B-CUBE MODEL IN AUTOMATED FUNCTIONAL DESIGN

Vicente Chulvi and Rosario Vidal
GID, Departament d'Enginyeria Mecànica i Construcció, Universitat Jaume I, Av. Sos Baynat, Castellón, Spain

Keywords: Knowledge Based Systems, Functional design, Ontologies.

Abstract: The present work proposes to use both the NIST's functional basis and the B-Cube model in the development of a KBS that is capable of automating functional design. For this purpose this article shows with an example how the system will work using the FBS framework. The starting point is defended by the terms of NIST functional basis. The evolution from functions to structures, that is, to the solution, is achieved by means of B-Cube model in the behaviour layer.

1 INTRODUCTION

Automated design is a current topic in most significant companies around the world and even more so in high technology companies, such as in the automotive or aerospace industries (Kochan, 1999; Liening and Blount, 1998). The last decade has witnessed the gradual introduction of automated design technologies into other manufacturing fields, with techniques like DFx (Design For Assembly, Design For Environments, etc.) or KBS (Knowledge Based Systems). Several authors have written on the advantages of these technologies (Bermell-García et al., 2001; Vidal and Mulet, 2006; Sainter et al., 2000; Chen, 1999).

Functional design is presented as a key step in the product design process and, as such, it is the focus of the main scientific efforts being made to automate design in artificial intelligence (AI) systems. The starting point of functional design is the concept of function, behaviour and structure. Although these concepts have been used for a long time, it was only in 1990 when they were clearly defined and when they were proposed as a modelling and representing framework, due to their functionality (Gero, 1990; Umeda et al., 1990). The FBS framework can be applied as a methodology for design process analysis, representing the evolution of the design state from protocol studies (Takeda et al., 1994). Within this framework, function represents the duties that the design fulfils, structure represents the physical elements of the solution and behaviour is the link between F and S. In solution synthesis, behaviour derives from a particular function, and the solution is achieved through this behaviour. Furthermore, when a solution has been defined, behaviour is inferred from that solution in order to evaluate whether the solution reaches the required degree of functionality.

The FBS framework allows computational modelling to be carried out. Thus, several authors have attempted to develop approaches and software applications to implement FBS-based procedures (Qian, 2002) or to model function and/or structure libraries to be implemented in functional reasoning processes (Lossack et al., 1998; Bracewell and Sharpe, 1996; Ying-Chieh et al., 2000; Chakrabarti et al., 2002; Lossack, 2002; Campbell et al., 2003). The possibilities of these systems have been increased by the use of taxonomies and ontologies.

Ontologies (and taxonomies consequently) can be understood as a form of knowledge representation. 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 (Gilchrist, 2003). Several function taxonomies have been developed, but the most significant reconciliation of function taxonomies nowadays are those provided by the National Institute of Standards and Technology (NIST) (Hirtz et al., 2002). The NIST's functional basis is a reconciliation and integration of other taxonomies, mainly from research carried out by the NIST (Szykman et al., 1999) and the functional basis effort (Stone and Wood, 2000), the purpose of which was to facilitate the development of a formal representation of functions with a hierarchical vocabulary of standardised terminologies, focused on mechanical design.

Chulvi V. and Vidal R. (2010).
B-CUBE MODEL IN AUTOMATED FUNCTIONAL DESIGN.
In *Proceedings of the 2nd International Conference on Agents and Artificial Intelligence - Artificial Intelligence*, pages 190-196
DOI: 10.5220/0002702001900196

An ontology can be described as an explicit specification of a shared conceptualisation, where concepts and relations are organised hierarchically and concepts are classified as classes and instances. Ontologies can be taxonomically or axiomatically based (Gruber, 1993). The B-Cube model (Chulvi and Vidal, 2009) is an ontology for the behaviour layer mainly based on the DOLCE's meta-ontology (Masolo et al., 2003; Ferrario and Oltramari, 2005), which uses three-dimensional vectors as terms. Each dimension of these vectors defines one characteristic of the behaviour, so this model uses definitions to define behaviours instead of one single word.

The aim of this article is to show how both the NIST's functional basis and the B-Cube model can work together to develop a KBS that is capable of automating the functional design. Section 2 introduces the B-Cube model developed by the authors, while section 3 uses an example to show the theoretical work of the proposed KBS. The article ends with some conclusions in section 4.

2 B-CUBE

The B-Cube model is used to represent the behaviour layer in the functional design process within the FBS framework. This model proposes a three-dimensional scheme that uses definitions as behavioural concepts. The key to this approach is that a behaviour is not defined with a word or a taxon, which could cause ambiguity and misinterpretation, but rather it is defined as a three-dimensional vector (X, Y, Z) that is set by its characteristics and qualities.

The starting point from which to define these parameters is the DOLCE's upper-level ontology. The definitions of the DOLCE have been slightly adapted to meet B-Cube's purposes. *Endurant* is defined as the entity or element (Structure) to which the B-Cube entry refers. It is supposed that there are an infinite number of endurants in the universe and they are differentiated as being *physical* (PEDs) and *non-physical endurants* (NPEDs). *Perdurant* (P) is a characteristic that defines a behaviour and it refers to the kind of behaviour that affects the above-mentioned entity. Ps are situated on the Y axis of the B-Cube model. Lastly, *qualities* are defined as characteristics linked to other entities which are going to be used to define the behaviour. There are three different sorts of qualities: *temporal* (TQs), *physical* (PQs) and *abstract qualities* (AQs). TQs are directly related to Ps, so they will be used to define a Behaviour, and are located on the Z axis. The B-Cube model is completed with the X axis, where the PQs will be, if the entry to the model was a PED, and the AQs, if the entry was an NPED.

Despite the fact that numerical values are used when working with the B-Cube model, all of these values have a term to define them. The terms have been taken mainly from the DOLCE's terminology (Masolo et al., 2003) and increased by Garbacz's work on it (Garbacz, 2006). As these terms were not enough to meet the needs of B-Cube, they were fulfilled with terms from Rasmussen's taxonomy (Rasmussen, 1983) and the NIST classification of flows (Hirtz et al., 2002, Nagel et al., 2007). As a result, they are defined as shown in Table 1.

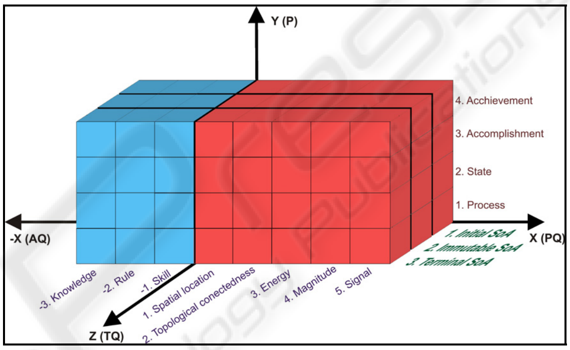

Figure 1: B-Cube model.

3 DESIGN REPRESENTATION MODEL BASED ON B-FES

The design representation model used to represent the knowledge in this project pretends to be useful for the understanding, transferring, and reproduction of this knowledge. There are a lot of methodologies created with the aim to represent the information and transfer it to computer systems to be integrated into a KBS. The IDEF language family consists in a set of modelling languages within the field of system engineering, which cover a wide range of needs from information's capture and modelled to net design. Due to their diversity they present a huge number of terminologies and symbols in order to represent knowledge. So it doesn't consist on choosing the best tool, but it is about choose the most adequate in the right moment. Several works have taken advantage from this adaptability, and they have create their own approaches to this modelled language using the standardized terminology (Kim and Jang, 2002, Romero et al., 2008). The proposal here applied is based on IDEF4, and it is adapted to represent the knowledge within a B-FES framework using both Functional basis and

Table 1: Values of B-Cube model with examples.

Axis	Value	Term	Significance	Examples
X	1	Spatial location	Position of a PED in space	To move an object
	2	Topological connectedness	The kind of connection at a topological level in which a PED finds itself	To break an object To join an object
	3	Energy	Energetic state of a PED	To freeze water To charge a battery
	4	Magnitude	A physical magnitude of the PED that is affected by the behaviour	To increase weight To change colour
	5	Signal	Actions referred to PEDs when they act as signals	To increase a wave A mobile phone sending a signal
	-1	Skill	The behaviour does not require conscious control by the subject	Driving a car Playing the piano
	-2	Rule	The behaviour requires conscious control by the subject, but this is subject to some process or “written rule”	Cooking following a recipe To tune an instrument
	-3	Knowledge	The behaviour requires conscious control by the subject, and this is not subject to any process or “written rule”	To compose a symphony To manage an enterprise
Y	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	1	Initial SoA	The behaviour makes the initial PQ or AQ decrease or disappear	To cool an object
	2	Immutable SoA	The behaviour doesn’t vary the grade or quantity of PQ or AQ affected by it	To convert energy
	3	Final SoA	The behaviour makes the grade or quantity of PQ or AQ increase or appear	To warm an object

B-Cube model’s terminologies. It pretends to achieve an intuitive and easy to understand representation for any designer. The figure 2 shows the boxes used to represent functions, behaviours, structures and restrictions (structures that belong to environment, not to design), and the main interactions between them.

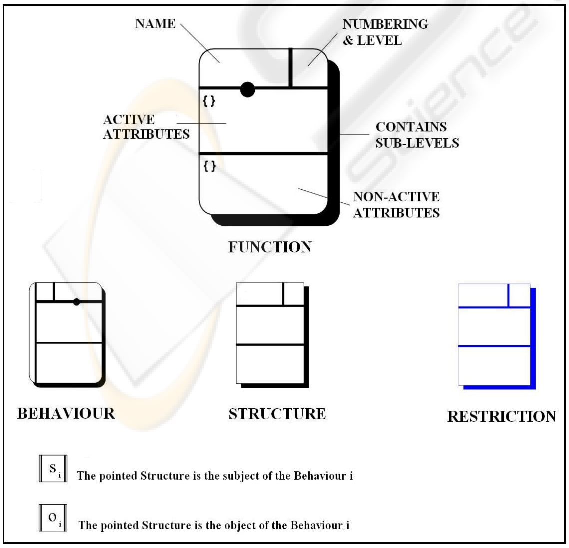

Figure 2: Representation boxes.

4 DISCUSSION

In an ideal system the designer would only need to introduce what he wants to design and the system would provide him with the best solution or several optimal solutions. This proposed ideal KBS needs a proper semantic tool, databases and evaluation system tool, besides the FBS-based design system, which has the B-Cube model as its core. As this section aims to show how the B-Cube model can work within the system, we are going to suppose that both the semantic tool and the evaluation system work ideally (or they are controlled directly by the designer) and that there are complete databases for the desired purpose. The NIST’s functional basis will be used to complete the function level.

The example developed for this purpose shows the development of the design of a device for writing and erasing (signs) on paper that can be used by one person and put easily in your lapel. In this case the designer or the semantic tool can distinguish four functions: write, erase, handle, and hold; and the restrictions corresponding to the environment level: person, hand, lapel, paper, and signs. Functions have been adapted to NIST terminology in order to standardise them. So, the function “write” can be interpreted as connecting signs with the paper, so it corresponds to the term “couple” of the functional basis. The function “erase”, which has the opposite meaning to “write”, can be defined as “remove”. In turn, the fact that an object is held can correspond to the terms “secure” and “(not) allow degree of freedom”. Despite the fact that it seems that only one term must be chosen, it will be shown below how it is advisable to indicate the two options. Lastly, the function “handle” corresponds to the terms “translate” and “rotate”.

A correlation between the terms used by the functional basis and those used by the B-Cube model was created to enable the behaviour level to be obtained from the function level. From this correlation it can be seen that the function “couple” corresponds to the behaviour (2,1,3). That is, X=2, that means that the behaviour affects the topological connectedness; Y=1, the behaviour represents a process that acts in a cumulative and non-homeomeric way; and Z=3, that represents that the topological connectedness is achieved by means of the behaviour. “Remove” can be represented by behaviours (1,1,1), (1,3,1), (2,1,1), (2,2,1), so a better definition is also required from the semantic tool, from the designer’s feedback or from any other automated system (e.g. algorithm libraries). In this case we chose (2,1,1), as it acts as the opposite of (2,1,3), which was unambiguously defined. The next function can be defined both for “secure” and “(not) allow degree of freedom”. Here, the first definition has a correspondence with the terms (1,2,2), (2,1,3), and (2,3,2), while the second one corresponds to (1,2,2). This value is chosen to represent the behaviour since it is the only value that matches the two possible functions. Lastly, functions “translate” and “rotate” only correspond to the term (1,1,2).

These behaviours, together with their relations with the environment level, are the starting point to define the structures that are needed for the desired design. For the development of this example it has been supposed that the available structure database is sufficient but limited. In this case, the objective consists in searching in the database for the structures that are able to carry out the required behaviour. Then the interaction with these structures with the environment’s restrictions is considered in order to obtain a first selection and, finally, the remaining solutions are evaluated in order to determine which ones can really be carried out and under which conditions. Tables 2 and 3 show the examples of the search for structures-solution for behaviours (2,1,3) and (2,1,1).

It can be observed in the example in Table 2 that five structures capable of carrying out behaviour (2,1,3) were found in the database. Two of these structures were rejected straight away because of their effects on the restriction, that is, they damage the paper. The remaining three were evaluated on the basis of their final purpose. As can be observed, two of them present one disadvantage and one requirement, and the other one presents three disadvantages and one requirement. So, this last one is rejected. As the other two options seem to be equally good, the system is split into two from now on and two different final solutions will be developed and then evaluated at the end of the process. The evolution from carbon lead is named “design A”, and the ink one is named “design B”.

Table 2: Structure search and evaluation from behaviour (2,1,3).

Structure	Compatibility with paper	Evaluation
Carbon lead	Good	It wears down. It requires a container.
Ink	Good	It wears down. It requires a container. It gets dry with air contact.
Chalk	Good	It wears down. It requires a container. Poor trace quality. It marks and smudges.
Carving	Bad	-
Welding	Bad	-

Table 3: Structure search and evaluation from behaviour (2,2,2).

Structure	Compatibility with ink	Evaluation
Hood: - Metal - Plastic - Wood - Cardboard	Good	It must be removable. - Viable - Viable - Viable - Viable. Ecological. Low mechanical strength. Ink can soak through it.
Cover	Good	Non-viable. It is not removable
Gel	Bad	-
Vacuum	Good	Very expensive system Difficult to implant into a small device.

Next evaluation phases must take this split into account. So, the next structure’s evaluation considers the structure’s utility with both carbon lead and ink. As an example, for the search and evaluation from behaviour (2,1,1), the structure “rubber” is the only one that has a good compatibility with paper. This structure is useful within design A, but it is useless in design B. In this case a structure with regular or dubious compatibility must be chosen as a solution in design B. The selected structure is again the one with the lowest number of disadvantages or requirements. Figures 3a and 3b show the differences in this first step in the development of designs A and B.

The system now evolves from this first step by turning the disadvantages and requirements of the new structures into new behaviours, which can be carried out by the existing structures or new ones may be required. The existing structure will always be preferred to a new one since no new behaviours will be needed and the number of structures will be lower. So, the new structures “carbon lead” and “ink” generate two new behaviours: one to show that they wear down (4,2,1) and the other one to show the need to be contained (2,2,2). Hence, behaviour (4,2,1) is derived from behaviours (2,1,3) and (2,1,1), which correspond to normal device use. As it proceeds from a disadvantage, it is considered as a non-desirable behaviour and it will need to be considered differently to the other behaviours. On the other hand, behaviour (2,2,2) proceeds from a requirement, so it is considered as a desirable behaviour and it will be treated in the same way as the behaviours that come from functions.

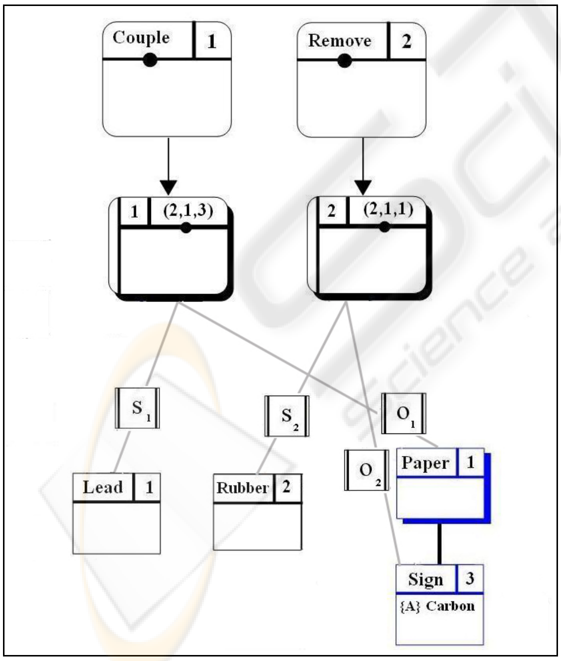

Figure 3a: First step in the evolution of design A.

If we centre on case B, the structure “ink” has generated a non-desirable behaviour. Here, a new behaviour has to appear in order to solve the requirement of the non-desirable behaviour, whose object is the “ink”, and it will need a new structure able to carry out this new behaviour (Table 3)

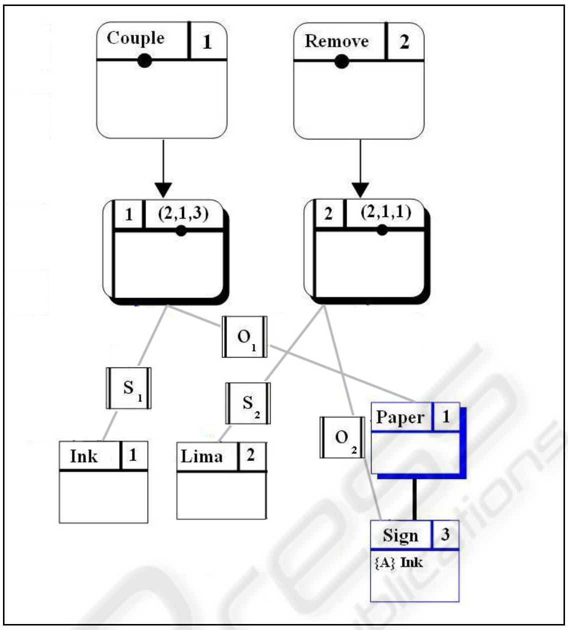

Figure 3b: First step in the evolution of design B.

The development of the designs will evolve in the way described until all the required behaviours can be carried out by some existing structure and all the existing structures need no new, as yet undefined, behaviours. At this point, the design process is considered to have ended and all the designs are ready to be evaluated, both by the designer and by an external application built for this purpose. The development of the design B obtained in this example is represented in Figure 4.

5 CONCLUSIONS

In this paper the authors have defended the use of both the NIST’s functional basis and the B-Cube model for developing a KBS that is capable of automating functional design. The B-Cube model is shown as one of the tools that are needed to create a KBS to automate the design process within an FBS framework. The importance of behaviour lies in its concreteness (in contrast to the generality of function) and also in its direct relation with structures. For this purpose, the B-Cube model offers the advantages of the lower ambiguity of vectors compared to taxa and the facility of computer programs to work with vectors. It has also proved its ability to work well with the NIST’s

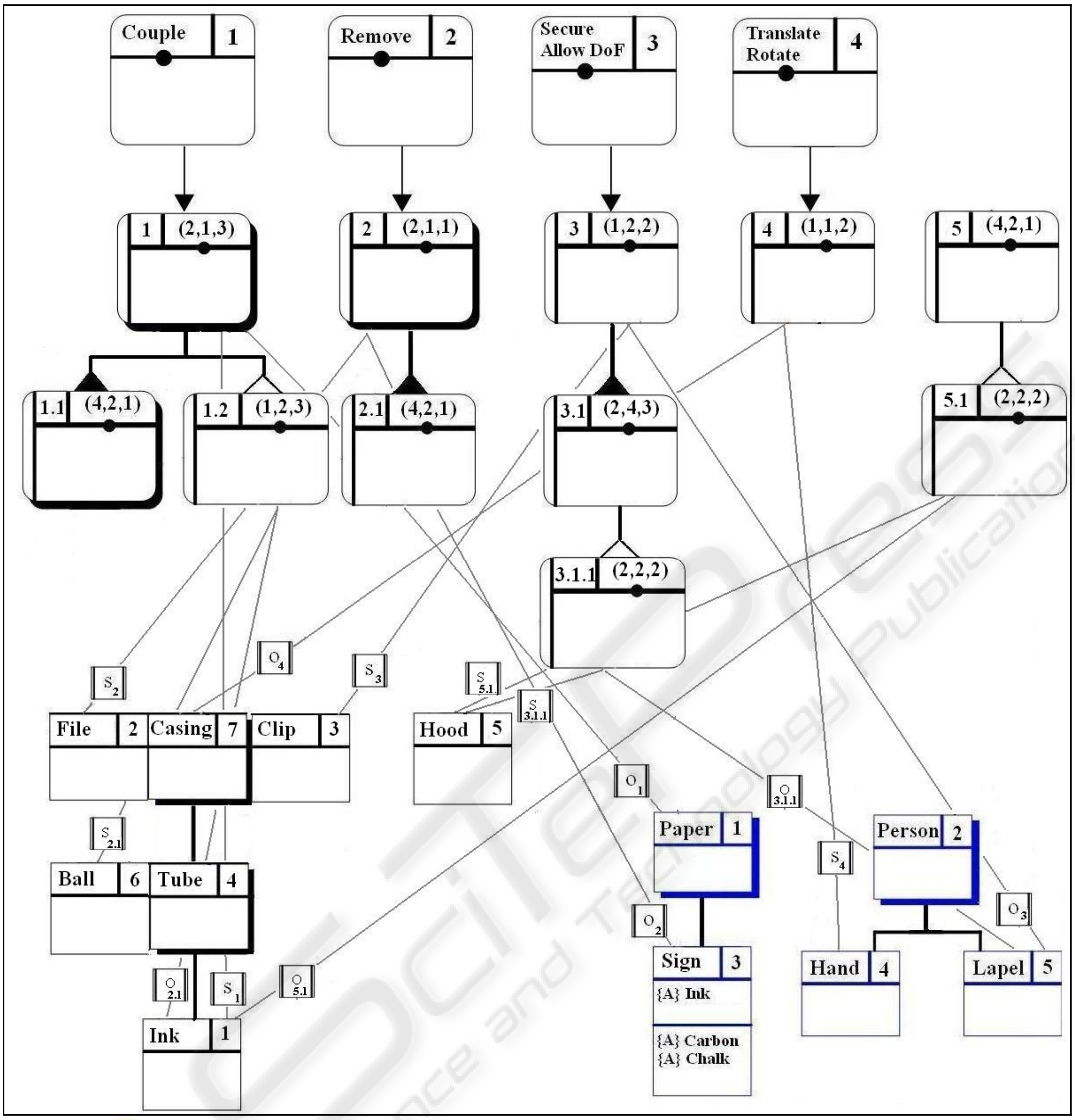

Figure 4: Final design B result.

functional basis, which is considered to be the most useful correlation of function terms, as well as being suitable for automated functional design.

As well as the B-Cube model and the NIST's functional basis, complete databases and evaluation and semantic tools are also needed to automate the design process. The use of ontologies is the easiest way to interact with these tools and share the knowledge in an appropriate manner. Future work will be directed along these lines, with emphasis on both the structures level and the link between functional design and Computer Aided Inventing (CAI) tools in order to innovate and improve the design process.

ACKNOWLEDGEMENTS

The authors would like to acknowledge the Ministry of Education and Science in Spain (ref. DPI2006-15570-C02-00) and the European Fund for Regional Development (FEDER) for their funding.

REFERENCES

Bermell-García, P., Fan, I.-S., Li, G., Porter, R. & Butter, D., 2001. Effective abstraction of engineering knowledge for KBE implementation. *13 International Conference on Engineering Design.* Glasgow.

Bracewell, R. & Sharpe, J., 1996. Functional Descriptions used in Computer Support for Qualitative Scheme Generation-"Schemebuilder". *AIEDAM,* 10**,** 333-346.

Campbell, M., Cagan, J. & Kotovsky, K., 2003. The A-Design Approach to Managing Automated Design Synthesis. *Research in Engineering Design,* 14**,** 12-24.

Chakrabarti, A., Langdon, P., Liu, Y. & Bligh, T., 2002. An Approach to Compositional Synthesis of Mechanical Design Concepts using Computers. In Chakrabarti, A. (Ed.) *Engineering Design Synthesis. Understanding, approaches and tools.* London, Springer-Verlag.

Chen, Y. M., 1999. Integrating Knowledge, Data and Geometry: A Computer-Aided Concurrent Engineering Infrastructure. *Integrated Computer-Aided Engineering,* 6**,** 189-212.

Chulvi, V. & Vidal, R., 2009. Functional basis and B-Cube: alternative or complementary models? *International Conference On Engineering Design, ICED'09.* 24-27 August. Stanford, Ca, USA.

Ferrario, R. & Oltramari, A., 2005. Towards a Computational Ontology of Mind.

Garbacz, P., 2006. Towards a standard taxonomy of artifact functions. *Applied Ontology,* 1**,** 221-236.

Gero, J. S., 1990. Design prototypes: A knowledge representation schema for design. *AI magazine,* 11**,** 26-36.

Gilchrist, A., 2003. Thesauri, taxonomies and ontologies – an etymological note. *Journal of Documentation,* 59.

Gruber, T. R., 1993. Toward Principles for the Design of Ontologies Used for Knowledge Sharing. In K. A. (Ed.) Padova, Italy, Knowledge Systems Laboratory, Stanford University.

Hirtz, J., Stone, R., Mcadams, D., Szykman, S. & Wood, K., 2002. A functional basis for engineering design: Reconciling and evolving previous efforts. *Research in Engineering Design,* 13**,** 65 - 82.

Kochan, A., 1999. Jaguar uses knowledge-based tools to reduce model development times. *Assembly Automation,* 19**,** 114-117.

Liening, A. & Blount, G. N., 1998. Influences of KBE on the aircraft brake industry. *Aircraft Engineering and Aerospace Technology,* 70**,** 439-444.

Lossack, R., 2002. Design Process and Context for the Support of Design Synthesis. In Chakrabarti, A. (Ed.) *Engineering Design Synthesis. Understanding, approaches and tools.* London, Springer-Verlag.

Lossack, R., Umeda, Y. & Tomiyama, T., 1998. Requirement, Function and Physical Principle Modelling as the basis for a Model of Synthesis. *Computer Aided Conceptual Design'98.*

Masolo, C., Borgo, S., Gangemi, A., Guarino, N. & Oltramari, A., 2003. WonderWeb Deliverable D18. Laboratory For Applied Ontology-ISTC-CNR.

Nagel, R. L., Bohm, M. R., Stone, R. B. & Mcadams, D. A., 2007. A representation of carrier flows for functional design. *16th International Conference on Engineering Design.* 28-30 August Paris, France.

Qian, L., 2002. Creative Design by Analogy. In Chakrabarti, A. (Ed.) *Engineering Design Synthesis. Understanding, approaches and tools.* London, Springer-Verlag.

Rasmussen, J., 1983. Skills, rules, and knowledge; Signals, signs and symbols, and other distinctions in human performance models. *IEEE Transactions on systems, man, and cybernetics,* SMC-13**,** 257-266.

Sainter, P., Oldham, K. & Larkin, A., 2000. Achieving benefits from knowledge-based engineering systems in the longer term as well as in the short term. *ICE 2000, Proceedings of the 6th International Conference on Concurrent***,** 143-150.

Stone, R. B. & Wood, K. L., 2000. Development of a Functional Basis for Design. *Journal of Mechanical Design* 122**,** 359-371.

Szykman, S., Racz, J. & Sriram, R., 1999. The representation of function in computer-based design. *Design Engineering Technical Conferences.* September 12-15, 1999, Las Vegas, Nevada, ASME.

Takeda, H., Yoshioka, M., Tomiyama, T. & Shimomura, Y., 1994. Analysis of design processes by function, behaviour and structure. *The Delft Protocols Workshop, conference proceedings.*

Umeda, Y., Takeda, H., Tomiyama, T. & Yoshikawa, H., 1990. Function, behaviour, and structure. In Gero, J. (Ed.) *Applications of Artificial Intelligence in Engineering V.* Berlin, Springer.

Vidal, R. & Mulet, E., 2006. Thinking about Computer Systems to support design synthesis. *Communications of the ACM,* 49**,** 100-104.

Ying-Chieh, L., Chakrabarti, A. & Bligh, T., 2000. A computational framework for concept generation and exploration in mechanical design. *Artificial Intelligence in Design'00.*