



Título artículo / Títol article: B-Cube, Behavioural modelling of technical artefacts

Autores / Autors Chulvi Ramos, Vicente ; Vidal, Rosario

Revista: Computers in Industry, 2013, vol. 64, no 1

Versión / Versió: Preprint del autor

Cita bibliográfica / Cita bibliogràfica (ISO 690): CHULVI, Vicente; VIDAL, Rosario. B-Cube, behavioural modelling of technical artefacts. Computers in Industry, 2013, vol. 64, no 1, p. 68-79

url Repositori UJI: <http://hdl.handle.net/10234/87290>

B-Cube, behavioural modelling of technical artefacts

Abstract

A new model, B-Cube, is described for managing knowledge at the behaviour level of the function-behaviour-structure framework. The model proposes a three-dimensional approach to the behavioural modelling of technical artefacts using definitions based mainly on the meta-ontology DOLCE as behaviour concepts.

The present work aims to show the complementarity between these terms and those from the NIST functional basis in functional design, assuming that this model achieves similar objectives with behaviours to those obtained by the NIST functional basis with functions: representation of behaviours in CAD and KBS, a scheme for the modelling of behaviours and a universal set of behaviours. The modelling language IDEF was adapted to exemplify graphically the modelling of technical artefacts in the FBS framework using B-Cube terminology at the behaviour level.

Keywords: Knowledge management; Behaviour; Functional design; FBS; Ontology

1 Introduction

The aim of this article is to present this new model to functional design through the FBS (Function-Behaviour-Structure) framework that can be implemented in a solution-synthesis system and act as a link between different libraries of software applications. This model proposes a three-dimensional approach that uses definitions as behaviour concepts. It is assumed that this model achieves similar objectives with behaviours to those obtained by the NIST (National Institute of Standards and Terminologies) functional basis with functions, i.e. the representation of behaviours in Computer Aided Design (CAD) and Knowledge Based Systems (KBS), a scheme for the modelling of behaviours and a universal set of behaviours. It is able to solve, a priori, the shortcomings found in previous research, when trying to link functional design with TRIZ based Computer Aided Inventing (CAI) tools [1], e.g. loss of information within the use of taxonomies at the functional level (such as ambiguities, synonyms and functions without correlations). The present work aims to show the complementarity between these terms and those from the NIST, the Reconciled Functional Basis (RFB), [2] in functional design. The B-Cube model is based mainly on DOLCE's meta-ontology (Descriptive Ontology for Linguistic and Cognitive Engineering) [3, 4] and on Garbacz's functional development of it [5, 6].

The concepts of function and behaviour, as a part of the Function-Behaviour-Structure (FBS) framework [7, 8], are at the core of our research. The FBS framework is widely used among designers for design process analyses, as it can represent the evolution of the design state from the study of protocols [9]. More recently, Gero and many other researchers have extended the study of FBS representation [10-14]. Within this framework, function is the abstract purpose the design is oriented towards. When the function is carried to a lower level of abstraction and defines how the device or its components will be related to the uses for which they are employed and designed, we are defining the term behaviour. Devices and their parts have physical structures and these structures and their relation with the environment determine the behaviours, which are related to the functions of the device [5, 15-17].

The FBS framework allows computational modelling to be carried out, that is, software applications can be produced that are able to use search and explore procedures in order to find and combine design-solving procedures for a problem represented by functions. Thus, several authors have tried to develop approaches and software

applications to implement FBS-based procedures [18], while others have attempted to model function and/or structure libraries to be implemented in functional reasoning processes [19-23]. The possibilities of these systems have been increased by function classifications achieved by means of hierarchies [15] and the use of taxonomies.

A taxonomy consists of a group of concepts and relationships that are organised hierarchically and whose concepts can be arranged as classes with sub-classes [24]. Taxonomies were introduced into the industrial world by the hand of Gershenson and Stauffer [25], but Szykman et al. [26] were the first to differentiate functions with their extensive review of function terminologies inside the engineering context from 1976 until 1998. Since then, several function taxonomies have been developed [27, 28], as well as behaviour taxonomies [5, 29-31] and structure taxonomies [32]. In the case of function taxonomies, the most significant are those provided by the NIST (National Institute of Standards and Technology) [2] and those based on DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) [6, 33].

The RFB allows overall product functions (especially from the electromechanical and mechanical domain) to be modelled as sets of connected elementary sub-functions. Function is described in a verb-object form and represented by a black-box operation on flows of materials, energies and signals. A sub-function is also described in verb-object form but it is represented by a well-defined basic operation on well-defined basic flows of materials, energies and signals. The black-box operations on general flows representing product functions are derived from customer needs, and the basic operations and basic flows representing sub-functions are arranged in libraries. Some authors have remarked that the descriptions of operations-on-flows may be better understood as representing the behaviour of products and of their components rather than functions [15, 34, 35].

RFB supports a number of engineering tasks, including the archiving, comparison and communication of functional descriptions of existing products and the engineering design of new products. For example, RFB has been used to develop and refine a web-based repository of design knowledge. This repository (which includes descriptive product information such as functionality, physical parameters of components, manufacturing processes, failure modes and component connectivity) contains detailed design knowledge on consumer products and the components that comprise them. Design generation tools, like function-component matrices and design structure matrices, can be readily created from single or multiple products and used in a variety of ways to enhance the design process [36]. The functional basis is applied even outside the engineering field, for modelling functional processes, manual operations and human-centric procedures [37].

So, a formal function representation is needed to support functional modelling, which helps to clarify the meanings of terms and also to support representation of device knowledge for automated reasoning [38]. In this regard, research efforts are being made to move Functional Basis towards a functional modelling language [39]. Another line of research is the work of Garbacz [6], who reviewed RFB and refined it formally, with the help of the conceptual framework of the DOLCE ontology [3]. This research line served as the inspiration for our model, B-Cube (Behaviour's Cube), which is introduced in the present article.

Also, in section 7 a graphical modelling approach is proposed to model technical artefacts in the FBS framework by adapting the modelling language IDEF4 [40] and using the B-Cube's terminology for behaviour's representation. The aim is to achieve an intuitive, easy-to-understand model that can be used by any designer, and to represent the examples that are included at the end of this article to illustrate more clearly how behaviours are modelled with B-Cube.

2 Functions and behaviours

Despite the importance of function and behaviour in engineering design, there are still some fundamental ambiguities and confusion regarding their definition. The disadvantage of lacking conceptual consensus becomes an important issue when functional and behavioural descriptions have to be shared. This occurs when, for instance, designing is modelled as a procedure in which existing knowledge about the relations between the functions, behaviour and physical structure of artefacts is partially retrieved from knowledge bases. In such cases, having a common set of definitions is essential [41].

Chandrasekaran distinguishes between two general approaches towards defining the functions of technical artefacts, called the functional representation approach and the functional modelling approach [34]. The two approaches involve performing research that is mutually complementary. First, functional representation research provides the basic layer for the device ontology in a formal framework that helps to clarify the meanings of terms such as function and structure, as well as supporting representation of the device knowledge for automated reasoning. Second, functional modelling research provides another layer in the device ontology by attempting to identify behaviour primitives that are applicable to subsets of devices, with the hope that functions can be described in those domains with an economy of terms. This can lead to useful catalogues of functions and devices in specific areas of engineering. With increased attention to formalization, work on function modelling can provide domain-specific terms for function representation research in knowledge representation and automated reasoning.

Functional modelling and functional representation might merge over time. Ontologies of the sort being developed by function modellers are certainly going to be useful for device knowledge representation because the current body of representational primitives in artificial intelligence do not have terms for the properties, behaviours and functions of devices in specific domains [34].

In relation to the functional representation approach, Chandrasekaran and Josephson [42] isolated five meanings of behaviour and two of function. The meanings of behaviour are characterized using the primitive notion of state variable:

- Behaviour as the value of some state variable of the artefact or a relation between such values at a particular instant.
- Behaviour as the value of a property of the artefact or a relation between such values.
- Behaviour as the value of some state variable of the artefact over an interval of time.
- Behaviour as the value of some output state variable of the artefact at a particular instant or over an interval.
- Behaviour as the values of all the described state variables of the artefact at a particular instant or over an interval.

The two meanings of function distinguished by Chandrasekaran and Josephson are called the device-centric and environment-centric meanings. Without going into detail, a device-centric function of an artefact is a behaviour of the artefact that is selected and intended by some agent (in device terms). It is a function that is described in terms of the properties and behaviours of the artefact only; an example of a device-centric function is 'making sound' in the case of an electrical buzzer. An environment-centric function is, in contrast, an effect or an impact of this behaviour of the artefact on its environment, provided this effect or impact is selected and intended by some agent

(the 'why' of the device). This kind of function is conceptually separate from the artefact that performs or is expected to perform the function; 'enabling a visitor to a house to inform the person inside the house that someone is at the door' is an environment-centric function of the buzzer.

Functional modelling includes the functional basis model by Stone and Wood [43] and the RFB [2]. Stone and Wood modelled the overall product functions of technical artefacts, especially from the electromechanical and mechanical domain, as sets of connected elementary sub-functions. An overall product function of an artefact is defined as a general input/output relationship of the artefact having the purpose of performing an overall task, and is represented by a black-box operation on flows of materials, energies and signals. A sub-function performs a part of that overall task and is represented by a well-defined basic operation on well-defined basic flows of materials, energies and signals, which are arranged in libraries that list all the possible basic operations and basic flows.

Deng [15] attempts to address the problem of integrating the purpose and the operation functions (sub-functions) in the functional modelling approach with function and behaviour in the functional representation approach by restricting the designs to the conceptual stage of mechanical products. According to Deng, the function at the same level of abstraction as behaviour is the operational function and it can be more specifically referred to as an action function, which is defined as 'a physical interaction between two objects of interest, each of which may be a component of a design or the design itself and its environment'. Hence, function can be semantically classified into two types: a purpose function and an action function. A purpose function is 'a description of the designer's intention or the purpose of a design', whereas an action function is 'an abstraction of intended and useful behaviour that an artefact exhibits'. In the same line, Vermaas [35] argued that the operations-on-flows descriptions (sub-functions), as used by Stone and Wood, may be better understood as representing the behaviour of products and of their components.

3 Reconciled functional basis

The RFB [2] is a reconciliation and integration of other taxonomies but mainly of the research conducted at the NIST [26] and the functional basis effort [43]. The main aim of the NIST is to develop a formal representation of functions with a hierarchical vocabulary of standardized terminologies, focused on mechanical design.

The NIST's work stems from three specific needs: the representation of functions in Computer Aided Design (CAD), a fixed scheme for modelling functions and a universal set of functions performed by mechanical systems. The greatest achievements reached by NIST are reduced ambiguity and increased uniformity. To reduce ambiguity they defend that the more terms are used to refer to the same concept (synonyms), the greater the number of different ways to model a given concept there will be. Increased uniformity attempts to facilitate the exchange of function information among different applications.

RFB follows the classic paradigm of Pahl and Beitz [44] in defining artefact functions in terms of flows. Pahl and Beitz define a function as a relation between an input and an output of an artefact (under a specific goal). A flow is either a conversion of material, a conversion of energy or a conversion of signal.

This hierarchically organized vocabulary arranges the terms onto three levels of specification. The higher level term provides full coverage of the meaning of the terms included in this class. Moreover, terms at the same level within the same class are mutually exclusive (Table 1).

Table 1. Extract of the Reconciled Functional Basis [2]

Primary Function	Secondary Function	Tertiary Function
Branch	Separate	Divide Extract Remove
	Distribute	
Channel	Import	
	Export	
	Transfer	Transport Transmit
	Guide	Translate Rotate ...
...		

Garbacz [6] criticized RFB for having a number of different shortcomings such as the lack of principles to explain the three divisions, the lack of exclusivity, the lack of exhaustiveness in some divisions, and problems of ambiguity in terms and examples. Garbacz proposed the philosophical category of states of affairs as the most adequate conceptual category of artefact functions and the use of the DOLCE ontological taxonomy [3].

4 Dolce ontology

DOLCE [3] is presented as an upper-level ontology or meta-ontology. In order to understand the development of the B-Cube model, it is necessary to start by defining some concepts belonging to DOLCE (Figure 1) that were used to develop the B-Cube model, namely the descriptions of endurant (ED), perdurant (P) and quality (Q).

An endurant is an entity, all the parts of which are present whenever the entity is present at any time in its existence, that is, the entity is wholly present throughout time. This meaning corresponds to the definition of structure in the FBS framework, so endurants are used as elements (artefacts or parts) that behaviours are related to. Since the B-Cube model refers to the behaviour level, endurants do not appear directly in the model because they are related to the structure level. Despite this, endurants help to define the input of the B-Cube and, hence, the distinction between physical endurants (PEDs) and non-physical endurants (NPEDs) is still meaningful. PEDs correspond to tangible structures like a human person, a bottle or a screwdriver, while NPEDs correspond to the abstract domain, like a legal person, a social agent or a role carried out by a human.

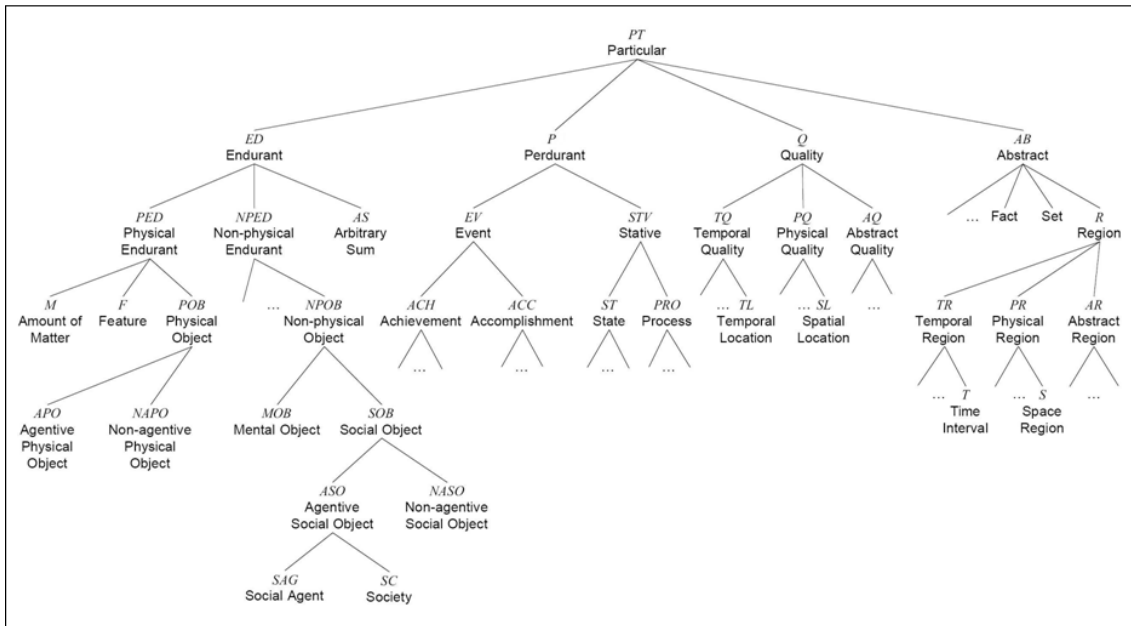


Figure 1. DOLCE's ontology [3].

A perdurant (P) was originally defined as an entity that is not wholly present over time, that is, not all its parts are present throughout all the time in which the entity is present. A perdurant “happens” in time, so during all the time that the entity is present, only some temporal parts of it are present at any given moment. In B-Cube, perdurants refer to the kind of behaviour that the structure (endurant) carries out. Thus, perdurants are necessary but are not enough on their own to describe behaviours. The list of perdurants provided by DOLCE seems to cover all the needs in B-Cube in this respect. These perdurants are achievement, accomplishment, state and process. Furthermore, if we look at the work of Garbacz [6], it can be seen that he agrees with these definitions and applies them to define the sub-taxonomy of participation functions, where they are understood as occurring when an endurant participates in a perdurant.

Figure 2 shows the classification algorithm of these terms according to the DOLCE definition: Cumulative (or stative) is when the mereological sum of two cases of the same type maintains that same type. For example: an occurrence of corrosion is cumulative since the sum of two instances of corrosion is still an occurrence of corrosion. Within cumulative occurrences, we distinguish between states and processes according to homeomerity. Homeomeric is when all the temporal parts are described by the same expression used for the whole occurrence, so corroding is classified as a state but shaping is classified as a process, since there are (very short) temporal parts of a shaping that are not themselves shaping. Non-cumulative (or eventive) occurrences are classified according to their atomicity. Atomic is when the case is immeasurably short in time, that is, it has no proper parts. Thus, the event is an achievement if it is atomic, otherwise it is called accomplishment. So ‘to switch off’ is atomic ($t \rightarrow 0$), while ‘to extract’ is non-atomic (the process of extracting something is clearly supposed to be > 0).

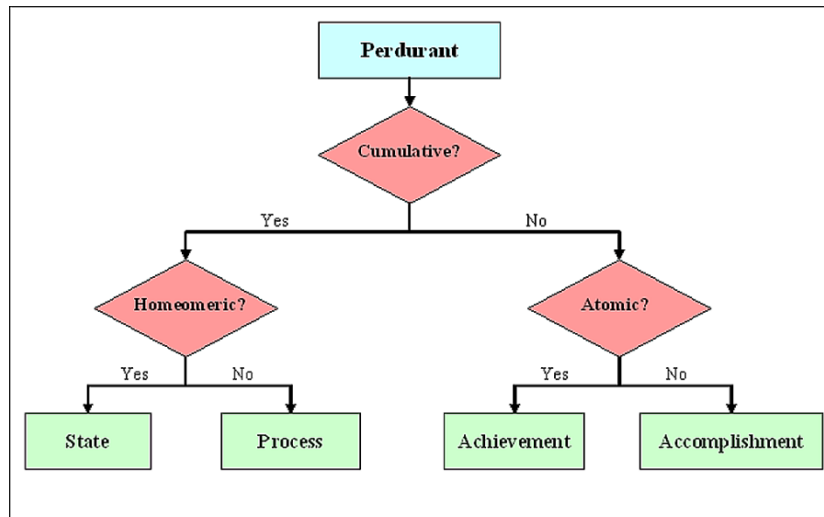


Figure 2. Classification algorithm of Perdurants according Masolo et al. [3].

Finally, qualities are defined as the basic entities that can be measured and are inherent to other entities (endurants or perdurants). Thus, every entity has a certain number of qualities that define it. DOLCE distinguishes between physical qualities (PQs), abstract qualities (AQs) and temporal qualities (TQs). PQs are those that are inherent in PEDs (e.g. the weight, the position in space or the energy state). DOLCE provides only one definition within this group (spatial location), but it leaves the group open to new parameters, that is, it explicitly makes room for other members in some of its divisions. Garbacz [6] took advantage of this fact to increase the number of PQs to three, with the addition of topological connectedness and energy. However, these three concepts do not fulfil all the needs that B-Cube generates in order to define all behaviours. For this reason the concept of the Black Box model, defended by several authors and in agreement with the NIST research, is used here [39, 44, 45]. This model emphasizes the fact that for each different kind of flow, when it acts over a specific function which is represented by a Black Box with input and output flows, this function displays a different behaviour depending on the type of flow that has acted on it. It can therefore be deduced that the three kinds of flows, e.g. energy, signal and material, are suitable for defining behaviours and, consequently, these terms can be added to the group of PQs. To sum up, the PQs used in B-Cube are:

- Spatial location: related to the position of a PED in space. Moving an object belongs to this category.
- Topological connectedness: concerned with the sort of connection at the topological level on which the PED is located. Breaking or joining an object corresponds to this group.
- Energy: refers to the energy state of the PED. Freezing water or charging a battery are examples of behaviours classified within the energy group.
- Material: related to the physical magnitude or material property of the PED that is affected by the perdurant. Increasing the weight or changing the colour of an object are examples that correspond to this group.
- Signal: related to actions involving PEDs when they act as signals. Examples could be increasing a wave or a mobile phone that is sending a signal.

On the other hand, when behaviour is related to an NPED instead of a PED, it must be defined by an AQ. Although both DOLCE's and Garbacz's work defines AQs as those related to NPEDs, it does not put forward any value for them, since this region does not affect the artefacts themselves. Thus, the AQ region is set aside for future work.

TQs are those that are inherent in perdurants. They refer to the way that PQs or AQs are affected by perdurants over time. As in the case of PQs, DOLCE provides only one definition within this group, temporal location (TL), and it leaves the group open to new parameters. But in this case, instead of increasing the group's parameters, Garbacz [6] developed TL with the terms initial SoA (State of Affairs) and final SoA. In the B-Cube model, TL is taken as being equal to TQ in order to take advantage of Garbacz's approach. Thus, initial SoA is defined as a behaviour which a perdurant performs by eliminating or reducing an initial PQ, e.g. cooling an object makes it lose its initial energy. Final SoA works in the opposite way, that is, the PQ is obtained (or increased) as a consequence of the perdurant. Thus, heating an object causes it to obtain a final energy. In order to cover all the range of behaviours, it is necessary to add a term to describe those which do not act by reducing or increasing PQ. For example, converting energy does not change the energy level (First Law of Thermodynamics), but it is a behaviour that needs to be described. This new term has been named immutable SoA. At this point it is important to emphasize that the application of the definition of TL in the PQ spatial location needs to be carried out in agreement with the other terms. It is therefore easy to see that initial SoA in energy means that the energy is at the beginning and the action decreases it; in topological connectedness, it means that the object is connected initially and then loses this connection. But it may not be so easy to perceive how TL affects spatial location, because an object always has a position in space. Thus, by analogy to other terms, TL is a relative position. In other words, initial SoA means that the initial position of an object is close to another element and the action separates the object; in the same way, when an object has topological connectedness with another, they are closer than if they had lost that connection, although this occurs on a microscopic scale.

5 B-Cube

B-Cube is a model for the representation of knowledge of the behaviours layer within the Function-Behaviour-Structure framework based on the DOLCE ontology. The basic idea when using the DOLCE ontology is that a behaviour b of a technical artefact a in a perdurant e , where a perdurant is an entity that is only partially present at any time it is present, is the specific way in which a occurs in e [41].

Although Borgo [41] formalize artefact functions as intended behaviours, in this state of the research the authors have preferred to use Deng's concept of Purpose Function [15] in the function layer and use the RFB taxonomy directly for them.

In the previous section we described the terms endurant (ED) – physical (PED) and non-physical (NPED), perdurant (P), physical qualities (PQs), abstract qualities (AQs) and temporal qualities (TQs) as they are used in the B-Cube model. Endurants were defined as structures or elements. A structure that is carrying out a function has a specific behaviour, so it is used as input in B-Cube. As the number of endurants in the universe tends to infinity, it does not seem to be practical to classify behaviour in classes. But it has been shown that endurants are related to qualities (PEDs to PQs, and NPEDs to AQs), and these exist in a finite number. Therefore, PQs and AQs describe the X-axis of the B-Cube. In the present article the X-axis will refer only to its physical part, since, as said above, the abstract part has been set aside for future work. On the other hand, perdurants were defined as 'the kind of behaviour' and so, by definition, perdurants can be used to define a behaviour. They are assigned to the Y-axis. Moreover, TLs were also related to the previous terms, since they were defined as the 'direction' in which a perdurant affects a PQ. As a result, TLs are set on the Z-axis, and they complete the three dimensions needed to build all the vectors that define behaviours in the B-cube model. Figure 3 shows the framework of the B-Cube model with the three axes that make up the model and where the previously-mentioned values will be allocated.

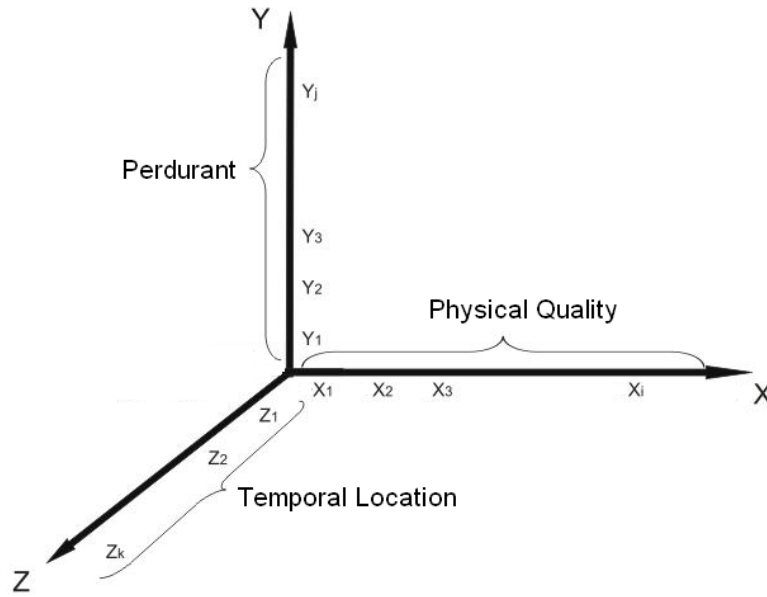


Figure 3. Axes for the representation of Behaviours.

Therefore, behaviour is now represented as (x_i, y_j, z_k) , where x_i refers to the PQ affected by the behaviour, y_j is the kind of perdurant, and z_k means the temporal location of the perdurant. Figure 4 represents the physical part of the B-Cube model with all its values, which is better explained in Table 2. So, for example, a blowtorch that is used to weld pipes has a main behaviour $(2, 3, 3)$. That is, $X = 2 =$ topological connectedness, due to the fact that welding refers to a physical state of connection. $Y = 3 =$ accomplishment, from the algorithm in Figure 2, it can be seen that welding pipes is non-cumulative, because to weld a pipe + to weld a pipe = to weld two pipes, and it is non-atomic, because to weld pipes requires more than an instantaneous action. And $Z = 3 =$ final SoA, because of the fact that topological connectedness is not present at the beginning, but it is obtained in the end.

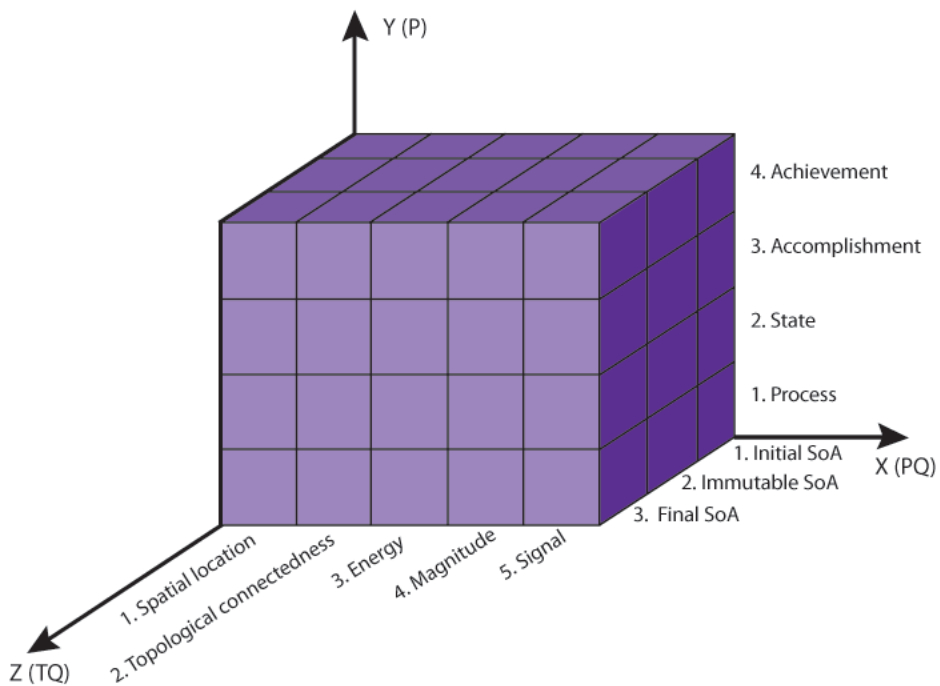


Figure 4 B-Cube model

Table 1. Definition of B-Cube terms

Axis	Value	Term	Significance	Examples
X (Physical Quality)	1	Spatial location	Position of a PED in space	Move an object
	2	Topological connectedness	The kind of connection at a topological level on which a PED finds itself	Break an object Join an object
	3	Energy	Energetic state of a PED	Freeze water Charge a battery
	4	Material	A physical magnitude of the PED that is affected by the behaviour	Increase weight Change colour
	5	Signal	Actions referred to PEDs when they act as signals	Increase a wave A mobile phone sending a signal
Y (Perdurant)	1	Process	The behaviour is cumulative and non-homeomeric	To run
	2	State	Cumulative and homeomeric	To sit
	3	Accomplishment	Non-cumulative and non-atomic	To give a lecture
	4	Achievement	Non-cumulative and atomic	To break a glass
Z (Temporal location)	1	Initial SoA	The behaviour makes the initial PQ or AQ reduce or disappear	To cool an object
	2	Immutable SoA	The behaviour does not vary the degree or quantity of PQ or AQ affected by it	To convert energy
	3	Final SoA	The behaviour makes the degree or quantity of PQ or AQ increase or appear	To warm an object

The B-cube model can also present an abstract part, which is not represented in the RFB, by using AQs instead of PQs. NIST does not take the abstract actions into consideration because it is focused on mechanical design. This fact is of no importance when designing physical objects, devices or artefacts, where the NIST functional basis works perfectly well, but the abstract part of B-Cube leaves an important door open to organizational and process design. Here there are some structures that can act as physical things, but they can also support an abstract behaviour or role [46, 47], like the case of human beings or artificial intelligence devices.


6 B-Cube version of Reconciled Functional Basis

The B-Cube model was deployed with each of the possible values for the three axes X, Y and Z (according to Table 2) and for each of these elements a search was conducted to find the verb that fitted the secondary and tertiary levels of the RFB taxonomy best. Despite the fact that tertiary level terms are supposed to be included in secondary level ones, the usage of both levels for establishing correlations is justified by two main reasons: the level of abstraction expected to achieve with B-Cube model makes that the slight differences of meanings between levels become quite important, and that NIST also uses these levels when establishing correlations with other taxonomies [2].

Correlations were made by taking the B-Cube terms one by one. For each of these terms, a function from the NIST functional basis is sought for which the behaviour can be a concrete action of its abstract meaning. As a result of these correlations, we can appreciate that one function verb from the RFB can be performed in several specific ways, that is, behaviours, in B-Cube. Table 3 shows an example of two different ways of carrying out the same function expressed by the RFB verb 'remove'. Both structures (objects or substances) act by causing the target to lose its initial (Z=1) topological connectedness (X=2), but while the sander machine entails a long-term continuous and

non-homeomeric process ($Y = 1$), ambient corrosion acts in a long-term continuous homeomeric way ($Y = 2$).

Table 2. Examples of two different behaviours that perform the function ‘remove’

		
FRB	Remove	Remove
B-CUBE	(2,1,1)	(2,2,1)

Another case is that two different verb functions in RFB can lead to the same behaviour in B-Cube. This happens when one of the possible ways to achieve a function can be described by the same three properties as another one. For example, when the function ‘translate’ means a process ($Y = 1$) of moving an object in space ($X = 1$) without a defined direction, that is, neither approaching nor moving away ($Z = 2$), the behaviour is the same as the function ‘rotate’, where it acts as a process of spinning an object, and so it neither approaches nor moves away from the reference point. This does not mean that the B-Cube model lacks unambiguity. On the contrary, it aims to increase the specificity presented by the FRB, since the term ‘translate’ could also mean to bring the object closer ($Z = 3$) or to move it away ($Z = 1$). The simple word ‘translate’ can also be understood as a movement carried out in a more instantaneous way ($Y = 4$).

Table 4 shows the assignments of the B-Cube terms to RFB verbs. The third column shows other corresponding terms in order to make it easier to grasp the specific meaning of each behaviour. In the same table it can be seen that there are some behaviours represented in the B-Cube model that have no correlation with RFB, since the definitions of behaviours in B-Cube were developed by combining their possible parameters, which ensures that no term or meaning can be omitted.

Table 3. Assignment of B-Cube terms to RFB verbs

B-Cube	RFB	Other correspondences
(1, 1, 1)	Remove; Extract	Bail out
(1, 1, 2)	Translate; Rotate	Move; Vibrate; Spin;
(1, 1, 3)		Implant; Insert; Install
(1, 2, 1)		Repel; Keep separated
(1, 2, 2)	Allow DOF; Secure	Not Allow DOF; Hold up; Orient
(1, 2, 3)		Attract; Keep close
(1, 3, 1)	Guide; Remove	Shift
(1, 3, 2)	Position	Align, Orient
(1, 3, 3)		Introduce; Put in; Approach
(1, 4, 1)	Separate; Export	
(1, 4, 2)	Stop	Locate
(1, 4, 3)	Import	Reach
(2, 1, 1)	Remove	Carve; Polish; Clean; Erode

B-Cube	RFB	Other correspondences
(2, 1, 2)	Stabilize	Stabilize
(2, 1, 3)	Secure; Couple	Fix; Fuse; Bind; Screw
(2, 2, 1)	Remove	Cleave; Corrode
(2, 2, 2)	Contain	Protect; Shield
(2, 2, 3)	Mix	
(2, 3, 1)	Separate	Disjoin; Disincrust
(2, 3, 2)	Inhibit; Secure	Insulate; Retain
(2, 3, 3)	Join	Assemble
(2, 4, 1)	Divide	Split; Tear; Rip; Disincrust
(2, 4, 2)		Block
(2, 4, 3)	Link	Touch; Prick; Stick; Attach
(3, 1, 1)	Collect	Consume
(3, 1, 2)	Transmit; Convert	Conduct; Channel; Transform
(3, 1, 3)	Convert	Generate; Energize
(3, 2, 1)	Convert	Cool; Freeze; Condense; Solidify
(3, 2, 2)	Store	Conserve; Transform
(3, 2, 3)	Convert	Warm; Evaporate; Melt
(3, 3, 1)		Unload; Discharge
(3, 3, 2)	Regulate	Regulate
(3, 3, 3)		Load; Charge
(3, 4, 1)	Prevent	Switch off; Stop; Turn off
(3, 4, 2)	Guide	Switch
(3, 4, 3)	Actuate	Switch on; Turn on
(4, 1, 1)	Shape	Compact; Compress
(4, 1, 2)	Condition; Shape	Adapt; Prepare; Deform
(4, 1, 3)		Expand; Stretch; Enlarge
(4, 2, 1)	Decrement	Reduce; Dampen; Weaken
(4, 2, 2)	Measure	
(4, 2, 3)	Increment	Magnify; Strengthen; Increase
(4, 3, 1)	Decrement	Decrement; Attenuate
(4, 3, 2)	Change	Normalize; Adjust
(4, 3, 3)	Increment	Amplify
(4, 4, 1)	Decrease	
(4, 4, 2)	Change	Invert
(4, 4, 3)	Increase	
(5, 1, 1)	Display; Export	Emit
(5, 1, 2)	Transfer	Identify
(5, 1, 3)	Detect; Import	
(5, 2, 1)	Display	
(5, 2, 2)	Transmit	Conduct; Guide
(5, 2, 3)	Sense	
(5, 3, 1)	Display	Emit
(5, 3, 2)	Process	Compare

B-Cube	RFB	Other correspondences
(5, 3, 3)	Indicate	Record; Register; Import
(5, 4, 1)	Indicate	Show
(5, 4, 2)	Process	Check
(5, 4, 3)	Measure	Locate

B-Cube was built from the DOLCE meta-ontology, meaning that there is knowledge hidden inside the term, or taxon. Thus, the functional basis is a taxonomy, which means that the tertiary class term 'join' belongs to the secondary level group 'couple', which is at the same time a part of the primary group 'connect'. As can be seen in Figure 5, when we structure the B-Cube terms hierarchically, they are systematically repeated. But this is because it is not a simple classification. Here, each chosen branch provides the final term with a specific and unambiguous meaning. This is one reason for defending the idea that the path is more important than the final term in the B-Cube model. As a result, when we are talking about the path (2, 4, 3), we are referring to an action that affects the topological connectedness ($X = 2$) of an object, in a non-cumulative and atomic way ($Y = 4$), and which makes the object achieve that connectedness ($Z = 3$), regardless of the name we have chosen to designate that path. By definition, several terms as 'touch', 'prick', 'stick', 'attach', 'link', 'join', and others can fit that meaning, but the same terms can also be understood as another path depending on the user. This is one of the main problems that authors find when trying to set up a taxonomy, as could be seen in the work on the RFB set of functions carried out by Hirtz [2]. There, it can be seen how several terms in correspondences appear in different positions within the reconciled taxonomy, depending on the author's perception of it. For example, one author can understand the verb 'join' as (2, 3, 3), another as (2, 1, 3) and a third one as (2, 4, 3), but the way to achieve this final joined state is different. Hence, a single word could be set for each term within a taxonomy or ontology just to make the model easier to understand, but for the time being it seems better to keep a vector terminology in order to avoid the ambiguity that may be present in the interpretation of one simple verb.

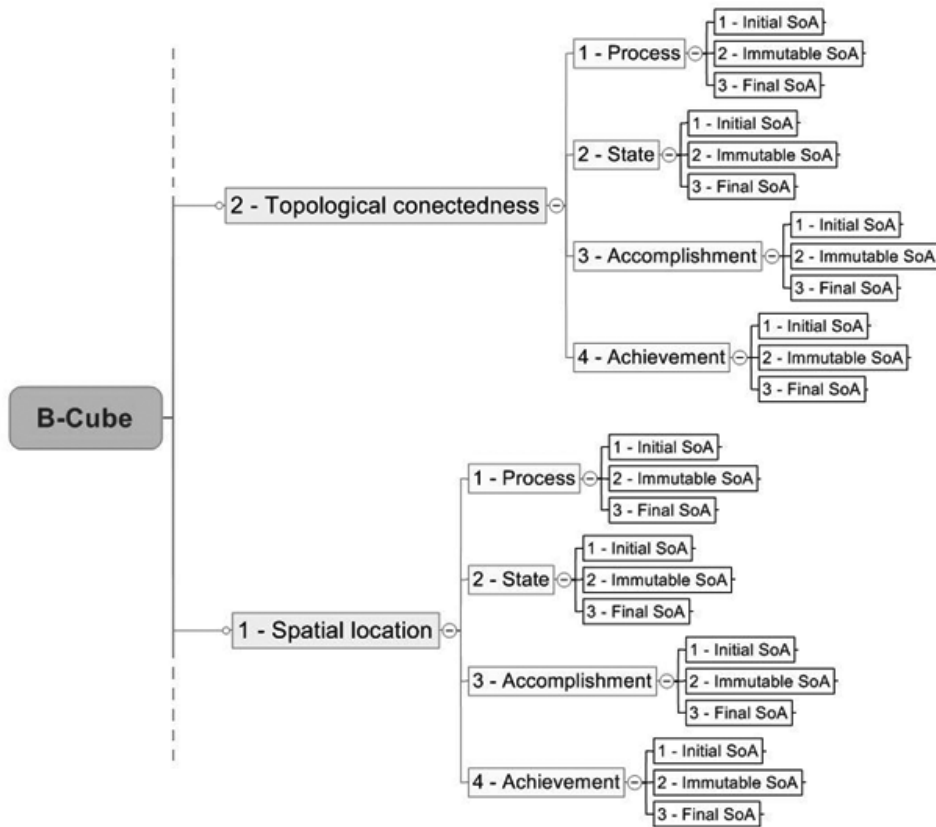


Figure 5. Extract from the B-Cube ontology.

7 Graphical modelling

In order to represent the models graphically, the authors have resorted to the concepts of the IDEF3 [48] and IDEF4 [40] modelling languages. IDEF3 is oriented towards process design and IDEF4 towards the design of objects.

The basic IDEF4 box, which has rounded corners, was selected to represent functions. The same kind of box was chosen to represent behaviours, but in this case a left lateral band was added in order to differentiate them from functions. Regarding structures, the basic IDEF4 box, which has square corners, was taken in this case. Structures regarding solution design and environment restrictions are differentiated by means of a left lateral band in the latter, as in the case of functions and behaviours. All these boxes have their header divided into two differently sized spaces, the wider one for the name of the function/behaviour/structure and the narrower one for its number and level, as in the IDEF3 modelling language. This name function, behaviour or structure corresponds to the purpose function in the NIST RFB taxonomy in the case of representing functions, to the terms of the B-Cube model when representing behaviours, and structures are called by their usual names as they appear in dictionaries. On the other hand, the numbers that are used in the boxes are simple units for first level components (1, 2, 3, ...), tens for the second level, where the first digit corresponds to the preceding first level component, hundreds for the third level, and so on. Thus, for example, a behaviour numbered 11 is the first behaviour derived from the behaviour labelled as 1. Numbers are situated in the upper right corner in the case of functions and structures, and in the upper left corner in the case of behaviours and restrictions, so as to be able to differentiate between them better. The body of the boxes is also divided into two parts, but in this case the division is horizontal. The upper part is for showing the active attributes in the design being represented, while the lower part is for

showing the rest of the possible but not active attributes. Additionally, if the box is represented with a black shadow, it means that this function/behaviour/structure comprises a sub-diagram. A sub-diagram is defined as a high-detail representation of one part of a diagram. For example, a wheel can be a complete part of the diagram representing a bicycle, while it can present a sub-diagram where it is sub-divided into its smaller parts: rim, tyre, spokes, and so forth. Sub-diagrams are represented on separate sheets and each numbering in that new diagram is presented as a numbered scheme with the first digit indicating the source component. For example, if structure 3 has a sub-diagram, all structures within this new diagram are preceded by a 3 (3.1; 3.2; 3.21; etc.). Figure 6 shows the four different kinds of boxes.

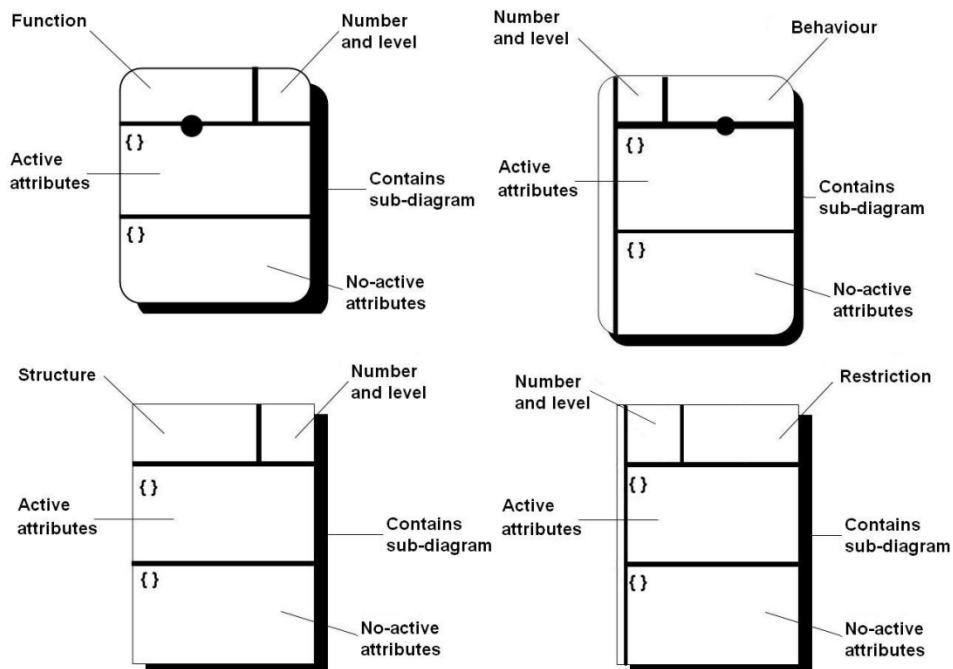


Figure 6. Box representations for functions, behaviours, structures and restrictions.

These boxes are represented in three layers corresponding to function, behaviour and structure (restrictions are represented in the same layer as structures). Different kinds of relationships between different components of the diagram are shown in Figure 7. Functions represent their correspondence with behaviours by means of arrows. The relations between one behaviour and those on a lower level can be represented in two different ways, depending on whether the new lower behaviour is caused or required by the upper level one. The relation of a behaviour caused by the previous one is represented by a line that ends in a black triangle, and the relation of a behaviour required by the previous one is represented by a line that ends in a white triangle. The relations between behaviours and structures or restrictions are represented with a line with a small box like the ones used in IDEF3 to indicate the 'and' and 'or' sentences, but in this case an 'O' will appear inside the boxes if the structure or restriction is the object of the corresponding behaviour and an 'S' if it is the subject. Furthermore, the letters 'O' and 'S' also have a subscript with the number of the corresponding behaviour that affects the structures or restrictions, in order to make the diagram easier to understand. Lastly, also with the aim of simplifying the diagrams, two implicit behaviours commonly present in all artefacts are represented there directly without the need for the corresponding behaviour boxes. These behaviours are (2, 2, 2), which refers to the state of one structure containing another one, and (2, 2, 3), referring to the state of one structure attached to another one. The first case is represented by an upper-lower level relation between structures in the form of a simple line. The second case is represented by drawing the two boxes joined by one side.

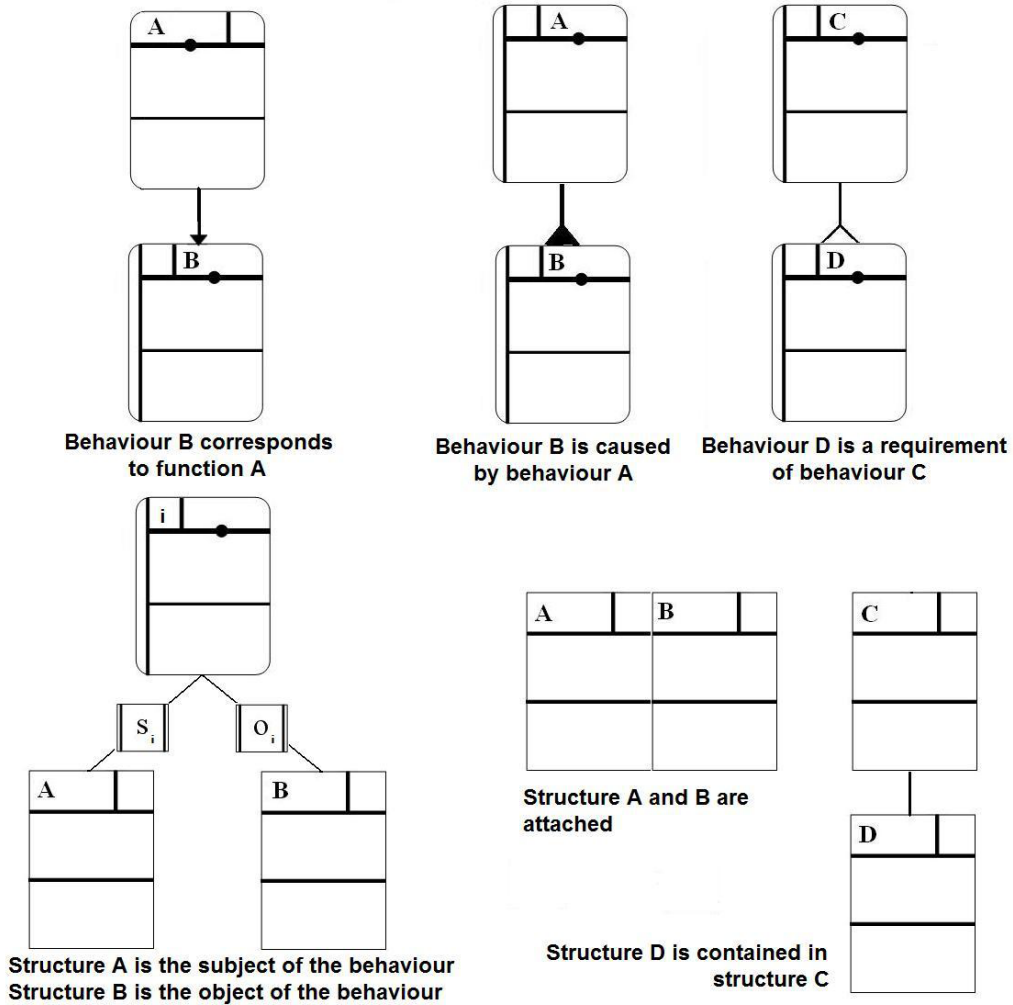


Figure 7. Relationships between the different components of the diagram model.

8 Examples

Two examples are given in this section in order to clarify the use of the B-Cube model together with RFB for design knowledge representation within an FBS framework.

The first example is a sander machine (Figure 8). The purpose function is 'remove', which is on the third level of the RFB (Table 1) derived from the second level function 'separate', which in turn stems from the first level function 'branch'. From the correspondence between terms shown in Table 4 it can be seen that this specific function can be performed in four different ways: (1, 1, 1), (1, 3, 1), (2, 1, 1), and (2, 2, 1). As the action of the sander affects the topological connectedness of the object (for example, the surface of a piece of wood) and not its position in space, the value of X must be 2. Moreover, the way the action is performed can be considered a cumulative and non-homeomeric process, that is, Y = 1, so the only behaviour remaining is (2, 1, 1). Topological connectedness is present at the beginning of the action and is lost while it is being performed (Z = 1), so the choice that was made is the correct one. It can be seen in the image that the structure that performs this behaviour is the sanding wheel, which is connected to a casing that contains a motor. This motor moves the sanding wheel in a (1, 1, 2) way, that is, the behaviour affects the spatial location (X = 1) with a continuous process (Y = 1), but neither approaching nor moving it away (Z = 2), just rotating it. This motor needs to be powered, so a structure capable of conducting energy is needed. The wire fulfils these requirements, since it performs the behaviour

related to the state ($Y = 2$) of allowing the energy flow ($X = 3$) without increasing or decreasing it ($Z = 2$). Lastly, the casing that is connected to the wire and the sanding wheel, and which contains the motor, is the part that is held by the user and represented by the behaviour (1, 2, 2).

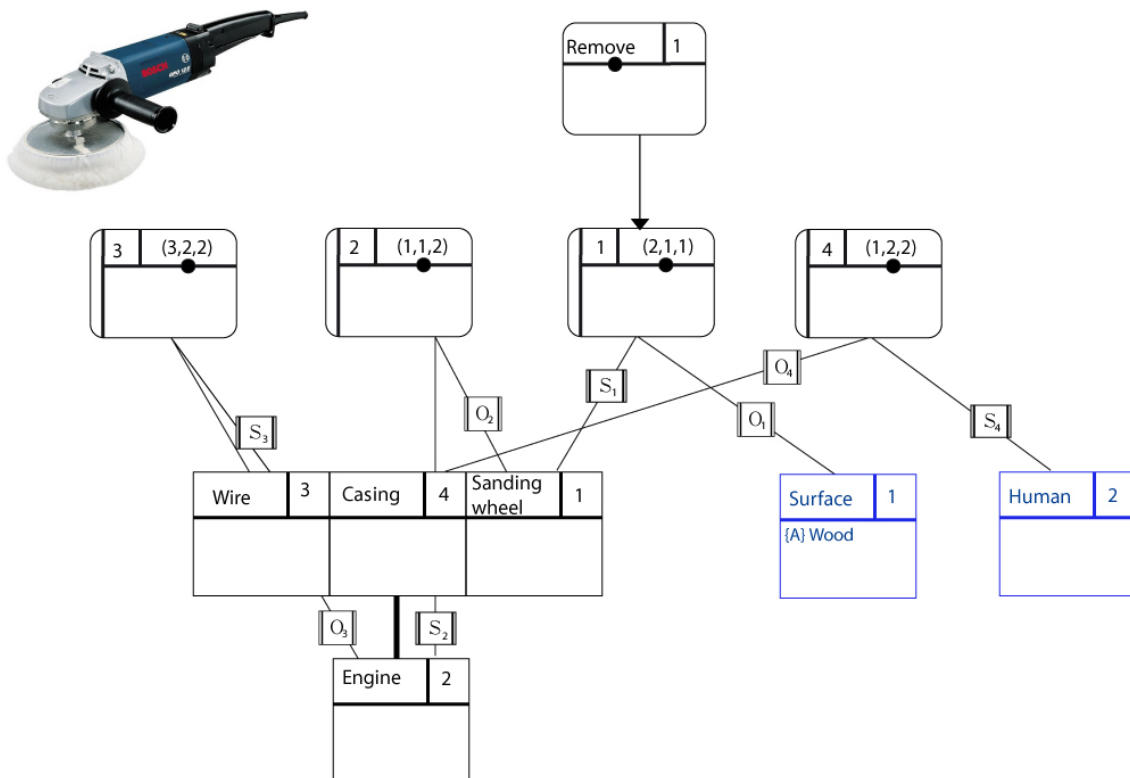


Figure 5. Representation of the scheme of a sander machine.

The second example is the case of a guillotine paper cutter. In Figure 9 it can be seen that the main function is represented by the RFB verbs 'branch'/'separate'/'divide', so it shares first and second levels of the taxonomy with the last example, but it is different in the third level term. Searching for correlations in Table 4, the only B-Cube term corresponding to 'divide' is (2, 4, 1), since it affects the topological connectedness ($X = 2$) in a non-cumulative and instantaneous way ($Y = 4$) and it causes loss of the initial topological connectedness ($Z = 1$). The structure chosen for carrying out this behaviour is a blade, which has a handle at one end in order to protect users from hurting themselves when using it. This use is represented by the behaviour (1, 4, 3), that is, the user changes the spatial location ($X = 1$) of the handle in a non-cumulative and instantaneous way ($Y = 4$) in order to move it closer to the paper ($Z = 3$).

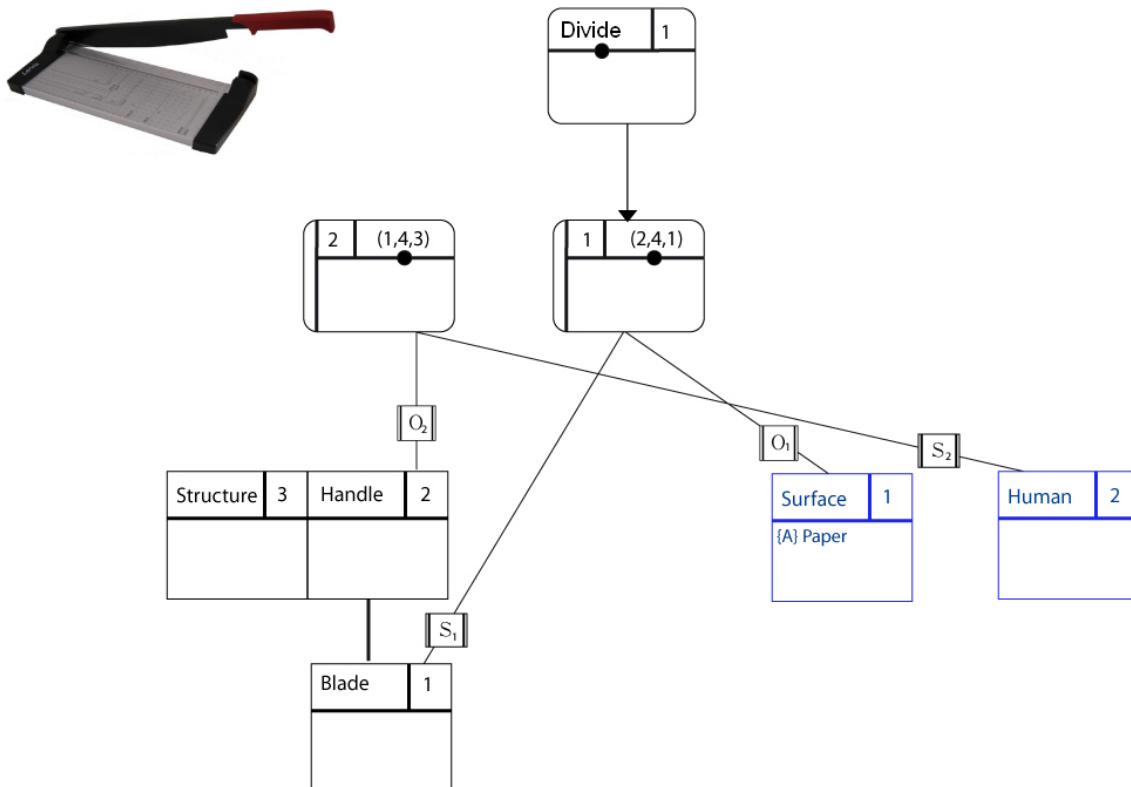


Figure 6 Representation of the scheme of a guillotine paper cutter

A more complex example of a functional design expressed using the B-Cube model's terminology can be seen in [49], where it is defended the usability of this kind of model for allowing the link between the functional design and the CAI tools.

9 Conclusions

The main objective achieved in present work is to develop a model for representing, managing and modeling the knowledge on the behaviour level of the function-behaviour-structure framework. This model is supposed to be clear and unequivocal, and it is focused towards optimization and automation of the decision taking of the product design process by means of knowledge management techniques or tools and towards a better knowledge transference by minimizing the lost of information.

In the present article, the B-Cube model represents the behaviour level and it acts as the link between the function level, represented by NIST's Functional Basis, and the structure level. From the comparison between the NIST functional basis and the B-Cube model of section 6, where B-Cube was expanded in each of the possible values for the three axes, and the best secondary and tertiary levels of the RFB taxonomy were assigned, if this were possible, to each vector, some conclusions can be made. The first one is that, while terms in the functional basis are generic, the ones in the B-Cube are more specific, since it requires the behaviours. This higher level of concreteness presented by behaviours has shown that there are possible actions that are not reflected in the NIST functional basis. This fact could also act as a check for the NIST functional basis, so the combinatory property of the B-Cube model ensures that no meaning is omitted.

Another way to explain this difference in level of specificity between the NIST functional basis and the B-Cube model is that, while the former is clearly a taxonomy, B-Cube acts more like an ontology. This seems logical because B-Cube was built from

DOLCE's meta-ontology. As can be seen in Figure 5, each chosen branch provides the final term with a specific meaning. This is one reason for defending the idea that the path is more important than the final term in the B-Cube model. A single word could be set for each term, just to make the model easier to understand, but for the time being it seems better to keep the vector terminology in order to avoid the ambiguity that may be present in the interpretation of one simple verb.

The importance of behaviour lies in its concreteness (in contrast to the generality of function) and also in its direct relation with structures. In the B-Cube model, behaviours are defined by a (x, y, z) vector, which offers several advantages. On the one hand, it provides more information than a single taxon, since there are three defining parameters instead of one. On the other hand, vectors do not have the ambiguity that may be present in the interpretation of taxa. Moreover, it is easier for software applications to manage vectors. As has been seen earlier, by managing 12 factors (five X-values, four Y-values and three Z-values), B-Cube is able to represent 60 different kinds of behavior thoroughly defined. This low number of factors to be managed will make software applications easier to build and faster to work with.

This fact opens the chance of enabling the implementation of software tools that support behavioural modelling within the FBS framework, while at the same time achieving the objective of developing a common basis of standardized terms for the exchange of behaviour-based information. The possibilities of the B-Cube model include its use in KBS to automate the design process and the link of this kind of system with other CAD, CAM or CAI tools through model libraries in order to improve it.

Following the DOLCE's structure in the development of the model it can be seen that similar results can be achieved with the role behaviours if the Abstract Qualities are considered instead of the Physical Qualities. Future work will be oriented in this line, as it is considered that the functional design including role behaviours can drive to new horizons in the fields of Artificial Intelligence and videogames..

Acknowledgements

This work was supported by the Spanish Ministry of Education and Science (ref. DPI2006-15570-C02-00) and the European Fund for Regional Development (FEDER).

References

- [1] V. Chulvi, R. Vidal, TRIZ on Design-oriented Knowledge-based Systems. A study on function level, *The TRIZ Journal*, (2009).
- [2] J. Hirtz, R. Stone, D. McAdams, S. Szykman, K. Wood, A functional basis for engineering design: Reconciling and evolving previous efforts, *Research in engineering design*, 13 (2002) 65 - 82.
- [3] C. Masolo, S. Borgo, A. Gangemi, N. Guarino, A. Oltramari, *WonderWeb Deliverable D18*, in, Laboratory For Applied Ontology - ISTC-CNR, 2003.
- [4] R. Ferrario, A. Oltramari, *Towards a Computational Ontology of Mind*, (2005).
- [5] S. Borgo, M. Carrara, P.E. Vermaas, P. Garbacz, Behaviour of a technical artifact: An ontological perspective in engineering, *Frontiers in Artificial Intelligence and Applications*, 150 (Formal Ontology in Information Systems) (2006) 214–225.
- [6] P. Garbacz, Towards a standard taxonomy of artifact functions, *Applied Ontology*, 1 (3) (2006) 221-236.
- [7] J. Gero, Design prototypes: A knowledge representation schema for design, *AI Magazine*, 11 (4) (1990) 26 - 36.
- [8] Y. Umeda, H. Takeda, T. Tomiyama, H. Yoshikawa, Function, behaviour, and structure, in: J. Gero (Ed.), *Applications of Artificial Intelligence in Engineering V*, Springer, Berlin, 1990, pp. 177-194.
- [9] H. Takeda, M. Yoshioka, T. Tomiyama, Y. Shimomura, Analysis of design processes by function, behavior and structure, in, *Proceedings of the The Delft Protocols Workshop*, conference proceedings., 1994.
- [10] J.S. Gero, U. Kannengiesser, The situated function-behaviour-structure framework, *Design Studies*, 25 (4) (2004) 373-391.

- [11] Y. Kitamura, R. Mizoguchi, Ontology-based systematization of functional knowledge, *Journal of engineering design*, 15 (4) (2004) 327-351.
- [12] P.E. Vermaas, K. Dorst, On the conceptual framework of John Gero's FBS-model and the prescriptive aims of design methodology, *Design Studies*, 28 (2) (2007) 133-157.
- [13] Y. Umeda, S. Kondoh, Y. Shimodura, T. Tomiyama, Development of design methodology for upgradable products based on function-behavior-state modeling, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 19 (2005) 161-182.
- [14] Q.L. Xu, S.K. Ong, A.Y.C. Nee, Function-based design synthesis approach to design reuse, *Research in engineering design*, 17 (2006) 27-44.
- [15] Y. Deng, Function and behavior representation in conceptual mechanical design, *Artificial intelligence for engineering design, analysis and manufacturing (AIEDAM)*, 16 (2002) 343-362.
- [16] S.B. Tor, G.A. Britton, W.Y. Zhang, Y.M. Deng, Guiding functional design of mechanical products through rule-based causal behavioural reasoning, *Int J Prod Res*, 40 (3) (2002) 667-682.
- [17] W.Y. Zhang, S.B. Tor, G.A. Britton, Y.M. Deng, Functional design of mechanical products based on behaviour-driven function-environment-structure modeling framework, in, Singapore, 2002, pp. 8.
- [18] L. Qian, *Creative Design by Analogy*, in: A. Chakrabarti (Ed.), *Engineering Design Synthesis. Understanding, approaches and tools*, Springer-Verlag, London, 2002, pp. 245 - 269.
- [19] R. Bracewell, J. Sharpe, Functional Descriptions used in Computer Support for Qualitative Scheme Generation-"Schemebuilder", *AIEDAM*, 10 (4) (1996) 333-346.
- [20] L. Ying-Chieh, A. Chakrabarti, T. Bligh, A computational framework for concept generation and exploration in mechanical design, in, *Proceedings of the Artificial Intelligence in Design'00*, Worcester, Massachusetts, USA, June, Springer, 2000.
- [21] A. Chakrabarti, P. Langdon, Y. Liu, T. Bligh, An Approach to Compositional Synthesis of Mechanical Design Concepts using Computers, in: A. Chakrabarti (Ed.), *Engineering Design Synthesis. Understanding, approaches and tools*, Springer-Verlag, London, 2002, pp. 179 - 194.
- [22] R. Lossack, Design Process and Context for the Support of Design Synthesis, in: A. Chakrabarti (Ed.), *Engineering Design Synthesis. Understanding, approaches and tools*, Springer-Verlag, London, 2002, pp. 213 - 227.
- [23] M. Campbell, J. Cagan, K. Kotovsky, The A-Design Approach to Managing Automated Design Synthesis, *Research in engineering design*, 14 (2003) 12 - 24.
- [24] A. Gilchrist, Thesauri, taxonomies and ontologies - an etymological note, *Journal of Documentation*, 59 (1) (2003).
- [25] J.A. Gershenson, L.A. Stauffer, The creation of a taxonomy for manufacturability design requirements, in, *Proceedings of the Design engineering technical conferences*, Boston, Massachusetts, September 17-20 1995, pp. 305 - 314.
- [26] S. Szykman, J. Racz, R. Sriram, The representation of function in computer-based design, in: ASME (Ed.), *Proceedings of the Design Engineering Technical Conferences*, Las Vegas, Nevada, September 12-15, ASME, 1999.
- [27] I.J. Golden, Function Based Archival and Retrieval: Developing a Repository of Biologically Inspired Product Concepts, in, Department of Mechanical Engineering, University of Maryland, 2005.
- [28] N. Feyngenson, Function synthesis: new methodological tool and case studies, in, *Proceedings of the ETRIA TRIZ Futures*, Kortrijk, Belgium, October 2006, 2006.
- [29] J. Gero, K.W. Tham, H.S. Lee, Behaviour: A link between Function and Structure in Design, in: D.C. Brown, M. Waldron, H. Yoshikawa (Eds.), *Proceedings of the IFIP - Intelligent Computer Aided Design*, North Holland, Elsevier Science Publishers, 1992, pp. 193-220.
- [30] S. Borgo, A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, WonderWeb Deliverable D15, in, *Laboratory For Applied Ontology - ISTC-CNR*, 2002.
- [31] J. Rasmussen, Skills, rules, and knowledge; Signals, signs and symbols, and other distinctions in human performance models, *IEEE Transactions on systems, man, and cybernetics*, SMC-13 (3) (1983) 257 - 266.
- [32] A. Bonaccorsi, Grammars of creation. Mapping search strategies for radical innovation, in, *Proceedings of the Innovation Pressure Conference*, Tampere, Finland, March 16-19, 2006.
- [33] A. Gangemi, N. Guarino, C. Masolo, A. Oltramari, Sweetening WORDNET with DOLCE AI Magazine, 24 (3) (2003) 13.24.
- [34] B. Chandrasekaran, Representing function: Relating functional representation and functional modeling research streams, *AIEDAM Artificial Intellig. for Engineering Design*, 19 (2) (2005) 65-74.

- [35] P.E. Vermaas, The functional modelling account of Stone and Wood: Some critical remarks, in, International Conference on Engineering Design, ICED'07, Paris, France, 2007.
- [36] M.R. Bohm, R.B. Stone, Representing functionality to support reuse: conceptual and supporting functions, in, Proceedings of the ASME DETC & CIE Conferences, Salt Lake City, Utah, September 28 – October 2, 2004.
- [37] R.L. Nagel, R.B. Stone, D.A. McAdams, A process modeling methodology for automation of manual and time dependent processes, in, Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Philadelphia, PA, USA, September 10-13, 2006.
- [38] Y. Kitamura, Roles of ontologies of engineering artifacts for design knowledge modeling, in, Proceedings of the 5th International Seminar and Workshop Engineering Design in Integrated Product Development (EDIProD 2006), Gronów, Poland, 21-23 September, 2006, pp. 59-69.
- [39] R.L. Nagel, M.R. Bohm, R.B. Stone, D.A. McAdams, A representation of carrier flows for functional design, in, Proceedings of the 16th International Conference on Engineering Design, Paris, France, 28-30 August 2007.
- [40] R.J. Mayer, C.P. Menzel, M.K. Painter, P.S. deWitte, T. Blinn, B. Perakath, Information integration for concurrent engineering (IICE) IDEF4 object-oriented design method report in, Knowledge Based Systems, Inc., Texas, 1995.
- [41] S. Borgo, M. Carrara, P. Garbacz, P.E. Vermaas, A formal ontological perspective on the behaviors and functions of technical artifacts, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 23 (Special Issue 01) (2009) 3-21.
- [42] B. Chandrasekaran, J.R. Josephson, Function in device representation, Engineering with Computers, 16 (2000) 162–177.
- [43] R.B. Stone, K.L. Wood, Development of a Functional Basis for Design, Journal of mechanical design, 122 (4) (2000) 359-371.
- [44] G. Pahl, W. Beitz, Engineering Design. A Systematic Approach, Springer, London, 1996.
- [45] C.R. Bryant-Arnold, R.B. Stone, J.L. Greer, D.A. McAdams, T. Kurtoglu, M.I. Campbell, A function-based component ontology for systems design, in, Proceedings of the 16th International Conference on Engineering Design, Paris, France, 28-30 August 2007.
- [46] C. Masolo, L. Vieu, E. Bottazzi, C. Catenacci, R. Ferrario, A. Gangemi, N. Guarino, Social roles and their descriptions, in, Proceedings of the Ninth International Conference on the Principles of Knowledge Representation and Reasoning,, Whistler, Canada, 2004.
- [47] C. Masolo, G. Guizzardi, L. Vieu, E. Bottazzi, R. Ferrario, Relational roles and qua-individuals, in, Proceedings of the AAAI Fall Symposium "Roles, an Interdisciplinary Perspective: Ontologies, Languages, and Multiagent Systems", Arlington, Virginia, 3-6 November, 2005, pp. 103–112.
- [48] R.J. Mayer, C.P. Menzel, M.K. Painter, P.S. deWitte, T. Blinn, B. Perakath, Information integration for concurrent engineering (IICE) IDEF3 process description capture method report, in, Knowledge Based Systems, Inc., Texas, 1995.
- [49] V. Chulvi, R. Vidal, B-Cube model in automated functional design, in, Proceedings of the 2nd International Conference on Agents and Artificial Intelligence, Valencia, Spain, January 22 - 24, 2010, pp. 190-196.

