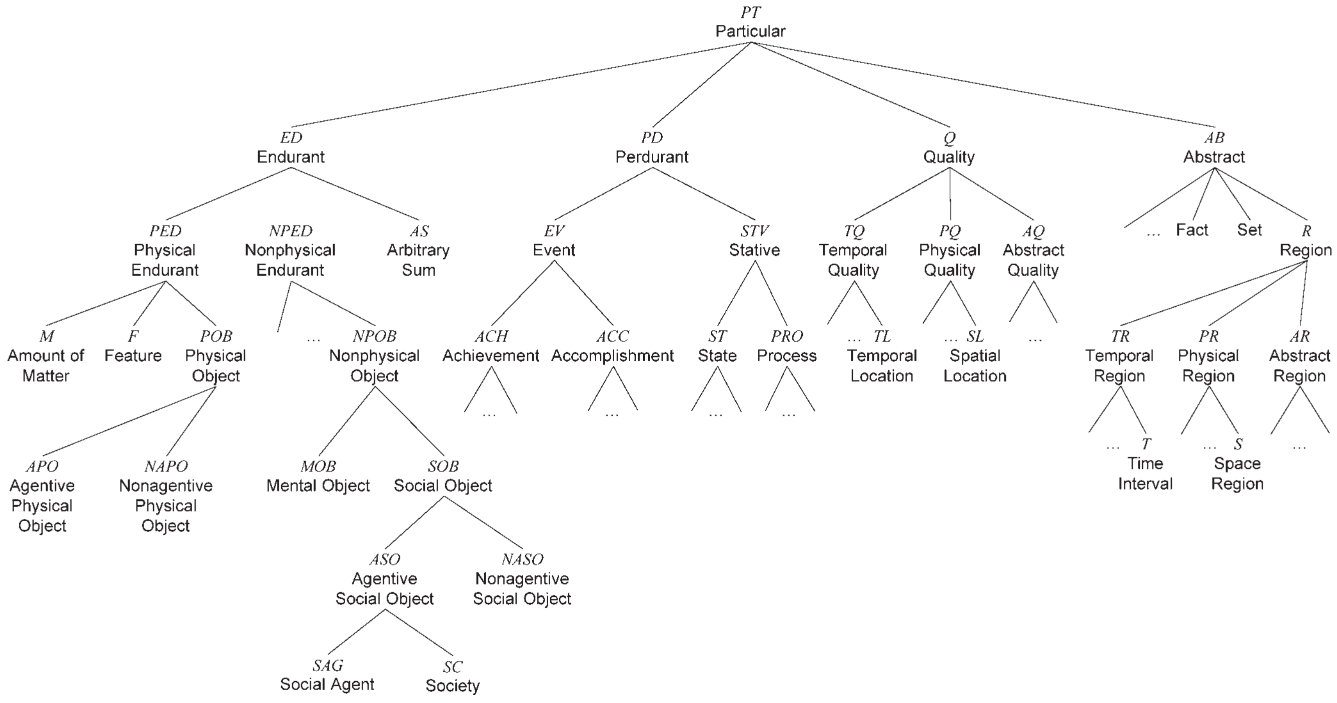

Fig. 1. The DOLCE basic categories according to Masolo et al. (2003).

Both endurants and perdurants are associated with a collection of qualities. The exact list of qualities may depend on the entity: *shape* and *weight* are usually taken as qualities of endurants, *duration* and *direction* as qualities of perdurants. An individual quality, for example, the weight of your newest car is a quality associated with *one and only one* entity; it can be understood as the particular way in which the entity instantiates the corresponding property, for example, the general property of "having weight." For example, the endurant hammer_#321 (a token) has its own individual instantiation of property "having weight."[6] This instantiation is the individual weight quality of hammer_#321. The change of an endurant in time is explained through changes in some of its individual qualities. For example, if a component of hammer_#321 is substituted for another, the hammer may change its weight. This means that the individual weight quality of this entity was first associated with, or classified in, a position *a* and later to (in) a position *b* of a given weight quality space. Note that *a* and *b* should not be considered weight measures, for example, not an amount of weight such as 5 kg. Instead, they are elements of, positions in, a quality space whose primary role is to partition individual qualities in equivalent, or similar, depending on the space, entities before committing to numeric values and units of measure. Thus, the same *a* may be associated with 5 kg in one measure space and with 11.1 lb. in another. Finally, note that the hammer cannot exist without its individual qualities: DOLCE forces a strict existential dependence between individual qualities and their hosts.

The example of the hammer makes clear that the position associated with an individual quality can change over time. DOLCE calls such positions qualia, quale in singular form. A quality space is the collection of all possible qualia, that is, distinct positions, that an individual quality can assume. The notion of quality space is quite flexible: for example, if in an application it is important to classify endurants as heavier, equal, or lighter than a given endurant *a*, then the adopted weight quality space could have just three distinguished positions: the position corresponding to individual weight qualities of endurants (whose weight quality is) lighter than *a*, the position of those not distinguishable from that of *a*, and the position of the individual weight qualities heavier than *a*.

3.2. DOLCE categories and relations

In this section we present the categories of Figure 1 that are relevant to our work. Note that the terminology adopted departs sometimes from usage in engineering design, knowledge representation, or conceptual modeling areas, because it has been affected, in part, by discussions presented in the philosophical literature.

1. ED(x) stands for "x is an endurant."[7] An endurant is an entity that is wholly present at any time it is present. It is

[6] The codes we use to refer to artifacts are nonsense codes. They are used, however, to emphasize that we refer thereby to particular instances of artifacts and not to the types thereof.

[7] In DOLCE one uses abbreviated names, like ED, for both the class, that is, ENDURANT, and the predicate "being an endurant." Because it does not cause confusion, we also follow this practice.

physical if located in space and time: a *hammer_#321*, a *mover machine_#111*, an *amount of plastic*, and *the cavity in which a piston moves*. Endurants may lack a spatial or temporal location. *Intentions*, *beliefs*, *desires*, *laws*, *plans*, and so forth, are examples of nonphysical endurants.

- PED(x) stands for "x is a physical endurant." PED is a subcategory of ED. A *hammer*, a *mover machine*, an *amount of plastic*, and *the cavity in which a piston moves*, are all examples of physical endurants.[8]
- APO(x) stands for "x is an agentive physical object," that is, endurants that have spatial and temporal location and such that we are justified in ascribing intentions, beliefs, or desires to them. *People* and *organizations* fall into this category while *hammers* and *engines* do not.
- NAPO(x) stands for "x is a nonagentive physical object," that is, endurants that have spatial and temporal location but do not entertain intentions, beliefs, or desires. *Hammers* and *engines* fall into this category while *people* and *organizations* do not.

2. PD(x) stands for "x is a perdurant," that is, an entity that is only partially present at any time that it is present. For instance, consider the perdurant *producing an item of type #234* that consists of attaching two metal pieces together with screws and painting the resulting piece. Although the *painting* goes on, the (temporal) part corresponding to *attaching the two pieces* is no longer present and when this is present, the *painting* still has to come. Perdurants can also have spatial parts. The category of perdurants is divided in four subcategories: achievements and accomplishments, the eventive perdurants, and states and processes, the stative perdurants.[9]

   - ACHIEVEMENT (ACH) is the class of perdurants characterized by anticumulativeness, that is, the sum of two achievements of, say, kind A is not an achievement of kind A, and atomicity, that is, they do not have temporal subparts. The guiding idea is that an achievement is an instantaneous change: *the moment in which a window is closed* is an event classified as achievement, but the whole event of *closing the window* is not.
   - ACCOMPLISHMENT (ACC) is the class of anticumulative perdurants that have temporal parts, that is, nonatomic perdurants. For example, the perdurant *closing the window* is composed of subevents like *pushing the window* and *turning the handle*. Accomplishments are anticumulative: consider two perdurants regarding the selling of a house, say, two different houses. Then, the sum of these two perdurants is not a perdurant regarding the selling of a house.
   - STATE (ST) is the class of cumulative perdurants; thus, it is closed under mereological sum in the sense that the sum of two perdurants, for example, two *drilling* events, is a perdurant of the same kind (a *drilling*). These perdurants are homeomeric. Events like *drilling*, *moving*, and *sitting* are perdurants in this class.
   - PROCESS (PRO) is the class of cumulative perdurants that also are nonatomic and nonhomeomeric. We have seen that *sitting* is classified as a state; instead *running* is classified as a process: there are short temporal subparts of a *running* that are not "runnings."

In the engineering domain, one needs to deal with a variety of processes and operations. In DOLCE these are events, that is, specific perdurants (tokens) and not types. The distinction among perdurants is driven by such properties as temporal subdivision. Consider an operation like a *machine configuration*. It requires the completion of a process that is obtained once a specific state is reached. If this does not happen, the configuration does not occur. Thus, if a configuration operation is divided in two temporal parts, only one of the two suboperations is a configuration operation (if any). We refer the reader to Masolo et al. (2003) and (Borgo & Leitao, 2007) for more information on these aspects.

A different type of entities finds a place in the QUALITY category.

- Q(x) stands for "x is an individual quality." Recall that in DOLCE qualities are particulars and should not be confused with properties, which are universals. Note that informally, one can think of an individual quality as an *instantiated property*. Qualities can be seen as instantiations of basic aspects of endurants or perdurants. Usually they can be perceived or measured, for example, weight, shape, or energy. The term "individual" is used to mark the essential relationship between the entity and its qualities.

We have seen that qualities are associated with quality spaces where comparisons like "truck A and truck B have a different weight" can be carried out. In the study of engineering behavior some qualities, like duration of perdurants, are central. For this reason, we will make use of the DOLCE relation $ql_{\mathrm{T,PD}}$. Expression $ql_{\mathrm{T,PD}}(t, x)$ means that t is the temporal location of a perdurant x. For instance, if a process of cooling a combustion chamber lasts 60 s, then the temporal location of this process corresponds to a temporal interval which, when evaluated in the standard measure system for time, corresponds to 60 s.

Another important relation for our analysis is the *parthood* relation: x is part of y, written P(x, y). The formal theory based

[8] Because it has a spatiotemporal location, a hole is physical, and, of course, nonmaterial.

[9] We present these types of perdurants only for explicatory purposes as we do not use them in what follows.

on parthood is called *mereology* (Simons, 1987). It consists of the *proper part* and *overlap* relations, both defined in terms of P. In DOLCE the *parthood* relation applies to pairs of endurants, for example, to state that an object is part of another, and to pairs of perdurants, for example, to state that an event is part of another. For instance, if $a =$ "writing the paper ABC" and $b =$ "writing the introduction to the paper ABC," then $\mathrm{P}(b, a)$ holds. For pairs of endurants, the relation of parthood is temporalized because an endurant may loose and gain parts throughout its existence. The relation $\mathrm{P}(e, e', t)$ represents the fact that an endurant e is part of an endurant e' at time interval, possibly atomic, that is, possibly at an instant t. For the sake of simplicity, in what follows we will assume that the time interval for all mereological relations among endurants that we consider is fixed. Consequently, the temporal relativization of mereological parthood between endurants will be neglected.

We will now give a number of auxiliary mereological definitions together with some examples. Again, recall that we neglect the temporal parameter. Note that throughout the paper the symbol $\triangleq$ is reserved to indicate definitions.

$$\mathrm{PP}(x, y) \triangleq \mathrm{P}(x, y) \wedge \neg\mathrm{P}(y, x). \tag{1}$$

A perdurant is a *proper part* (formally PP) of another if the first is part of the second and not vice versa.

Example: reading this section is a proper part of reading the whole paper.

$$\mathrm{O}(x, y) \triangleq \exists z(\mathrm{P}(z, x) \wedge \mathrm{P}(z, y)). \tag{2}$$

Two perdurants *overlap* (formally O) if an entity exists which is simultaneously part of both.

Example: "My fastening a screw in my toaster" and "the fixing of my toaster," these perdurants overlap because "the changing of the screw position" is part of both.

Sum (+) and *fusion* (σ) are two special operators. The sum of x and y is a perdurant z such that each part of x and each part of y are parts of z, and if a perdurant w overlaps z, then w also overlaps x (or y). Finally, the fusion operation is a direct extension of the binary sum to all the perdurants that exhibit a property ϕ (Simons, 1987).

$$x + y \triangleq \iota z \quad \forall w(\mathrm{O}(w, z) \leftrightarrow (\mathrm{O}(w, x) \vee \mathrm{O}(w, y))) \tag{3}$$

Example: closing a window is the sum of moving the window and turning the handle.

$$\sigma x\phi(x) \triangleq \iota z \quad \forall y(\mathrm{O}(y, z) \leftrightarrow \exists w(\phi(w) \wedge \mathrm{O}(y, w))) \tag{4}$$

Example: my staying in Paris is the fusion of all the days I spend in Paris.

We draw the reader's attention to the fact that definition 3 is functional, indicated formally by "ιz," that is, the assumption is made that exactly one z exists such that the formula "$\forall w(\mathrm{O}(w, z) \leftrightarrow (\mathrm{O}(w, x) \vee \mathrm{O}(w, y)))$" is satisfied. An analogously remark can be made for definition 4.

We use the following theorems (Simons, 1987) of mereology extensively without further remarks:

$$\mathrm{P}(x, x + y) \tag{5}$$

$$\phi(x) \rightarrow \mathrm{P}(x, \sigma y\phi(y)) \tag{6}$$

$$\mathrm{P}(y_1, x) \wedge \mathrm{P}(y_2, x) \rightarrow \mathrm{P}(y_1 + y_2, x) \tag{7}$$

The main relation involving both endurants and perdurants is called *participation*, formally PC. This relation captures the simple fact that an endurant "lives" in time by participating in some perdurant. For example, a machine (endurant) may participate in a production process (perdurant). A car's "life" is also a perdurant in which that car participates throughout the duration of its life, that is, the time span between the construction of the car untill its destruction. If endurant a participates in perdurant e at each instant of period t, we write $\mathrm{PC}(a, e, t)$ which reads "a participates in e during all of t," note: here t may be just a part of the duration of e and participation and parthood are distinct relations. An endurant is never part of a perdurant, only perdurants can be parts of perdurants, analogously, only endurants can be parts of endurants. Participation is also time indexed in order to account for the varieties of participation in time like temporary participation, constant participation and so on (see Masolo et al., 2003).

4. DEFINING ARTIFACT BEHAVIORS IN DOLCE

Our approach to formalizing artifact behavior is to model it as a quality of artifacts that relates the artifact, that is, an endurant, and the perdurants to which the artifact participates. Informally, we take behavior to mean *the specific way in which an artifact participates in some perdurant*. The relationship between the artifact and the perdurant is twofold: it captures the inherent contribution that the artifact brings to the perdurant at hand, which explains, at least in part, why the latter happens in the way it does, and it also reveals the effects that the perdurant brings to the artifact. However, a formal description of this relationship is extremely hard to provide in its generality. In our alternative approach we view behavior as a primitive quality and study it within an ontological framework. In this setting, the properties of the artifact and the perdurant can be exploited to characterize and constrain this primitive. It follows that to talk of the behavior of an artifact, we need to refer to a perdurant, perhaps for as long as the life of the artifact itself. We point out from the start that our formalization of behavior is not limited to actual perdurants; it is customary, especially in the engineering domain, to discuss of the behavior of an object in nonactual situations, that is, situations that are merely possible and, as we will see, sometimes even physically impossible.[10] To account for these cases, our notion of perdurant will be quite general.

Perdurants can be actual, that is, events in the real world, or just possible, that is, events in some imagined world. The latter category is very broad as it contains all perdurants that are logically possible, including those that are physically impossible, for example, transmitting information at triple the speed of light. This broad category of perdurants is used to make room for all perdurants that could be considered by (rational) engineers. We assume that a (rational) engineer may be fallible as far as the laws of physics are concerned, but he or she is not fallible with respect to such basic laws of thought as logical principles. Thus, the perdurant that is the mereological fusion of rotating and nonrotating is assumed to be outside the scope of our considerations as it is meaningless from the logical perspective. We will make a similar distinction in the class of behaviors.

Technically, all notions we define are provided in first-order logic (Lemmon, 1965). That is, although in informal explanations we may speak of sets and classes, and their elements, we made a significant effort to avoid second-order terms in the formalization. The choice to adopt first-order logic is motivated by two considerations: one, DOLCE, is a first-order theory so we have no reason to constrain ourselves to weaker logical systems; two, the metalogical properties of richer logics, like second-order logics, make their implementation in information systems highly problematic.

Finally, we add a proviso: in DOLCE the most specific category for technical artifacts is that of nonagentive physical objects (NAPO) which is, admittedly, too general because it includes nonartificial entities like human bodies and planets. Unfortunately, a category comprising technical artifacts only has not been characterized in DOLCE. Fortunately, our theory does not depend on the specific characterization of technical artifacts. All we need to assume is that there is a way to distinguish a technical artifact from other entities. Clearly, this is a reasonable assumption in engineering applications. Then we can develop the theory by assuming that a specific category of technical artifacts, here called TechArt, has been given. In principle, the category TechArt may be part of a future extension of the DOLCE ontology (e.g., along the lines of Borgo & Vieu, in press) or can be defined by practitioners according to their specific interests.

4.1. A formal definition of artifact behavior

Let a be a technical artifact, that is, $\mathtt{TechArt}(a)$.[11] Let e be a perdurant, that is, PD(e). As anticipated, we take the behavior b of a in a perdurant e to be *the specific way in which a occurs in e.* In this view, behavior depends on the chosen a and e and is seen as a qualification of the participation relation. For instance, if a is a capacitor, then the way in which a occurs or exists in a given process of storing electric energy is a behavior of precisely *this capacitor a*.[12] Our definition of behavior links behaviors with two categories of entities: endurants and perdurants. Consequently, we are able to take into account inherent conceptual connections between behaviors and the entities to which we ascribe behaviors. We are also in a position to say that two different endurants behave differently in the same perdurant. For instance, if two capacitors in an electrical circuit participate in a process, we can say that they exhibit different behaviors, despite referring to the very same process. Formally, we take behavior b to be an element of a new quality category $\mathtt{B}$, that is, a new subcategory of Q: it does not hold for a single endurant or perdurant, but for pairs of endurants and perdurants.[13] In this way, it captures the special relationship between the artifact and the event in which it "behaves."

To formalize this relationship, we introduce a ternary relation

$$\mathtt{Beh}(a, e, b)$$

which reads "b is the behavior of the technical artifact a in event e" and is taken as primitive in our theory. It will be helpful later on to use the following binary and unary definitional reductions:

$$\mathtt{Beh}(a, b) \triangleq \exists e\, \mathtt{Beh}(a, e, b) \tag{8}$$

$$\mathtt{Beh}(b) \triangleq \exists e, a\, \mathtt{Beh}(a, e, b) \tag{9}$$

The expression "$\mathtt{Beh}(a, b)$" means that b is a behavior of a $\mathtt{TechArt}$ entity a, whereas the expression "$\mathtt{Beh}(b)$" means that b is a behavior of some unspecified entity in TechArt. Similar definitions may be constructed for all other types of behavior we provide below. In particular, $\mathtt{Beh}(b)$ if and only if $\mathtt{B}(b)$.

We talk of the behavior of a in a perdurant e if and only if a participates in e for *all the duration of e*, and in the logic we impose that, for a pair endurant–perdurant satisfying this condition, the corresponding behavior b exists. A behavior b is uniquely identified by the pair endurant–perdurant, although a perdurant e may have several participants and an entity a may participate in several perdurants.

[10] The notion of impossible perdurant may look strange. We take the perspective of the engineering science and consider "impossible" any perdurant that does not comply to natural, and in particular, physical, laws. As it happens, a single or a group of engineers can erroneously consider that some perdurant may happen because the available knowledge of data and of natural laws does not prevent it. Indeed, knowledge may be imprecise, erroneous, or even lacking in some aspect so that someone cannot rule out in his or her reasoning, an event that is, to all effect, impossible. Of course, these "impossible perdurants," although not part of reality, are part of the spectrum of engineering activity, and we must include them in our formalization. Below we will qualify these as *physically impossible* perdurants to distinguish them from *logically impossible* perdurants.

[11] The formal predicates and relations introduced by the theory are written ABC or abc. The predicates and relations already in DOLCE are written ABC or abc.

[12] Indeed, we begin by looking at capacitor instances (tokens) and do not address the behavior of a *type* of capacitors.

[13] The ontological classification of "behavior" is still an open problem in the literature. As far as we know, the approach we follow in this paper is new.

Given a perdurant e, let us write tm for the function that takes as argument a perdurant and returns the period of time that it spans, that is, the temporal location of the perdurant (see Section 3.2). In other words, $tm(e)$ is the whole of the period during which perdurant e occurs.

Using ι, the iota operator, we can formally define tm as follows

$$tm(e) \triangleq \iota t ql|_{\mathrm{T,PD}}(t, e) \tag{10}$$

Using these notions, we capture our initial observations via the following axioms

$$\mathtt{Beh}(a, e, b) \rightarrow \mathrm{PC}(a, e, tm(e)) \tag{11}$$

$$\mathtt{Beh}(a, e, b) \wedge \mathtt{Beh}(a, e, b') \rightarrow b = b' \tag{12}$$

$$\mathtt{Beh}(a, e, b) \wedge \mathtt{Beh}(a', e', b) \rightarrow a = a' \wedge e = e' \tag{13}$$

$$\mathrm{PC}(a, e, tm(e)) \rightarrow \exists b\ \mathtt{Beh}(a, e, b) \tag{14}$$

These axioms have the following informal readings, respectively: if b is the behavior of a in e, then a participates in e during all the time spanned by e; if b and b' are the behavior of a in e, then b and b' must be the same behavior; if b is the behavior of a in e and of a' in e', then a, a' (e, e') must be the same artifact (perdurant, respectively); and, if a participates in e during all the time e spans, then b exists, which is the behavior of a in e.

4.2. Formal definitions of actual, possible, desired, and input/output behaviors

Looking at engineers' activities, we need to distinguish different kinds of behavior. Although an "actual behavior" of an artifact is what it actually does during (a part of) its life, the more general notion of "possible behavior" deals with what an artifact can possibly do. A pen may be destroyed before it happens to write; still, the pen could have participated to a writing event, that is, writing is part of its behavior although not of its "actual" behavior. Furthermore, although a pen may possibly not write because of a design flaw, still engineers, not aware of the flaw, talk about its writing behavior. Now we will look at how to make room for these cases in the formalism.

As we have seen, perdurants can be divided in actual, physically possible, and physically impossible perdurants.

The last class includes all the perdurants believed possible by rational agents, in particular, engineers, but in which the artifacts at stake, designed or constructed, cannot do what is assumed of them. The above three subclasses of the PD category are not explicitly defined in DOLCE because the ontology is not restricted to the contingent laws, namely, those of physics, which are at the core of these distinctions.

For this reason, we introduce the class GEPD of *generalized engineering perdurants*, that is, the large class of perdurants in PD that can be relevant to engineers regardless of their physical possibility.[14] This subcategory of PD collects all actual, physically possible, and physically impossible perdurants. We exclude from GEPD only those perdurant that cannot be believed by rational agents, like the logically impossible perdurants, which do not comply with the laws of logic. We write APD and EPD for the subclasses of actual and physically possible perdurants respectively. Then,

$$\mathtt{APD} \subseteq \mathtt{EPD} \subseteq \mathtt{GEPD} \subseteq \mathrm{PD} \tag{15}$$

We add three constraints (16)–(18) revealing the mereological structure of perdurants with respect to their modal status:

$$\mathtt{APD}(e) \wedge \mathrm{P}(e', e) \rightarrow \mathtt{APD}(e') \tag{16}$$

This constraint says that any part of an actual perdurant is actual.

$$\mathtt{APD}(e) \wedge \mathtt{APD}(e') \rightarrow \mathtt{APD}(e + e') \tag{17}$$

This constraint say that the sum of two actual perdurants is actual. Notice that although the sum of any two actual perdurants is always actual, the sum of two possible perdurants need not be possible; for example, a process of neutralizing an acid, at a given spatiotemporal location, and a process of neutralizing an alkali, at the same spatiotemporal location. Moreover, we do not assume either that the sum of two generalized perdurants is a generalized perdurant. The reason is that although the domain of generalized perdurants contains entities that defy physical laws, it does not contain entities that defy logical principles. Thus, if we consider the perdurant of writing of a pen, at a certain spatiotemporal location, and a perdurant of nonwriting of the same pen, at the same spatiotemporal location, then their sum, that is, the perdurant of writing and nonwriting the pen, is logically impossible and therefore is not a generalized perdurant.

$$\mathtt{EPD}(e) \wedge \mathrm{P}(e', e) \rightarrow \mathtt{EPD}(e') \tag{18}$$

This axiom says that any part of a possible perdurant is possible. We can now constrain the domain of Beh.

$$\mathtt{Beh}(a, e, b) \rightarrow \mathtt{TechArt}(a) \wedge \mathtt{GEPD}(e) \wedge \mathrm{B}(b) \tag{19}$$

Some combinations of perdurants are meaningless in engineering practice. It is possible that an air-conditioning system cools the room, at a given time, and it is also possible that the

[14] Of course, we do not claim that each of these perdurants is considered by engineers, yet the class itself is important in engineering and in particular in engineering design. An engineer may not know if perdurant "this pen can write and also unwrite" is possible or not, yet she can conceive the perdurant itself (it is not logically inconsistent) and consider it in her design activity: she would not prevent herself to think about it just because *now* she does not know how to build such a device. Similarly, there might not be enough material in the universe to build a designed device, yet engineers can describe in detail how the device would work if it were to exist.

same system heats the room, at the same time. However, these two perdurants cannot "belong to the content" of the real world. To account for this condition, we say that two or more perdurants are *coherent* when their sum is a (physically) possible perdurant.

$$\mathtt{Coh}(e, e') \triangleq \mathtt{EPD}(e + e') \tag{20}$$

Because of (5), axiom 18 implies that only possible perdurants are coherent

$$\mathtt{Coh}(e, e') \rightarrow \mathtt{EPD}(e) \wedge \mathtt{EPD}(e') \tag{21}$$

Note that the relation of coherence is reflexive and symmetric in the class of possible perdurants, but in general, it need not be transitive. Also, all actual perdurants turn out to be coherent:

$$\mathtt{APD}(e) \wedge \mathtt{APD}(e') \rightarrow \mathtt{Coh}(e, e')$$

It follows from (7) and (18) that all parts of any possible perdurant are coherent:

$$\mathtt{EPD}(x) \wedge \mathrm{P}(y_1, x) \wedge \mathrm{P}(y_2, x) \rightarrow \mathtt{Coh}(y_1, y_2) \tag{22}$$

We say that two perdurants are *cotemporal* if they occur at the same time period, for example, the perdurants "Paulo's attending Robert's talk today" and "Robert's giving a talk today."

For any a in NAPO, let C_a be the class of all generalized perdurants e such that PC(a, e, $tm(e)$). From DOLCE, it follows that C_a is nonempty. Fix a perdurant e in C_a for some fixed a. We say that e is *minimal* in C_a if for each perdurant e' in C_a cotemporal with e, not PP(e', e). For instance, a "soccer game" and the "playing of agent a in the soccer game" are two perdurants in the same class C_a but only the latter is minimal in the class.

To define this notion formally, we first introduce the relation of "wholly participation"

$$\mathrm{PC}_{\mathtt{WH}}(a, e) \triangleq \mathtt{GEPD}(e) \wedge \mathrm{PC}(a, e, tm(e)) \tag{23}$$

Given a NAPO a, the predicate $\mathtt{PC}_{\mathtt{WH}}$ characterizes the class of all (generalized) perdurants in which a participates throughout all of its life. When $\mathtt{PC}_{\mathtt{WH}}$ (a, e), we will say that *a wholly participates in e*. Let a wholly participate in some perdurant e, then we define a new relation

$$\begin{aligned}\mathtt{MIN}(a, e) \triangleq \mathrm{PC}_{\mathtt{WH}}(a, e) \wedge \forall e'(\mathrm{PC}_{\mathtt{WH}}(a, e') \wedge tm(e) \\ = tm(e') \rightarrow \neg\mathrm{PP}(e', e))\end{aligned} \tag{24}$$

Informally, $\mathtt{MIN}(a, e)$ means that e is a minimal perdurant in C_a with respect to parthood. Note: e does not need to cover the whole life of a.

We use this notion and the previous classification of perdurants to specialize the DOLCE relation lf(x, a), which reads "perdurant x is the life of endurant a":

1. $\mathtt{Alf}(e, a)$ stands for "perdurant e is the actual life of endurant a" and is formally defined as the fusion of the actual perdurants that are minimal in C_a;
2. $\mathtt{Plf}(e, a)$ stands for "perdurant e is a possible life of endurant a" and is formally defined as the fusion of a maximal class, with respect to inclusion, of coherent engineering possible perdurants which are minimal in C_a;
3. $\mathtt{Glf}(e, a)$ stands for "perdurant e is a generalized life of endurant a" and is formally defined as the fusion of a maximal class, with respect to inclusion, of logically consistent engineering generalized perdurants that are minimal in C_a.

$\mathtt{Alf}$ is easily stated once we ensure the endurant participates in an actual perdurant.

$$\begin{aligned}\mathtt{Alf}(e, a) \triangleq \exists e'(\mathtt{APD}(e') \wedge \mathrm{PC}_{\mathtt{WH}}(a, e')) \wedge e \\ = \sigma y(\mathtt{APD}(y) \wedge \mathtt{MIN}(a, y))\end{aligned} \tag{25}$$

it follows that a NAPO, and thus a technical artifact, has at most one actual life.

$$\mathtt{Alf}(e_1, a) \wedge \mathtt{Alf}(e_2, a) \rightarrow e_1 = e_2 \tag{26}$$

Assuming that a perdurant e exists such that $\mathtt{APD}(e)$ and PC(a, e), we write $\mathtt{Alf}(a)$ to denote "the actual life of a."

To define "a possible life" of an endurant a we must be able to identify a maximal class of perdurants that are globally coherent. To this goal, we first define a *coherent fusion* of engineering possible perdurant for a

$$\begin{aligned}\sum_{\mathtt{Coh}}(a, e, s) \triangleq{}& \mathtt{EPD}(e) \wedge \mathtt{EPD}(s) \wedge \mathrm{PC}_{\mathtt{WH}}(a, e) \\ & \wedge \forall e'((\mathrm{PC}_{\mathtt{WH}}(a, e') \wedge \neg\mathrm{P}(tm(e'), tm(e))) \\ & \rightarrow \neg\mathtt{EPD}(e + e')) \wedge \mathrm{P}(s, \sigma y(\mathtt{EPD}(y) \wedge \mathtt{MIN}(a, y) \\ & \wedge \mathtt{Coh}(e, y))) \wedge \forall z((\mathtt{EPD}(z + s) \wedge \mathtt{MIN}(a, z) \\ & \wedge \mathtt{Coh}(e, z)) \rightarrow \mathrm{P}(z, s))\end{aligned} \tag{27}$$

This definition says that endurant a wholly participates in the engineering possible perdurant e but not to any longer perdurant, that is, e spans the possible life we are considering, and that s is a maximally coherent part of the fusion of all minimal perdurants that are compatible to e. In particular, all perdurants that compose s are coherent with one another. Then,

$$\mathtt{Plf}(e, a) \triangleq \exists e' \sum_{\mathtt{Coh}}(a, e', e) \tag{28}$$

It remains to define $\mathtt{Glf}$, that is, "a generalized life" of an endurant a. We proceed by defining $\mathtt{Plf}$. Define *consistent fusion*, $\Sigma_{\mathtt{Cons}}$, to identify fusions of maximal consistent classes of (minimal) generalized perdurants that are pair wise logically compatibles. In this way, we avoid to sum logically inconsistent perdurants like “the drill works at *t*” and “the drill does not work at *t*.” Recall that $\mathtt{PC}_{\mathtt{WH}}(a, c)$ ensures $\mathtt{GEPD}(c)$ and so does $\mathtt{MIN}(a, c)$.

$$\sum_{\mathtt{Cons}} (a, e, s) \triangleq \mathtt{PC}_{\mathtt{WH}}(a, e) \wedge \forall e'(\mathtt{PC}_{\mathtt{WH}}(a, e') \rightarrow \mathrm{P}(tm(e'), tm(e))) \wedge \mathrm{P}(s, \sigma y(\mathtt{MIN}(a, y) \wedge \mathtt{GEPD}(e + y))) \wedge \forall z((\mathtt{GEPD}(z + s) \wedge \mathtt{MIN}(a, z)) \rightarrow \mathrm{P}(z, s)) \qquad (29)$$

Then, the definition of $\mathtt{Glf}$ is

$$\mathtt{Glf}(e, a) \triangleq \exists e' \sum_{\mathtt{Cons}} (a, e', e) \qquad (30)$$

Note that, in general, an entity in NAPO may have more than one possible, or generalized, life. In the case of technical artifacts different possible lives may be determined, among other things, by imagining the different ways in which the users of artifacts may use them.

An *actual behavior* of *a* is the way in which *a* occurs in some actual perdurant. The *maximal actual behavior* of a is the way in which *a* occurs in its actual life.

$$\mathtt{ActBeh}(a, e, b) \triangleq \mathtt{Beh}(a, e, b) \wedge \mathtt{APD}(e) \qquad (31)$$

$$\mathtt{ActBeh}_{\mathtt{max}}(a, b) \triangleq \mathtt{Beh}(a, \mathtt{Alf}(a), b) \qquad (32)$$

Given (11) and $\mathtt{Alf}(a)$, it can be shown that definition (32) is consistent, that is, that there exists a unique *b* for which it is the case that $\mathtt{Beh}(a, \mathtt{Alf}(a), b)$.

A *possible behavior* of *a* is the way in which *a* occurs in some possible perdurant. A *maximal possible behavior* of *a* is the way in which *a* occurs in one of its possible lives.

$$\mathtt{PossBeh}(a, e, b) \triangleq \mathtt{Beh}(a, e, b) \wedge \mathtt{EPD}(e) \qquad (33)$$

$$\mathtt{PossBeh}_{\mathtt{max}}(a, e, b) \triangleq \mathtt{Beh}(a, e, b) \wedge \mathtt{Plf}(e, a) \qquad (34)$$

Note: a maximal possible behavior need not be unique, that is, it is possible that

$$\mathtt{PossBeh}_{\mathtt{max}}(a, e_1, b_1) \wedge \mathtt{PossBeh}_{\mathtt{max}}(a, e_2, b_2) \wedge b_1 \neq b_1$$

An engineering *impossible behavior* of *a* is the way in which *a* occurs in some engineering impossible perdurant.

$$\mathtt{ImBeh}(a, e, b) \triangleq \mathtt{Beh}(a, e, b) \wedge \mathtt{GEPD}(e) \wedge \neg\mathtt{EPD}(e) \qquad (35)$$

A *general behavior* of *a* is the way in which *a* occurs in some generalized engineering perdurant. A *maximal general behavior* of *a* is the way in which *a* occurs in a generalized life.

$$\mathtt{GenBeh}(a, e, b) \triangleq \mathtt{Beh}(a, e, b) \wedge \mathtt{GEPD}(e) \qquad (36)$$

$$\mathtt{GenBeh}_{\mathtt{max}}(a, e, b) \triangleq \mathtt{Beh}(a, e, b) \wedge \mathtt{Glf}(e, a) \qquad (37)$$

All types of perdurants and behaviors discussed so far constitute objective parts of the real world, that is, they do not depend in the existence of mental attitudes for any agent. Still, engineering practice heavily depends on beliefs and desires. To incorporate this dependence into our framework, we assume two primitive notions of agent-related perdurants. Given an agent, or group of agents,[15] $\mathcal{G}$ we write $\mathtt{PD}_{\mathcal{G}}(e)$ to state that $\mathcal{G}$ believes that *e* is a possible perdurant. Similarly, we write $\mathtt{INT}_{\mathcal{G}}(e)$ to state that $\mathcal{G}$ desires perdurant *e* to be actual. Of course, $\mathcal{G}$ is an agentive physical endurant, that is, APO($\mathcal{G}$). The former will be called *$\mathcal{G}$-possible perdurants* and the latter will be called *$\mathcal{G}$-desired perdurants*. The following two axioms establish the ontological categories of $\mathcal{G}$-possible and $\mathcal{G}$-desired perdurants.

$$\mathtt{PD}_{\mathcal{G}}(e) \rightarrow \mathtt{GEPD}(e) \qquad (38)$$

$$\mathtt{INT}_{\mathcal{G}}(e) \rightarrow \mathtt{GEPD}(e) \qquad (39)$$

We use these notions to define the behavior of a technical artifact *for a group $\mathcal{G}$*. Let $\mathcal{G}$ be an agent. A *$\mathcal{G}$-behavior* is a generalized engineering behavior believed by $\mathcal{G}$ to be possible.

$$\mathtt{Beh}_{\mathcal{G}}(a, e, b) \triangleq \mathtt{Beh}(a, e, b) \wedge \mathtt{PD}_{\mathcal{G}}(e) \qquad (40)$$

Roughly speaking, a *maximal $\mathcal{G}$-behavior* of a is a generalized engineering behavior that involves the fusion of a coherent class of $\mathcal{G}$-possible perdurants, which is maximal with respect to inclusion. As before (cf. 27), we first define the fusion of a coherent class of $\mathcal{G}$-possible perdurants that are involved in behaviors (*$\mathcal{G}$-coherent fusion*)

$$\sum_{\mathcal{G}-\mathtt{Coh}} (a, e, s) \triangleq \mathtt{PD}_{\mathcal{G}}(e) \wedge \mathtt{PD}_{\mathcal{G}}(s) \wedge \mathtt{PC}_{\mathtt{WH}}(a, e) \wedge \forall e'(\mathtt{PC}_{\mathtt{WH}}(a, e') \rightarrow \mathrm{P}(tm(e'), tm(e))) \wedge \mathrm{P}(s, \sigma y(\mathtt{PD}_{\mathcal{G}}(y) \wedge \mathtt{MIN}(a, y) \wedge \mathtt{Coh}(e, y))) \wedge \forall z((\mathtt{PD}_{\mathcal{G}}(z + s) \wedge \mathtt{MIN}(a, z) \wedge \mathtt{Coh}(e, z)) \rightarrow \mathrm{P}(z, s)) \qquad (41)$$

As in case (27), because of equation (22), if $\sum_{\mathcal{G}-\mathtt{Coh}} (a, e, s)$, then all perdurants that compose *s* are coherent with one another, that is, *s* is the fusion of a coherent class of perdurants.

$$\mathtt{MaxBeh}_{\mathcal{G}}(a, e, b) \triangleq \mathtt{Beh}_{\mathcal{G}}(a, e, b) \wedge \exists e' \sum_{\mathcal{G}-\mathtt{Coh}} (a, e', e) \qquad (42)$$

Obviously, $\mathtt{MaxBeh}_{\mathcal{G}}(a, e, b)$ means that *b* is a maximal $\mathcal{G}$-behavior of *a*. Notice that a NAPO may exhibit more than one maximal $\mathcal{G}$-behavior with respect to the same agent $\mathcal{G}$.

One of Chandrasekaran’s definition of behavior mentions the notion of output state variable. In order to model it in our

[15] We allow that any belief of this kind may exhibit collective aspects, but for the sake of simplicity we neglect these aspects and avoid speaking about groups of agents.

framework we need the notion of output, respectively, input, behavior. These notions are also of crucial importance for other engineering models of artifact functions (e.g., Pahl & Beitz, 1998; Stone & Wood, 2000). Here we define them with the help of two primitive, that is, undefined, notions of input/output perdurant. It is assumed that the, actual or possible or generalized, life of any artifact contains two perdurants such that one of them corresponds to the class of input variables and the other corresponds to the class of output variables. For instance, the input perdurant of a thermostat (we discuss this example later on in more detail) can be the thermostat's detecting of a decline in temperature, and the output perdurant can be its closing of an electrical circuit connecting the thermostat and the furnace. Each such pair of input and output perdurants is selected by an agent involved in an engineering process, for example, in conceptual design. Then an input/output behavior is a behavior that involves such input/output perdurants. Let $\mathtt{InPD}_{\mathcal{G}}(a, e)$ mean that e is an input perdurant of an artifact a from the engineering perspective of an agent $\mathcal{G}$; similarly, $\mathtt{OutPD}_{\mathcal{G}}(a, e)$ means that e is an output perdurant of an artifact a from the engineering perspective of $\mathcal{G}$.

$$\mathtt{InPD}_{\mathcal{G}}(a, e) \rightarrow \mathrm{P}(e, \mathtt{Glf}(a)) \wedge \mathtt{PD}_{\mathcal{G}}(e) \tag{43}$$

$$\mathtt{OutPD}_{\mathcal{G}}(a, e) \rightarrow \mathrm{P}(e, \mathtt{Glf}(a)) \wedge \mathtt{PD}_{\mathcal{G}}(e) \tag{44}$$

Using the standard mereological technique we can also define the maximal input and output perdurants as fusions of, respectively, input and output perdurants.

An *input behavior* as seen from the engineering perspective of an agent $\mathcal{G}$, is any $\mathcal{G}$-behavior that involves some input perdurant. The *maximal input behavior* is such behavior that involves the maximal input perdurant. An *output behavior* as seen from the engineering perspective of an agent $\mathcal{G}$, is any $\mathcal{G}$-behavior that involves some output perdurant. The *maximal output behavior* is such behavior that involves the maximal output perdurant.

$$\mathtt{InBeh}_{\mathcal{G}}(a, e, b) \triangleq \mathtt{Beh}_{\mathcal{G}}(a, e, b) \wedge \mathtt{InPD}_{\mathcal{G}}(a, e) \tag{45}$$

$$\mathtt{OutBeh}_{\mathcal{G}}(a, e, b) \triangleq \mathtt{Beh}_{\mathcal{G}}(a, e, b) \wedge \mathtt{OutPD}_{\mathcal{G}}(a, e) \tag{46}$$

$$\begin{aligned}\mathtt{MaxInBeh}_{\mathcal{G}}(a, e, b) \triangleq{}& \mathtt{InBeh}_{\mathcal{G}}(a, e, b) \wedge \forall e'(\mathtt{InPD}_{\mathcal{G}}(a, e') \\ &\rightarrow \mathrm{P}(e', e))\end{aligned} \tag{47}$$

$$\begin{aligned}\mathtt{MaxOutBeh}_{\mathcal{G}}(a, e, b) \triangleq{}& \mathtt{OutBeh}_{\mathcal{G}}(a, e, b) \wedge \forall e'(\mathtt{OutPD}_{\mathcal{G}}(a, e') \\ &\rightarrow \mathrm{P}(e', e))\end{aligned} \tag{48}$$

4.3. Formalizing the five meanings of artifact behavior

Now we use the above ontological framework to formalize the five meanings of behavior and their examples, described by Chandrasekaran and Josephson (2000), as presented in Section 2.1. Here, state variables are indirectly captured through the given $\mathcal{G}$-behavior b, where $\mathcal{G}$ is an agent involved in some engineering task.

1. The behavior b of an artifact a from the engineering perspective of $\mathcal{G}$ is a $\mathcal{G}$-behavior such that if $\mathtt{Beh}(a, e, b)$, then the duration of e is an instant.[16]
   Chandrasekaran and Josephson's example of "the car rattled when the driver hit the curve": let e be the event described by the sentence "the car rattled when the driver hit the curve" and $\mathtt{Inst}$ a predicate corresponding to the property "being an instant" in the adopted theory of time, then

$$\mathtt{Beh}(\text{car}, e, b) \wedge \mathtt{PD}_{\mathcal{G}}(e) \wedge \mathtt{Inst}(tm(e)). \tag{49}$$

2. The behavior b of an artifact a from the engineering perspective of $\mathcal{G}$ is a $\mathcal{G}$-behavior such that $\mathtt{Beh}(a, e, b)$ implies $\mathrm{ST}(e)$.
   The example: let e be the event described by "the lintel's distributing the load to two sides," then

$$\mathtt{Beh}(\text{lintel}, e, b) \wedge \mathtt{PD}_{\mathcal{G}}(e) \wedge \mathrm{ST}(e) \wedge \mathrm{P}(e, \mathtt{Alf}(a)) \tag{50}$$

3. The behavior b of an artifact a from the engineering perspective of $\mathcal{G}$ is a $\mathcal{G}$-behavior such that if $\mathtt{Beh}(a, e, b)$, then $tm(e)$ is a period of positive length.
   The example: let $e =$ "the increasing of the BHP of the artifact for a while and its decreasing afterwards," then[17]

$$\mathtt{Beh}(\text{artifact}, e, b) \wedge \mathtt{PD}_{\mathcal{G}}(e) \wedge |tm(e)| > 0 \tag{51}$$

4. The behavior b of an artifact a from the engineering perspective of an agent $\mathcal{G}$ is a $\mathcal{G}$-behavior that is an output behavior of a.
   The example: let e be the event described by "the amplifier performing its function," then

$$\begin{aligned}&\mathtt{Beh}(\text{amplifier}, e, b) \wedge \mathtt{PD}_{\mathcal{G}}(e) \wedge \forall t, t'(\mathrm{P}(t, tm(e)) \\ &\wedge \mathrm{P}(t', tm(e)) \rightarrow \forall r, r'(ql(r, b, t) \wedge ql(r', b, t') \rightarrow r = r')\end{aligned} \tag{52}$$

5. The behavior b of an artifact a from an engineering perspective of an agent $\mathcal{G}$ is a maximal $\mathcal{G}$-behavior of a, restricted to the given period of time.

[16] What counts as an instant depends on the notion of time adopted in the application domain. It can be, for instance, an atomic period, a period of length zero, a period of positive length that satisfies some minimal constraint. These cases are all compatible to both the DOLCE ontology and our theory.

[17] This formalization does not capture the specific nature of the example given by Chandrasekaran and Josephson. Namely, we abstract from the fact that some parameter first increases and then decreases. The reason for this abstraction is that our goal is to identify a notion of behavior, not the particularities of this specific behavior b. For the interested reader, the change of the BHP, its increasing or decreasing, can be formalized by using the relations that occur in example 4. Contrary to example 3, in 4 these relations are explicitly used because they are necessary to capture the notion of behavior as discussed there.

Chandrasekaran and Josephson did not give an example. But if e is the perdurant involved in this case, then

$$\texttt{MaxBeh}_{\mathcal{G}}(a, e, b) \tag{53}$$

Admittedly, the formalization of the examples from Chandrasekaran and Josephson does not do justice to the richness of our framework and its advantages. For this reason, we will now formalize another example taken from Dretske (1988). Consider a mechanical thermostat in a room and assume that the room temperature drops to 17°C. The thermostat responds by turning the furnace on. This event characterizes a behavior of the thermostat: a fall in a room's temperature causes a bimetal strip in the thermostat to bend. When the bimetal strip bends to a certain angle A, here associated with 17°C, it closes an electrical circuit that connects the furnace to the thermostat and the furnace ignites. The event sequence can be illustrated in the following way:

1. The temperature drops to 17°C.
2. The strip bends to angle A.
3. The switch closes.
4. Current flows to the furnace,
5. The furnace ignites.

"The thermostat's behavior," Dretske observes, "is the bringing about of furnace ignition by events occurring in the thermostat; in this case, the closure of a switch by the movement of a temperature-sensitive strip" (Dretske, 1988, p. 86).

In our framework we represent the thermostat behavior as $\texttt{Beh}(\text{thermostat}, e, b)$ where b is the general behavior of the thermostat for the event e corresponding to the sequence 1–5 above. If we want to model the behavior for a subevent e', say 3, we write $\texttt{Beh}(\text{thermostat}, e', b')$ where e' is "Switch closes." Instead, the behavior of the switch at e' is introduced by writing $\texttt{Beh}(\text{switch}, e', b'')$.

Finally, note that we have not characterized the relationship between the behavior b of an artifact in an event e and its behavior b' in a subevent of e. There is an obvious dependence between the two that can be modeled according to well-known ontological approaches (e.g., Simons, 1987; Fine, 1995; Thomasson, 1999).

5. DEFINING ARTIFACT FUNCTIONS IN DOLCE

Besides distinguishing different meanings of behavior, Chandrasekaran and Josephson (2000) define the notion of artifact function. They assume a theoretical perspective in which artifact functions are construed as intended behaviors and define two concepts: *device-centric* function and *environment-centric* function. We will show in this section to what extent the ontological approach outlined above is suitable for grasping these concepts. Leaving aside some minor aspects, this section is intentionally very close to the approach of Chandrasekaran and Josephson. Admittedly, the formalization we obtain can be further elaborated and, perhaps, simplified. However, our goal here is to show how to translate the engineering work of Chandrasekaran and Josephson into a formal ontological system maintaining a clear link between the engineering and the formal approaches.

5.1. A formal definition of device-centric artifact function

To characterize both notions of functions, Chandrasekaran and Josephson start with the definition of behavioral constraint. Let X be a class of technical artifacts. It is stated in (Chandrasekaran & Josephson, 2000) that a *behavioral constraint* in X is any constraint on the behaviors of the elements of X. As their examples suggest, a behavioral constraint may be absolute, that is, unconditional, for example, that the value of output voltage is greater than 5 V, or conditional, for example, if the input voltage is above 5 V, the output voltage is a sinusoid.

We define the notion of behavioral constraints for entities in the class $\texttt{TechArt}$ by making use of the mereological notion of fusion (see Section 3.2). Namely, given a subclass X of $\texttt{TechArt}$, we consider the element a, which is the fusion of all elements of X, that is, $a = \sigma x(x \in X)$. We will call a the *environment* for the behavioral constraint. Formally, we define a *behavior environment* as follows

$$\texttt{BehEnv}(a) \triangleq \exists X(X \subseteq \texttt{TechArt} \wedge a = \sigma x(x \in X)) \tag{54}$$

Following the arguments presented in Chandrasekaran and Josephson (2000), we formalize behavioral constraints in environment a as pairs of behaviors. If pair $<b_0, b_1>$ is one of such constraints, then we say that b_0 is a condition for b_1 provided that two entities exist, a' and a'', for which $\text{P}(a', a)$, $\text{P}(a'', a)$ with b_0 a behavior of a' and b_1 a behavior of a''. For the sake of homogeneity, we also use pair $<b_0, b_0>$ to represent an unconditional constraint b_0. Thus, to summarize, when $b_0 = b_1$, pair $<b_0, b_1>$ indicates that behavior b_0 is an *unconditional constraint on a* and when $b_0 = b_1$, $<b_0, b_1>$ indicates a *conditional constraint on a*, namely, that behavior b_0 is a condition for behavior b_1.

Let us express the relationship "being a condition" using the primitive predicate $\texttt{Cond}$. We do not formally characterize this notion here besides requiring that its arguments are behaviors.

$$\texttt{Cond}(b, b') \rightarrow \texttt{Beh}(b) \wedge \texttt{Beh}(b') \tag{55}$$

Formally, we define a behavioral constraint in environment a, where a is a fusion of elements in $\texttt{TechArt}$, by

$$\begin{aligned}\texttt{CrBeh}(a, b_0, b_1) \triangleq {} & \texttt{BehEnv}(a) \wedge (b_0 = b_1 \rightarrow \exists a'(\text{P}(a', a) \\ & \wedge \texttt{Beh}(a', b_0))) \wedge (b_0 \neq b_1 \rightarrow \exists a', a''(\text{P}(a', a) \\ & \wedge \text{P}(a'', a) \wedge \texttt{Cond}(b_0, b_1) \wedge \texttt{Beh}(a', b_0) \\ & \wedge \texttt{Beh}(a'', b_1)))\end{aligned} \tag{56}$$

The notion of behavioral constraint is introduced to capture intended states of the world. As a result, any such constraint may, or may not, be satisfied. Such a conceptual category is crucial if one wants to allow for the possibility that some engineering models are not realizable because of various possible flows or oversimplifications. In other words, the predicate $\mathtt{Cond}$ does not express the objective relationship between behaviors but only an intended relationship. Therefore, within the domain of behavioral constraints we need to distinguish those constraints that are satisfied. At first blush, a behavioral constraint is satisfied when it corresponds to an objective regularity in the real world. To define this notion, we introduce the predicate $\mathtt{SatCrBeh}$ for *satisfied behavioral constraint*.

$$\begin{aligned}\mathtt{SatCrBeh}(a, b_0, b_1) \triangleq\, & \mathtt{CrBeh}(a, b_0, b_1) \wedge \exists a', a'', e', e''(\mathrm{P}(a', a) \\ & \wedge \mathrm{P}(a'', a) \wedge \mathtt{PossBeh}(a', e', b_0) \\ & \wedge \mathtt{PossBeh}(a'', e'', b_1)) \\ & \wedge (b_0 = b_1 \rightarrow \exists e\, \mathtt{ActBeh}(a', e, b_0)) \\ & \wedge (b_0 \neq b_1 \wedge \exists e\, \mathtt{ActBeh}(a', e, b_0) \\ & \rightarrow \exists e' \mathtt{ActBeh}(a'', e', b_1)) \end{aligned} \tag{57}$$

We are now in a position to grasp the device-centric notion of function defined in Chandrasekaran and Josephson (2000): "Let *F* be a class of behavioral constraints defined on, and satisfied by, an object *D*. If *F* is intended or desired by an agent *A*, then *D* has function *F* for *A*" (Chandrasekaran & Josephson, 2000, p. 172).

Note that *D* plays the role of the environment *X* and it is seen as a single object. In our terminology, the informal characterization given above tell us that a function of *a*, fusion of *a*, technical artifact, say *a*, for an agent $\mathcal{G}$ is a behavioral constraint on *a*, provided that this behavioral constraint is satisfied and is desired by $\mathcal{G}$. Thus, we need a notion of behavioral constraint *desired by* an agent. To this end, we will define the predicate $\mathtt{DES}_{\mathcal{G}}$ behaviors. The expression "$\mathtt{DES}_{\mathcal{G}}(a, b_0, b_1)$" means either that a behavioral constraint b_0 in environment *a* is desired by an agent $\mathcal{G}$ (when $b_0 = b_1$) or that a behavioral constraint $<b_0, b_1>$ (in *a*) is desired by $\mathcal{G}$ (when $b_0 \neq b_1$).

Recall that unconditional constraints are simply behaviors and that $\mathtt{Beh}_{\mathcal{G}}$ expresses the fact that a behavior is desired by $\mathcal{G}$ (see definition 40). Thus, we already possess the formal concepts to describe an unconditional behavioral constraint desired by an agent. We say that $<b_0, b_1>$ for environment *a* is *desired by* an agent $\mathcal{G}$ if there are a_0, a_1 in *a* such that for any e_0 and e_1 such that $\mathtt{Beh}(a_0, e_0, b_0)$ and $\mathtt{Beh}(a_1, e_1, b_1)$, then e_0 and e_1 are $\mathcal{G}$-desired perdurants. Formally,

$$\begin{aligned}\mathtt{DES}_{\mathcal{G}}(a, b_0, b_1) \triangleq\, & \mathtt{CrBeh}(a, b_0, b_1) \wedge \exists a', a''(\mathrm{P}(a', a) \\ & \wedge \mathrm{P}(a'', a) \wedge \forall e_0, e_1((b_0 \neq b_1 \\ & \wedge \mathtt{Beh}(a', e_0, b_0) \wedge \mathtt{Beh}(a'', e_1, b_1)) \\ & \rightarrow \mathtt{INT}_{\mathcal{G}}(e_0) \wedge \mathtt{INT}_{\mathcal{G}}(e_1)) \\ & \wedge \forall e((b_0 = b_1 \wedge \mathtt{Beh}(a', e, b_0)) \\ & \rightarrow \mathtt{Beh}_{\mathcal{G}}(a', e, b_0)) \end{aligned} \tag{58}$$

Now we have all the conceptual tools we need to define the notion of device-centric function. The particular feature of this definition is the fact that a function of a technical artifact is a behavioral constraint imposed in an environment that coincides with the artifact.

$$\mathtt{DevFunc}_{\mathcal{G}}(a, b_0, b_1) \triangleq \mathtt{SatCrBeh}(a, b_0, b_1) \wedge \mathtt{DES}_{\mathcal{G}}(a, b_0, b_1) \tag{59}$$

5.2. A formal definition of environment-centric artifact function

To define the environment-centric notion of function we need to introduce the notion of mode of deployment. According to Chandrasekaran and Josephson's explanation, a mode of deployment for an artifact a consists of what they call "the specifications of the ways in the causal interactions" between *a* and some objects from its environment. More perspicuously speaking, a mode of deployment for an artifact *a* consists of the structural relationships between *a* and the objects in the environment and the actions in which *a* and these objects are involved. We represent such modes of deployment using our notion of perdurant. As for behavioral constraints, we define the notion of model of deployment in relation to a single entity, that is, the fusion of the given environment. Then a mode of deployment of a technical artifact *a* in an environment *a*′ is a perdurant in which both *a* and *a*′ wholly participate, note: *a* is always included in the environment, thus P(*a*, *a*′). To be more specific, a *mode of deployment for* an artifact *a* in an environment *a*′ is any generalized perdurant *e* such that there exists a_1 with $\mathrm{P}(a_1, a')$ and $a \neq a_1$ and both *a* and a_1 participate in *e*. Informally, we capture the relationships between *a* and elements in the environment, as well as the actions among them, via a perdurant in which they all participate. Note that it suffices to change the perdurant to select different sets of entities, actions and relationships.

$$\begin{aligned}\mathtt{MD}(e, a, a') \triangleq\, & \mathtt{TechArt}(a) \wedge \mathtt{BehEnv}(a') \\ & \wedge \mathrm{P}(a, a') \wedge \exists a_1(\mathrm{P}(a_1, a') \wedge a \neq a_1 \\ & \wedge \mathtt{PC}_{\mathtt{WH}}(a, e) \wedge \mathtt{PC}_{\mathtt{WH}}(a_1, e)) \end{aligned} \tag{60}$$

Then, a *feasible mode of deployment* is an (engineering) possible perdurant which is a mode of deployment

$$\mathtt{FMD}(e, a, a') \triangleq \mathtt{EPD}(e) \wedge \mathtt{MD}(e, a, a') \tag{61}$$

Chandrasekaran and Josephson define environment-centric functions as follows: "Let *F* be a class of behavioral constraints that an agent, say *A*, desires or intends to be satisfied in some *W* [i.e., in some world *W*]. Let *D* be an object introduced into *W*, in a mode of deployment *M*(*D*, *W*). If *D* causes *F* to be satisfied in *W*, we say that *D* has, or performs, the function *F* in *W*" (Chandrasekaran & Josephson, 2000, p. 171).

We chose to formalize these functions, because as one can represent the causal relations at stake in various ways, there was not need to employ any particular theory of causality except to assume that the relation of causality is a binary relation that relates perdurants (cf. Lehmann et al., 2004). Let $\mathtt{Cause}(e_0, e_1)$ mean that a perdurant e_0 causes a perdurant e_1, that is,

$$\mathtt{Cause}(e_0, e_1) \rightarrow \mathtt{GEPD}(e_0) \wedge \mathtt{GEPD}(e_1) \tag{62}$$

In our framework we may add that the domains of actual perdurants and possible perdurants are closed downward and upward with respect to the relation of causality. Note: causality is not a bridge between these domains. More specifically,

- all causes of actual perdurants are actual,
- all results of actual perdurants are actual,
- all causes of engineering possible perdurants are engineering possible, and
- all results of engineering possible perdurants are engineering possible.

In formal terms these constraints have the following form:

$$\mathtt{Cause}(e, e') \rightarrow (\mathtt{APD}(e) \leftrightarrow \mathtt{APD}(e')) \tag{63}$$

$$\mathtt{Cause}(e, e') \rightarrow (\mathtt{EPD}(e) \leftrightarrow \mathtt{EPD}(e')) \tag{64}$$

Given this interpretation of causality, instead of saying that a technical artifact in a certain mode of deployment causes a behavioral constraint to be satisfied, we will say that a mode of deployment, that is, a perdurant, in which an artifact is involved causes a behavioral constraint to be satisfied. Let $\mathtt{CrBeh}(a, b_0, b_1)$ express a behavioral constraint in some environment a and let e be a feasible mode of deployment, of some artifact, in this environment. We will say that e causes a behavioral constraint $<b_0, b_1>$, to be satisfied in a, if and only if

1. $<b_0, b_1>$ is satisfied in a, and
2. e causes both perdurants involved in b_0 and b_1.

Formally,

$$\begin{aligned}\mathtt{Cause}_{\mathtt{MD}}(a, e, b_0, b_1) \triangleq\ & \mathtt{SatCrBeh}(a, b_0, b_1)\\ & \wedge \forall a_0, e_0(\mathtt{Beh}(a_0, e_0, b_0) \rightarrow \mathtt{Cause}(e, e_0))\\ & \wedge \forall a_1, e_1(\mathtt{Beh}(a_1, e_1, b_1) \rightarrow \mathtt{Cause}(e, e_1))\end{aligned} \tag{65}$$

A behavioral constraint is said to be a function of a technical artifact in a certain environment relative to a mode of deployment e if e causes the behavioral constraint and there is an agent $\mathcal{G}$ that desires this behavioral constraint. Then, an *environment-centric function* is defined as follows

$$\begin{aligned}\mathtt{EnvFunc}(b_0, b_1, a, a', e) \triangleq\ & \mathtt{Cause}_{\mathtt{MD}}(a, e, b_0, b_1) \wedge \mathtt{FMD}(e, a, a')\\ & \wedge \exists \mathcal{G} \mathtt{DES}_{\mathcal{G}}(a', b_0, b_1)\end{aligned} \tag{66}$$

6. CONCLUSIONS

In this paper we have presented a formal characterization of the engineering concepts of *behavior* and *function* of technical artifacts. The goal of our formalization was to capture the informal meanings these concepts have in the engineering practice and to characterize them as part of a foundational ontology. Of the various informal meanings of behavior and function that can be found in the engineering and design methodology literature, we formalized the five meanings of artifact behavior and the two meanings of function that Chandrasekaran and Josephson (2000) identified as part of what they call the functional representation approach. We used the DOLCE ontology (Masolo et al., 2003) to formalize these meanings.

More specifically, in Section 4 we introduced general concepts of artifact behavior in DOLCE. The basic idea of our formalization is that a behavior b of a technical artifact a in a perdurant e, where a perdurant is an entity that is only partially present at any time it is present, is the specific way in which a occurs in e. For example, if a is a capacitor, then the way in which a occurs or exists in a given process of storing electric energy is a behavior of this capacitor a. Looking at engineering we distinguished different classes of behavior. Engineers consider in, for instance, designing, artifact behaviors in actually, in physically possible, and in physically impossible perdurants, which lead us to define actual behaviors of artifacts, possible behaviors of artifacts, and behaviors that are believed to be artifacts behaviors. Moreover, engineers can isolate input and output behaviors of artifacts, and we captured these concepts using twoprimitive notions. In Section 4.3 we then used these general concepts to formalize Chandrasekaran and Josephson's five identified engineering meanings of behavior.

In Section 5 we moved to the task of introducing the concept of artifact function in DOLCE. In Section 5.1 we defined, using DOLCE, the notions of behavioral constraint and behavioral constraint desired by an agent, and then captured Chandrasekaran and Josephson's notion of a device-centric function of a technical artifact function. Finally, in Section 5.2 we defined the notion of mode of deployment of an artifact in DOLCE and captured Chandrasekaran and Josephson's second notion of an environment-centric function of a technical artifact.

We have shown that an ontological characterization of artifact behavior and function, using a foundational ontology, can provide the means to make precise and formalize the informal meaning these concepts have in engineering. These meanings may be used ambiguously in engineering designing, redesigning, reverse engineering, and knowledge bases, yet they can be accommodated and distinguished in a foundational ontology such as DOLCE. Our results thus bring us closer to the ideal that engineers from different disciplinary backgrounds, and working with different engineering models, can share and exchange their behavioral and functional descriptions of technical artifacts without getting caught in conceptual misunderstanding. This idea is, however, still an ideal. Our formalizations have not yet been implemented as engineering tools for facilitating the sharing and exchanging of descriptions of the behavior and functions of artifacts, and information about their practical use is still required. The DOLCE ontology has already been applied by a variety of independent organizations in several projects covering a variety of areas like, for example, e-commerce, manufacturing, social cooperation, linguistics, and the Semantic web.[18] In addition, the ontology has been positively evaluated in independent multipurpose surveys and comparison studies like the MITRE Technical Report,[19] the Ontospace Project,[20] and the SmartWeb Project,[21] just to name a few. Our formalization of behavior and function in DOLCE is fairly direct and did not brought up the need to changes in the DOLCE ontology. The formalization we have provided fits thus quite naturally with the DOLCE structure and, because pivoted by engineering studies, its implementation is expected by us to obtain a positive evaluation.

In this paper we have formalized, using DOLCE, the engineering meanings of behavior and functions as identified by Chandrasekaran and Josephson (2000), and as acknowledged, these meanings are part of what they call the functional representation approach. The engineering meanings of artifact behavior and functions as they figure in the rival functional modeling approach still need to be accommodated in our framework. The results of our current analysis provide us with the means to achieve this task. In the functional basis model of designing of Stone and Wood (2000), which may be taken as one of the more important representatives of the functional modeling approach, functions are, as discussed in Section 2.2, defined as general input/output relationships between flows of materials, energies, and signals. Now, taking these relationships as relationships between input and output behaviors of artifacts, we arrive at a first approach to accommodate also this concept of function in DOLCE. The question of whether this can be done has still to be answered and this defines our next project. If it can be done, a foundational ontology as DOLCE will prove its value by allowing us to make unambiguous communications about the behaviors and functions of technical artifacts possible in engineering.

ACKNOWLEDGMENTS

Stefano Borgo was partially supported by the Provincia Autonoma di Trento (PAT) and by the TOCAI.IT project, Pawel Garbacz by the Marie Curie Intra-European Fellowship (EIF-006550), and Pieter Vermaas by the Netherlands Organization for Scientific Research (NWO).

[18] A list of users can be found at http://www.loa-cnr.it/DOLCE

[19] See http://www.mitre.org/work/tech_papers/tech_papers_04/04_0603/04_1175.pdf

[20] See http://www.ontospace.uni-bremen.de/twiki/bin/view/Main/Spatial Ontology

[21] See http://smartweb.semanticweb.org

REFERENCES

Bhatta, S., & Goel, A. (2002). Design patterns and creative design. In *Engineering Design Synthesis* (Chakrabarti, A., Ed.) pp. 271–284. Berlin: Springer.

Bhatta, S., Goel, A., & Prabhakar, S. (1994). Innovation in analogical design: a model-based approach. *Artificial Intelligence in Design—1994* (Gero, J., & Sudweeks, F., Eds.), pp. 57–74. Dordrecht: Kluwer.

Borgo, S. (2007). How formal ontology can help civil engineers. In *Ontologies for Urban Development* (Teller, J., Lee, J., & Roussey, C., Eds.), pp. 37–45. Berlin: Springer.

Borgo, S., Carrara, M., Vermaas, P.E., & Garbacz, P. (2006). Behavior of a technical artifact: an ontological perspective in engineering. *Formal Ontology in Information Systems: Proc. 4th Int. Conf. (FOIS 2006). Frontiers in Artificial Intelligence and Applications* (Bennett, B., & Fellbaum, C., Eds.), Vol. 150, pp. 214–225. Amsterdam: IOS.

Borgo, S., & Leitao, P. (2007). Foundations for a core ontology of manufacturing. In *Ontologies: A Handbook of Principles, Concepts and Applications in Information Systems. Integrated Series in Information Systems* (Kishore, R., Ramesh, R., & Sharman, R., Eds.), Vol. 14, pp. 752–776. Dordrecht: Springer.

Borgo, S., & Vieu, L. (in press). Artefacts in formal ontology. In *Handbook of Philosophy of Technology and Engineering Science*. Amsterdam: Elsevier.

Bryant, C.R., McAdams, D.A., Stone, R.B., Kurtoglu, T., & Campbell, M.I. (2006). A validation study of an automated concept generator design tool. *Proc. 2006 ASME IDETC/CIE Conf.*, Paper No. DETC2006-99489, Philadelphia, PA, September 10–13.

Bucciarelli, L.L. (1994). *Designing Engineers*. Cambridge, MA: MIT.

Chandrasekaran, B. (1994). Functional representation and causal processes. *Advances in Computers 38*, 73–143.

Chandrasekaran, B. (2005). Representing function. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 19(1)*, 65–74.

Chandrasekaran, B., & Josephson, J.R. (2000). Function in device representation. *Engineering With Computers 16(3/4)*, 162–177.

Chandrasekaran, B., Josephson, R., & Benjamins, V. (1999). What are ontologies, and why do we need them? *IEEE Intelligent Systems 14*, 20–26.

Chittaro, L., & Kumar, A.N. (1998). Reasoning about function and its applications to engineering. *Artificial Intelligence in Engineering 12*, 331–336.

Dretske, F. (1988). *Explaining Behavior*. Cambridge, MA: MIT.

Fine, K. (1995). Ontological dependence. *Proceedings of the Aristotelian Society 95*, pp. 269–290.

Garbacz, P. (2006). A formal model of functional decomposition. *Proc. 2006 ASME IDETC/CIE Conf.*, Paper No. DETC2006-99097, Philadelphia, PA, September 10–13.

Gero, J.S. (1990). A knowledge representation schema for design. *AI Magazine 11(4)*, 26–36.

Gero, J.S., & Kannengiesser, U. (2004). The situated function–behavior–structure framework. *Design Studies 25*, 373–391.

Goel, A. (1991). A model-based approach to case adaptation. *Proc. 13th Annual Cognitive Science Conf.*, pp. 143–148. Mahwah, NJ: Erlbaum.

Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S., & Wood, K.L. (2002). A functional basis for engineering design: reconciling and evolving previous efforts. *Research in Engineering Design 13*, 65–82.

Kitamura, Y., Koji, Y., & Mizoguchi, R. (2005/2006). An ontological model of device function: industrial deployment and lessons learned. *Applied Ontology 1*, 237–262.

Kitamura, Y., Sano, T., Nambo, K., & Mizoguchi, R. (2002). A functional concept ontology and its application to automatic identification of functional structures. *Advanced Engineering Informatics 16(2)*, 145–163.

Lehmann, J., Borgo, S., Masolo, C., & Gangemi, A. (2004). Causality and causation in dolce. *Proc. 3rd Int. Conf. FOIS 2004* (Varzi, A.C., & Vieu, L., Eds.), pp. 273–284. Amsterdam: IOS.

Lemmon, E.J. (1965). *Beginning Logic*. Indianapolis, IN: Hackett Publishing Company.

Masolo, C., Borgo, S., Gangemi, A., Guarino, N., & Oltramari, A. (2003). Wonderweb deliverabled18. Accessed at http://www.loa-cnr.it/Papers/D18.pdf

Mizoguchi, R., Sunagawa, E., Kozaki, K., & Kitamura, Y. (2007). Model of roles within an ontology development tool: Hozo. *Journal of Applied Ontology 2*, 159–179.

Pahl, G., & Beitz, W. (1998). *Engineering Design*. Berlin: Springer.

Rosenman, M.A., & Gero, J.S. (1998). Purpose and function in design. *Design Studies 19*, 161–186.

Simons, P. (1987). *Parts: A Study in Ontology*. Oxford: Clarendon.

Staab, S., & Studer, R. (2004). *Handbook of Ontologies; International Handbooks on Information Systems*. Berlin: Springer.

Stone, R.B., & Wood, K. (2000). Development of a functional basis for design. *Journal of Mechanical Design 122(4)*, 359–276.

Thomasson, A. (1999). *Fiction and Metaphysics*. Cambridge: Cambridge University Press.

Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., & Tomiyama, T. (1996). Supporting conceptual design based on the function–behavior–state modeler. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 10*, 275–288.

Umeda, Y., Kohdoh, S., Shimomura, Y., & Tomiyama, T. (2005). Development of design methodology for upgradable products based on function–behavior–state modeling. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing 19*, 161–182.

Umeda, Y., & Tomiyama, T. (1997). Functional reasoning in design. *IEEE Expert: Intelligent Systems and Their Applications 12*, 42–48.

Vermaas, P.E. (2007). The functional modelling account of Stone and Wood: some critical remarks. *Proc. 16th Int. Conf. Engineering Design, Design for Society: Knowledge, Innovation and Sustainability*, Abstract, Paris, 28–30 August, pp. 851–852, full paper on accompanying CD, Paris: Ecole Centrale.

Vermaas, P.E., & Dorst, K. (2007). On the conceptual framework of John Gero's FBS-model and the prescriptive aims of design methodology. *Design Studies 28*, 133–157.

Stefano Borgo is a Researcher in the Laboratory for Applied Ontology (LOA, ISTC-CNR) in Trento, Italy. He works in the area of artificial intelligence, with special interests in the formalization of domains like enterprise modeling, process modeling, geography, and multiagent systems. His competence areas are logic, conceptual modeling, and formal ontology.

Massimiliano Carrara is an Assistant Professor of logic and philosophy of science in the Department of Philosophy at the University of Padua, Italy. He graduated and received his PhD in philosophy from the University of Padua. His main research interests are in philosophical logic and metaphysics, in particular, the notions of identity and existence.

Pawel Garbacz is an Assistant Professor in the Department of Philosophy at John Paul II Catholic University, Lublin, Poland. His research focuses on the interdisciplinary problems among philosophy, logic, and artificial intelligence.

Pieter Vermaas is a Researcher in the Department of Philosophy at Delft University of Technology, Delft, The Netherlands. Since 2000 his research focus has been the concept of technical function as used in engineering and design methodologies and as analyzed in the philosophy of technology.

APPENDIX A: FORMAL TERMINOLOGY

Table A.1 contains the most important notions of our formal ontology of artifact behaviors and functions. Some auxiliary notions that occur in the text above have been neglected. If a given term is not defined in this paper, we label it "primitive," irrespective of whether it is primitive or defined in DOLCE.

Table A.1. *Formal terminology*

Term	Explanation	Status
	Foundation	
ED(x)	x is an endurant	Primitive (DOLCE)
PED(x)	x is a physical endurant	Primitive (DOLCE)
APO(x)	x is an agentive physical endurant	Primitive (DOLCE)
NAPO(x)	x is a nonagentive physical endurant	Primitive (DOLCE)
PD(x)	x is a perdurant	Primitive (DOLCE)
APD(x)	x is an actual perdurant	Primitive
EPD(x)	x is an engineering possible perdurant	Primitive
$tm(e)$	The period of time spanned by the perdurant e	Defined by (10)
Inst(e)	The perdurant e is an instant	Primitive
Coh(e, e')	e is coherent with e'	Defined by (20)
Q(x)	x is a quality	Primitive (DOLCE)
$ql_{\mathrm{T,PD}}(t, x)$	t is the temporal location of a perdurant x	Primitive (DOLCE)
PC(a, e, t)	a participates in e through t	Primitive (DOLCE)
$\mathrm{PC}_{\mathrm{WH}}(a, e)$	a wholly participates in e	Defined by (23)
MIN(a, e)	e is a minimal perdurant for a	Defined by (24)
P(x, y)	x is part of y	Primitive (DOLCE)
PP(x, y)	x is a proper part of y	Defined by (1)
O(x, y)	x overlaps y	Defined by (2)
$x + y$	The mereological sum of x and y	Defined by (3)
$\sigma x \phi(x)$	The mereological sum of all entities that satisfy ϕ	Defined by (4)
$\Sigma_{\mathrm{Coh}}(a, e, s)$	s is the fusion of a's perdurants coherent with e	Defined by (27)
$\Sigma_{\mathrm{Cons}}(a, e, s)$	s is the fusion of a's perdurants consistent with e	Defined by (29)
$\Sigma_{\mathcal{G}-\mathtt{Coh}}(a, e, s)$	s is the fusion of a's $\mathcal{G}$-possible perdurants coherent with e	Defined by (41)
Alf(e, a)	e is the actual life of a	Defined by (25)
Plf(e, a)	e is a possible life of a	Defined by (28)
Glf(e, a)	e is the generalized life of a	Defined by (30)

Behavior		
Cause(e_0, e_1)	e_0 is a cause of e_1	Primitive
Beh(a, e, b)	b is the behavior of a in e	Primitive
Beh(a, b)	b is the behavior of a in some perdurant	Defined by (8)
Beh(b)	b is the behavior of an artifact in some perdurant	Defined by (9)
B(x)	x is an (individual quality) behavior	Primitive
ActBeh(a, e, b)	b is the actual behavior of a in e	Defined by (31)
ActBeh$_{\max}$(a, b)	b is the maximal actual behavior of a	Defined by (32)
PossBeh(a, e, b)	b is a possible behavior of a in e	Defined by (33)
PossBeh$_{\max}$(a, e, b)	b is a maximal possible behavior of a in e	Defined by (34)
ImBeh(a, e, b)	b is an impossible behavior of a in e	Defined by (35)
GenBeh(a, e, b)	b is a generalized behavior of a in e	Defined by (36)
GenBeh$_{\max}$(a, e, b)	b is a maximal generalized behavior of a for e	Defined by (37)
PD$_{\mathcal{G}}$(e)	e is a $\mathcal{G}$-possible perdurant	Primitive
INT$_{\mathcal{G}}$(e)	e is a $\mathcal{G}$-desired perdurant	Primitive
Beh$_{\mathcal{G}}$(a, e, b)	b is a $\mathcal{G}$ behavior of a in e	Defined by (40)
MaxBeh$_{\mathcal{G}}$(a, e, b)	b is a maximal $\mathcal{G}$ behavior of a in e	Defined by (42)
InPD(e, a)	e is an input perdurant of a	Primitive
OutPD(e, a)	e is an output perdurant of a	Primitive
InBeh$_{\mathcal{G}}$(a, e, b)	b is an input $\mathcal{G}$ behavior of a in e	Defined by (45)
OutBeh$_{\mathcal{G}}$(a, e, b)	b is an output $\mathcal{G}$ behavior of a in e	Defined by (46)
MaxInBeh$_{\mathcal{G}}$(a, e, b)	b is the maximal input $\mathcal{G}$ behavior of a in e	Defined by (47)
MaxOutBeh$_{\mathcal{G}}$(a, e, b)	b is the maximal output $\mathcal{G}$ behavior of a in e	Defined by (48)
Function		
BehEnv(a)	a is a behavior environment	Defined by (54)
Cond(b, c)	b is a condition of c	Primitive
CrBeh(a, b, c)	$<b, c>$ is a behavioral constraint in a	Defined by (56)
SatCrBeh(a, b, c)	$<b, c>$ is a satisfied behavioral constraint in a	Defined by (57)
DES$_{\mathcal{G}}$(a, b, c)	$<b, c>$ is a behavioral constraint in a desired by $\mathcal{G}$	Defined by (58)
DevFunc$_{\mathcal{G}}$(a, b, c)	$<b, c>$ is a device-centric function of a for $\mathcal{G}$	Defined by (59)
MD(e, a, a')	e is a mode of deployment of a in a'	Defined by (60)
Cause$_{\text{CrBehMD}}$(e, b, c)	e causes $<b, c>$	Defined by (65)
EnvFunc(b, c, a, a', e)	$<b, c>$ is an environment-centric function of a in a' under e	Defined by (66)