INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'07
28 - 31 AUGUST 2007, CITÉ DES SCIENCES ET DE L'INDUSTRIE, PARIS, FRANCE

FUNCTION AND BEHAVIOUR REPRESENTATION FOR SUPPORTING FLEXIBLE EXPLORATION AND GENERATION IN A FUNCTIONAL MODEL FOR CONCEPTUAL DESIGN

Camelo, D.[1], Mulet, E.[1] and Vidal, R.[1]
[1]Universitat Jaume I - Spain

ABSTRACT
For years, CAD systems have assisted many design tasks. These systems help to capture and control information in the final stages, when the product has a defined form and function. In spite of the advances that have been made, knowledge in the preliminary stages is still an incipient issue for engineering design researchers. For this reason, functional representation remains a challenge that has still to be solved.
Many of the synthesis models that are currently available perform cycles of exploration, generation and evaluation, but they do not allow for designer intervention [1]. Yet, the most recent models consider that designer interaction during synthesis is essential to achieve better results [2-5].
From the point of view of knowledge representation, few models are capable of capturing/representing such a wide range of information as that used during a real synthesis process. Most of the models that have been studied restrict their representation to the mechanical design area and most of them synthesize concepts by means of specific reasoning methods (e.g. input-output relation, cause-effect relation, bond-graph, function-means tree, etc.).
In this work we took an analysis of function and behaviour representation of the most significant synthesis models as the basis for some potential improvements. The aim of the present work is therefore to define a representation scheme which considers these improvements and seeks to accomplish the following goals: facilitate the intervention of designers during synthesis cycles and return to higher levels of abstraction in the design process in order to generate alternatives.
This research work introduces a knowledge representation and reasoning scheme to support the design process in the early stages of design; this will in turn facilitate the interaction of the designer during synthesis and the representation of a wide range of knowledge within different knowledge domains.

Keywords: function and behaviour representation, support conceptual design, functional reasoning

1 INTRODUCTION
The design of engineering products is one of the most important contributions to wealth generation in industrial nations and a strategic aspiration of many developing economies. It is therefore not surprising that a great deal of scientific research is currently being conducted in order to understand, optimise and support the processes and activities involved in the design and production of engineering products [6].
It is accepted that the design activity can be divided into four phases: clarification of the task, conceptual design, embodiment design and detail design. The aim of the conceptual design phase is to devise the concept of the artefact. By the end of this phase, the designer should have a clear, but not necessarily complete, conception of the basic structure, working principles and constituent elements, and a layout for the artefact. This phase is thus crucial to define new, creative products.
CAD systems were introduced into the area of engineering design in order to automate some routine tasks. Management of requirements, functional modelling or combination of working principles is supported only by certain tools, which are designed mainly for individual use and to deal with specific problems. Although a number of computer-based conceptual design systems have been proposed, nearly all of them are focused on a specific type of design problem. Most of these tools use the

principles of artificial intelligence to automate the generation of design solutions, thus ruling out the creative intervention of the user. The theoretical and practical challenges of specifying and creating computer programs for generic conceptual design is therefore still a relatively unexplored issue [7].
The knowledge managed by these tools hardly takes the creative intervention of the user into account, and for this reason this article contributes to knowledge representation for conceptual design activities. With this idea in mind, we present a reasoning scheme and its knowledge representation framework that enable us to obtain a more flexible model of synthesis.

1.1 Function-Behaviour-Structure model

Since 1990, the Function-Behaviour-Structure (FBS) approach has represented a common language with which to analyse and compare knowledge representation models. In this line of thinking, function represents the functions performed by the design, structure represents the states or physical elements, and behaviour connects the two levels. Gero defends that structure expresses the internal and external states of a physical element [8] and Umeda holds that this level represents the elements of an artefact and relationships among them [9]. As a complement for this comparison, other authors maintain that functions can also present one focus on the designer's intention and another focus on the operation of the system [10-14].

1.2 Related works

To analyse and describe the most significant representation models, the analysis adopts the FBS approximation as a reference for the comparison between models. A set of aspects is used to describe each representation model, such as semantics, syntax and relationships between the levels of abstraction. The semantics criterion describes the characteristics of each level of abstraction. From the computational point of view, these characters express the attributes of each level. Syntax describes how each attribute can be represented in a computer program so that it can be easily defined, retrieved, edited and manipulated by a designer or a computer program in specific design situations. Relationships indicate the links between the knowledge in a database and show the procedure to be adopted during the synthesis.
The analysis presented here considers the following codifications for the syntax: in natural language, in mathematical form or direct referent. Natural language is simple and promotes interaction with the user. In contrast, however, it is ambiguous and lacks precision in its interpretation. The verb + complement scheme is the commonest form presented by most functional approaches. The mathematical form solves part of the language problems. Its format is precise but complex for the user to understand. It always presents the common form of numeric values of constants, variables, ranges or parametric equations. And direct reference represents a restrictive connection of knowledge inside a database.
The models by Williams [15], Chakrabarti [16,17] and Zhang [18] introduce three levels of FBS abstraction. Two other models proposed by Kota [19] and Deng [13] introduce only two levels of function and behaviour. For most of these models, the relationships are restrictive, linear (from the more abstract to the more physical levels) and rigid.
On referring to the models that consider the FBS levels, Williams describes the functions in terms of dynamic movement and the flow of systems. For both cases, the function considers the system as having a single input and single output (SISO) and the mathematical form predominates on every level of abstraction. The behavioural level of his model encapsulates the structural elements. Based on this encapsulated structure, the model adopts the relationship form $F \to BS$. The functional level in the Chakrabarti model also represents the SISO system. As regards its syntax, this functional level uses both natural language and the mathematical form, which is also used by the other levels of abstraction. Due to the clear-cut division of the levels, the relationships occur under the form $F \to B \to S$. The function of Zhang's model considers the designer's intention by means of natural language and the behavioural level encapsulates the structures. The syntax of the behavioural level considers both natural language and the mathematical form at the same time. The structural level lacks a clear description. The relationships between the levels occur in the form $F \to BS$.
As far as the models that consider the FB levels are concerned, Kota uses the mathematical form for both the function and behaviour levels. The model by Deng considers natural language and the mathematical form for both the functional and behavioural levels. Unlike other models, Deng proposes

a functional level with two focuses, i.e. intentions and operations. Both Kota's and Deng's models adopt the relationships in the form $F \rightarrow B$.

1.3 Problems detected

Based on the previous study, Deng's model was considered to be the one that was best suited to representing functions with two focuses and behaviour with more precision. The objective of the main study is to adapt this model to a more flexible synthesis scheme. To accomplish this goal, the proposed model will have to solve the following problems: difficulty to adapt the model to a synthesis process focused on the user, difficulty to maintain the interaction between the user and the information, restriction to explore the knowledge that has been stored, restriction on the domain of the knowledge represented, and difficulty to reuse the knowledge from past projects.

To solve them, this article introduces a representation scheme that considers improvements to facilitate designers' intervention during the synthesis cycles, to return to higher levels of abstraction in the design process and thus improve the design solution, and to provide better control over the process of generating alternatives.

2 PROPOSED MODEL

The FBS scheme outlined above allows certain information to be represented for the synthesis. Although it presents a wide approach, its scheme lacks precision and it cannot be adapted easily to more flexible reasoning models. Nonetheless, we propose a knowledge representation model to improve the flexibility of the exploration during the synthesis. The following sub-section describes the levels of abstraction that go to make up our scheme. Next, the second sub-section shows how the model uses relationships to connect the levels of abstraction in a more flexible manner.

2.1 Levels of abstraction

The proposed model is based on the levels of abstraction defined by Deng: called purpose function, describes the designer's intention; the second level, called action function, abstracts the desired behaviours; the third level, called behaviour, abstracts the physical states of a component; the fourth level, called structure, abstracts the geometry of a physical component or mechanism; and the environmental input, which describes the external flows that affect the execution of the product. Based on these levels, several attributes and relations between them are defined in order to allow for a more flexible use.

2.1.1 Functional level

To facilitate interaction between the user and knowledge during synthesis, the model suggests the reasoning process start with a description of the problem, which can be expressed by means of the designer's intentions or the product activity. For this reason, the level of function presents two focuses: the designer's intention, called the purpose function (f_i), and the operation of the product, called the action function (a_j). This distinction between these two functions is necessary and important. First, the purpose function is a higher level of design abstraction and, as such, it makes it possible to choose from a wider range of design solutions. With carefully elaborated functional vocabulary, this function may be used to enable wider and more flexible design synthesis to take place. On the other hand, the action function is focused on a specific design where the behaviour of an artefact is related to certain actions. This kind of function has a more direct relation with physical principles which may be used to seek a more specific design solution.

The purpose function is the most abstractive level of those that describe the intention of the designer or the objectives of a design. Its semantics consider only one attribute with the same name (purpose function). Its syntax uses natural language in the form of verb + complement. For the verb attribute, we should use the second class of verbs introduced by Hirtz [20]. This contribution of the representation model allows the user to define the intention of the product and to avoid ambiguity. Table 1 shows the semantics, syntax and an example of this kind of function.

Table 1 – Semantics of the purpose function level and an example of applying this scheme

Semantics	*Example*
Purpose function (natural language)	"regulate occupied volume"

The action function (a_j) abstracts the behaviour of the product and it is operation-oriented. Based on this level, designers can guarantee the achievement of an intention. Its semantics consist of the following attributes:

- Description: this attribute describes the objectives of the function.
- Driving input and functional output: flows of input and output that affect the performance of the product, mechanisms or components. Unexpected flows are not yet considered at this level.
- Initial and ending states: effects expressed during the operation of the components.
- Purpose functions: List of functions that can be met by the action function.

In its syntax, this level considers both natural language and mathematical representation forms. When natural language is used to indicate a flow, the attribute uses the verb + complement form. To avoid ambiguity, as in the case of the purpose function, the driving input and functional output should use verbs of the third class according to Hirtz's classification [20]. This representation scheme proposes two contributions: the integration of flows, i.e. object (material, energy or signal) and action (movement), and the dependence relationship between an output flow and the final state. For this reason, an output flow represents a consequence of an indicated final state. Table 2 describes the semantics and an example of the current level of abstraction.

Table 2 – Semantics of the action function level and an example of the action function with this scheme

Semantics	*Example*
Description	To regulate or reduce the volume of a component to its minimum value
Purpose functions	Regulate volume
Input-output flow: - Driving input - Functional output	Driving input: Actuate after twisting movement or Stabilise weight Functional output: Change dimension of the component

2.1.2 Behaviour level

Behaviour (b_k) is the level of abstraction that describes the intermediate states of a component or physical mechanisms by means of the input and output flows. In this approach, the states can be the internal states of a component, the internal interaction between two or more components, or the interaction between a component and its surroundings. Because it focuses on the physical element, this abstraction presents more precise information. To make this information readable by the user, the level includes attributes to describe the objectives of the behaviour in a language that is more familiar to the user.

From the computational point of view, the level of behaviour is similar to the action function. But there are differences in some characteristics, e.g. it considers the unexpected or harmful flows of the system, describes the intermediate states of a physical component and describes precise values for the flows and states. Due to the specific character of the states, the user can identify all the intermediate states and organise them in chains or networks. This network representation scheme contributes to allow the user to describe variations in the way the product behaves [21].

The semantics of the behavioural level presents the following attributes:

- Description: similar to the description of the action function.
- Driving input and functional output: flow of an object (material, energy or signal) or an action that affects the operation of the system.
- Harmful input and side-effects: flows of an object (material, energy or signal) or an action that hinders the correct operation of the product. Harmful input and side-effects are considered to be unexpected flows for the system.

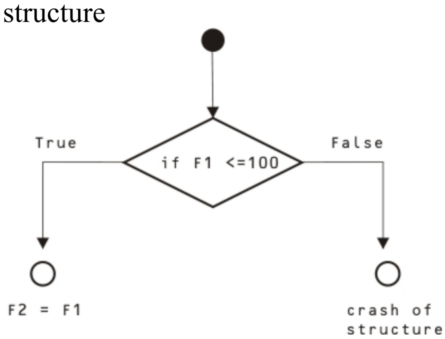
- States: besides the normal initial and ending states, the current attribute can take into account intermediate states and more than one final state. This attribute can be organised into networks of chains of states.
- Action functions: list of action functions met by the behaviour element.
- Purpose functions: list of purpose functions attended to by the behaviour.

The syntax of the behavioural level generally uses both a mathematical form and natural language in its attributes except for the description and purpose functions, which use only natural language. This mathematical form considers constants, ranges or parametric functions as common formats with which to express the precise numeric values.

During synthesis, behaviours can be organised in the form of chains or networks to generate variations of a unique design alternative.

This level of abstraction represents behaviour by means of chains or networks of states. The result of these states defines variations in the output of object flow and action flow. Table 3 describes the semantics and an example of a behavioural element.

Table 3 – The semantics of the behavioural level and an example of behaviour with this scheme

Semantics	*Example*
Description	To support a vertically distributed load or which acts on a certain point on a smooth surface
Driving input (expected input)	Weight: 100 kg * Unit: kg Range: 0-100
Functional output (expected output)	Normal force perpendicular to the surface
Harmful input (unexpected input)	Weight equal to or greater than 100 kg
Side effect (unexpected output)	Break-up of structure
Initial and ending states (chain or network)	

* These natural language values are just for demonstration purposes.

2.1.3 Structure level

Structure (s_l) is the least abstract level and describes the geometry of a physical component or mechanism. Based on this level, the user can get a formal idea of the solution from the model. During the synthesis process, this level can consider a parametric model of the component to be adapted to the designer's needs. With this parametric model, the user receives a visual preview of the component or the mechanism formed during the synthesis. It is accepted in this research work that this parametric model would provide the basis for future extensions of the representation model. By means of this extension the user will be able to consider the surfaces of contact, which allow the shape of the product to be optimised. In its semantics, the current level presents the following attributes:

- Description: Attribute that gives a literal description of the component or mechanism.
- Purpose functions: List of functions that can be executed by the structure.
- Action function: List of action functions that can be met by the structure.
- Behaviours: List of behaviours performed by the structure.
- Geometric definition: Geometric representation of the physical element.
    - Solid geometry: Parametric abstraction of the geometry of the component. A parametric model can be encapsulated to make a preview of the geometry.
    - List of surfaces of contact:

- ID: Identification code of the surface.
- Surface type: Type of surface that allows the contact, e.g. plane, cylinder, concave surface, convex surface, etc. [22].
- Degree of freedom (DOF): degree of freedom between the two surfaces [23].
- Compatible surfaces: Surfaces that participate in the composition of the product.

In its syntax, the structure level allows for the use of three types of codification: natural language, a mathematical form and direct reference between elements. Similar to the behavioural level, the mathematical form is used in every sub-attribute of the geometric definition and direct reference is used in the lists of action functions and behaviours.

2.1.4 Environmental input level

Environmental inputs are the external elements that may be needed to perform a particular behaviour. By means of this level, the user can describe these external elements and control the solution space. To guide the synthesis process, the model connects the environmental level to both the action function and behaviour levels. These connections occur when the external elements fulfil the requested inputs that affect the functionality of the system. Through this method, the model allows chains and networks of action functions and behaviours to be controlled during the generation of alternatives.

From the computational point of view, the environment level controls synthesis. During reasoning, the model starts the exploration with action functions and behaviours in order to implement the intentions. Certain action functions and behaviours are given priority if their required inputs match the environmental inputs of the system.

To connect the two levels of action function and behaviour, the environment also considers two levels of representation: one with more precise information to connect with the behavioural level and another with less accurate information to connect with the action function level.

For the action function level, the semantics of the environment considers the attributes of description, type of flow and its values (unique or range). For the behaviour level, the semantics inherit the same attributes but specify the precision of these values. Each specialisation depends on the type of flow, e.g. action, material, energy or signal.

2.2 Relationships

To represent a product bearing in mind its intentions, behaviours and physical abstraction, the model stores the necessary data and relates them in a logical sequence. Without them, the result can be an incomplete solution, misuse of the available computational resources or problems in the synthesis of the solution. To better guide synthesis and make better use of the available resources, this representation scheme has to consider relationships that link the stored information in a flexible manner. This representation scheme extends Deng’s assumptions and allows the user to participate on an intensive basis during the synthesis, as well as increasing the flexibility of the exploration process. With these relationships, the user can abstract from complex problems and encounter no restrictions to start the synthesis from any level of abstraction.

This section shows the relationships available at each level of abstraction. Each relationship presents its attributes of connection and benefits. Figure 1 presents the logical sequence of exploration by means of the connections that are available. The bold arrows indicate the contributions of the current representation scheme.

2.2.1 Relationships on the purpose function level

Before starting the reasoning process, the user has to describe the design problem. The model outlined here suggests starting the activity with the following information: intentions of the product and flows of the environment. We have to take into account that this procedure cannot be considered a rule. Rather, it gives the freedom to select a suitable level of abstraction to describe the design problem.

After the initial definition, the user can explore the knowledge of the three subsequent levels, according to the search sequence that is defined. This search can be carried out based on two methods: one automatic, without the intervention of the user, and other guided by the user. In the automatic process, the model looks for action functions that attend to the intentions. If no action function is found, the model extends this search to the next level of behaviour and subsequently to the structure level. If the search process cannot match this intention, the function is considered to be complex and subsequently has to be sub-divided into simple intentions. To connect two purpose functions, the model uses natural language, based on the verb + complement format. This kind of relationship is similar to the Behavioural and Functional decomposition proposed by Zhang [18]. According to this author, combining both methods can prevent the design problem from being broken down too finely, thus avoiding combinatorial explosion during the behavioural configuration. This usually results in a more compact and less costly design.

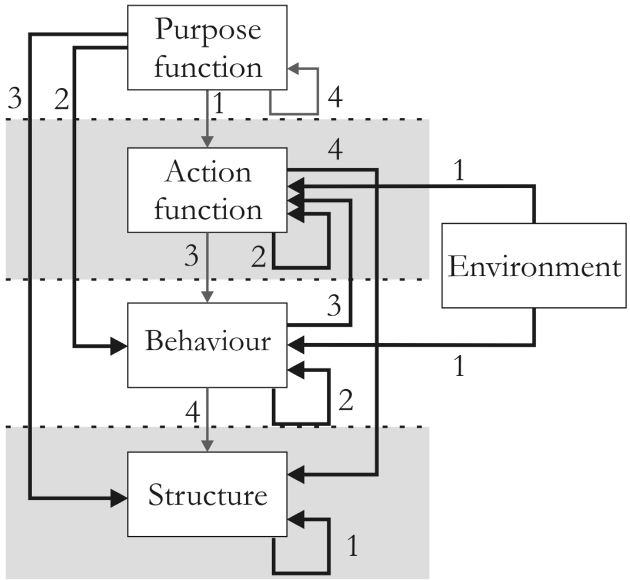


Figure 1 - Overview of the representation scheme and relationships available to connect the levels of abstraction

2.2.2 Relationships on the action function level

When the model finds an action function, the user is prompted to provide the driving input port with an available environmental input. Should the process fail to find the requested flow, the exploration of action functions is activated to meet this requested input. Chains of action functions are formed until they again find an environmental input to meet the inputs of the last combined action function. In this example, an initial action function requires electric energy to generate rotation. As presented, the initial step looks for an external element that can supply this energy input. If this kind of flow cannot be matched by any available environmental input, the model explores other action functions to supply energy. Action functions with energy output and input related to any available environmental input are given higher priority in the selection. The result of this combination favours the formation of chains of action functions to attend to their inputs with an available external flow.

If an external flow meets the last input in the functional chain, this chain is considered to be completed, and subsequently the process combines behaviours or structures for each action function in the chain. The connection between an action function and a behavioural element occurs in the following situation: the input of the action function matches the input of the behaviour and the output of the action function matches the output of the behaviour. This process is illustrated in Figure 2. If the action function cannot be attended to by an equivalent behaviour, the synthesis forces networks or chains of behaviours to form in order to equalise their inputs and outputs. Another contribution of the current representation scheme is that it allows the user to create variations of the same alternative based on this network of behaviours. In brief, the combination between the action functions and behaviours allows the following combinations to occur:

- A behavioural element matches the input and output of an action function.
- A chain of behaviours is formed when the extreme inputs and outputs match the input and output of an action function.
- Or a network of behaviours is formed to attend to the action functions. This combination increases the number of alternatives by means of variations of the solution.

If no available behaviour is found by the model, structures are combined. The connection between an action function and a structure occurs only through a direct reference.

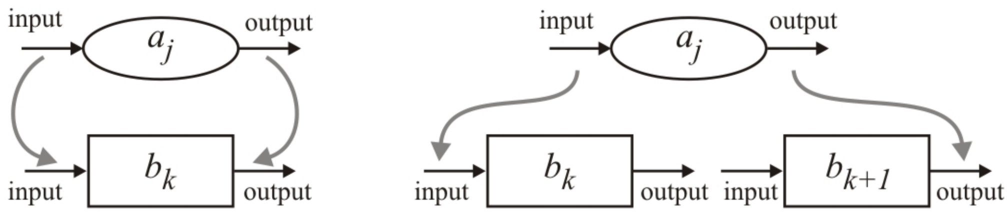


Figure 2 - Connection between an action function and a behaviour or between an action function and two or more behaviours

2.2.3 Relationships on the behavioural level

The level of behaviour explores connections in the following order:

- Behaviour – environment
- Behaviour – behaviours
- Behaviour – action function – behaviours
- Behaviour – structures

The two initial explorations are similar to the relationships existing on the action function level. The initial behaviour may form chains or networks of behaviours to match its required input with a flow of the environmental input.

During the exploration, the model may find redundant searches of behaviour. To avoid this problem, we propose a relationship that allows a return to more abstract levels, which thus enables an abstract re-definition of a design problem to be obtained. As a contribution, this representation scheme allows the user to return to this level of abstraction. This is achieved by means of the behaviour–action function–behaviour relationship.

One of the processes that end the implementation of an intention is the relationship between behaviour and structure. By means of this combination, the user may find the structures that present the expected behaviours. These structures may be a unique component or a group of components that form a mechanism. In this latter situation we have to consider that some kinds of behaviour exist due to the interaction between two or more components, which are called mechanisms here.

3 CASE STUDY

In this section, we will study one design case to prove the efficiency of the proposed representation scheme. In this case study, the functional design process will be explained. This case will describe the implementation of an initial intention and define the possible solution variations that can be generated during a synthesis process.

The resolution was divided into phases of knowledge capture and problem reasoning. The knowledge captures stage involved including the previous knowledge and information incorporated during the synthesis. The database of past knowledge consisted of items of furniture taken from a catalogue. This information was decoded and distributed over the four levels of abstraction. The knowledge captured during the synthesis included the designer knowledge gained from interaction with the model. Both knowledge capture processes make it possible to store the information needed to form the knowledge database. Based on this information, the designer could use knowledge from past projects to solve routine design problems or to adapt them to new intentions or principles.

The experiment started with the intention “Support writing activity” and two external flows {“Human force”, “Vertical load”}. This initial intention supported the creation of 14 distinct solutions.

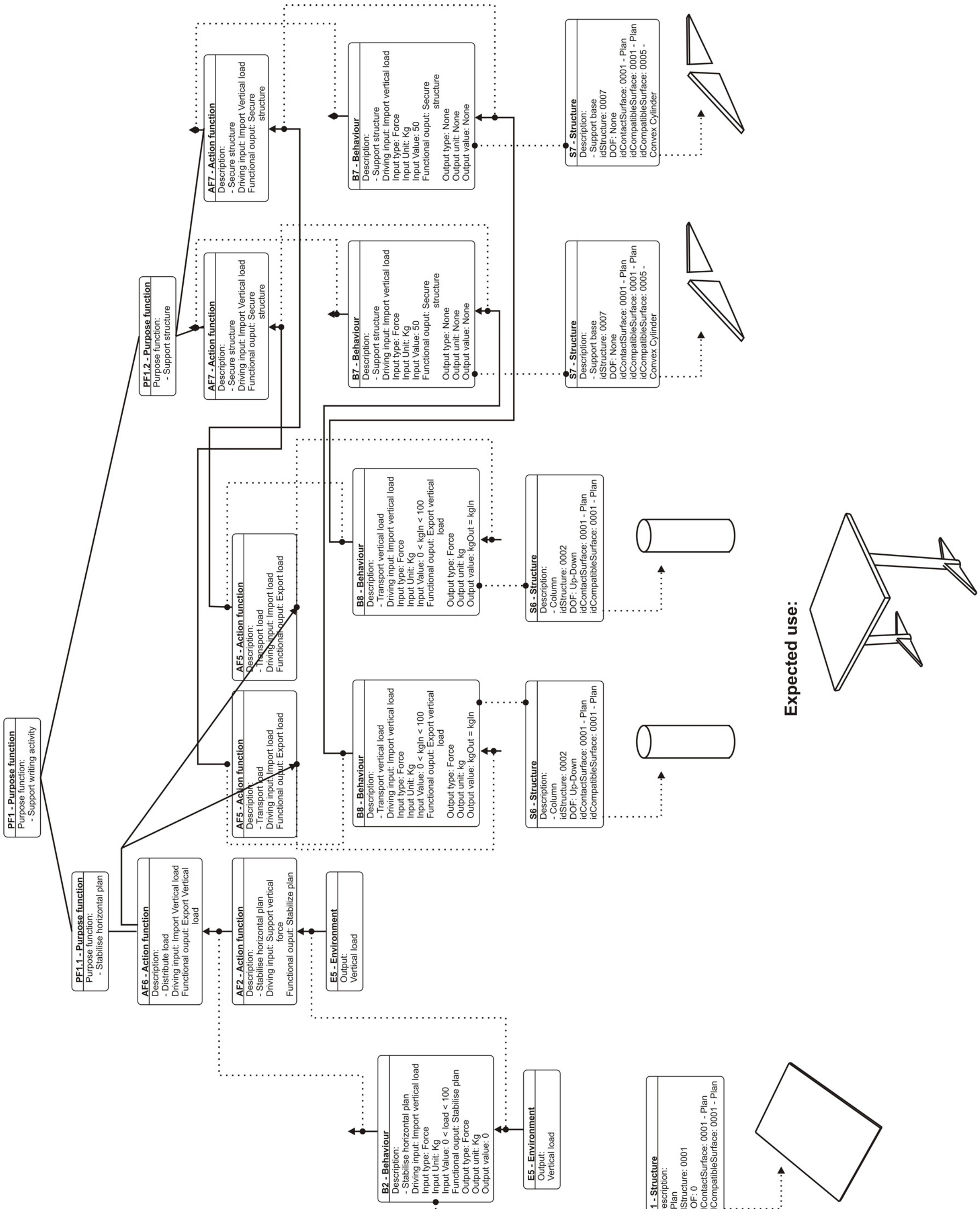


Figure 3 – Alternative generated for the design problem

After defining the intention and the external flows, the model did not present the knowledge needed to implement the intention. Next, this complex function was sub-divided into two other sub-functions {“Stabilise horizontal plan”, “Support structure”}. Subsequently, the model attends the sub-functions with action function of input ports of “Vertical load”. Some of these action functions formed chains with the set of action functions {“Distribute load”, “Stabilise horizontal plan”, “Transport load”, “Secure structure”} to attend to their requested input with the available environmental inputs. Based on the chains that were formed, some behavioural elements were explored and combined, and then some structures were used to implement these behaviours.

Due to the flexibility of the model, an intention could be included and variations of the same alternative could be generated. One variation incorporated the purpose sub-functions of “Regulate height”. By means of this function, the solution could synthesize new mechanisms to raise or lower the system. This new functionality thus adapted the product to a new user need. Figure 5 illustrates the new alternative that was generated.

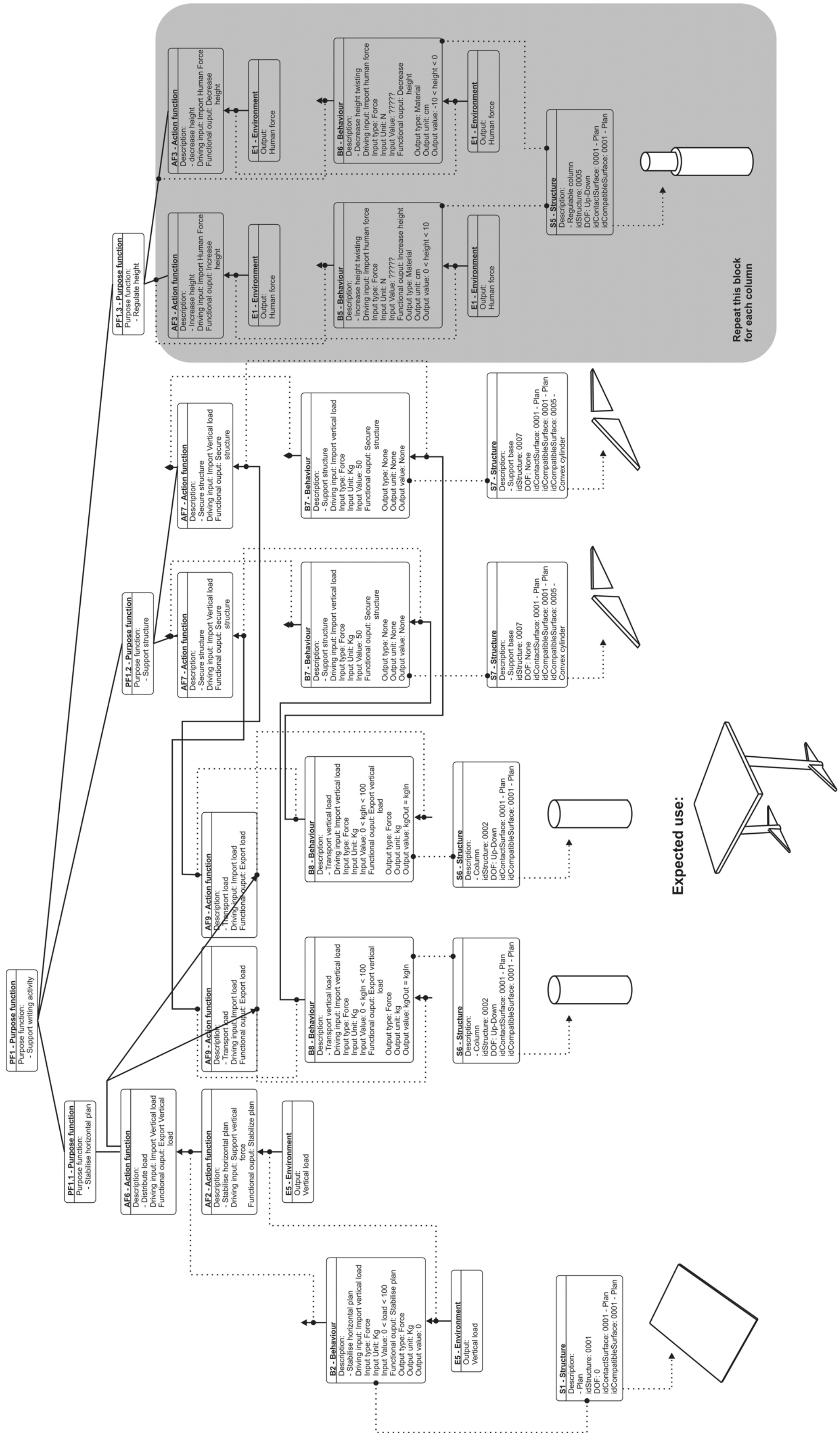


Figure 4 - Variation of the alternative for the same design problem.

4. CONCLUSIONS

This research work proposes a procedure for exploring and synthesizing a model based on the FBS framework and shows its application to a design case. The proposed model satisfies the following objectives:

- The user can go back to higher levels of abstraction in order to abstract the design problem and to create variations of the same design alternatives. With this method, the model adopts either a top-down or a bottom-up exploration.
- The user can start the design problem from any level of abstraction. As suggested by the model, the user can describe the problem by means of the intentions of the product and, if necessary, the problem may be described by means of an action function, behaviour or structure. With this method, the model has no restrictions to start reasoning and allows the user to tackle the problem with predefined behaviours or structures.
- The user controls the granularity of the alternative by means of the environmental input level. The proposed representation scheme takes into account the external flows in order to control the divergence of the design space during synthesis.

REFERENCES

[1] Campbell M., Cagan J. and Kotovsky K. The a-Design Approach to Managing Automated Design Synthesis. *Research in Engineering Design*, 2003, 14 12 - 24.

[2] Xu Q.L., Ong S.K. and Nee A.Y.C. Function-Based Design Synthesis Approach to Design Reuse. *Research in Engineering Design*, 2006, 17 27-44.

[3] Chakrabarti A., Sarkar P., Leelavathamma B. and Nataraju B.S. A Functional Representation for Aiding Biomimetic and Artificial Inspiration of New Ideas. *Artificial intelligence for engineering design,analysis and manufacturing(AIEDAM)*, 2005, 19 113-132.

[4] Zhang W.Y., Tor S.B. and Britton G.A. Funcdesigner a Functional Design Software System. *International Journal of Advanced Manufacture Technology*, 2003, 22 295-305.

[5] Parmee I.C. *Evolutionary and Adaptive Computing in Engineering Design*, 2001

[6] Al-Salka M., Cartmell M. and Hardy S. A Framework for a Generalized Computer-Based Support Environment for Conceptual Engineering Design. *Journal of Engineering Design*, 1998, 9 (1), 57 - 88.

[7] Horvath I. Conceptual Design - inside and Outside. In *2nd Seminar and Workshop EDIPro*, *Vol.* Poland, pp.10

[8] Gero J. Design Prototypes: A Knowledge Representation Schema for Design. *AI magazine*, 1990, 11 (4), 26 - 36.

[9] Umeda Y., Takeda H., Tomiyama T. and Yoshikawa H. Function, Behaviour, and Structure In (Eds) J. Gero. *Applications of Artificial Intelligence in Engineering V*, 1990 (Springer)

[10] Dorst K. and Vermaas P.E. John Gero'S Function-Behaviour-Structure Model of Designing: A Critical Analysis. *Research in Engineering Design*, 2005, 16 17-26.

[11] Far B.H. and Elamy A.H. Functional Reasoning Theories: Problems and Perspectives. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM)*, 2005, 19 75-88.

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

[13] Deng Y. Function and Behavior Representation in Conceptual Mechanical Design. *Artificial intelligence for engineering design,analysis and manufacturing(AIEDAM)*, 2002, 16 343-362.

[14] Chakrabarti A. Supporting Two Views of Function in Mechanical Designs. In *The Fifteenth National Conference on Artificial Intelligence - AAAI'98*, *Vol.* Madison, Wisconsin, July 26–30, 1998,

[15] Williams B.C. Interaction-Based Design: Constructing Novel Devices from First Principles. In *Intelligent Computer Aided Design*, *Vol.* North-Holland, pp.255-282

[16] Chakrabarti A., Langdon P., Liu Y. and Bligh T. An Approach to Compositional Synthesis of Mechanical Design Concepts Using Computers In (Eds) A. Chakrabarti. *Engineering Design Synthesis. Understanding, Approaches and Tools*, 2002 (Springer-Verlag)

[17] Liu Y., Chakrabarty A. and Bligh T. A Computational Framework for Concept Generation and Exploration in Mechanical Design Further Developments of Function. *Artificial Intelligence in Design '00*, 2000, 499 - 519.

[18] Zhang W.Y., Tor S.B. and Britton G.A. A Heuristic State-Space Approach to the Functional Design of Mechanical Systems. *The International Journal of Advanced Manufacturing Technology*, 2002, 19 (4), 235-244.
[19] Kota S. and Chiou S.-J. Conceptual Design of Mechanisms Based on Computational Synthesis and Simulation of Kinematic Building Blocks. *Research in Engineering Design*, 1992, 4 75-87.
[20] Hirtz J., Stone R., McAdams D., Szykman S. and Wood K. A Functional Basis for Engineering Design: Reconciling and Evolving Previous Efforts. *Research in Engineering Design*, 2002, 13 65 - 82.
[21] Martel A. Applications of Ergonomics and Consumer Feedback of Product Design at Whirlpool In (Eds) *Human Factors in Consumer Products*, 1998 (Taylor & Francis)
[22] Mulet E. *(Phd) Análisis Experimental Y Modelización Descriptiva Del Proceso De Diseño*, 2003 (Politechnical of Valencia).
[23] Bruno F., Giampà F., Muzzupappa M. and Rizzuti S. A Methodology to Support Designer Creativity During the Conceptual Design Phase of Industrial Products. In *International Conference on Engineering Design, ICED 03*, *Vol.* Stockholm, Sweden,

Supported by the Programme Alban, the European Union Programme of High Level Scholarships for Latin America, scholarship no. E04D030611BR

Contact: Dr. Elena Mulet
Universitat Jaume I. Department of Mechanical and Construction Engineering
Av. Sos Baynat, s/n. 12071, Castellón de la Plana. Spain
Phone: +34 964 72 8114
Fax: +34 964 72 8106
e-mail: emulet@emc.uji.es