Proceedings of TMCE 2010 Symposium, April 12-16, 2010, Ancona, Italy, ed. by I. Horváth, F. Mandorli and Z. Rusák

FORMALIZATIONS OF FUNCTIONS WITHIN THE DOLCE ONTOLOGY

Stefano Borgo
Laboratory for Applied Ontology
Institute of Cognitive Sciences and Technologies CNR
Italy
borgo@loa-cnr.it

Massimiliano Carrara
Philosophy Department
University of Padua
Italy
massimiliano.carrara@unipd.it

Pawel Garbacz
Philosophy Department
John Paul II Catholic University of Lublin
Poland
garbacz@kul.lublin.pl

Pieter E. Vermaas
Philosophy Department
Delft University of Technology
The Netherlands
p.e.vermaas@tudelft.nl

ABSTRACT

In this paper we give formalizations of two engineering concepts of technical functions and present in more general terms the project of supporting engineering functional reasoning by means of ontological analyses. The concepts that we formalize are the concepts of function as defined in the Functional Representation *approach by Chandrasekaran and Josephson and in the* Functional Basis *approach by Stone and Wood. These two concepts represent two main ways of understanding functions in engineering: the first by means of the behavior of artifacts, and the second by means of operations on flows as preformed by artifacts. Both formalizations are given within the foundational* DOLCE *ontology. Our choice to formalize existing concepts of functions within a single foundational ontology, is one strategy towards the goal of ontological analyses of functions. This goal of enabling the development of tools for automated functional reasoning, may be realized by other strategies as well, such as defining a single formalized concept of function, either for replacing existing concepts or for use as a reference to which existing concepts should be related. We compare these strategies briefly and discuss the merits and shortcomings of our strategy.*

KEYWORDS

function, formal ontology, DOLCE, functional basis, functional representation

NOMENCLATURE

ED	=	is an endurant
PD	=	is a perdurant
PED	=	is a physical endurant
POB	=	is a physical object
P	=	is part of
PP	=	is a proper part of
PC_T	=	participates
`TechArt`	=	is a technical artifact
`Beh`	=	is a behavior in an event
`APD`	=	is an actual perdurant
`EPD`	=	is an engineering perdurant
`GPD`	=	is an generalized perdurant
`Cond`	=	is a condition for
`CrBeh`	=	is a behavioral constraint in
`SatCrBeh`	=	is a satisfied behavioral constraint in
`DES`	=	behavioral constraint is desired in
`BehEnv`	=	is a environment
`CS`	=	causally brings about

`DevFunc`	=	is a device-centric function
`EnvFunc`	=	is an environment-centric function
`MD`	=	is a mode of deployment
`FMD`	=	is a feasible mode of deployment
`Req`	=	design requirements
`CustNeeds`	=	customer needs
`EngFun`	=	engineering function
`BasicFunc`	=	basic function
`ProdFunc`	=	product function
`Dev`	=	is a technical device
`InFlow`	=	is an input flow for
`OutFlow`	=	is an output flow for
`MFlow`	=	is a flow of matter
`EFlow`	=	is a flow of energy
`SFlow`	=	is a flow of signal
`Perform`	=	performs

Subscripts:

$\mathcal{G}$	=	refers to group of agents
$\mathtt{T}$	=	refers to time

1. INTRODUCTION

The formalization of technical functions in engineering ontologies is generally seen as furthering our understanding of this concept. Functions are used ubiquitously in engineering but despite their central importance, the concept is associated with various and, sometimes, possibly conflicting meanings (e.g., [10] and [21] in this volume). Formalization can clarify these meanings and the way in which they may differ from one another. This clarification may in turn lead to the means necessary to build automated tools for engineering, such as tools for supporting reasoning in engineering designing of new technical artifacts and tools for archiving and retrieving functional descriptions of existing artifacts.

In this paper we present two of such formalizations. We describe the results of earlier work [3] on functions as understood by the *Functional Representation* approach formulated by Chandrasekaran and Josephson [8]. And we present recent work [5], [6] on functions as understood by the *Functional Basis* approach as defined by Stone and Wood [19], which is similar to the way in which this is done by Pahl and Beitz [18]. These two approaches have been identified by Chandrasekaran [7][1] as the two main rival ones in engineering.

The ontology we use for both formalizations is DOLCE, the *Descriptive Ontology for Linguistic and Cognitive Engineering* [15]. By giving these formalizations with one ontology, we obtain two important results: an analysis of the conceptual structure of the individual approaches and a framework for a direct comparison of these approaches. This brings us to our larger project. Our broader goal is to build a series of integrated formal systems that capture various ways in which engineers use the concept of function. These formal systems may, from a practical engineering perspective, pave the way towards building tools for supporting engineering, and towards comparisons and translations of the functional concepts between different approaches. From a more theoretical perspective they may enable us to compare our work with related work on the formalization of functions, notably work by Mizoguchi, Kitamura and their collaborators [13], [12], and by Arp and Smith [1].

This paper makes one step in this direction: the goal is to present the formalization of the concepts of function as advanced by the Functional Representation and Functional Basis approaches. We proceed as follows. After a brief survey of our topic and the work done on it, we discuss our assumptions. In section 4 we then present the main line of the formalization of functions in the Functional Representation approach,[2] and in section 5 we give it for functions in the Functional Basis approach. We end with a discussion and conclusions.

2. SURVEY

The variety of meanings in which the concept of function is used in current engineering is recently surveyed in [10]. A thorough way of carrying out our project would imply formalizing all 18 proposals considered in this survey, an effort which may be simplified a bit by highlighting similarities between the proposals. In order to make progress in a more sophisticated way, we however focus on only two main approaches in understanding functions, as defined by Chandrasekaran [7]. These approaches are the *Functional Representation* (FR) approach formulated by Chandrasekaran and Josephson [8], and the *Functional Basis* (FB) approach as advanced by Stone and Wood [19]. We introduce these two notions in the subsections 2.1 and 2.2.

Work on the formalization of technical functions by

[1] In [7] the Functional Basis approach is actually called the *Functional Modeling* approach.

[2] Section 4, and some other parts of this paper originate from [4].

means of engineering ontologies is currently being carried out in different ways. In principle two choices needs to be made in this work, one concerning the ontology to be used and the second concerning the meaning of the concept of function that is formalized. The first choice includes a choice between a light-weight '*terminological*' *ontology*, consisting of a clear and well-related standard engineering terminology, and a *foundational ontology*, formulated in terms of general, axiomatized and philosophically motivated concepts [2]. The second choice involves making a normative assessment of the current meanings attached to the concept of function in engineering. Because these current meanings are ambiguous and imprecise, one can opt for introducing a new but clear and formalized concept of function that *ipso facto* has a meaning different to the meanings that engineers use. This new concept may be introduced for replacing existing meanings in engineering, or for acting as a reference against which the existing engineering meanings are analyzed, related to one another, and eventually translated into one another. Alternatively, one can opt for directly formalizing the existing ways in which engineers use functions without the help of auxiliary notions.

The main group of researchers involved in the formalization of functions is the group of Mizoguchi and Kitamura [13], [12]; Arp and Smith [1] have recently included technical functions into their ontological work; and our contribution is given in [3], [5], [6]. All these authors make the first choice in favor of foundational ontologies, although not in favor of one and the same ontology: Mizoguchi and Kitamura rely on YATO, Arp and Smith on BFO, and we have opted for DOLCE.[3] With respect to the second choice there is however not consensus. Mizoguchi and Kitamura introduce one formalized meaning of function, acknowledge current engineering meanings attached to the concept, and develop translation rules between their formalized meaning of function and these engineering meanings [14]. Arp and Smith define a formalized meaning that may be taken as replacing existing meanings. Our choice is to formalize directly the existing ways in which engineers use functions as we will explain in section 3.

[3]See, respectively, http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/upperOnto.htm, http:/www.ifomis.org/bfo and http:/www.loa-cnr.it/DOLCE.html.

2.1. The functional representation approach

Functions of artifacts as defined in the FR approach advanced by Chandrasekaran and Josephson [8] are defined in terms of the behavior of the artifacts. Chandrasekaran and Josephson isolate five engineering meanings of behavior and two of function. The meanings of behavior are characterized with the help of the primitive notion of *state variable*:

- 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.
- behavior as the value of a property of the artifact or a relation between such values.
  Example: a lintel distributes the load to two sides.
- behavior as the value of some state variable of the artifact over an interval of time.
  Example: the BHP[4] increased for a while, but then started decreasing.
- 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.
- behavior as the values of all the described state variables of the artifact at a particular instant or over an interval.
  (No example given.)

Notice that for all five meanings, a behavior of a technical artifact is partially objective and partially subjective. It has an objective aspect because it eventually depends on the properties or features of the artifact. Still, the very same behavior has a subjective aspect: it depends on the designer(s) and, indirectly, on engineering practice for the choice of the state variables.

The two meanings of function that Chandrasekaran and Josephson distinguish are called the *device-centric* and *environment-centric* meanings. 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 case of an electrical buzzer. An *environment-centric function* is in turn an effect or impact of this behavior of

[4]BHP stands for Brake Horse Power and it is described as the amount of real horsepower going to the pump.

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. The functional basis approach

In the rival FB approach the concept of behavior has no role and the meaning of function is given in a more uniform way. The approach of Stone and Wood [19] propounds to model overall product functions, especially from the electromechanical and mechanical domain, as sets of connected elementary sub-functions. In line with the design methodology of Pahl and Beitz [18] an overall product function is described by means of a verb-object form and graphically represented by a black-boxed operation on flows of materials,[5] energies, and signals. Sub-functions are also described by verb-object forms but they correspond to well-defined basic operations on well-defined basic flows of materials, energies, and signals. Basic operations and basic flows are limited in number and listed in shared libraries, given in [11], which define the functional design space. The whole system is called a “Functional Basis”.[6]

Stone and Wood present their proposal as supporting the archiving, comparison, and communication of functional descriptions of existing artifacts, as well as the engineering designing of new artifacts. Archiving, comparison, and communication is assisted since the sub-functions into which overall product functions are decomposed, are described by means of the limited number of basis operations and basic flows. Designing of new artifacts is supported since the FB design methodology allows designers to make critical design decisions about the architecture of artifacts, that is, decisions about the decomposition of the overall product functions in sub-functions, in the early conceptual stage of designing at which only functional descriptions are considered. Moreover, this approach provides the means to find overall design solutions quickly since Stone and Wood require the sub-functions to be small and easily solvable in designing.

[5]The notion of material is construed here rather broadly as it comprises also such objects as screws, air, human beings, and their parts.

[6]FB is the result of reconciliation of two previous taxonomies: the NIST taxonomy (cf. [20]) and the older versions of Functional Basis developed in, for example, [16].

In the designing of new artifacts the overall product function is given in terms of input/output flows and may initially be coarse-grained. Consider the following example: the overall product function of loosening/tightening screws, which is performed or ascribed to a power screwdriver, is defined by several input flows: electricity, human force, on/off signals and screws; and one output flow: looseness/tightness of screws. But when this overall product function is decomposed in terms of connected sub-functions with well-defined input and output flows, the overall product function can be modeled in more detail. The input flows are extended to include also relative rotation, and the output flows to include also heat and noise. The full-fledged functional decomposition is depicted in figure 1.

Stone and Wood [19] provide a glossary of the terms used in their account, which we resume here for later reference. Note that the term ‘function’ *simpliciter* is by Stone and Wood defined to indicate only an operation. Formally this definition has the awkward consequence that neither product functions nor sub-functions (as described above) are instances of functions in the FB terminology. More generally this definition may create the wrong impression that the FB approach is merely one about operations, whereas it actually has a much broader scope and use in engineering.

- *Product function*: the general input/output relationship of an artifact having the purpose of performing an overall task, typically stated in verb-object form.
- *Sub-function*: a description of a part of an artifact’s overall task stated in verb-object form. Sub-functions are decomposed from the product function and represent the more elementary tasks of the artifact.
- *Function*: a description of an operation to be performed by an artifact. It is expressed as the active verb of the sub-function.
- *Flow*: a change in material, energy, or signal with respect to time. Expressed as the object of the sub-function, a flow is the recipient of the function’s operation.

3. ASSUMPTIONS AND METHOD

In our ontological analysis of functions we focus only on the meanings as advanced in the Functional Representation and Functional Basis approaches. This selection may be taken as somewhat limited and arbitrary, since by the survey [10] there may be more

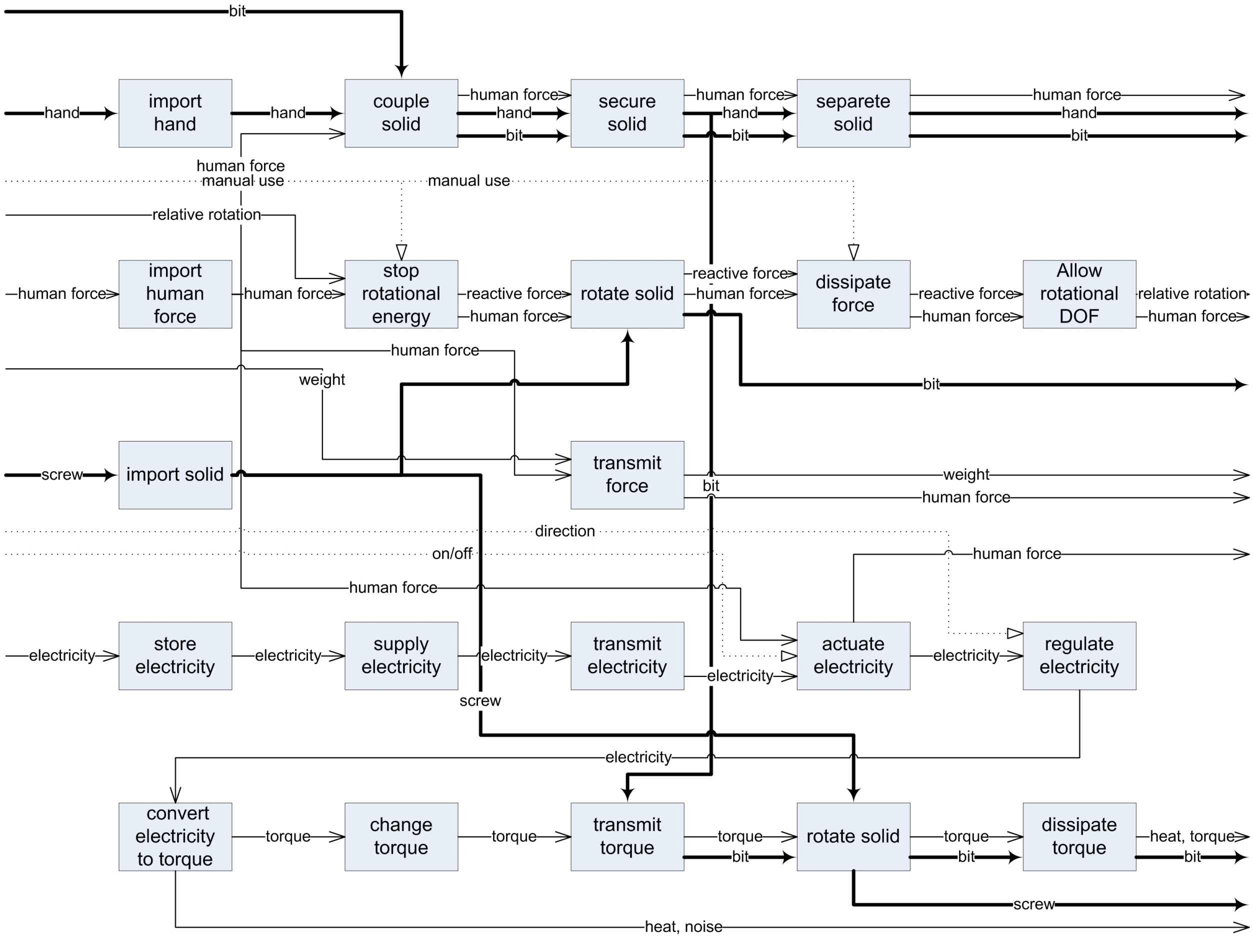

Figure 1 Functional decomposition of a screwdriver in FB

than 18 meanings around, and since other selections may be reasonable as well.[7] Relying on Chandrasekaran [7] we however assume that these two meanings cover a larger part of practices in which engineers reason about functions. Moreover, the two meanings that are selected are quite different. In the FR approach functions of artifacts are related to behaviors of artifacts, and then these behaviors are related to structural-physical descriptions of the artifacts. This approach thus grants a pivotal conceptual role to the term 'behavior', and suggests a clear ontological ordering: technical artifacts have their physical structure; this structure, in interaction with a physical environment, gives rise to the artifacts' behavior; and this behavior then determine in some way the artifacts' functions. Behavior is denied this pivotal role in the FB approach. Functions of artifacts are modeled in the FB approach in terms of inputs and outputs, and through this modeling, the functions are directly related to the structural-physical descriptions of artifacts. In this second approach there is thus no explicit role for the concept of behavior in relating functions of artifacts with the structure of artifacts. By these differences between the two approaches, we cover thus two quite distinct meanings of function.

In our formalization we use DOLCE as our ontology. This usage again introduces arbitrariness, which this time is unavoidable since a "one and only", best ontology is currently not available, and may easily never come to be available. We describe DOLCE in a separate subsection.

3.1. DOLCE

The DOLCE ontology — the *Descriptive Ontology for Linguistic and Cognitive Engineering* — is part of the *WonderWeb* effort.[8] The vision of this effort is

[7]For instance, in [21] in this volume there are for a purpose different to formalization, three meanings selected as archetypical to the variety of engineering meanings of function. Two of these archetypical meanings correspond to the meanings considered in this paper.

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

to have a library of foundational ontologies reflecting different ontological choices. For that reason extensive attention is given to a careful isolation of the ontological options and their formal relationships.

The DOLCE's ontology is the most studied module of this library. Formalizing the FR and FB meanings of function by means of DOLCE in part means that we also want to extend DOLCE to capture crucial notions in the area of engineering and so to use this ontological framework to analyze, clarify, and possibly improve the work in this area. Here we can provide just a minimal introduction to the whole ontology, the interested reader can find in [15] the underlying motivations and a throughout discussion of technical aspects. DOLCE ontology concentrates on particulars as opposed to universals. Roughly speaking, a *universal* is an entity that is instantiated or concreted by other entities (like "being human" and "being an event"). A *particular* (an element of the class PARTICULAR) is an entity that is not instantiated by other entities (like the Eiffel Tower in Paris). That is, your car is a particular as opposed to the model of your car, provided that the latter is interpreted as a type being instantiated by a number of entities among which one is your car. 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 like products and events like operations; it includes a differentiation among individual qualities (e.g., the weight of a specific material item), quality types (weight, color, and the like), quality spaces (spaces to classify weights, colors, etc.), and quality positions or qualia (informally, locations in quality spaces). These, together with measure spaces (where the quality positions get associated to a measure system and, thus, to numbers), are important to describe and compare artifacts. Indeed, among the motivations to use DOLCE, an important element was its robustness and flexibility which allows to capture in a natural way the views proper of engineering practice.

The DOLCE ontology category (class) ENDURANT comprises objects (like a *hammer*) or amounts of matter (like *an amount of plastic*), while the category PERDURANT comprises events like *making a hole* or *playing a soccer game*, that is, things that happen in time. (Formally, the category ENDURANT is represented by the constant ED and we write $\mathsf{ED}(x)$ to mean that x is an endurant. Analogously, we use PD for perdurants.) The term 'object' is used in the ontology to capture a notion of unity as suggested by the partition of the class PHYSICAL ENDURANT (formally, PED) into classes AMOUNT OF MATTER (M), FEATURE (F), and PHYSICAL OBJECTS (POB). Both endurants and perdurants are associated with a bunch of individual qualities (elements of the category QUALITY). The exact list of qualities may depend on the entity: *shape* and *weight* are usually taken as qualities of physical endurants,[9] *duration* and *direction* as qualities of perdurants. An individual quality, e.g., the weight of the car you are driving, is a quality associated with *one and only one* entity; it can be understood as the particular way in which that entity instantiates the general property "having weight". For example, the endurant Hammer_#321 (a token) has its own individual instantiation of property "having weight", namely, the individual weight-quality of Hammer_#321.

The change of an endurant in time is explained through the change of some of its individual qualities. For example, with the substitution of a component, Hammer_#321 may change its weight. This means that the individual weight-quality of this entity was first associated to (or classified in) a position p and later to (in) a position q of a given weight-quality space. Note that p and q should not be considered weight measures like, say, $5kg$. 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 measure units. Thus, the same p may be associated to $5kg$ in one measure space and to $11.1lb$ in another. Quality spaces are elements of the REGION category, a subcategory of ABSTRACT.

DOLCE's taxonomic structure is pictured in figure 2. Each node in the graph is a category of the ontology. A category is a subcategory of another if the latter occurs higher in the graph and there is an edge between the two. PARTICULAR is the top category. The class of subcategories of a given category forms a partition except where dots are inserted.

Another important relation for our work in this paper is the *parthood* relation: "x is part of y", written: $\mathsf{P}(x,y)$, with its cognates the *proper part* (written: $\mathsf{PP}(x,y)$) and *overlap* relations (written: $\mathsf{O}(x,y)$). In DOLCE the *parthood* relation applies to pairs of endurants (e.g., one endurant is part of another) as well as to pairs of perdurants (to state that an event is

[9] In DOLCE they are called physical qualities and are denoted by means of the PQ predicate.

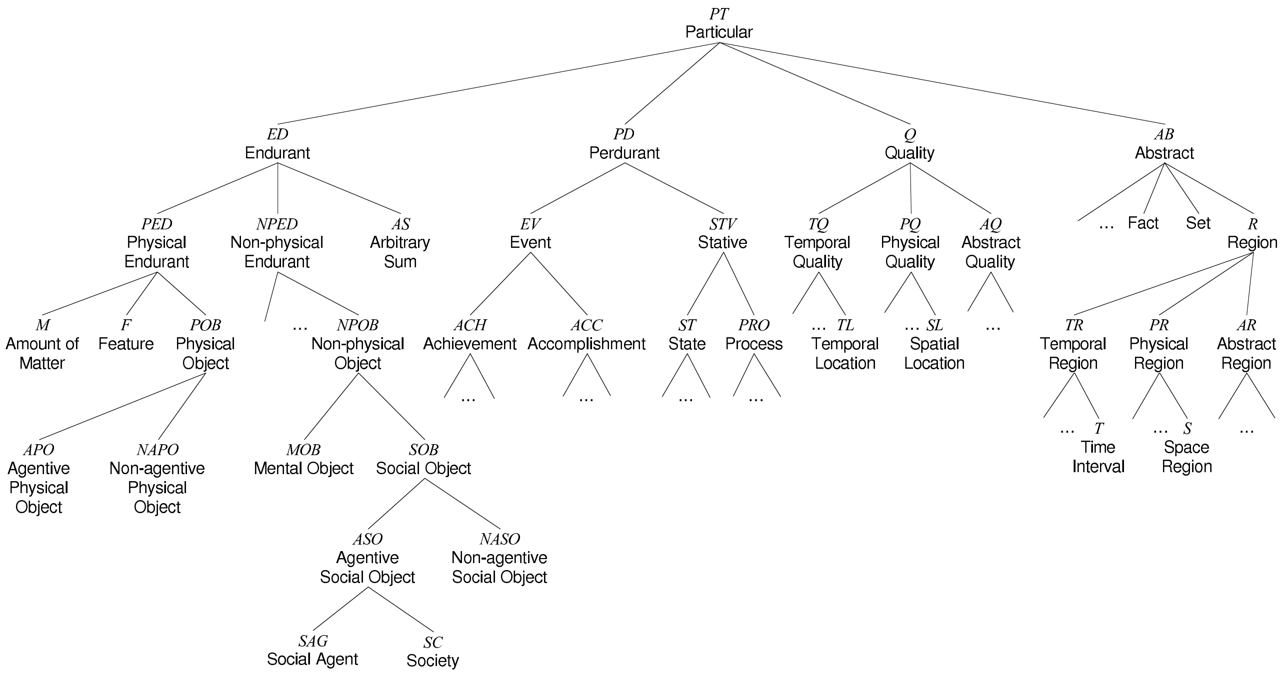

Figure 2 DOLCE taxonomy of entities

part of another). For pairs of endurants, the relation of parthood is temporalized (i.e., with the third argument for time objects) since an endurant may loose and gain parts throughout its existence.

4. FORMALIZING FUNCTIONAL REPRESENTATION FUNCTIONS

4.1. Behaviors in functional representation

Let x be a technical artifact, formally $\texttt{TechArt}(x)$. Let y be a perdurant, that is, $\mathsf{PD}(y)$. We take the behavior z of x in a perdurant y to be *the specific way in which x occurs in y*. In this view, behavior depends on the chosen x and y and is seen as a qualification of the participation relation. For instance, if x is a capacitor, then the way in which x occurs or exists in a given process of storing electric energy is a behavior of precisely *this capacitor x*.[10]

Our definition of behavior links behaviors with two categories of entities: endurants and perdurants. Formally, we take behavior z to be a new kind of individual quality: it does not hold for a single endurant or perdurant, but for pairs of an endurant and a perdurant.[11] In this way, it captures the special relationship between the artifact and the event in which it 'behaves'. 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) because they are associated to different entities z. To formalize this relationship, we introduce a ternary relation

$$\texttt{Beh}(x, y, z) \tag{1}$$

which reads "z is the behavior of the technical artifact x in event y" and is taken as primitive in our theory.

Looking at engineers' activity, we need to distinguish different kinds of behavior. While an 'actual behavior' of the artifact is what it actually does during (a part of) its life, the more general notion of 'possible behavior' deals with what the artifact can possibly do. A pen may be destroyed before it happens to write, still the pen could have participated to a writing event, i.e., writing is part of its behavior although not of its 'actual' behavior. Furthermore, although a pen may not possibly write due to a design flaw, still

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

[11] The ontological classification of 'behavior' is still an open problem in the literature. As far as we know, our approach is new.

engineers (not aware of the flaw) talk about its writing behavior. To make room for these cases in design activity, perdurants are divided in actual, physically possible, and physically impossible (yet logically meaningful). The category $\mathtt{GPD}$ of *generalized engineering perdurants* is the class of all perdurants which can be relevant to engineers regardless of their physical possibility. The categories $\mathtt{APD}$ and $\mathtt{EPD}$ collect those that are actual and physically possible, respectively. Then, $\mathtt{APD} \subseteq \mathtt{EPD} \subseteq \mathtt{GPD} \subseteq \mathsf{PD}$.

Having to deal with possible perdurants, we use the notion of *coherent perdurants* to indicate perdurants that live in the same possible world (cf. [3]). Using coherence and a minimality condition on the class $\mathcal{C}_x$ of perdurants to which an artifact x participates, one can define the actual (possible, generalized) life of x. The function (in the mathematical sense) $\mathtt{Alf}(x)$ gives the perdurant y which is the actual life of x, essentially the fusion of the actual perdurants which are mereologically minimal (with respect to space) in $\mathcal{C}_x$. In [3] we specialized behavior into *actual*, *possible*, *impossible*, and *general behavior* by considering the way in which an artifact x occurs in an actual, possible, (engineering) impossible, and generalized engineering perdurant, respectively.

To incorporate dependence on beliefs, needed to model Chandrasekaran and Josephson's notions of behavior, two further notions of agent-related perdurants are introduced. Given a group of agents $\mathcal{G}$, we isolate the *$\mathcal{G}$-possible perdurants* using the predicate $\mathtt{PD}_{\mathcal{G}}(y)$ whose meaning is "$\mathcal{G}$ believes that y is a possible perdurant" and that we partially characterize in [3]. Then, a generalized engineering behavior believed possible by $\mathcal{G}$, called a *$\mathcal{G}$-behavior*, is defined by (formally, $\triangleq$):

$$\mathtt{Beh}_{\mathcal{G}}(x, y, z) \triangleq \mathtt{Beh}(x, y, z) \land \mathtt{PD}_{\mathcal{G}}(y). \tag{2}$$

4.2. Functions in functional representation

Besides distinguishing different meanings of behavior, Chandrasekaran and Josephson [8] define the notion of artifact function. They presuppose a theoretical perspective in which artifact functions are construed as intended behaviors and define two concepts: *device-centric* function and *environment-centric* function. In this subsection we show to what extent the ontological approach outlined above is suitable for grasping these concepts.

In order 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 said in [8] that a *behavioral constraint* in X is any constraint on the behaviors of the elements of X. As the examples given in [8] suggest, a behavioral constraint may be absolute, i.e., unconditional (e.g., that the value of output voltage is greater than 5 volts), or conditional (e.g., that if the input voltage is above 5 volts, the output voltage is a sinusoid). Call *behavioral environment* an object x, written $\mathtt{BehEnv}(x)$, obtained as the sum of all the technical artifacts of interest in the environment; formallly, $\exists X(X \subseteq \mathtt{TechArt} \land x = \sigma y(y \in X))$, where σ is the mereological fusion operator. Then, we can formalize behavioral constraints in environment x as pairs of behaviors [3]: the first behavior plays the role of a constraint for the latter (in absolute constraints the two behaviors coincide). Let $\mathtt{Cond}(z, z')$ mean "perdurant z is a condition for perdurant z'", then a *behavioral constraint in environment x with condition (z, z')*, written $\mathtt{CrBeh}(x, z_0, z_1)$, is defined by formula:

$$\begin{aligned}&\mathtt{BehEnv}(x)\land \qquad (3)\\ &(z_0 = z_1 \rightarrow \exists x', y\,(\mathsf{P}(x', x) \land \mathtt{Beh}(x', y, z_0))) \land\\ &(z_0 \neq z_1 \rightarrow \exists x', x'', y, y'\,(\mathsf{P}(x', x) \land \mathsf{P}(x'', x) \land\\ &\mathtt{Cond}(z_0, z_1) \land \mathtt{Beh}(x', y, z_0) \land \mathtt{Beh}(x'', y', z_1)))\end{aligned}$$

The formula states that if the constraint is unconditional ($z_0 = z_1$), then z_0 is the behavior of an element in x. If the constraint is conditional both z_0 and z_1 are behaviors of two (possibly equal) elements of x. Now it is easy to define the predicate *satisfied behavioral constraint*, written $\mathtt{SatCrBeh}$, to indicate objective regularities in engineering possible perdurants.

The notion of device-centric function is described as follows:

> "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." ([8], p. 172)

Here D plays the role of the environment X and is seen as a single object (from which our definition of behavioral environment). To model the function we lack only a notion of *behavioral constraint desired by* an agent. The predicate $\mathtt{DES}_{\mathcal{G}}(x, z, z')$, meaning that "the behavioral constraint (z, z') in environment x is desired by an agent $\mathcal{G}$", is introduced for this reason (in [3]) we discuss the technical aspects of this predicate). Finally, we reach the sought definition:

$$\mathtt{DevFunc}_{\mathcal{G}}(x, z, z') \triangleq \quad (4)$$
$$\mathtt{SatCrBeh}(x, z, z') \wedge \mathtt{DES}_{\mathcal{G}}(x, z, z').$$

The environment-centric notion of function requires also the notion of *mode of deployment* for an artifact: it consists of (i) the structural relationships between the artifact and the objects in the environment and (ii) the actions in which the artifact and these objects are involved. Since these aspects are captured by perdurants, a mode of deployment for an artifact x in an environment x' is simply any generalized perdurant y such that there exists x_1 with $\mathsf{P}(x_1, x')$ and $x \neq x_1$ and both x and x_1 participate in y, formally:

$\mathtt{MD}(y, x, x') \triangleq \mathtt{TechArt}(x) \wedge \mathtt{BehEnv}(x') \wedge \mathsf{P}(x, x') \wedge \exists x_1(\mathsf{P}(x_1, x') \wedge x \neq x_1 \wedge \mathtt{PC_{WH}}(x, y) \wedge \mathtt{PC_{WH}}(x_1, y))$

where $\mathtt{PC_{WH}}(x, y)$ means that x participates in the whole perdurant y. Then a *feasible mode of deployment*, FMD, is an (engineering) possible perdurant which is a mode of deployment. From [8]:

"Let F be a class of behavioral constraints that an agent, say A, desires or intends to be satisfied 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." (p. 171)

Now fix a causal relation CS and specialize it to a satisfied behavioral constraint over x, z, z' caused by a perdurant y to obtain predicate

$\mathtt{CS}_{MD}(x, y, z, z') \triangleq \mathtt{SatCrBeh}(x, z, z') \wedge \forall x', x'', y', y''(\mathtt{Beh}(x', y', z) \wedge \mathtt{Beh}(x'', y'', z')) \rightarrow (\mathtt{CS}(y, y') \wedge \mathtt{CS}(y, y''))$

We can finally formalize the environment-centric function as follows:

$$\mathtt{EnvFunc}(z, z', x, x', y) \triangleq \mathtt{CS}_{MD}(x, y, z, z') \wedge \quad (5)$$
$$\mathtt{FMD}(y, x, x') \wedge \exists \mathcal{G}\, \mathtt{DES}_{\mathcal{G}}(x', z, z').$$

The ontological framework sketched above has been used in [3] to formalize the five meanings of behavior and their examples, described by Chandrasekaran and Josephson [8].

Here, for reason of space we limit ourselves to formalizing one example taken from Dretske [9]. 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 the 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 which connects the furnace to the thermostat and the furnace ignites. The event sequence can be illustrated in the following way:

(I) Temperature drops to $17°C$,
(II) Strip bends to angle A,
(III) Switch closes,
(IV) Current flows to furnace,
(V) 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." ([9], p. 86) In our framework we represent the thermostat behavior as $\mathtt{Beh}(\mathrm{thermostat}, \mathtt{e}, \mathtt{b})$ where b is the general behavior of the thermostat for the event e corresponding to the sequence (I)–(V) above. If we want to model the behavior for a subevent $\mathtt{e}'$, say (III), we write $\mathtt{Beh}(\mathrm{thermostat}, \mathtt{e}', \mathtt{b}')$ where $\mathtt{e}'$ is "Switch closes". Instead, the behavior of the switch at $\mathtt{e}'$ is introduced by writing $\mathtt{Beh}(\mathrm{switch}, \mathtt{e}', \mathtt{b}'')$.

5. FORMALIZING FUNCTIONAL BASIS FUNCTIONS

We now move to the second approach which, as we saw earlier, rely on a very different framework of concepts.

5.1. Design requirements

A key concept in the formalization of FB is the notion of *design requirements*. This notion is used by Pahl and Beitz ([18], Chapters 4.2 and 5) in a general sense comprising explicit and implicit requirements; it covers topics as different as demands and wishes of the user, economic feasibility, safety and performance requirements, and so on. However, when focusing on functional modeling, as in the work of Stone and Wood, the important requirements are just those that constrain the notion of engineering function broadly considered.

We write $\mathtt{CustNeeds}(x, y)$ to indicate that product function x and the device y[12] satisfy the customer

[12] In our formalization of FB functions, we speak about devices rather than about artifacts. Stone and Wood [19] themselves use the concepts of artifact, device, and product side-by-side, with-

needs [19], and $\texttt{Req}(x, y, v, z)$ to mean that the design requirements are satisfied by the engineering function x and device y with an input flow v and an output flow z. Note that this relation is ontological and not epistemological since y has no actual role. In the functional design process, as envisioned in the functional basis approach, the device is simply ignored.[13]

Since the customer needs are mandatory for the design requirements, we have:

$$\texttt{Req}(x, y, v, z) \to \texttt{CustNeeds}(x, y). \tag{6}$$

We constrain the notion of customer needs even further so that they are consistent and make sense from the FB perspective:

$$\texttt{CustNeeds}(x, y) \to \exists v, z\ \texttt{Req}(x, y, v, z). \tag{7}$$

Design requirements and functions

We say that a perdurant x is an engineering function with input flow y_1 and output flow y_2 if there is some device z which participates to the whole perdurant x such that $\texttt{Req}(x, z, y_1, y_2)$.

$$\texttt{EngFun}(x, y_1, y_2) \triangleq \exists z \texttt{Req}(x, z, y_1, y_2). \tag{8}$$

An engineering function *simpliciter* is a perdurant for which there exist input and output flows that satisfy definition (8). Formally,

$$\texttt{EngFun}(x) \triangleq \exists y_1, y_2 \texttt{EngFun}(x, y_1, y_2). \tag{9}$$

Among all functions the FB model distinguishes a certain set of functions which are called subfunctions. As the term "sub-function" suggests an implicit relationship to functional decomposition, we will use the more neutral term "basic function" to denote FB sub-functions.

$$\texttt{BasicFunc}(x, y, z) \to \texttt{EngFun}(x, y, z).$$
$$\texttt{BasicFunc}(x) \to \texttt{EngFun}(x). \tag{10}$$

FB is not very clear on its classification of functions. However, it seems coherent with FB to assume that all and only non-basic engineering functions are product functions.

$$\texttt{ProdFunc}(x, y, z) \triangleq \texttt{EngFun}(x, y, z) \land \neg\texttt{BasicFunc}(x, y, z). \tag{11}$$

$$\texttt{ProdFunc}(x) \triangleq \texttt{EngFun}(x) \land \neg\texttt{BasicFunc}(x). \tag{12}$$

Design requirements and flows

FB partitions flows in three disjoint types: material, energy, and signal (cf. [18], pp. 29-30). This classification may be called ontological because it accounts for what flows *are* as we will see later. In the perspective of function descriptions, instead, we have seen that a (general) flow can participate to functions as input or as output flow. In other terms, the function at stake provides the contextual perspective to classify flows as input or output flows. Again, we can define general flows via the notion of design requirements:

$$\texttt{InFlow}(z, x) \triangleq \exists y, v \texttt{Req}(x, y, z, v). \tag{13}$$

$$\texttt{OutFlow}(v, x) \triangleq \exists y, z \texttt{Req}(x, y, z, v). \tag{14}$$

where $\texttt{InFlow}(w, x)$ and $\texttt{OutFlow}(w, x)$ mean that w is an input (output, respectively) flow for the engineering function x.

The correlation between the notions of input/output flow and that of function is now obvious:

$$\texttt{EngFun}(x, y, z) \leftrightarrow (\texttt{InFlow}(y, x) \land \texttt{OutFlow}(z, x)). \tag{15}$$

Since FB classifies flows in types, we enforce that any flow entity must belong to one of the allowed types, namely, material, energy, or signal flow:

$$\texttt{Flow}(x) \leftrightarrow (\texttt{MFlow}(x) \lor \texttt{EFlow}(x) \lor \texttt{SFlow}(x)). \tag{16}$$

Of course, types must be distinct:

$$\texttt{MFlow}(x) \to \neg(\texttt{EFlow}(x) \lor \texttt{SFlow}(x)) \tag{17}$$
$$\texttt{EFlow}(x) \to \neg(\texttt{MFlow}(x) \lor \texttt{SFlow}(x)) \tag{18}$$
$$\texttt{SFlow}(x) \to \neg(\texttt{MFlow}(x) \lor \texttt{EFlow}(x)) \tag{19}$$

Design requirements and devices

As discussed, FB is not clear on its basic assumptions. This is the case even when the notion of device is used. Here we define a device as an object involved in a given context of design requirements remarking, in this way, the centrality of the Req relationship:

$$\texttt{Dev}(y) \triangleq \exists x, z, v \texttt{Req}(x, y, z, v). \tag{20}$$

One can then express the fact that the design requirements stipulate that a certain device performs a certain function:

$$\texttt{Perform}(x, y) \triangleq \exists v, z \texttt{Req}(y, x, v, z). \tag{21}$$

out indicating whether there are differences between these concepts.

[13] To highlight this fact, we could simply write $\texttt{Req}(x, v, z)$ and introduce an existential quantifier in formula (6). This choice would, however, hide the distinction between what is requested (Req) and what is realized (EngFun). There are different ways to make the distinction and the introduction of a reference to device y in Req is perhaps the simplest.

5.2. Functional basis and DOLCE

Ontological categorization of FB functions

As anticipated before, the engineering functions listed in FB are here interpreted as types whose instances are perdurants of certain kinds.

$$\texttt{EngFun}(x) \rightarrow \mathsf{PD}(x). \tag{22}$$

The class of perdurants used by FB is not arbitrary and can be characterized at least to some extent. The proposed match between engineering functions and a subclass of perdurants allows us to provide a coherent and ontological framework for the three notions used in FB: function, sub-function, and product function. For instance, given our assumption that an engineering function is the most basic in the ontological sense and is formalized by means of a class of perdurants, an FB sub-function is modeled as the subclass collecting all and only the perdurants that have as participants the flows requested in the (often informal) engineering description of the sub-function.

Given axiom (22) and Definition (8) it seems natural to constrain that an entity performing a function must participate to the whole perdurant:

$$\texttt{Perform}(x, y) \rightarrow \mathsf{PC}_\mathsf{T}(y, x). \tag{23}$$

Ontological categorization of FB flows

For obvious reasons it is not possible to assign the general notion of flow (i.e., Flow) to any specific ontological category of DOLCE. The following three conditions make clear the ontological categories of FB *flow kinds*. They state that material flows are (a subclass of) endurants, e.g., a screw, that energy flows are (a subclass of) physical qualities of endurants, e.g., pressure, and that signal flows are (a subclass of) perdurants, e.g., visual signal:

$$\texttt{MFlow}(x) \rightarrow \mathsf{ED}(x). \tag{24}$$

$$\texttt{EFlow}(x) \rightarrow \mathsf{PQ}(x). \tag{25}$$

$$\texttt{SFlow}(x) \rightarrow \mathsf{PD}(x). \tag{26}$$

Our full formalization extends the above axioms and definitions with a number of additional constraints whose aim is to tie down the formal meaning of the concepts we use (cf. [5]).

The above classification of flows needs some discussion. By taking energy flows as physical qualities we may counter the engineering practice to see energy or energy flows as a separate ontological entity that exists or occurs independent of other entities. The mere fact that in FB engineers speak about energy flows that are distinct of the matter and signal flows illustrates this practice. Yet, also in engineering it is acknowledged that energy flows in reality do not come separately. Pahl and Beitz ([18]; p. 30) confirm that the conversion of electricity in a power plant is associated with a material conversion, although that latter conversion may not be visible in a nuclear power plant. We choose to see energy flows as qualities since this gives a clear ontological motivation of the flow distinction embraced by FB. If one insists on having energy flows as a separate category, this would lead to a revision of DOLCE or to the adoption of a different ontology.

For completeness, we point out an alternative option which is compatible with DOLCE: to model energy as an entity on a par with entities that are seen as ‘amounts’ like water and air. This view has some awkward consequences like making independent the change of energy and the change of water location when water is moving down a waterfall. Perhaps, the uneasiness that some engineers show with this ontological classification is due to the constraint of looking at flows as objects: flows of energy are intrinsically associated by some to processes or events which have very different ontological properties. The whole issue is anyway subtle and might require some further discussion. Here we recognize the different positions while deciding to stay close to the view acknowledged by Pahl and Beitz. In FB this position means to take energy as a primary flow, which presupposes a second carrier flow that has the purpose to transport the primary flow [17].

Also, the claim that signal flows are perdurants should be explained. Clearly, not all perdurants are signal flows. Generally speaking, in engineering a signal flow refers to the presentation of data (geometric information, instruction list, etc.) and is typically provided by static perdurants or processes (a light is on, an increasing sound is emitted, a figure is shown, a flowchart is depicted in steps etc.). The actual interpretation of the information (a light on means the temperature is reached, a sound means the device is malfunctioning, a figure means that the output port is on the back etc.) is a much more complicated perdurant and is not modeled by the FB approach.

6. DISCUSSION

The formalizations given in the two previous sections show that an ontological analysis of the concept of function as used in engineering leads to highlighting the essential elements and, subsequently, to their logical expressions. Yet it should be acknowledged that these results are about the theoretical side of the project we described in the introduction; empirical proofs, consisting of validations of the practical usefulness of the formalizations in engineering are still to be given. This validation may be given after addressing a number of further issues raised by the two formalizations. For instance, if by a comparison of the two formalizations, precise rules can be given for the translation of functional descriptions given in the FR approach into functional descriptions given in the FB approach, and *vice versa*, then one would obtain clear results by which the usefulness of this work can be determined. More generally we see a number of further issues to be addressed.

A first issue is whether we have sufficiently covered the domain of engineering function descriptions by formalizing FR and FB functions. Second, the two formalizations should be compared in detail, in order to analyze their differences also from the ontological perspective. Third, it is still unclear how our formalizations relate to the one developed by Mizoguchi and Kitamura in [12] and [13], and to the one given by Arp and Smith [1].

The focus in our work on engineering meanings of the concept of function forces us to consider the first issue of "completeness". We have relied on Chandrasekaran [7] when presenting the Functional Representation and the Functional Basis approaches towards functions as approaches that indeed cover the main part of engineering. Yet it should be acknowledged that in principle 18 engineering concepts of function can be distinguished [10]. These 18 concepts may be sufficiently similar to single out a more limited number of archetypical engineering approaches. Yet the number of archetypes can be larger than two, as is, for instance, exemplified by the analysis of functions in [21] in this volume. Conversely, it may be argued that we have been too elaborated by formalizing two concepts of functions. Chandrasekaran [7], for instance, makes a distinction between the FR and FB approaches but also conjectures that research within these approaches is converging. Similarly, in the survey in [10] it is suggested that the 18 meanings identified may be partly integrated. And in [21] in this volume, it is argued that one can define one overall account of function that captures at least three different engineering meanings of functions. To some extent, the answer is that we are not in a position to determining whether the ontological analysis of functions will lead to a couple, 18, or eventually only one concept of function. Further work is needed to understand the key element behind each proposed perspective.

The second issue of comparing the different formalizations is, as already said, important to the validation of our work. The separate formalizations may already provide the means to develop computer-supported tools for supporting functional descriptions in engineering. But by also comparing the formalizations, tools for translating FR functional descriptions to FB functional descriptions and *vice versa*, could be developed, thus relieving the burden of the use of multiple meanings of functions in engineering.

The evaluation of the third issue, the relationship of our work with that of Mizoguchi and Kitamura, depends on how one considers the new formal concepts that ontological analyses may introduce into engineering. If these new concepts are again engineering meanings of functions, as is arguably the case for Mizoguchi and Kitamura, then comparing our work with that of Mizoguchi and Kitamura is needed to address the second issue just addressed. If, however, these new concepts are not engineering meanings, as may be the case with the one defined by Arp and Smith, then the comparison should be carried out on different grounds. The motivation may rest on noting that in the work of Arp and Smith coherence between the concepts of *biological* function and *technical* function is central, which would mean that by comparing our formalizations with the one by Arp and Smith, the differences between FR functions, FB functions, and biological functions can be determined. These differences may then be relevant to engineering fields in which biology plays a role, like biotechnology and biomimetic engineering.

7. CONCLUSIONS

The term "function" is associated in engineering with various and sometimes conflicting meanings, as the survey [10] of the 18 different engineering concepts of function seems to imply. One consequence of this variety is that in engineering a communication problem arises when engineers exchange functional descriptions using different meanings. Ontological analysis can provide a solution to this communication problem: by laying down the different meanings within a single foundational ontology, the different concepts can be made explicit enabling automated functional reasoning using one meaning, and enabling automated translation of functional descriptions using different meanings for the term. But it is acknowledged that this solution still has a promissory character.

A first concluding remark is that as described in this paper, ontology provides more strategies to deal with the problem. We advanced the strategy to formalize existing concepts of function but alternatively one can introduce a new formalized concept of function, for relating the existing engineering concepts or for replacing them. Which strategy will be most successful is not a topic we considered, yet we can note that the third strategy may become in conflict with engineering practices when it is beneficial for engineering to use the concept of function with more than one meaning. Function may be a term that has a flexible meaning for a good reason ([21] in this volume), or it may be a Wittgensteinian family resemblance concept, in the sense that it has a set of similar meanings that exist side-by-side in engineering.

A second concluding remark is that the work presented in this paper within the strategy of formalizing existing concepts of function is not completed. Other concepts that the two advanced in the Functional Representation and the Functional Basis approaches may need to be formalized, and the presented formalizations may need to be developed to more detail on the basis of engineering literature on functions not considered in this paper. Moreover, the number and nature of practical illustrations of our formalizations, show that the translation of our ontological analyses to applications that are useful on the engineering work floor is still in an initial phase. Results presented in this paper are therefore still primarily on the conceptual level of explicating the various meanings engineers attach to the concept of technical function; their potential to solve the communication problem for functional descriptions in engineering is conjectured but still has to be substantiated by practical application.

ACKNOWLEDGMENT

Research by Pieter Vermaas for this contribution is supported by the Netherlands Organization of Scientific Research (NWO).

REFERENCES

[1] Arp, R., and Smith, B., (2008), "Function, Role, and Disposition in Basic Formal Ontology", Proceedings of Bio-Ontologies Workshop (ISMB 2008), Toronto, Canada, pp. 45–48.

[2] Borgo, S., (2007), "How Formal Ontology can help Civil Engineers"; in Ontologies for Urban Development, ed. by Teller, J., Lee, J., and Roussey, C., Springer, pp.37–45.

[3] Borgo, S., Carrara, M., Garbacz, P., and Vermaas, P.E., (2009), "A Formal Ontological Perspective on the Behaviors and Functions of Technical Artifacts", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 23(1), 3–21.

[4] Borgo, S., Carrara, M., Garbacz, P., and Vermaas, P.E., (2009), "Steps Towards a Formal Ontology for Engineering Functions of technical artifacts"; *paper at the ICCME conference, 2009, Salzburgh*.

[5] Borgo, S., Carrara, M., Garbacz, P., and Vermaas, P.E., (2009), "Towards an Ontological Representation of Functional Basis in DOLCE"; in Interdisciplinary Ontology, Vol.2, ed, by Okada, M., and Smith, B., Open Research Centre for Logic and Formal Ontology, Keio University, pp.3–16.

[6] Borgo, S., Carrara, M., Garbacz, P., and Vermaas, P.E., (2010), "An Ontological Interpretation of the Functional Basis Approach", *manuscript*.

[7] Chandrasekaran, B., (2005), "Representing Function: Relating Functional Representation and Functional Modeling Research Streams", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 19, 65–74.

[8] Chandrasekaran, B., and Josephson, J.R., (2000), "Function in Device Representation", Engineering with Computers, 16(3/4), 162–177.

[9] Dretske, F., (1988), Explaining Behavior, MIT Press.

[10] Erden, M.S., Komoto, H., VanBeek, T.J., D'Amelio, V., Echavarria, E., and Tomiyama, T., (2008), "A Review of Function Modeling: Approaches and Applications", Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 22, 147–169.

[11] Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S., and Wood, K.L., (2002), "A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts", Research in Engineering Design, 13, 65–82.

[12] Kitamura, Y., Koji, Y., and Mizoguchi, R., (2006), "An Ontological Model of Device

Function: Industrial Deployment and Lessons Learned”, Applied Ontology, 1(3-4), 237–262.

[13] Kitamura, Y., and Mizoguchi, R., (2004), “Ontology-Based Systematization of Functional Knowledge”, Journal of Engineering Design, 15(4), 327–351.

[14] Kitamura, Y., Takafuji, S., and Mizoguchi, R., (2007), “Towards a Reference Ontology for Functional Knowledge Interoperability”; Proceedings of the ASME 2007 IDETC/CIE conference, September 4-7, Las Vegas, Nevada, USA, DETC2007-35373.

[15] Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., and Schneider, L., (2002), “Wonderweb Deliverable d17: The Wonderweb Library of Foundational Ontologies”, *technical report*.

[16] McAdams, D., Stone, R., and Wood, K., (2000), “Functional Interdependence and Product Similarity Based on Customer Needs”, Journal of Research in Engineering Design, 11, 1–19.

[17] Nagel, R.L., Bohm, M.R., Stone, R.B., and McAdams, D.A., (2007), “A Representation of Carrier Flows for Functional Design”; Proceedings of ICED’07, August 28-31, Paris, France.

[18] Pahl, G., Beitz, W., Feldhusen, J., and Grote, K.H., (2007), Engineering Design. A Systematic Approach, 3rd edition, Springer.

[19] Stone, R.B., and Wood, K., (2000), “Development of a Functional Basis for Design”, Journal of Mechanical Design, 122(4), 359–370.

[20] Szykman, S., Racz, J., and Sriram, R., (2000), “The Representation of Function in Computer-Based Design”; Proceedings of the 1999 ASME IDETC/CIE Conferences, September, Las Vegas, Nevada, USA, DETC99/DTM-8742.

[21] Vermaas, P.E., (2010), “Technical Functions: Towards Different Practical Meanings and one Theoretical Account”; In this volume.