006550
FOFIS
Formal Ontology for Information Systems

EIF

Deliverable 3 (Final) General Ontology of Artefacts Including Subontology of Information Objects

Period Number:1	Due date of deliverable: 17.03.2006
Period covered: from 1.02.2005 to 31.01.2006	Date of preparation: 10.03.2006
	Date of submission: 12.03.2006
Start date of project: 1.02.2005	Duration: 12 months

Project coordinator name: Nicola Guarino
Project coordinator organisation name: Istituto di Scienze e Tecnologie della Cognizione del Consiglio Nazionale delle Ricerche

Organisation name of lead contractor
for this deliverable: Istituto di Scienze e Tecnologie della Cognizione del Consiglio Nazionale delle Ricerche

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)
Dissemination in level PU

Contents

1 Introduction

In the previous two deliverables I reported two major achievements of the project: a general semi-formal theory of artefact and a formal theory of documents. This deliverable is aimed to unify these two views and to develop the further on. The first two section summarises tha main results from the previous deliverables. Section 3 extends the general theory of artefact to caputer the engineering notions of behaviour and function. Section 4 shows how to incorporate the theory of documents into the broad picture provided by the theory of artefacts. The last section is devoted to evaluation of the project outcomes and suggestions for further work.

2 General theory of artefacts

2.1 Formal tools

The general theory of artefacts uses two formal tools: the theory of consequence operation and the theory of states of affairs and objects.

Let $\mathfrak{L}_{\mathfrak{A}}$ be a language to speak about artefacts. The consequence operation $C : \wp(\mathfrak{L}_{\mathfrak{A}}) \rightarrow \wp(\mathfrak{L}_{\mathfrak{A}})$ satisfies the following axioms:

$$
\begin{gathered}
(i)\ X \subseteq C(X), \\
(ii)\ X \subseteq Y \rightarrow C(X) \subseteq C(Y), \\
(iii)\ C(C(X)) \subseteq C(X), \\
(iv)\ \varphi \in X \rightarrow \exists Y \subseteq X[Y \text{ is finite } \wedge \varphi \in C(Y)].
\end{gathered} \tag{1}
$$

The theory of states of affairs and objects is based on two primitive (i.e. undefined) notions: '$Occ(x,y)$' and "$x \leqslant y$". The former means that an object x occurs in a state of affairs y and the latter means that a state of affairs x is part of a state of affairs y. The theory is defined by the following definitions and axioms:

$$
\begin{gathered}
(i)\ Obj(x) \equiv \exists y\ Occ(x,y), \\
(ii)\ Soa(x) \equiv \exists y\ Occ(y,x).
\end{gathered} \tag{2}
$$

$$Obj(x) \equiv \neg Soa(x). \tag{3}$$

$$x \leqslant y \rightarrow Soa(x) \wedge Soa(y). \tag{4}$$

$$Soa(x) \rightarrow x \leqslant y. \tag{5}$$

$$x \leqslant y \wedge y \leqslant x \rightarrow x = y. \tag{6}$$

$$x \leqslant y \wedge y \leqslant z \rightarrow x \leqslant z. \tag{7}$$

$$x \leqslant y \rightarrow \forall z\ [Occ(z,x) \rightarrow Occ(z,y)]. \tag{8}$$

2.2 Theory

Artefacts are entities produced on purpose. We produce artefacts in order to achieve by means of them some aims we find important. We produce them on the ground of their designs, which are supposed to make our production more efficient. We produce artefacts and construct their designs referring to some background knowledge relevant for this kind of artefacts. This knowledge is supposed to guarantee that artefacts help us to achieve the aims for which they were produced, or at least it is supposed to make it more probable that we achieve those aims. We may achieve those aims if we follow the instructions of use determined by artefacts designs and the respective background knowledge. Subsequently, any artefact should be characterised with respect to four dimensions: teleological (i.e. purposes), intentional (i.e. design), epistemic (i.e. background knowledge), and operative (i.e. instructions of use).

Teleological dimension I argued that the category of artefact purposes should be relativised to a community that imposes these purposes. $Purpose(x, y, z) \subseteq S_A$ means that a state of affairs x is ascribed by a community z as a purpose of an artefact x. A set $\mathfrak{Purpose}(x, y)$ will contain any sentence that represents a purpose of an artefact x in a community y. Since it turned out that artefacts are social entities, I will represent them by means of a binary predicate "Art": $Art(x, y)$ means that an object x is an artefact in a community y. Below I present the axioms that determine the categories of arguments of $Purpose$ and Art (9 and 31) and secure that any artefact is endowed with at least one purpose (32 and 33).

$$Art(x, y) \rightarrow Obj(x) \wedge Obj(y). \tag{9}$$

$$Purpose(x, y, z) \rightarrow Soa(x) \wedge Art(y) \wedge Obj(z). \tag{10}$$

$$Art(x, y) \rightarrow \exists z\ Purpose(z, x, y). \tag{11}$$

$$Art(x, y) \rightarrow \mathfrak{Purpose}(x, y) \neq \emptyset. \tag{12}$$

Intentional dimension I argued that any artefact is related to some design that determines its characteristic features. I discriminated between engineering specifications, which are material objects in which designs are "encoded", from design in the proper sense of the word. I classified the former to the philosophical category of intentional states of affairs. I show what it means that design represents artefact. I also argued at some length that at least some arefacts are related to more than one design, different designs related to one artefacts being ordered with respect to their specificity.

The resulting theory of artefact design may be rendered in the first-order language. Let "design(x, y)" means that x is a *design of* an artefact y.

$$design(x, y) \rightarrow Soa(x) \wedge Obj(y). \tag{13}$$

Artefact designs may be ordered with respect to their specificity. I will identify the relation of being less specific with the relation $<\restriction \{x : \exists y \text{ design}(x, y)\}$. i.e the relation $<$ restricted to the set of artefact designs. It is obvious that every artefact has the most specific design (14), which is the design according to which the artefact is manufactured. If we agree that some artefacts have also less specific design representations, we should acknowledge the existence of the least specific representation (15). In order to support this claim let me observe that any artefact design consists of finitely many elements, i.e. of finitely many states of affairs, as a product of an intentional agent (or a finite group of intentional agents) with strictly limited representational capabilities. This entails that any artefact has a (possibly non-unique) $<$-minimal design. Since all these minimal designs represent one artefact, they have something in common, i.e. the set of states of affairs which are parts of all minimal designs is not empty. This common core is represented here by the notion of least specific design.

$$design(x, y) \equiv design(x, y) \wedge \forall z[design(z, y) \rightarrow z \leq x]. \tag{14}$$

$$design_0(x, y) \equiv design(x, y) \wedge \forall z[design(z, y) \rightarrow x \leq z]. \tag{15}$$

Axiom 16 states that every artefact has the most and the least specific design.

$$Art(x) \to \exists y \exists z\ [design_0(y,x) \wedge design(z,x)]. \tag{16}$$

Definitions 14 and 15, and axiom 16 entail that for any artefact x, the most and the least specific design of x are unique. I will denote them by, respectively, "Design(x)" and "design$_0$(x)". The former will be called the *full design* of x; the latter will be called the *minimal design* of x.

A set $\mathfrak{design}(x,y) \subseteq \mathfrak{L}_{\mathfrak{A}}$ will contain all sentences representing a design x of an artefact y. Subsequently,

$$\mathfrak{design}(x) := \{y : \mathfrak{design}(y,x)\}. \tag{17}$$

$$\mathfrak{design}_{\mathfrak{o}}(x) := \bigcup\{\mathfrak{design}(x,y) : design_0(x,y)\}. \tag{18}$$

$$\mathfrak{Design}(x) := \bigcup\{\mathfrak{Design}(x,y) : Design(x,y)\}. \tag{19}$$

I introduced a number of constraints which are to exclude the most obvious cases of irrational designs.

$$Occ(x, design(y)) \wedge Art(x) \to design_0(x) \leq design(y). \tag{20}$$

$$design(x,y) \to \neg Occ(y,x). \tag{21}$$

$$design_0(x) < design(y) \wedge design_0(y) < design(x) \to x = y. \tag{22}$$

$$design(x,y_1) \wedge design(x,y_2) \to y_1 = y_2. \tag{23}$$

$$Occ(x,y) \wedge design_0(z,x) \to z \leq y. \tag{24}$$

$$Design(x_1) = Design(x_2) \to \forall z\ [Purpose(z,x_1,y) \equiv Purpose(z,x_2,y)]. \tag{25}$$

$$\mathfrak{Design}(x_1) = \mathfrak{Design}(x_2) \to \mathfrak{Purpose}(x_1,y) = \mathfrak{Purpose}(x_2,y). \tag{26}$$

Epistemic dimension The third element of my conceptual model of artefacts is background knowledge. When Smith designs some artefact, her designing is not a chaotic sequence of independent actions. Her designing forms a relatively compact structure of actions linked together by her conceptual decisions based on some background knowledge. The information she refers to does not determine her action exhaustively, but the more influential impact it exerts, the less accidental the resulting design turns out to be. Similarly, when Brown produces the artefact designed by Smith, Brown is guided by information from some source of information. The sum of information from both sources will be called the background knowledge relevant for a given artefact. The background knowledge relevant for an artefact x contains not only general theorems about the nature of objects of the same kind as x, but also practical rules of thumb relevant for the production of x. The majority of artefacts we use are designed and manufactured on the ground of vague psychological and sociological observations concerning our desires, fears, preferences, beliefs, etc., but we may generically determine the content of the relevant background knowledge by using labels: mathematics, quantum chemistry, physiology of hearing, etc.

I will represent the background knowledge relevant for an artefact x by a set $\mathfrak{Knowledge}(\mathfrak{x})$ of sentences from $\mathfrak{L}_{\mathfrak{A}}$.

$$Art(x,y) \to \mathfrak{Knowledge}(x) \neq \emptyset. \tag{27}$$

It seems plausible to assume that the background knowledge is a theory with respect to the consequence operation C. I define an auxiliary extension of C:

$$C_{K(x)} := C(X \cup \mathfrak{Knowledge}(x)). \tag{28}$$

I stipulated that

$$C_{K(x)}(\mathfrak{Purpose}(x,y) \cup \mathfrak{Design}(x)) \neq \mathfrak{L}_{\mathfrak{A}}. \tag{29}$$

The background knowledge relevant for an artefact x is determined uniquely by the full design of x.

$$\mathfrak{Design}(\mathfrak{x}_1) = \mathfrak{Design}(\mathfrak{x}_2) \to \mathfrak{Knowledge}(x_1) = \mathfrak{Knowledge}(x_2). \tag{30}$$

2.3 Operative dimension

The last element of my conceptual model is instructions of use. As in the case of designs, I distinguish a technological instruction of use from a philosophical instruction of use. The former is usually a sentence in the imperative mood, the latter is the ontic representation of the declarative counterpart of that sentence. $Use(x,y,z) \subseteq S_A$ means that a state of affairs x is ascribed by a community z as a purpose of an artefact x. A set $\mathfrak{Use}(x,y)$ will contain any sentence that represents a purpose of an artefact x in a community y. Here are the constraints on the operative dimension I found necessary.

$$Use(x,y,z) \to Soa(x) \wedge Art(y) \wedge Obj(z). \tag{31}$$

$$Art(x,y) \to \exists z Use(z,x,y). \tag{32}$$

$$Art(x,y) \to \mathfrak{Use}(x,y) \neq \emptyset. \tag{33}$$

$$C_{K(x)}(\mathfrak{Use}(x,y) \cup \mathfrak{Purpose}(x,y) \cup \mathfrak{Design}(x,y)) \neq \mathfrak{L}_{\mathfrak{A}}. \tag{34}$$

$$Design(x_1) = Design(x_2) \to Use(x_1) = Use(x_2). \tag{35}$$

$$\mathfrak{Design}(x_1) = \mathfrak{Design}(x_2) \to \mathfrak{Use}(x_1,y) = \mathfrak{Use}(x_2,y). \tag{36}$$

$$\mathfrak{Purpose}(x,y) \subseteq C_{K(x)}(\mathfrak{Design}(x) \cup \mathfrak{Use}(x,y)) \setminus C_{K(x)}(\emptyset). \tag{37}$$

2.4 Definition of artefact

The four-dimensional ontology defines an artefact type as a quadruple consisting of its purposes, design(s), background knowledge, and instructions of use. More precisely speaking,

$$Art(x,y) \equiv \exists z\ Purpose(z,x,y) \wedge \exists z\ design(z,x) \wedge \mathfrak{Knowledge}(x) \neq \emptyset \wedge \exists z\ Use(z,x,y). \tag{38}$$

Thus, an artefact type x within a community y will be represented as a quadruple $< \mathfrak{Purpose}(x,y), \mathfrak{design}(x), \mathfrak{Knowledge}(x), \mathfrak{Use}(x,y) >$.

3 Behaviour and function in general theory of artefacts

3.1 Behaviour and function in engineering design

Any adequate engineering model of technical devices is bound to describe their behavioral and functional characteristics. In engineering design there exist different, sometimes incompatible, definitions of function and behaviour. Here is a short survey of the more influential accounts.[1]

1. FR (cf. [3])
    - Behaviour

[1] The abbreviations I use here are explained in the first deliverable.

- Definitions

  Beh-i. The value(s), or relations between values, of state variables of interest at a particular instant [...]
  Beh-ii. The value(s), or relations between values, of properties of an object. This is closely related to sense (i) above. [...] In such descriptions, however, time is not explicitly mentioned. Thus, instead of thinking of these behaviors as state variable values at any specific instant, it is more perspicuous to think of them as relations between specified properties of an object.
  Beh-iii. The value(s) of state variables of interest over an interval of time. [...]
  Beh-iv. The value(s) of state variable(s) specifically labeled output state variables, either at an instant or over an interval of time. [...]
  Beh-v. The values of all the state variables in the object description, either at an instant or over an interval of time. [...]
  Beh-vi. The causal rules that describe the values of the variables under various conditions.

  Let W be an object or an object configuration. A behavioral constraint is any constraint on the behaviors in W (behaviors defined in any of the senses defined earlier).

- Examples

  Beh-i. [...] How did the car behave? It rattled when I hit the curve. How does the widget behave? The ratio of output to input voltage is greater than one.
  Beh-ii. [...] For example, one might say, a lintel distributes the load to the two sides. A window transmits the light from outside to inside the house. A paperweight keeps the paper in place. [...] object.
  Beh-iii. [...] What did you notice about the behavior? The BHP increased for a while, but then started decreasing.
  Beh-iv. [...] The amplifier is behaving well – the output voltage is constant.
  Beh-v. [...] A graph that plots all the variables over time is often called the behavior graph.
  Beh-vi. [...] For example, one can say that Ohms Law, i.e., the causal model in (3), describes the behavior of an electrical resistor.

  Behavioral constraint i. $P(B)$ is a behavioral constraint. Examples: The value of output voltage is greater than 5 volts. The average value of the variable P over time is 6.5 psi.
  Behavioral constraint ii. If $C, P(B)$ is a behavioral constraint. Example. If the input voltage is above 5, the output voltage is be a sinusoid. If the switch is pressed, the voltage goes up to 5 volts, and stays at that level.
  Behavioral constraint iii. If C_1, then $P_1(B)$; then if C_2, then $P_2(B)$, ...then if C_m, then "$P_m(B)$. Example: If switch is pressed, the ATM flashes, insert card. If card inserted, ATM flashes, Enter ID. If ID is entered, ATM...

- Function
  - Definitions

The mode of deployment of an object (device) D in some world W, represented as $M(D, W)$, is the specification of all the ways in causal interactions between D and the entities in W are instantiated. Such a specification might include one or more of the following:
(i) Structural relations between D and the entities in W.
(ii) Actions or action sequences by entities in W on D.

Let F be a set of behavioral constraints that an agent, say A, desires or intends to be satisfied in some 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.

Let F be a set 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.

  – Examples

    ad i) Example: Locate part p of device D at location L_1, electrically-connect terminal t of D to terminal of electrical outlet, ...
    ad ii) Examples: 1. Push *Switch* of D. 2. Insert card into slot of D, when insert card flashes on screen. (3)...

    For instance, an electrical battery may deliver the function, Provide a voltage of B volts between external electrical terminals p_1 and p_2, under the mode of deployment, p_1, p_2 electrically connected to electrical terminals of battery.

    Let F_E be the following behavioral constraint: F_E: T_v, the temperature of a specific volume of space Vol in some world $W > T_{ambient}$, the ambient temperature. (That is, we wish to heat a given volume of air in a space.) A device-centered functional description for heater, corresponding to the above F_E, is: F_D: If *Switch* is closed, $T_S(t) > T_S(t_0)$, where T_S is the temperature of the surface and t_0 is the initial time.

2. FBRL (cf. [13]

   - Behaviour
     - Definition
       A *behaviour* is represented by relations among input and output objects.
     - Examples
       A shift of a kind of energy from one stream of energy to another stream.
   - Function
     - Definitions
       A *goal* is an intended desirable state of a component of a system.
       A *function* is the interpretation of behaviour under a goal.
     - Examples
       goal: temperature of the input coolant does not exceed 500 degrees centigrade
       function: to make the temperature of a thing put on a cooking stove to be more than a certain degree (a function of the cooking stove)

3. Kitamura and Mizoguchi (cf. [6])

   - Behaviour

- Definitions
  An *object* is something that can be considered as what it goes through a device from the input port to the output port during which it is processed by the device. A b*ehaviour* is a change of an attribute value of an object from that at the input port of a device to that at the output of the device.
- Examples
  object: heat
  behaviour: the increase of the temperature of steam occurred during it goes through a superheater

• Function

- Definitions
  A *function* of a device is a result of teleological interpretation of a behaviour under an intended goal.
- Examples
  to vaporize water
  to heat water

4. FBS (cf. [16]

• Behaviour

- Definition
  Behaviour is a change of states or a sequence of changes of states.
- Example
  A is supporting B.

• Function

- Definition
  Function is a description of behaviour abstracted by human through recognition of the behaviour in order to utilise it.
- Example
  to support something for manufacturing

5. Pahl and Beitz (cf. [11])

• Flow

- Definition
  A *flow* is a conversion of energy, material or information.
- Examples
  flows: carpet tiles, electricity, torque

• Function

- Definition
  A *function* is an input/output relation of a system whose purpose is to perform a task.
- Examples
  to reduce speed
  to transfer torque

3.2 Towards ontological understanding of behaviour and function

I submit that the engineering categories of behaviour and function may be modelled as states of affairs, either static, e.g. that the window transmits the light from outside to inside the house, or dynamic, e.g. that the BHP increased for a while, but then started decreasing. Functions are construed then as those behaviours that are intentionally selected by designers. This modelling solution has the following advantages:

- We obtain a broad picture in which purposes, behaviours, functions, and instructions of use are unified within a well-established conceptual category.
- The category of state of affairs is broad enough to accommodate both the definitions from Functional Representation approach ([3], [2]) and the definitions from Functional Modelling ([14], [5]).
- In this picture, behaviours and functions are construed as complex entities.
- If we align this solution to a sufficiently broad ontological taxonomy, we may obtain a number of distinctions and relationships.

It is not the case that every state of affairs in which a given artefact occurs is its behaviour or function because some states of affairs in which artefacts occurs are accidental to them. It seems that only those states of affairs that are parts of the full design of an artefact may be classified as the behaviours of that artefact. I do not claim however that any state of affairs from the full design may be called a behaviour of that artefact.

One of the possible ways of making the distinction between behaviours and functions is to say that a function of a device is such behaviour of this device which is intended by its designer(s) (cf. [3]). It is claimed that when the designer chooses a behaviour with an intention of transforming it into a function, he abstracts from or disregards some aspects of the behaviour (e.g. [7], p. 400; [16], p. 183; [17], p. 340-341). For example, the function "to make a sound" is abstracted from the behaviour "the oscillator collides with the boss repeatedly" ([16], p. 187). Due to the fact that research in engineering design does not characterise this operation of abstraction in sufficient detail, I have to stop this search for a definition of function at this, clearly unsatisfactory point:

- A behaviour of an artefact is a state of affairs from the full design of the artefact.
- A function of an artefact is a result of the operation of abstraction performed by the designer(s) of the artefact.

Nonetheless, at this point we may use the above results to define the notion of functional parthood.

3.3 First-order theory of functional parthood

It is obvious that some behaviours specified by the design of an artefact do not concern the functions of the artefact. Still, it is equally obvious that for any rational design process, any object mentioned in this design, e.g. a bolt, piston, etc., plays some function in the artefact; otherwise, the object would be redundant and as such would not occur in a rational design. In other words, even if a design mentions more than just artefact functions, each object mentioned in the design performs some function in the artefact which is represented by this design.

This argument requires four comments. First, the argument is based on the distinction between entities that are specified (or qualified) by a design, such as bolts, pistons, and capacitors, and specifications (or qualifications), such as diameters, temperatures, etc. Fox example, when an engineer requires that the diameter of a bolt be 10 mm, then the bolt is the entity which is thereby specified and the diameter is the corresponding specification. I assume that only the former entities, which later will be called objects, are, so to speak, eligible candidates for function bearers. Thus, it is the bolt, and not its diameter, of which we may say that it has a function. The notion of object at stake is roughly equivalent to the notion of bearer of properties. Nonetheless, having certain qualities as their properties, objects participate in processes, are related by relations, etc.

Secondly, one should distinguish between the broad and the narrow notion of function. A function in the broad sense is anything which is desired or intended by some agent (cf. [3], p. 172-177). If an engineer speaks about functions or functional requirements, he usually has in mind this broad meaning (cf. [12], p. 166-167). A function in the narrow sense is a role which something plays in some structure as opposed to a purpose of the structure construed as a whole. Peter McLaughlin explains this distinction as follows:

> In the case of the functions of whole artifacts the determination of their functions or purposes is completely external. It lies in the actual intentions of the designer, manufacturer, user, etc., however socially determined these intentions may in fact be. [...] On the other hand, the functions of parts of an artificial system are in a sense internal and somewhat more objective insofar as these functions are always relative to their contribution to the capacities of the system of which they are part, and this contribution is part of the causal structure of the material world. It is the implied reference to the containing system [...] that distinguishes such (relative) functions from purposes. We can plausibly distinguish between a knife that has a purpose [...] and a gear that has a function within a machine [...]. ([10], p. 52)

The argument presupposes the narrow notion of function. The broad notion of function is the "sum" of the narrow notion and the notion of purpose.

Still, and this is the third comment, the notion of function at stake is broad enough to include the aesthetic and ergonomic functions. Thus, even if a decorative trim around your car does not physically contributes to the overall function of being a means of transportation, still it performs some function, which justifies the designer's decision to fix the trim to the chassis.

Fourthly, it is worth to emphasise that the argument requires a relatively modest assumption concerning the rationality of designing practice. I do not claim that any detail of an artefact token performs some function because the theory espoused here is consistent with the claim that some details of the artefact token are not specified by its design.[2] Moreover, I do not deny that some functions are redundant, in which case we may eliminate them from the respective designs. Similarly, I do not deny that functions may be inconsistent, in which case performing one of them inhibits performing the other. I just claim that if an engineer mentions some object in his design, this means that he considers this object as performing some function in the artefact he designs and that this opinion of his is, so to speak, ontologically reliable for the notions at stake in the sense that the object actually performs some function.

This allows us to define the notion of functional parthood. Assuming that any object which occurs in a design of an artefact is not redundant within this design, we may say that x performs some function in y iff there is a state of affairs in which x occurs and which is part of the full design of y.

$$Func(x,y) \equiv \exists z[Occ(x,z) \wedge z \leq Design(y)]. \tag{39}$$

[2] Subsequently, I do not commit to a principle which is called by D. Dennett a default assumption of reverse engineering. According to Dennett, a reverse engineer must start his analysis with the assumption that any detail of an artefact whose design he is to reconstruct is there for some reason, i.e. it performs some function in the artefact [4], p. 212.

The formal theory of functional parthood (FTFP) may be expressed in a first order language (with identity) with three primitive binary predicates: "Occ", "$<$", and "design".

The solution to the effect that a theory of functional parthood is in fact a theory of design has three advantages over other approaches. First, the notion of design is far better understood and far less controversial than the notion of function. Being aware of the aforementioned problems with an adequate definition of the latter notion, one may appreciate an approach in which the notion of function is not taken for granted.

Secondly, in the logical properties of functional parthood are not simply assumed as axioms but may be derived as theses. This fact is of some importance for those who esteem the epistemological value of the debates on the mereological principles. We are in a position to discuss the controversies over the relation of functional parthood in a broad framework which makes room for discovering the sources of disagreement. For instance, it can be shown that the relation defined by 39 is irreflexive and asymmetric (40 and 41) but in general not transitive (42). Still, we can show why or when the relation is transitive (43 and 44).

$$\vdash_{\text{FTFP}} \neg Func(x, x). \tag{40}$$

Proof. Assume that for some $x_0, Func(x_0, x_0)$. Definition 39 entails that for some z_0, $\text{Occ}(x_0, z_0)$ and $z_0 \leq Design(x_0)$. Consequently, $Occ(x_0, Design(x_0))$. Axiom 21 entails now that $\neg Design(Design(x_0), x_0)$, which is inconsistent with the definitions of FTFP. □

$$\vdash_{\text{FTFP}} Func(x, y) \rightarrow \neg Func(y, x). \tag{41}$$

Proof. Suppose that for some x_0 and y_0, $\text{Func}(x_0, y_0)$ and $Func(y_0, x_0)$. The former entails (*) and the latter entails (**).
(*) $\exists z[Occ(x_0, z) \land z \leq Design(y_0)]$,
(**) $\exists v[Occ(y_0, v) \land v \leq Design(x_0)]$.
It follows from (*) that $Occ(x_0, Design(y_0))$. It follows from (**) that $Art(x_0)$. (*) and axiom 20 entail now (***):
(***) $design_0(x_0) \leq Design(y_0)$.
Similarly, (****) follows from (**).
(****) $design_0(y_0) \leq Design(x_0)$.
(***), (****), and axioms 22 and 23 entail that $x_0 = y_0$ what contradicts both assumptions of the proof (cf. 40). □

$$\nvdash_{\text{FTFP}} Func(x, y) \land Func(y, z) \rightarrow Func(x, z). \tag{42}$$

Proof. Consider the following model of FTFP. The model consists of ten elements: $1, 2, 3, 4, 5$, $A, B, C, D,$ and E.

1. $Occ(1, A), Occ(1, C)$,
2. $Occ(2, B), Occ(2, C), Occ(2, D), Occ(2, E)$,
3. $Occ(3, C)$,
4. $Occ(4, D), Occ(4, E), Occ(5, E)$,
5. $A < C$,
6. $B < C, B < D, B < E$,

7. $D < E$,
8. $design(A, 3)$,
9. $design(B, 4), design(C, 4)$,
10. $design(D, 5)$.

It is easy to verify that all axioms of FTFP are satisfied in this model. Moreover, for some x, y, and, z if the formulas "$Func(x, y)$" and "$Func(y, z)$" are satisfied, then the formula "$Func(x, z)$" is not. □

$$\vdash_{\text{FTFP}} \forall x\ design_0(x) = Design(x) \rightarrow$$
$$\rightarrow [Func(x, y) \wedge Func(y, z) \rightarrow Func(x, z)]. \tag{43}$$

Proof. Assume that for all x, $design_0(x) = Design(x)$. Now let $Func(x, y)$ and $Func(y, z)$. The former entails (*) and the latter entails (**).
(*) $\exists v_1[Occ(x, v_1) \wedge v_1 \leq Design(y)]$,
(**) $\exists v_2[Occ(y, v_2) \wedge v_2 \leq Design(z)]$.
As in the proof of 41, (**) entails that $design_0(y) \leq Design(z)$. Since $design_0(y) = Design(y)$, (*) gives us that $v_1 \leq Design(z)$. Consequently, we get (***):
(***) $\exists v_1[Occ(x, v_1) \wedge v_1 \leq Design(z)]$,
This obviously completes the proof (cf. 39). □

$$\vdash_{\text{FTFP}} \forall x, y\ [Func(x, y) \rightarrow \text{Design}(x) \leq Design(y)] \rightarrow$$
$$\rightarrow [Func(x, y) \wedge Func(y, z) \rightarrow Func(x, z)]. \tag{44}$$

Proof. Assume that for all x and y, $[Func(x, y) \rightarrow \text{Design}(x) \leq Design(y)]$. Now let $Func(x, y)$ and $Func(y, z)$. The former entails (*) and the latter entails (**).
(*) $\exists v_1[Occ(x, v_1) \wedge v_1 \leq Design(y)]$,
(**) $\exists v_2[Occ(y, v_2) \wedge v_2 \leq Design(z)]$.
Moreover, both assumptions entail that
(***) $Design(y) \leq Design(z)$.
(*) and (***) gives us that $v_1 \leq Design(z)$. Consequently, as in the previous proof we get (****):
(****) $\exists v_1[Occ(x, v_1) \wedge v_1 \leq Design(z)]$,
□

43 reveals one of the sufficient conditions for the transitivity of functional parthood: if each artefact in a set X has exactly one design, then the relation of functional parthood is transitive in X. 44 reveals another condition: if for any artefact from a set X, its full design contains the full designs of all its functional parts, then the relation of functional parthood is transitive in X. Although neither of these conditions is necessary for the transitivity, we may treat theses 43 and 44 as claims which reveal some sources of the non-transitivity of the relation at issue: the multiplicity of designs of a single artefact and underdetermination of artefact designs in general. In a similar way, one can discuss in FTFP other principles of the standard mereology, e.g. the principle of extensionality.

The third advantage of FTFP over other approaches consists in the fact that FTFP does not take for granted such crucial logical properties of functional parthood as transitivity or extensionality, but relates them to the actual engineering designs. Thus, whether this relation is transitive or extensional depends eventually on the way in which artefacts are designed.

3.4 Taxonomy of artefact functions

Exploring the opportunity of getting acquainted with the DOLCE ontology, I will characterise artefact functions in more detail. DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) (cf. [9]) is the first module of the WonderWeb Foundational Ontology Library. The WonderWeb project is aimed at developing language architecture for representing ontologies in the Semantic Web. Within this library DOLCE plays the role of a reference module to be adopted as a starting point for comparing and elucidating the relationships with other modules in the library.

The most characteristic feature of DOLCE is the fact that it is an axiomatic theory of particulars, as opposed to universals. Roughly speaking, a *universal* (in another terminology: type or kind) is an entity that is instantiated in a number of other entities, called particulars. For example, a car model is (usually) instantiated in a number of particular cars. DOLCE provides a taxonomy of particulars and sketches a net of relations between different types thereof by means of more than 80 definitions and 40 axioms expressed in first-order logic. The figure below illustrates the basic categories of particulars acknowledged by DOLCE.

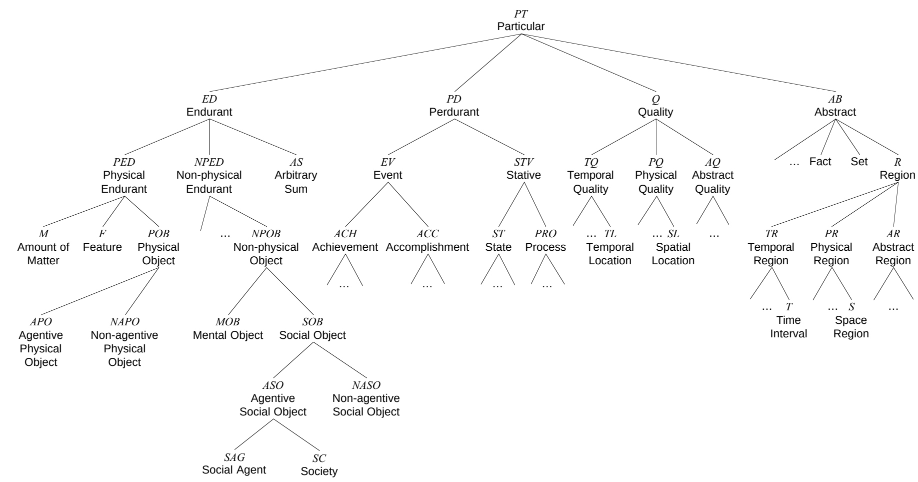

Figure 1: DOLCE basic categories from [9].

An *endurant* is defined as an entity whose all (essential) parts are present at any time at which the entity is present. A *perdurant* is defined as an entity whose some parts are not present at some time at which the entity is present. A shaft is an example of an endurant and a rotation of the shaft is an example of a perdurant. A *quality* is an entity that inheres in another entity. The weight of a shaft is an example of a quality. In what follows I will restrict the intended domain of qualities to the so-called direct qualities, i.e. to such qualities that are not themselves qualities of other qualities. An *abstract entity* is an entity that does not have any spatial or temporal qualities and that is not a quality itself. In DOLCE we distinguish between qualities and quales. The weight of a shaft is a quality of the shaft and the actual "value" of this weight is a quale. A *quale* is an abstract entity that is a part of a quality region. The colour quales, for instance, compose the colour space.

The domain of endurants is related to the domain of perdurants by the primitive relation of participation: an endurant *participates* in a perdurant (at a time interval). The domain of qualities is related to the domain of carriers of qualities by the primitive relation of inherence: a quality *inheres* in a carrier of qualities, or equivalently, a carrier of qualities has a quality. The domain of qualities is related to the domain of quales by the relation of having a value: a quality has a value (in a time interval) which is a quale region. DOLCE advances the constraint to the effect that the relation of inherence is atemporal. This means that a quality inheres in an entity during the whole time interval in which the entity exists. In other words, an entity has a quality always or never.

A *physical endurant* is an endurant that has some spatial qualities. A non-physical endurant does not have any spatial qualities. The category of physical endurants is divided into subcategories by means of the notion of whole. Roughly speaking, an entity x is a *whole* under a relation R iff x is a maximal (under the relation of mereological parthood) mereological sum of entities that belong to the domain of R and are related to each other by R. An *amount of matter* is a physical endurant that is not a whole. A *physical object* is a physical endurant that is a whole under some unifying relation. Different kinds of physical objects are wholes under different unifying relations.

If the (mereological) sum of any two perdurants of a kind φ is a perdurant of the kind φ, then all perdurants of this kind are stative (or cumulative); otherwise they are eventive. If all (mereological) parts of a stative perdurant of a kind φ are of the kind φ, then the perdurant is a *state*; otherwise the perdurant is a *process*. If an eventive perdurant has no proper parts, then it is called an *achievement*; otherwise we call it an *accomplishment*. Conferences, ascents, and performances are examples of accomplishments. Acts of reaching (e.g. a reaching of the summit of K2), departures, and deaths are examples of achievements.

A quality is *temporal* if it inheres in a perdurant. One of the most crucial examples of temporal qualities is a temporal location of a perdurant. A quality is *physical* if it inheres in a physical endurant. One of the most crucial examples of physical qualities is a spatial location of an endurant. Moreover, it seems that at least some topological qualities, e.g. topological connectedness, also belong to this category although they are not explicitly mentioned in DOLCE. A quality is abstract if it inheres in a non-physical endurant.

The DOLCE taxonomy leads to three basic types of states of affairs. If an entity has a quality, then I will say that the respective state affairs (i.e. that the entity has the quality) is of the *inherence type*. For instance, if a hammer has a weight (as one of its properties), then this fact, i.e. that a hammer has a weight, is a state of affairs of the inherence type. If an endurant participates in a perdurant, then the respective state affairs is of the *participation type*. If a quality has a value that is a quale region, then I will say that the respective state affairs is of the *value type*. All these types of states of affairs refer to particular states of affairs because all entities that constitute any state of affairs are particulars. Still, when the engineer speaks about functions, he usually has in mind not a particular state of affairs, but a type of states of affairs. Thus, identifying functions with states of affairs, I claim that behaviour and functions are types of states of affairs that may be represented as sets of particular states of affairs.

Now I will present a taxonomy of artefact functions based on DOLCE. The taxonomy develops the so-called Reconciled Functional Basis approach. The Reconciled Functional Basis (RFB) is one of the recent efforts towards establishing a standard taxonomy of artifact functions (see [14] and [5]). RFB is the result of reconciliation of two previous taxonomies: the NIST taxonomy (cf. [15]) and the older versions of Functional Basis developed in [8]. Each of these taxonomies is a result of empirical generalisation of engineering specifications. RFB follows the classic paradigm of Pahl and Beitz ([11]).

Because the RFB taxonomy of functions is flawed in a number of respects, I will refine it with the help of my general theory of artefacts and DOLCE. Given that artifact functions are states of affairs, the taxonomy of artifact functions should copy the aforementioned taxonomy of states of affairs. However there is one important exception: due to the DOLCE notion of atemporal inherence, no state of affairs of the i-type is an artifact function. In DOLCE a quality inheres in an entity either always or never. Thus, (on the first layer) I divide artifact functions into participation functions (p-functions), i.e. states of affairs of the p-type, and value functions (v-functions), i.e. states of affairs of the v-type. Metaphorically speaking, the former category corresponds to dynamic artifact functions, while the latter corresponds to static functions. Then, I divide both kinds of functions with respect to the kind of entities involved in the respective states of affairs.

If a physical endurant participates in an achievement (respectively in an accomplishment, state, or process), then the state of affairs (that consists in) that the endurant participates in the achievement (accomplishment, state, or process) is a function of the Achieve (Accomplish, Maintain, or Process) type, provided that this state of affairs is an artifact function at all. Thus, there are exactly four kinds of p-functions.

Similarly, when a quality of spatial location (respectively, of energy, temporal location quality) has a value that is a quale region, then the (v-type) state of affairs is a function of the Locate (Energate, or Temporate) type. The category of topological connectedness represents a set of qualities which are modelled in DOLCE in a peculiar way. Since no entity can lose any of its qualities without ceasing to exist, therefore when an endurant becomes topologically disconnected (without ceasing to exist), we model this fact saying that the endurants quality of topological connectedness changes its value from Yes (or 1 or the value that corresponds to the fact that something is topologically connected) to No (or 0 or the value that corresponds to the fact that something is not topologically connected). In an analogous way, we model any fact to the effect that some endurant is topologically connected (or disconnected). Now, when a quality of topological connectedness has the value Yes, the respective (v-type) state of affairs is a function of the Connect type provided that this state of affairs is an artifact function at all. If a quality of topological connectedness has the value No, then the respective state of affairs is a function of the Branch type. Thus, there are at least five kinds of v-functions: Locate, Connect, Branch, Energate, and Temporate.

We may also divide the four categories of the p-type functions. To this, end I annotate every state of affairs of the p-type with a pair of (possibly identical) v-type states of affairs: the initial state of affairs and the terminal state of affairs. Both of these states of affairs involve the qualities that are changed throughout the state of affairs of the p-type. Any quality of this kind will be called a quality associated with the state of affairs (of the p-type). More precisely speaking, a quality x of a kind φ will be said to be associated with a state of affairs y of a kind ψ iff for any state of affairs y of the kind ψ, there is a quality x of the kind φ such that x is changed throughout y. In general, there may be more than one quality associated with a given state of affairs of the p-type. The initial state of affairs corresponds to the initial values of these qualities and the terminal state of affairs corresponds to the terminal values. More precisely speaking, an initial state of affairs associated with a state of affairs x of the p-type is a v-type state of affairs that a quality associated with x has such a value that corresponds to the moment at which x comes into being. Similarly, a terminal state of affairs associated with a p-type state of affairs x is a v-type state of affairs that a quality associated with x has such a value that corresponds to the moment at which x ceases to exists. Consider, for example, the state of affairs that a coffee maker transports water from its reservoir through its heating chamber to the filter basket. The quality associated with this state of affairs (of the p-type) is the spatial location of water. The initial state of affairs (of the v-type) is that the spatial location of water has the quale of the spatial location of the reservoir of the coffee maker. The terminal state of affairs (of the v-type) is that the spatial location of water has the quale of the spatial location of the filter basket of the coffee maker.

Since stative perdurants are cumulative, therefore for any stative perdurant, the initial state

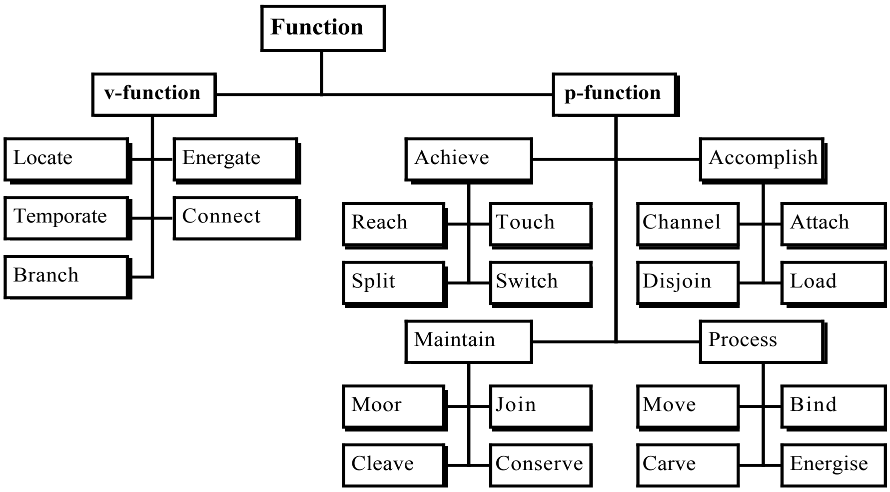

Figure 2: Taxonomy of artefact functions.

of affairs is identical with the terminal state of affairs; otherwise the mereological sum of two perdurants of a certain kind would not be of this kind. For the same reason, for any eventive perdurant, these states of affairs are different.

The results of this extension of the basic taxonomy of p-functions are presented in the following two tables.

	Spatial location	**Topological connectedness**		**Energy**	...
Achievement	Reach	Touch	Split	Switch	...
Accomplishment	Channel	Attach	Disjoin	Load	...
State	Moor	Join	Cleave	Conserve	...
Process	Move	Bind	Carve	Energise	...

	Initial state of affairs: Top. connect. is present	**Terminal state of affairs:** Top. connect. is present	**p-function**
Achievement	Yes	No	Split
Achievement	No	Yes	Touch
Accomplishment	Yes	No	Disjoin
Accomplishment	No	Yes	Attach
State	Yes	No	Join
State	No	Yes	Cleave
Process	Yes	No	Bind
Process	No	Yes	Carve

The final taxonomy of artefact functions is depicted in figure 2.

4 Information objects as artefacts

In the previous deliverable I constructed a formal ontology of documents, which ontology is expected to be a kind of paradigm for a future ontology of information objects of any kind. The starting point to the former ontology was the theory of document genres by J. Yates and W. Orlikowski.

I proposed to articulate the genre discourse by means of the following primitive notions:

1. two basic general ontological categories of endurants (ED) and perdurants (PD),
2. a specific relation of being a member of a community,
3. a complex general ontological category of situation-types (Sit),
4. a non-empty set $Time$ of time parameters (temporal moments or regions),
5. a specific ontological category of agents and three specific relations between agents' mental attitudes and situation-types (here represented as a set $Ment\_Sit$),
6. two specific relations of being a part of, one of which is atemporal ($\leqslant$) and the other is temporal ($\leqslant_t$).

I distinguished between a document genre and a communication genre. The former is instantiated by documents that are endurants; the latter is instantiated by documents that are perdurants. Any document of a document genre was called a document in the strict sense; any document of a communication genre will be called an act of communication or just a communication.

The genre was defined by its use and content.

The use element of a genre is to contain the recurrent situations in which the genre is referred to and the purposes for which it is referred to. The former aspect will be represented here by a set $Trigger$ of situation-types. $Trigger$ is to comprise all conditions that are necessary for production of a document of a given genre. Any element of $Trigger$ will be called a *trigger* both for the genre and for the documents of this genre. Because all triggers are situation-types, any document of a genre is associated with the same set of triggers. Similarly, the purpose aspect will be represented by a set $Purpose$ of situation-types. Each element of $Purpose$ will be called a *purpose* both of a given genre and of all documents of this genre.

The content of a genre consists of the medium and the language of the genre. The former is to represent the medium and structure components of the form aspect from the theory of Yates and Orlikowski. The latter is to represent the language component of theirs. The medium component of my concept of genre contains a set of genre supports and a relation among characteristic parts of these supports. A support for a genre is any document of this genre. Each support has its own mereological structure, which in the case of endurants may change over time. I argued that for each document from a given genre, there exists a set of its parts, which will be called *characteristic* for this genre, such that a set of characteristic parts of any other document from this genre is isomorphic to the former set. Any characteristic part of a document contributes to the structural specificity of this document as long as this specificity is determined by the genre to which this document belongs. I also argued that in the case of documents in the strict sense, it seems obvious that only their essential parts may be characteristic.

The content of a genre was represented as a pair $< Med, Lang >$, where a set Med characterises the medium aspect of the genre and a set $Lang$ characterises its linguistic dimension.

The medium of a document genre was represented as a pair $< Supp, \leqslant_{ch} >$, where

1. $Supp \subseteq ED$ is a non-empty set of supports of a given genre,

2. $\leqslant_{ch}$ is a subset of $\leqslant_{es}$ such that $\leqslant_{ch}$ is a partial order and

$$\forall x, y \in Supp < P_{\leqslant_{ch}}(x), \leqslant_{ch}> \text{ is isomorphic to } < P_{\leqslant_{ch}}(y), \leqslant_{ch}>, \quad (45)$$

   where $P_{\leqslant_{ch}}(x) := \{y \in ED : y \leqslant_{ch} x\}$.

The medium of a communication genre was represented as a pair $< Supp, \leqslant_{ch}>$, where

1. $Supp \subseteq PD$ is a non-empty set of supports of a given genre,

2. $\leqslant_{ch}$ is a non-empty subset of $\leqslant$ such that $\leqslant_{ch}$ is a partial order and condition 45 is satisfied, now $P_{\leqslant_{ch}}(x) := \{y \in PD : y \leqslant_{ch} x\}$

The language element of a document genre was modelled by a function *Lang* that maps a set of sets of equiform endurants into a set of sets of situation-types, i.e. if $X \subseteq \wp(ED)$, then $Lang : X \rightarrow \wp(Sit)$. Subsequently, if $X \in Lang(Y)$, then this means that any endurant from Y conveys a piece of information represented by X. The language element of a communication will be modelled by a function *Lang* that maps a set of sets of equiform perdurants into a set of sets of situation-types, i.e. if $X \subseteq \wp(PD)$, then $Lang : X \rightarrow \wp(Sit)$.

A genre x from a community y was defined a pair $< Use, Content >$ such that:

1. $Use =< Trigger, Purpose >$, where

   (a) $Trigger \subseteq Sit \wedge Trigger \cap Ment\_Sit \neq \emptyset$,

   (b) $Purpose \subseteq Sit \wedge Purpose \cap Ment\_Sit \neq \emptyset$,

   (c) $Ment\_Sit(y) \cap (Trigger \cup Purpose) \neq \emptyset$,

2. $Content =< Med, Lang >$, where $Med =< Supp, \leqslant_{ch}>$.

Assuming that situation-types are a kind of univeral states of affairs, if we compare this definition to 38, we can notice that

- genre triggers correspond to the operative dimension of artefacts because the former determine the conditions under which a given genre is evoked,

- for obvious reasons, genre purposes correspond to the teleological dimension,

- $\leqslant$ corresponds to the intentional dimension because characteristic parts of documents are fixed by documents' designs,

- *Lang* corresponds both to the intentional and epistemic dimension because the way in which documents are interpreted depends both on documents' designs and on linguistic competences of users of documents.

5 Evaluation and further work

The most important results of the work presented above include:

- a general theory of artefacts,

- a formal theory of documents,

- a formal theory of functional parthood,
- a taxonomy of artefact functions.

The work presented in this deliverable concludes the project aimed to provide a formal ontology of artefacts. This goal has been only partially achieved. The full formalisation is limited to the theory of design and functional parthood and the theory of documents. The general ontology of artefacts is in a semi-formal stage with a well-articulated conceptual basis. The missing parts include a better developed theory of agents and agentive causality including social agents and collaborative agency. Only when these theories are developed beyond their present embryonic stage, a formal theory of artefacts will be attainable.

Actually, the first step towards this end has been made. During the project I managed to build a logic of instrumental stit operator in which I defined the operator of agency: x sees to it that φ with the help of y. I focused on actions in which agents employs various instruments in order to achieve the desired outcomes. I explored the ontological structure of such actions and the semantic features of sentences by means of which we refer to them. The logical framework for this logico-philosophical enterprise is the theory of the so-called stit operator: …see to it that … (cf. [1]). I modified the original theory so that we could represent those events in which agents see to things with the help of physical objects. At the moment, however, the issue how to incorporate this the logic into the conceptual schema of the theory of artefacts is open.

References

[1] Nuel Belnap, Michael Perloff, and Ming Xu. *Facing the Future: Agents and Choices in Our Indeterminist World*. Oxford University Press, Oxford, 2001.

[2] B. Chandrasekaran. Representing function: Relating functional representation and functional modeling research streams. *Artificial Intelligence for Enigneering Design, Analysis and Manufacturing*, 19:65–74, 2005.

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

[4] D. C. Dennett. *Darwin's Dangerous Idea*. Norton, New York, 1995.

[5] J. Hirtz, R. B. Stone, D. A. McAdams, S. Szykman, and K. Wood. A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design*, 13:65–82, 2001.

[6] Y. Kitamura, T. Sano, K. Namba, and R. Mizoguchi. A functional concept ontology and its application to automatic identification of functional strucuteres. *Advanced Engineering Informatics*, 16:145–163, 2002.

[7] A. Kumar and S. Upadhyaya. Component-ontological representation of function for reasoning about devices. *Artificial Intelligence in Engineering*, 12:399–415, 1998.

[8] A. Little, Kristin Wood, and Daniel McAdams. Functional analysis: A fundamental empirical study for reverse engineering, benchmarking and redesign. In *Proceedings of the 9th ASME International Conference on Design Theory and Methodology*, September 1997.

[9] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. Wonderweb deliverabled18, 2003.

[10] Peter McLaughlin. *What Functions Explain. Functional Explanation and Self-Reproducing Systems*. Cambridge University Press, Cambridge, 2001.

[11] G. Pahl and W. Beitz. *Engineering Design: A Systematic Approach*. Springer, London, 1996.

[12] M. Rosenman and J. Gero. Purpose and function in design; from the socio-cultural to the technophysical. *Design Studies*, 19:161–186, 1998.

[13] M. Sasajima, Y. Kitamura, M. Ikeda, and R. Mizoguchi. Fbrl: A function and behavior representation language. In *Proceedings of IJCAI-95*, pages 1830–1836, 1997.

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

[15] Simon Szykman, Ram Sriram, and Janusz Racz. The representation of function in computer-based design. In *Proceedings of the 1999 ASME Design Engineering Conference (11th International Conference on Design Theory and Methodology)*, number Paper No. DETC99/DTM-8742, Las Vegas, Nevada, September 1999.

[16] Y. Umeda, H. Takeda, T. Tomiyama, and H. Yoshikawa. Function, behaviour, and structure. In J. S. Gero, editor, *Applications of Artificial Intelligence in Design*, volume 1, pages 177–194. Springer-Verlag, 1990.

[17] W. Y. Zhang, S. B. Tor, G. A. Britton, and Y. M. Deng. Efdex: A knowledge-based expert system for functional design of engineering systems. *Engineering with Computers*, 17:339–353, 2001.