Formal Ontology in Information Systems
R. Ferrario and W. Kuhn (Eds.)

doi:10.3233/978-1-61499-660-6-241

Formalizing and Adapting a General Function Module for Foundational Ontologies

Stefano BORGO [a,1], Riichiro MIZOGUCHI [b] and Yoshinobu KITAMURA [c]
[a] *Laboratory for Applied Ontology, ISTC CNR, Trento, Italy*
[b] *Research Center for Service Science, JAIST, Nomi, Japan*
[c] *College of Information Science & Engineering, Ritsumeikan University, Shiga, Japan*

Abstract. We introduce and formalize the key elements of a recent approach [1] to function definition that covers both biological and artefact functions. The paper concentrates on the steps from informal definitions to formal expressions and aims to show how to develop an ontological module for generic function representation. This part is done assuming the YAMATO ontology as background system. The second goal of the paper discusses how the module could be adapted to other foundational ontologies. In this part, we first point out the ontological assumptions on which the module relies and then discuss problems in reformulating them in other systems. This step is needed to verify whether and how the function module can be meaningfully used in other ontologies. We exemplify this case by discussing BFO and DOLCE, and by formalizing the module in the latter ontology.

Keywords. function, foundational ontology, behavior, process, system, BFO, YAMATO, DOLCE

1. Introduction

The power of foundational ontologies is in their capacity to guide a coherent conceptualization of the real world. To achieve this, they need to include cross-disciplinary notions, like function and artefact, that are necessary in modeling domains like engineering, biology and medicine. Cross-disciplinary notions can be hard to capture since each domain may provide a different characterization. When this happens, one usually searches for a unifying notion that can classify the other views. For instance, this happesn with notions like function and artefact across engineering and biology [1,2,3,4,5,6]

This paper concentrates on the definition of function presented in [1,6], because it gives an ontology-based unified definition of function which covers both biological and artefact functions. The goal is to turn this approach to function modeling into a framework for ontologists. We approach this goal in two steps: first we formalize the approach using the YAMATO ontology [6,7]; second, we discuss its incorporation into other foundational ontologies. The latter goal provides an original and challenging translation problem: instead of formalizing the categories of an ontology in terms of another ontology, we

[1]Corresponding Author: Stefano Borgo, E-mail: stefano.borgo@cnr.it

formalize a notion taken from application domains in terms of ontologies with different ontological commitments.

Structure of the paper. After an example on biological and artefact functions and a presentation of how functions can be understood, Section 3 informally introduces the core notions of [1] and Section 4 provides a formalization of these. This part is completed in Section 5 by giving the general notion of function. Section 6 discusses the ontological assumptions behind this system and investigates reformulations in terms of BFO [8,9] and DOLCE [10]. A system in DOLCE is partially formulated in the next section.

2. What are Functions?

A classical example of biological function is that of the human heart [11,12] which has the function to pump blood throughout the body via the circulatory system. An example of artefact function discussed in the literature is the function of a heat exchanger [13]. A heat exchanger has the function to cool or heat by transferring heat between two fluids. In both cases, we talk about the function of an object in a system: the heart connected to the human circulatory system, the device connected to two fluid circulatory systems. Also, in both cases we point to a goal of the function: the tissues have to be supplied with blood rich of oxygen and nutrients while the blood rich of carbon dioxide and other wastes must be taken away from the tissues; the temperature of one fluid system has to be reduced and the temperature of the other fluid system to be increased.

Many definitions of function have been proposed in engineering design, philosophy and ontology research [11,14,15,16]. Ontologically these definitions vary considerably. In many definitions in engineering design, function is directly associated with a *behavior* performed by an object in a system (e.g., [17,18]) or *effects* to the environment (e.g., [19]). On the other hand, in philosophy functions are typically regarded as special *properties (capacities)* of an entity [20], *dispositions* [8,21] or *traits* [22,23]. See [4] for a classification and comparison. The theory in [1] falls within the contribution theory, which relies on the causal relationship (e,g., about the oxygen of blood or the temperature of fluid) between an object and a system as many definitions in engineering design. The most important difference between this theory and other contribution approaches is that here the goal to achieve is determined in terms of a context, a step missing in both [24] and [20]. Also, functions are distinguished from capacities and dispositions.

In addition, much research has been carried out on the comparison of biological and artefact functions [25], and some attempts to unify them have been proposed [26]. The theory in [1] defines systemic function as the core notion for both artefact functions and biological functions. The definition satisfies the sets of desiderata in [22] for biological functions as well as those for artefact functions collected in [27].

3. A Unifying Notion of Function

The two examples of function in the previous section show an interesting similarity across biological and artefact function descriptions: both require a context and a goal. Typically one is not free to choose a context for a biological function: there is no designer of a biological organ and its function is somehow constrained in the needs of the overall system to survive. This is clearly not the case for artefact functions since a device does not have survival constraints and can always be used in ways not foreseen by the designer.

This sharp distinction and the relative difficulties in unifying biological and artefact functions require a suitable *formal framework* where functional assumptions can be formally reconstructed. This formalization is the first aim of the paper and is based on the work in [6,1] where the general definition has been motivated and analyzed.

3.1. Basic Categories and Relations for the Function Module

First, we take the core definitions from [1] and analyze them ontologically. The goal is to extract the core categories and relationships, which we then formalize within the YAMATO ontology [7].[2] In this step we observe what is actually needed for the definition in order to later be able to adapt the module to other ontologies.

We start from a set of primitive notions that are quite standard elements in an ontology: object (written `OBJCT`), event (`EVENT`), state (`STATE`) and description (`DESCRPTN`). These correspond to generic ontological notions and are not specifically constrained by a function theory. Of course, the precise choice of these notions is important. For instance, in an ontology that assumes that holes are a kind of object, one may claim that (some) holes perform a function. In ontologies that adopt the opposite stand, holes not being objects *per se* cannot perform functions. Nonetheless, in these latter cases one can still say that the object performs the function *due* to its hole-feature. A similar argument holds for events on issues like: can an event be instantaneous? Can an event be scattered in time? These choices affect some cases in which one may want to talk in terms of function but in general not the ontological stand of what a function actually is. Note that the participation relationship between objects and events is a primitive relation which is at the core of our ontological assumptions: an event in which a function is performed has two important participants, namely, the object performing the function and the system of which that object is part. Also, a state is often considered a kind of event in ontology but, strictly speaking, this is not necessarily so. All we need to assume is that a state is a temporal entity characterized by a property or relation, e.g., “being open”, ”being running” and “being attached to”. Thus, whether a state is a type of event or not is irrelevant to the approach. Finally, the notion of description we adopt is quite generic. Sometimes the corresponding category is called information object, i.e., the category which collects things like the content of a book, of a laboratory report or of a engineering blueprint.

In today’s ontologies the notions of agent or intentionality are often taken as primitive. The specific choice on these notions (or their reduction to others) is not important for our goals. What is needed in our framework is a way to discriminate between objects and agents as well as to talk about goals (states to be achieved or outcomes).

We add to this initial list the notions of process (`PROC`), role (`ROLE`), goal (`GOAL`) and context (`CONTXT`) which are more specific. We take the first from [28,29]. which see a process as a temporal entity characterized by instantaneous change.[3] Following [6], let us fix an event, a spatio-temporal entity, in which a person walks from A to B during a period of time $[t_0, t_1]$. While the event takes place, i.e. during the walk of the person from

[2] See also `http://www.ei.sanken.osaka-u.ac.jp/hozo/onto_library/upperOnto.htm`

[3] Note that the term ‘instantaneous’ refers to the change, not to the process identified by the change. The process itself is an evolving entity and at each point in time it coincides with the instantaneous change.

time t_0 to time t_1, the person undergoes a continuous change, namely, what the person experiences at each instant from t_0 to t_1. In natural language we use the expression <the person is walking> to indicate this ongoing (progressive) change." YAMATO calls it a process and opposes it to the walk event, the latter being a temporally extended entity, namely, the walk seen as a whole throughout $[t_0, t_1]$. It is this reading of the notion of process that allows us to treat them as object-like entities. An important consequence, as we will see in Section 4, is that processes as here understood can play roles. The second, the notion of role, is taken from [31], it extends the idea of roles to temporal entities as we will see later. The notion of goal is fairly standard although we consider it specialized since it relies on that of role: we take a goal to be a state to be achieved, i.e., a role played by a state. Note that we assume that all (non self-contradictory) states exist in the formal system. This allows us to talk about future states, and in particular about desired states, although most of them will never become actual. The last, the notion of context in general, is not yet well understood in the literature and we use just a minimal characterization (presented later). We will use this only to introduce a more specific notion, that of function context already discussed in [1,6]. All these notions have been partially formalized in [7] where the reader can also find some examples. Finally, another couple of categories we will need are those of system (SYSTM) and behavior (BEHVR). These are introduced later.

Table 1. Notions from YAMATO [30].

State	Time-indexed quality. Example: Feeling well at time T.
Goal	A state to be achieved.
Process	Temporal entity which evolves in time (ongoing) and exists as a whole at any time when it exists.
Event	Temporal entity which spans a temporal interval and that is only partially present at each point in the interval. An event is the global view of all the changes that identify one or more processes.
Role	Entity that is dependent on a context and can be played by another entity. Examples: a teacher (role played by an agent) or a goal (role played by a state). A role is said vacant if there is no player.

The approach uses a series of standard ontological relations like *is_a* (subcategory, only implicitly used in this paper), *instance_of* (instance of, also implicitly used), *parthood* and its cognates (written P, we write PP for proper parthood) and *participation* (between an object and an event, written PC). A relationship between states and objects is introduced (written STATEFOR) to claim that a given state is possible for the given object, providing a weak notion of modality for an object. We also include a relationship *describes* (between a description and an object which is described in it, written DESCRIBES). Some classical relations are here extended as a consequence of our modeling choices: for instance, the relation *plays* holding of an object and a role here applies also to a process and a role (as said, even processes can play roles in YAMATO). Another case is the parthood relationships: we will extend it on systems once the latter notion has been defined.

An important relation for this approach to function is called "causally contributes to" and applies to pairs of processes. Informally, "process P_1 contributes to a process P_2" (where P_2 is typically the process of achieving a certain state) when the happening of process P_1 is *a cause for* process P_2 to happen. For instance, the growing of blood clot in a blood vessel causally contributes to the process of reducing blood flow in that vessel. Note that there could be other processes causing (or co-causing) the reduction of the blood flow, e.g. the process of the vessel's cross section restricting or of the heart reducing its activity, but at least one of these processes must occur causally contributing to the first. Causality relations are notoriously complex to model since causality is not well understood. Sometimes the relationship is easy to establish (e.g. the contribution of the spinning of a drill's electric engine in the making of a hole) and harder in others (is the slight irregular spinning of the drill's engine causing my getting injured?). However, recall that we are interested in using the ontological relationship and not in addressing the epistemic problem of establishing when and where the relationship holds. Following the work done in YAMATO [7], we formally write $\mathtt{CCNTR}(P_1, P_2)$ to mean that "P_1 causally contributes to P_2". The remaining relations in the system will be introduced later as needed.

The overall ontological structure we use to define the function module is depicted in Figure 1. As said, the categories and relations should be understood depending on the underlying foundational ontology, in this case YAMATO, see Table 1. Finally, note that all categories in the paper are categories of particulars (individuals).

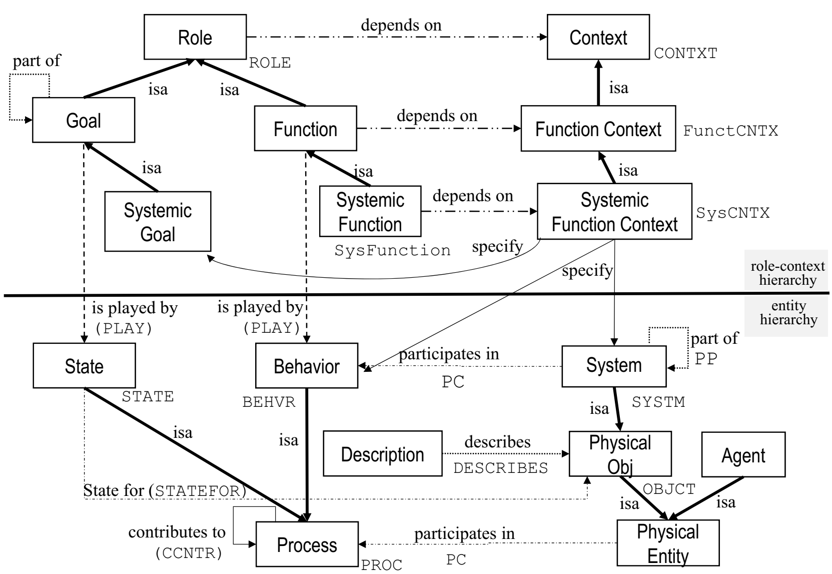

Figure 1. The core of the system used to define function in YAMATO. The upper-half shows role categories, the bottom-half the entity hierarchy. An isa-arrow from category C_1 to C_2 means "any instance of C_1 is an instance of C_2", while for the other relations R an R-arrow means "any $x \in C_1$ has R-relation with some $y \in C_2$". E.g., "any $x \in C_1$ participates in some $y \in C_2$" and "any $x \in C_1$ depends on some $y \in C_2$".

3.2. The Informal Definitions of the Function Module

We now report the basic definitions in [1] and highlight their important aspects. For lack of space, we cover only the necessary elements from that paper, e.g., we leave out the definition of design context and user context. Furthermore, where possible we simplify the original approach.

The starting point is the notion of system. This is a complex object enriched with a selection of some parts and relationships. The ontological characterization of systems is not important here, so we do not model systems ontologically and represent them as pairs of an object and a related description. The system is taken to be an individual (a token).

Definition 1 (System and components)**.** A *system* $\hat{O}$ is the combination of an object O and a description of O. The description identifies at least two distinct parts of O as well as some relationships between the parts and between O and these parts. The identified parts, called *components*, can be atomic or complex. An *atomic* component has no identified subparts, so it is not decomposed further. A *complex* component is itself a (sub-) system whose description is included in $\hat{O}$'s description. Every system is decomposed into atomic components and has at most a finite number of components.

Let us assume that the relationships between components and between components and their system are of two types: direct or indirect. The specific criteria for the distinction may depend on the ontological background (e.g., if one assumes holes are objects or features of objects). *Input* and *output* are two disjoint subclasses of the direct relationships which are regulated by physical laws. Since input and output are based on physical relationships and their identification is ontology dependent, we treat them as primitives. This allows us to define behavior:

Definition 2 (Behavior)**.** A behavior of an object O is the ongoing progressive change of the values of some O's qualities (these describe the evolution of the object's status), thus a process. More precisely, the behavior of O is the process of change of the values between some of the O's inputs and outputs.

Informally, a context is at least a set of objects with the description of some of their relationships (e.g., marking their distribution in a location). Here we take the notion as given provided it satisfies (or is enriched so to satisfy) the following constraints: a context C for object O is formally modeled as a tuple consisting of an entity C_Σ (the mereological sum of all the objects in C and called the *support* of C), an object O (also in C), a system $\hat{C}_\Sigma$ for C_Σ, and a state G (or set of states) of C_Σ. For example, in a cooling context the support is the sum of the heat exchanger artefact (a radiator), the connected pipes, pumps, fluids, heat generator(s) etc. Contexts can be nested. If x is part of the support of C we write $C_{\Sigma,x}$. Clearly, we have $C_{\Sigma,O}$ whenever C ia a context for O. (Finally, note that we use tuple as logical functions. In short, in the paper each tuple without open variables denotes an individual in the system.)

We specialize the notion of context, which is seen as an individual (a token) to function context:

Definition 3 (Function Context)**.** Given a context C for object O, a *function context C_f for O* is a 5-tuple whose first four arguments are those of the context C for O, and the fifth is a behavior B of O.

The *appropriateness* relationship is introduced between the behavior of a system and the behavior of one of its subsystems:[4]

Definition 4 (Context-Behavior appropriateness)**.** Let C_f be a a function context for O with behavior B. Let C'_f be a a function context for O' with behavior B'. We say that B is *appropriate for* B', if there exists a sequence of distinct function contexts $C_{f1}, C_{f2}, \ldots, C_{fn}$ with distinct objects O_i, and processes B_i, such that $C_f = C_{f1}, C'_f = C_{fn}$, and for $1 \leq i < n$ the object O_i is part of the object O_{i+1}, and the process B_i causally contributes to the process B_{i+1}.

Definition 5 (Systemic Function Context)**.** A *systemic function context C_s for O*, systemic context for short, is a function context C_f for O such that the goal G, specified by C, is a state for the support C_Σ.

Finally, we need two more auxiliary notions characterizing chains of nested systemic contexts.

Definition 6 (Direct Systemic Function Context)**.** A *direct systemic (function) context C* for O is a systemic context where the support's system $\hat{C}_\Sigma$ has no proper subsystem.

Definition 7 (General Systemic Function Context)**.** A *general systemic (function) context* is a finite sequence $C_1, \ldots, C_n$ ($C_i \neq C_j$) of direct systemic contexts such that the support system $\hat{C}_{\Sigma i}$ is the object O_{i+1} of C_{i+1}, i.e., $\hat{C}_{\Sigma i}$ is a component of the support system $\hat{C}_{\Sigma i+1}$ of the successor context C_{i+1}.
A *general context for O in S* is a general context $C_1, \ldots, C_n$ in which $O = O_1$ and $S = \hat{C}_{\Sigma n}$.

From the definition, let a steam generator system with the goal of generating steam be the context of a pump, and the electricity generator system with the goal of generating electricity be the context of the steam generator. Then, the sequence "steam generator system with the goal of generating steam" and "electricity generator system with the goal of generating electricity" form a general systemic context for the pump in the electricity generator system.

4. Formalizing the Basic Elements

In the previous section we have collected the basic ontological categories on which this approach to function modeling relies like object, process, goal and context. These anchor the function module to the rest of the ontology one is using, YAMATO in our case. We have also introduced dedicate notions (system, behavior, function context etc.) doing our best to highlight the important elements.

Here we propose a formalization. The goal is *not* to reach a satisfactory axiomatization. After all, any axiomatization falls short of being satisfactory when the notions one starts with are only partially characterized in the underlying ontology. Instead, the goal of the section is to give a framework for an axiomatization. We propose a *formal core* of a general function module that one can adapt to the target ontology and specify as needed.

[4]This relationship of appropriateness improves that in [1] by simplifying it.

4.1. The Formal Backbone Module

We start from a few basic constraints on roles limiting our attention to the categories that interest us here. For instance, we need some constraints on the new categories like behavior (a process) and on relations like play, since here a behavior (process) can play a role and a goal is a role played by a state.

$$\mathtt{BEHVR}(x) \rightarrow \mathtt{PROC}(x) \tag{1}$$

$$\mathtt{PLAY}(x,y) \rightarrow \mathtt{ROLE}(y) \wedge (\mathtt{OBJCT}(x) \vee \mathtt{PROC}(x)) \tag{2}$$

$$\mathtt{PLAY}(x,y) \wedge \mathtt{GOAL}(y) \rightarrow \mathtt{STATE}(x) \tag{3}$$

To make explicit our assumptions, we state the following (which holds in YAMATO):

$$\mathtt{STATE}(x) \rightarrow \mathtt{PROC}(x) \tag{4}$$

A context's support is an object and any object participates to a behavior:

$$\mathtt{CONTXT}(c) \rightarrow \mathtt{OBJCT}(c_\Sigma) \tag{5}$$

$$\mathtt{OBJCT}(x) \rightarrow \exists y.\ \mathtt{PC}(x,y) \wedge \mathtt{BEHVR}(y) \tag{6}$$

4.2. Formalizing the Functional Notions

Here we revisit the above definitions, propose a formalization and add further comments.

Definition 1): A system is a pair composed by an object and a description, and the latter identifies at least two distinct parts of the object:

$$\begin{aligned}\mathtt{SYSTM}(s) \equiv_{\text{def}} \exists o,d,x,y\ (s = <o,d> \wedge \mathtt{OBJCT}(o) \wedge \\ \mathtt{DESCRPTN}(d) \wedge \mathtt{DESCRIBES}(d,x) \wedge \mathtt{DESCRIBES}(d,y) \wedge \\ \mathtt{PP}(x,o) \wedge \mathtt{PP}(y,o) \wedge \neg \mathtt{P}(x,y) \wedge \neg \mathtt{P}(y,x))\end{aligned} \tag{7}$$

Following the informal notation of Def.1, we write $\hat{x}$ to indicate the system given by object x and an associated description, and write $\hat{x}_o$ for the object x itself and $\hat{x}_d$ for the description. Thus, we adopt these equivalent notations:
$\hat{x} \equiv <\hat{x}_o, \hat{x}_d> \equiv <x, \hat{x}_d>$.
This notation is natural when a context is fixed since in that case the association of the object with a description is given by the context (see below). We also extend the parthood relationship among systems by assuming:

$$\mathtt{P}(\hat{s},\hat{t}) \leftrightarrow \mathtt{P}(\hat{s}_o,\hat{t}_o) \wedge \mathtt{P}(\hat{s}_d,\hat{t}_d) \tag{8}$$

The parts of a system can be atomic or complex, i.e., subsystems. Note that we cannot characterize this form of atomicity within the first-order language but we can constraint it to some extent (recall that we write $\hat{x}$ only if x is an object):

$$\texttt{COMPONENTOF}(x,\hat{o}) \leftrightarrow (\texttt{PP}(x,\hat{o}) \vee (\texttt{PP}(x,o) \wedge \texttt{DESCRIBES}(\hat{o}_d,x))) \tag{9}$$

Here "$\texttt{PP}(x,o) \wedge \texttt{DESCRIBES}(\hat{o}_d,x)$" holds only if x is an atomic component, "$\texttt{PP}(x,\hat{o})$" when it is complex, i.e., a subsystem.

Definition 2): In Section 3.2 we introduced input and output as primitives since their characterization depends on the representation of properties and values in the underlying ontology. We now define the behavior of an object O as a behavior in which O's input and output participate:

$$\texttt{BEHVR}(b,o) \equiv_{\text{def}} \texttt{OBJCT}(o) \wedge \texttt{PC}(o,b) \wedge \exists x,y\ (\texttt{PC}(x,b) \wedge \texttt{PC}(y,b) \\ \wedge \texttt{INPUT}(x,o) \wedge \texttt{OUTPUT}(y,o)) \tag{10}$$

Definition 3): A function context is a 5-tuple in which there are a support, a system, the object of the system and a goal (these four forming a context) plus a behavior of the object. (Note that in the definition $s_\Sigma = s_o$.)

$$\texttt{FunctCNTX}(\bar{c},o) \equiv_{\text{def}} \exists s,g,b\ (\bar{c} =< s_\Sigma,o,s,g,b > \wedge \texttt{SYSTM}(s) \wedge \\ \texttt{COMPONENTOF}(o,s) \wedge \texttt{GOAL}(g) \wedge \\ \texttt{CONTXT}(< s_\Sigma,o,s,g >) \wedge \texttt{BEHVR}(b,o)) \tag{11}$$

Definition 4): The context-behavior appropriateness is not definable in first-order logic due to the arbitrary length of the sequences of function contexts. However, given a sequence of contexts $\vec{c} =< \bar{c}_1,\ldots,\bar{c}_n >$, with $\bar{c}_i =< c_{i_\Sigma},o_i,\hat{c}_i,g_i,b_i >$, the following states whether $\bar{c}_1,\bar{c}_n$ are appropriate, i.e., whether the context are distinct, the objects are linearly nested, and each behavior contributes to the behavior of the next larger context. Since the object are properly nested, the contexts must already be distinct, so we have:

$$\texttt{APPROPRT}(\vec{c}) \equiv_{\text{def}} \bigwedge_{1\le i<n} (\texttt{PP}(o_i,o_{i+1}) \wedge \texttt{CCNTR}(b_i,b_{i+1})) \tag{12}$$

Definition 5): A systemic context for an object is a function context for that object whose goal is a suitable state for the support of the context, that is, given $\bar{c} =< c_\Sigma,o,\hat{c},g,b >$

$$\texttt{SysCNTX}(\bar{c},o) \equiv_{\text{def}} \texttt{FunctCNTX}(\bar{c},o) \wedge \texttt{STATEFOR}(g,c_\Sigma)) \tag{13}$$

Definition 6): The notion of direct systemic context requires a direct relationship between the component and the system of the context, given $\bar{c} =< c_\Sigma,o,\hat{c},g,b >$

$$\texttt{DirCNTX}(\bar{c},o) \equiv_{\text{def}} \texttt{SysCNTX}(\bar{c},o) \wedge \\ \neg\exists x\ (\texttt{COMPONENTOF}(\hat{x},\hat{c}) \wedge \texttt{P}(o,x)) \tag{14}$$

Definition 7): It remains to define the notion of general systemic context. This is a sequence of contexts $\vec{c} =< \bar{c}_1,\ldots,\bar{c}_n >$ (with $1 \le n$) such that context at position $i+1$ is a direct systemic context of $\hat{c}_i$, i.e., the system for context at position i:

$$\mathtt{GenCNTX}(\vec{c},o) \equiv_{\text{def}} \mathtt{DirCNTX}(\bar{c}_1,o) \wedge \bigwedge_{i<n} \mathtt{DirCNTX}(\bar{c}_{i+1},c_{i_\Sigma}) \tag{15}$$

Furthermore, a general systemic context $\vec{c}$ for object o and system s, is defined by

$$\mathtt{GenCNTX}(\vec{c},o,s) \equiv_{\text{def}} \mathtt{GenCNTX}(\vec{c},o) \wedge s = \hat{c}_n \tag{16}$$

5. The Formalization of Function

We now have all the elements to define the notion of function, that is

Definition 8 (Systemic Function)**.** Given a function context C, we say that a behavior B of O *plays a functional role* in S with respect to C and, thus, O *performs a systemic function* in S with respect to C if there exists a general systemic context $< C_1,\ldots,C_n >$ for object O and system S with $B_1 = B$ where $< C_1,\ldots,C_n >$ is context-behavior appropriate.

Let us fix a sequence of contexts $\vec{c} =< \bar{c}_1,\ldots,\bar{c}_n >$. We write $\bar{c},o,b$ and s for the context, the object, the behavior and the system in the informal definition and $\bar{c}_i =< c_{i_\Sigma},o_i,\hat{c}_i,g_i,b_i >$. Then, the following formula formalizes the definition of function:

$$\begin{aligned}\mathtt{SysFunction}(o,b,\bar{c},s;\vec{c}) \equiv_{\text{def}} {} & o = o_1 \wedge b = b_1 \wedge \bar{c} = c_n \wedge \\ & s = \hat{c}_n \wedge \mathtt{GenCNTX}(\vec{c},o,s) \wedge \mathtt{APPROPRT}(\vec{c})\end{aligned} \tag{17}$$

Thus, a functional role is the entity formalized by the tuple: object, behavior of the object, context, and system of that context. An object performs a function in a context when it has a behavior that plays the corresponding functional role in the context's support. We now classify functions in terms of functional roles with respect to object O.[5]

- A function F is *irrelevant* to O when no behavior B of O performs the functional role F in any function context. Since we cannot quantify over general contexts, we give a weak characterization of the notion. A functional role is irrelevant for object o in context $\bar{c}$ if for all sequence of contexts $\vec{c}$ and all behaviors b:

$$\mathtt{APPROPRT}(\vec{c}) \rightarrow \neg\mathtt{SysFunction}(o,b,\bar{c},\hat{c}_n;\vec{c})$$

- A function F is *relevant* to O when there is a function context in which O performs the functional role F, that is, given obejct o, context $\bar{c}$ there is a sequence of contexts $\vec{c}$ and a behavior b such that:

$$\mathtt{APPROPRT}(\vec{c}) \wedge \mathtt{SysFunction}(o,b,\bar{c},\hat{c}_n;\vec{c})$$

6. Adapting the Model to Other Foundational Ontologies

6.1. Ontological Considerations

The function definition we have modeled is based on two key assumptions:

[5] The functional role and the corresponding function are distinct: the first is a role played by a behavior, the latter is a role-holder (the functional role plus the behavior that plays it) performed by an object, see [31].

(1) A function is a role played by behavior.
(2) A process exists as a whole at any time when it exists and it can change.

Assumption (1) implies that an ontology can be enriched with our function framework only if it includes, or is extended with, a theory of roles compatible with occurrent roles. An occurrent role is a role played by an occurrent. Roughly speaking, an occurrent role (e.g., a preparation phase in a procedure) describes the contribution an occurrent makes in the given context beyond its simple realization. In contrast, a continuant role (e.g., a teacher in a school) describes what the continuant has to do in the given context. For example, if 'cleaning a table' is performed as preliminary step for 'taking a meal on the table', when seen in that context the cleaning process plays the role of a preparation.

Assumption (2) has also ontological implications. It leads to distinguish two subtypes of the occurrent type: process and event. The first exists at each instant and lasts for a period of time, the latter exists only in the whole interval rather than at any time in the interval (what exist within the interval are its sub-events). In order to include our function definition, therefore, an ontology should be able to include (or suitably translate) this separation between process and event.

An analysis of the status of existing foundational ontologies with respect to these issues and a proposal on how to modify the function module to make it suitable in those systems cannot be pursued here for lack of space. In the rest of this section we briefly look at these points in two ontologies, namely, BFO and DOLCE.

6.2. BFO

BFO's definition of function clearly distinguishes between function (a disposition of the function bearer) and realization (an occurrence). Thus, any occurrent related to a function is a realization of a disposition, that is, a functioning. BFO's approach does not distinguish between behavioring and functioning since it focuses on the capacity/disposition to perform independently of how the bearer is behaving or whether it is functioning.

This observation already suggests that our function module needs to be modified to be included in BFO. If we divide BFO's functions in two types, e.g. capacity functions and actual functions, as discussed in [4], then we can try to incorporate the function module into BFO. A current BFO function would be considered a capacity function and a function as defined in our module would be an actual function. However, two fundamental issues remains on the treatments of occurrents and roles. First, BFO does not differentiate between process and event, furthermore occurrents cannot change. Second, the notion of role in BFO answers the question "what is the role of a teacher?" While in the role community one typically discusses the following "what is a teacher?" [32], which is the notion we have used. In summary, to incorporate our function module in BFO requires some conceptual work.

6.3. DOLCE

DOLCE does not include a notion of function although, e.g., engineering functions have been modeled within this framework [5] via extending individual qualities to relational qualities. We thus need to see whether and how the assumptions (1) and (2) can be understood with this system.

DOLCE's notion of social role [33] does not comprise nor excludes occurrent roles. The basic idea in DOLCE's role notion is that a role is a description that classifies endurants during some period of time while satisfying some basic properties. This way to formalize roles is compatible with what done in the previous sections.

On the contrary, the event definition in DOLCE does not allow to include a notion of 'ongoing' process, as requested by assumption (2). This shows that one has to adapt DOLCE and/or modify the function module of Section 5 before including it in DOLCE, if possible at all. (Note that DOLCE has also a category called 'process' and this is different from the notion used in this paper. We will never use the DOLCE notion so there is no danger of terminological clash.)

Recall that a process, being an occurrent, has a participant, call it o. In DOLCE our proposal is to model the notion of process used in the function module as a (complex) individual quality. Let us call q_o the process-quality of o. This quality q_o should gather the *tendency to change* of all the simple individual qualities of the object o. A behavior, in turn, is defined as an individual quality corresponding to a restriction of q_o to just some of the qualities of o. In particular, note that the quality q_o (and the behaviors associated with it) exists at any time as a whole, satisfying an important feature of processes in the function module.[6] In the next section, we show how this can be done in DOLCE.

7. A General Function Module for DOLCE

Let us now go back to Section 3.2 to see whether some definition needs to be adjusted and what are the consequences. We also provide a formalization of the modified elements. We refer to the DOLCE-CORE in [10] since it includes important notions like role and concept, the latter including description (recall our initial discussion in Section 3.1).

The notions of system and component (Def. 1 in Sec. 3.2) are based on a generic notion of object not characterized in details. Unless one is interested in including/excluding specific types of objects (e.g. holes as discussed in Section 3.1), that notion can be accepted taking objects to be DOLCE endurants (in order to have functions of features and of materials) or just physical objects. Thus, relations like DESCRPTN and DESCRIBES can be redefined from the relations needed to formalize the DOLCE description category.

As seen, the notion of behavior (an individual quality which is sum of direct qualities) changes and now reads: "a behavior of an object O is an indirect quality of O that at each point in time characterizes the tendency to change of some direct qualities of O."

Formally, we can rewrite Def. 2 of Sec. 4.2 as follows (here 'Q'' is a quality predicate based on the DOLCE predicate Q that holds for the mereological sum of direct physical qualities, while 'I' is the inheritance relation formalized in [10]):

$$\mathtt{BEHVR}(b,o) \equiv_{\text{def}} \mathtt{OBJCT}(o) \wedge \mathrm{Q}'(b) \wedge \mathsf{I}(b,o) \tag{18}$$

This change amounts to substitute a sum of qualities for processes. First of all, the CCNTR relation must now be defined on this kind of qualities. This change makes it easier to establish the causal contribution of the tendency to change of one set of physical qualities over the tendency to change of another. Second, the notion of function context does not

[6] Formally, this process-quality should be treated as a relational quality between the object o and the support of the context or some part of it.

depend on processes (neither in the YAMATO nor in the DOLCE sense) but on individual qualities (*b* is the fifth argument in a 5-tuple defining such contexts). Third, the notion of state should be changed to that of a behavior-quality with 'zero-tendency to change', so that a goal is now a role played by a quality. These changes lead to rewrite basic axioms like (6), and to rephrase other constraints:

$$\mathtt{CONTXT}(x) \rightarrow \exists y\, \mathtt{BEHVR}(y) \wedge \mathsf{I}(y, x_\Sigma) \tag{19}$$

$$\mathtt{OBJCT}(x) \rightarrow \exists y\, \mathtt{Q}'(y) \wedge \mathsf{I}(y, x) \tag{20}$$

The other definitions in our function module remain adequate. Yet, they now have different meanings. In particular, here behavior is ontologically a different entity.

Finally, regarding the general assumptions, we observe that adapting the function module to DOLCE led to a module in which condition (1) of Section 6.1 is satisfied only if we allow to understand behavior as a different category. Similarly, even condition (2) is now satisfied: a process, understood as an indirect individual quality measuring the tendency to change in the value of direct individual qualities, exists as a whole at each time in which it exists and can change its values over time.

8. Conclusions

After revising the general ontological definition of function in [1,6], we highlighted the basic elements on which it rests and provided a formalization obtaining a formal ontological module suitable suitable for function modeling in engineering and biology.

The functional world requires a rich conceptual structure in terms of properties, roles and processes. The difficulty to model functions explains the variety of definitions available today. The module is thus proposed as a source for ontologists to extend an ontological system with a structure for function modeling which is quite broad and ontologically sound. The module is presented as a framework and, as such, could be adapted to other ontologies. We investigated two cases, BFO and DOLCE, exemplifying what problem may arise when a more specific notion of function is already present (BFO) and how the framework can be modified to add a notion of function when one is missing (DOLCE). The framework is aimed to ontologists and knowledge engineers (not, e.g., system administrators or end-users) since its adaptation requires important ontological decisions.

In the future, we plan to investigate further ways to simplify the approach (e.g., by using a simpler notion of context) as well as the use of the module in other foundational ontologies. Furthermore, it is interesting to see how to add the module in systems that already include other notions of function. This goal requires a broader analysis of the ontological categories typically used for function definition.

References

[1] R. Mizoguchi, Y. Kitamura and S. Borgo. A unifying definition for artifact and biological functions *Applied Ontology*, 2016 (to appear).

[2] S. Borgo and L. Vieu. Artifacts in Formal Ontology. In A. Meijers, editor, *Handbook of the Philosophy of the Technological Sciences.*, volume 9, Elsevier, pages 273–307, 2009.

[3] S. Borgo, M. Franssen, P. Garbacz, Y. Kitamura, R. Mizoguchi and P.E. Vermaas. Technical artifact: An integrated perspective. *Applied Ontology*, 9(3–4), 217–235, 2014.

[4] Y. Kitamura and R. Mizoguchi. Characterizing functions based on phase- and evolution-oriented models. *Applied Ontology*, 8(2):73–94, 2013.

[5] S. Borgo, M. Carrara, P. Garbacz and P.E. Vermaas. A formalization of functions as operations on flows. *Journal of Computing and Information Science in Engineering*, 11(3):1–14, 2011.

[6] R. Mizoguchi, Y. Kitamura and S. Borgo. Towards a unified definition of function. FOIS 2012, volume 239 of *FAIA*, IOS Press, pages 103–116, 2012.

[7] S. Borgo and R. Mizoguchi. A first-order formalization of event, object, process and role in Yamato. In P. Garbacz and O. Kutz, editors, *International Conference on Formal Ontology in Information Systems (FOIS 2014)*, vol. 267 *FAIA*, pages 79–92, IOS Press, 2014.

[8] R. Arp and B. Smith. Function, role, and disposition in basic formal ontology. In *Proceedings of Bio-Ontologies Workshop (ISMB 2008)* (pp. 45–48), Toronto, 2008.

[9] B. Smith and P. Grenon. Basic formal ontology. `http://ontology.buffalo.edu/bfo`, 2002.

[10] S. Borgo and C. Masolo. Foundational Choices in DOLCE. In S. Staab and R. Studer, eds, *Handbook on Ontologies*, pages 361–381, Springer Verlag, 2009.

[11] A.G. Wouters. The function debate in philosophy. *Acta Biotheoretica*, 53(2):123–151, 2005.

[12] L. Wright. Functions. *The Philosophical Review*, 82(2):139–168, 1973.

[13] K.D. Forbus. Intelligent computer-aided engineering. *AI magazine*, 9(3): 23–36, 1988.

[14] M.S. Erden, H. Komoto, T.J. Van Beek, V. D'Amelio, E. Echavarria and T. Tomiyama. A review of function modeling: Approaches and applications. *AIEDAM*, 22(2):147–169, 2008.

[15] M. Perlman. The modern philosophical resurrection of teleology. *The Monist*, 87(1), 3–51, 2004.

[16] J. Röhl and L. Jansen. Why functions are not special dispositions: an improved classification of realizables for top-level ontologies *Journal of Biomedical Semantics* 5(1):27, 2014.

[17] Y. Kitamura, Y. Koji and R. Mizoguchi. An ontological model of device function: industrial deployment and lessons learned. *Applied Ontology*, 1(3-4):237–262, 2006.

[18] Y. Umeda, M. Ishii, M. Yoshioka, Y. Shimomura and T. Tomiyama. Supporting Conceptual Design based on the Function-Behavior-State Modeler. *AIEDAM*, 10, 275–288, 1996.

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

[20] R. Cummins. Functional analysis. *Journal of Philosophy*, 72:741–765, 1975.

[21] I. Johansson, B. Smith, K. Munn, N. Tsikolia, K. Elsner, D. Ernst, and D. Siebert. Functional anatomy: A taxonomic proposal. *Acta Biotheoretica*, 53(3):153–166, 2005.

[22] M. Artiga. Re-organizing organizational accounts of function. *Applied Ontology* 6(2):105–124, 2011.

[23] P. McLaughlin. What Functional Explanation and Self-Reproducing Systems. Cambridge: Cambridge Univ. Press, 2001.

[24] C. Boorse. A rebuttal on functions. In R.C.A. Ariew and M. Perlman, eds, *Functions: New essays in the Philosophy of Phychology and Biology*, pages 63–112, 2002.

[25] U. Krohs and P. Kroes. *Functions in biological and artificial worlds: comparative philosophical perspectives*. MIT Press, 2009.

[26] U. Krohs. Functions and fixed types: Biological and other functions in the post-adaptationist era. *Applied Ontology*, 6(2):125–139, 2011.

[27] W. Houkes and P.E. Vermaas. *Technical functions: On the use and design of artefacts*. Springer Verlag, 2010.

[28] A. Galton and R. Mizoguchi. The water falls but the waterfall does not fall: New perspectives on objects, processes and events. *Applied Ontology*, 4(2):71–107, 2009

[29] R. Stout. Processes. *Philosophy*, 72:19–27, 1997.

[30] R. Mizoguchi. Yet Another More Advanced Top-level Ontology. Available from: `http://download.hozo.jp/onto_library/YAMATO101216.pdf`, 2010.

[31] R. Mizoguchi, A. Galton, Y. Kitamura and K. Kozaki. Families of roles: A new theory of occurrent-dependent roles. *Applied Ontology*, 10 (3-4):367–399, 2015.

[32] F. Steimann. The role data model revisited. *Applied Ontology*, 2:89–103, 2007.

[33] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi and N. Guarino. Social roles and their descriptions. *KR*, AAAI Press, pages 267–277, 2004.