

CONVERGENCE APPROACH IN EXPERIMENTAL RESULTS

Camelo, D¹.; Vidal, R¹.; Mulet, E.¹; García, L.A.²

Grupo de Ingeniería del Diseño

Universitat Jaume I

Av. Sos Baynat s/n (12071) – Castellón de la Plana

Phone: +34 964 729252

E-mail: dioclecio.camelo@alumail.uji.es

RESUMEN

El proceso de síntesis está basado en fases de divergencia y convergencia. Diversos autores demuestran el proceso de divergencia en la forma de búsqueda para la obtención de mejores alternativas para los requerimientos de diseño. Todavía existen muy pocas manifestaciones en el proceso de convergencia para evitar las posibles explosiones combinatorias presentes en sistemas más complejos. La aproximación heurística es un proceso que puede gestionar esa complejidad en el proceso de resolución funcional (convergencia). Su uso difiere de los algoritmos (procedimientos matemáticos) por estar basada en reglas generales sacadas de las experiencias. Los programas heurísticos son conocidos por su proceso de auto-aprendizaje, lo que puede generar soluciones más optimizadas y eficientes para los requerimientos de los procesos de diseño.

Este artículo expone una investigación del Grupo de Ingeniería del Diseño de Castellón en la generación de una mejor solución en un caso real de diseño, a través del uso del algoritmo de búsqueda best-first search, presentado por Zhang. Además de aportar otro punto de vista en la creación de un modelo computacional más eficiente y eficaz para el diseño automatizado.

Palabras clave: aproximación heurística, resolución funcional, diseño automatizado

ABSTRACT

The process of synthesis is based on phases of divergence and convergence. Many authors illustrate the divergence process to find better alternatives for the design requirements. Although there are still few explanations to prevent the possible combinatorial explosion present in complex systems. The heuristic state-space approach is an estimative process that can manage the complexity in the functional reasoning process (convergence). Its use differs from that of algorithms (mathematical procedures) because it is based on commonsense general rules taken from experience. Heuristic programs are well known for their capacity for self-learning, which can generate better optimised and more efficient solutions for design requirements.

This article describes a research project carried out by the Engineering Design Group of Castellón on the generation of a better solution to a real design case, through the use of the best-first search algorithm presented by Zhang. We will therefore provide another point of view on the creation of a more efficient computational framework for the automated design process.

Key Words: heuristic search, functional reasoning, automated design

1 INTRODUCTION

The advance of CAD systems has provided us with tools that enable the detailed design phases of the development of a product to be widely optimised. Nevertheless, these support systems have shortcomings in the resolution of problems during the conceptual phase of the design process. The Engineering Design Group of Castellón devotes part of its research to the area of support tools for solving problems in the conceptual phase of product development, using methods and computational frameworks to aid designers in the routine development of products.

The purpose of this article is to explain the process of developing a product using the synthesis algorithm and the heuristic state-space approach presented by Zhang [1]. During the conceptual design phase, the functional syntaxes of Hirtz [2] and the functional decomposition of Zhang were used to aid in the search and definition of behaviours inside a knowledge database.

The present work is part of a project called FiBiuS, which consists of a synthesis system to aid the design activity. It intends to operate over a new Model Library structure, thus allowing not only collection of the knowledge available but also help in the routine design of new products. The system architecture is based on the Function-Behaviour-Structure (FBS) approach and on the use of ontology frameworks aimed at using and sharing knowledge between different applications. With this experience we expected to identify opportunities to improve the algorithm in the following aspects: to reach a more optimised solutions space showing more feasible alternatives, rather than just one solution; to identify other heuristic ways to obtain better combinations and to find more innovative alternatives in the design space.

The functional reasoning introduced here uses the assumption of multiple divergence-convergences, which can rule out many unfeasible alternatives during the synthesis process and thus reveal the best path to reach the goal and prevent combinatorial explosion. Zhang recommends a decomposition using a multiple level of abstraction to get better alternatives to the solution, and justifies the fact that poor alternatives can be generated when a design process occurs on a single level of divergence-convergence.

As a way to prove the efficiency of the algorithm, the real development case reported by Vidal et al. [3] was used to supply the system with information as a basis for the construction of a library with behaviours, functions, constraints and environment resources. This research, however, generated a feasible compact solution and aided designers in the selection of alternatives in the solution space through the use of the heuristic state-space approach.

2 FUNCTIONAL REASONING AND THE HEURISTIC STATE-SPACE APPROACH

In this section the Heuristic state-space approach is explained by the functional reasoning that solves a real design problem. This reasoning is the algorithm proposed by Zhang [1], which is based on the best-first search. In spite of its origins in the realm of

Artificial Intelligence, this search algorithm is frequently used in GIS problems, game solutions and applications that demand a search process to find the best route between the initial and the target state. The reasoning uses simple phases of exploring, combining and selecting the best solution from a list and, if necessary, expanding the exploration process until the functional reasoning has been completed. As its procedure is basically guided by exploration, this synthesis is recommended for use in routine and non-innovative solutions due to the fact that it is unable to create new devices, but just new combinations of available devices.

As an initial approach, Zhang based his algorithm on the A* Search algorithm. The precept of this search algorithm is to find the shortest path between two points (problem and solution) through a vector comparison and the heuristic estimate of costs.

For this estimate a Fuzzy Multi-Criteria Decision Making (FMCDM) calculation is used to qualify alternatives by means of the linguistic terminology present in the combination of function and behaviour. Each criterion in the database is a result of converting the qualitative (linguistic) values into quantitative (score) ones. A number of criteria like assembly, manufacturability, material costs, etc., are taken into account. This assumption can allow alternatives to be compared by extracting the best one to be explored. It leads to a linear way to the solution, consequently providing better and more compact alternatives [4].

2.1 Algorithm proposed by Zhang

Fig. 1 depicts the functional reasoning algorithm proposed by Zhang. It shows the steps needed to break down functions into sub-functions and, if possible, to compound behaviours into functional requirements for the representation of the solution.

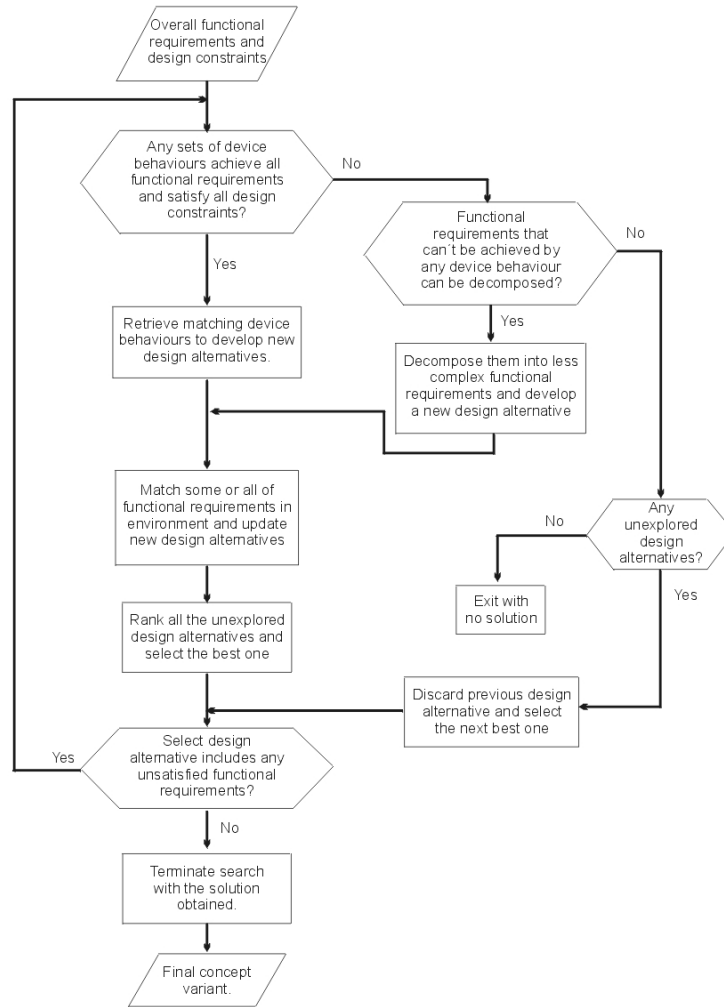


Fig. 1 - Heuristic state-space algorithm purposed by Zhang

Initially, the algorithm is supplied with all function requirements, constraints and resources from the environment, which provides sufficient information for the exploration process. Then, the system verifies the possible combination with existing behaviours that satisfy the constraints; otherwise, the functions are broken down into more sub-functions extracted from a knowledge database. After this expansion, the algorithm creates new alternatives and selects the best one from the list.

Following these steps, finally just one solved alternative is generated. As stated by the author, the solution is the best and most compact alternative for the problem.

2.2 Heuristic calculations

A heuristic estimative is used to compare each alternative in a rank. This approximation starts with the Fuzzy Multi-Criteria Decision Making problem that calculates a weight w for each criterion and the ratio r available for each behaviour in the alternative. This process translates the linguistic terms into a numbered score, thus allowing it to be compared and ranked in the next phase.

The following equation is used to compare one alternative with another, which represents the heuristic cost.

$$\hat{h} = \sum (1/\bar{r}_i) + k$$

Where:

- \bar{r} is the numeric value for the behaviour present in the alternative. $\bar{r} \in (0,1)$
- k represents 1 for each function left unexplored or incomplete function requirement in the alternative.

\bar{r} represents the classic use of the weighted average aggregation method. In the formula:

$$\bar{r}_i = \frac{\sum_{j=1}^n w_j r_{ij}}{\sum_{j=1}^n w_j}$$

Where:

- w represents the crisp weight of the criterion in the combination behaviour-function of the alternative. $w_j \in (0,1)$
- r represents the crisp performance ratio of the criterion in the same combination. $r_{ij} \in (0,1)$

Values of r and w are results from a linguistic calculation based on triangle or trapezoidal criteria defined in a range between [0, 1].

Table 1 shows initial values from one of the alternatives that were studied. These linguistic values, presented by the system, are the result of information available in the database from the function-behaviour combination, and then the estimative calculations are applied to rank and select alternatives.

						average
1	very low	0	0	0,1	0,2	0,08
2	low	0,1	0,2	0,2	0,3	0,20
3	fairly low	0,2	0,3	0,4	0,5	0,35
4	medium	0,4	0,5	0,5	0,6	0,50
5	fairly high	0,5	0,6	0,7	0,8	0,65
6	high	0,7	0,8	0,8	0,9	0,80
7	very high	0,8	0,9	1	1	0,93

Table 1 - Linguistic values

3 ALGORITHM APPLICATION TO A CASE STUDY

The present section introduces the application of Zhang's divergence-convergence approach to a real design problem. Previous research by Vidal et al. [3]

was used to verify the efficiency of the algorithm. This research is a protocol study of different brainstorming techniques in the divergence process for the design of a drawing support for designers, architects and engineers. It involved 12 experiments with 4 teams, each made up of 5 design students. To guide this experiment, some requirements were established to improve the characteristics of the product, namely: optimisation of space to be taken up, limits of the board, the need to be steady when not in use, and so on.

For the initial stage, a knowledge database of functions and behaviours was created to aid the system in the reasoning process. The structure of these data uses concepts proposed by Zhang [5] and Hirtz [2]. Specifically, Zhang defines a behaviour structure based on object-oriented analysis, with rules, qualitative and quantitative data. As the outcome of this exploration, a set of alternatives will be shown as a result of the reasoning process.

The functional structure uses the assumption of Hirtz [2], which makes use of verbs from the secondary and tertiary classes. The representation is:

Function representation

Function code – Infinitive verb + noun as a complement

Ex.:

F1 - Support a plan for drawing

The functional hierarchy in the database was defined like this:

- F1 - Support a plan for drawing
 - F1.1 - Stabilise plan for drawing action
 - F1.2 - Position plan (at a height)
 - F1.2.1 - Regulate structure in height
 - F1.2.2 - Guide structure
 - F1.3 - Regulate angle of tilt
 - F1.3.1 - Increase angle of tilt
 - F1.3.2 - Decrease angle of tilt
 - F1.4 - Change volume (space occupied)
 - F1.4.1 - Shape plans for compactness
 - F1.4.2 - Increase volume
 - F1.4.3 - Decrease volume

This library illustrates how each function could be broken down into sub-functions by the algorithm.

In addition, in this research behaviours were taken as being a physical interaction of the design component and its environment. Therefore, the behaviour library has a structure that interacts with the function entity and the environment through the driving input and the functional output. This could allow for a combination with resources supplied by designers. For this reason, behaviours have the following representation:

Behaviour representation

Behaviour code – Description

Purpose: Verb in infinitive + complement
Initial/ending state: Verb + complement
I/O flow of action
– Driving input: intended input action
– Functional output: intended output action

Ex:
B5 – Regulate by twisting
Initial/ending state: Change height of structure by twisting
Driving input: User intervention
Functional output: increase or decrease height

4 RESULTS - FUNCTIONAL REASONING IN A REAL CASE

In this section, the use of the algorithm for designing a drafting table will be outlined. All steps explained by the heuristic approach were applied and calculated to prove its efficiency. We will explain how the decomposition function was obtained and how it could solve the problem of the interpretation of functions. Then the behaviour search will be shown using the library available, and all the alternatives generated throughout the exploration and combination process will be calculated.

The process begins with information supplied by the designer. These data are the function requirements, constraints and environments that are provided.

After definition of the function requirements and environment, the system built the first alternative called A0, shown in Fig. 2, where F1 means “*Support a plan for drawing*”.

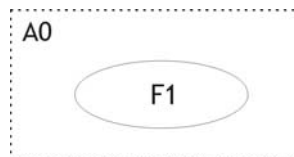


Fig. 2 - Alternative 0

4.1 Behavioural search

As presented by Zhang, this set might belong to a domain that is broad and could involve other types of problems, which could be of a hydraulic, thermal, electric, etc., nature. In our case, the system makes use of a behaviour set related to the mechanical domain, thus restricting the search process even more.

After the initial definition, a great deal of behaviour is found but none was capable of satisfying the functional requirement. In spite of this, the system tended to decompose the original function into more sub-functions by consulting the function library described above.

4.2 Decomposition of function

The decomposition is performed by means of an exploration of the knowledge database. This constitutes a new alternative A1, shown in Fig. 3, where functions mean:

- F1 – Support a plan for drawing
- F1.1 – Stabilise plan for drawing action
- F1.2 – Position plan (at a height)
- F1.3 – Regulate angle of tilt
- F1.4 – Change volume (space occupied)

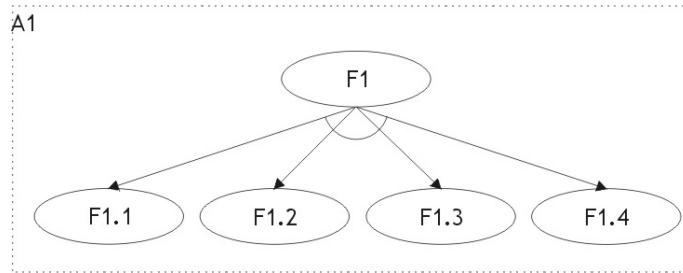


Fig. 3 - Decomposition of functions

After decomposition, the system seeks for other behaviours that should connect with the function requirements available in the selected alternative. The system found 4 feasible behaviours that satisfied the functions and the constraints, namely:

- B1 - Establish a surface
- B2 – Regulate length by twisting
- B3 – Divide components
- B4 – Rotate plan

Subsequently, the system combined the behaviours with the function requirements, as well as the environment available. With these combinations, new alternatives were included in a vector that was to be calculated and used afterwards. The composition process generated 5 alternatives, as illustrated in Fig. 4.

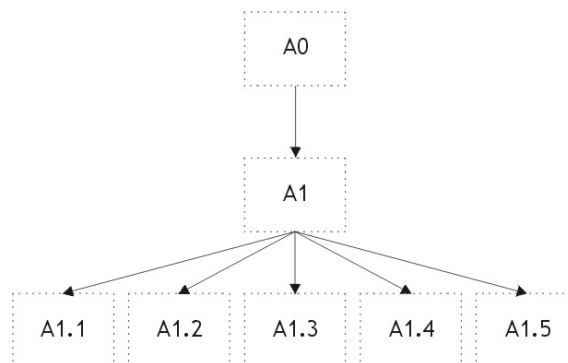


Fig. 4 - Alternative set

An example of the alternatives generated is shown in detail in Fig. 5.

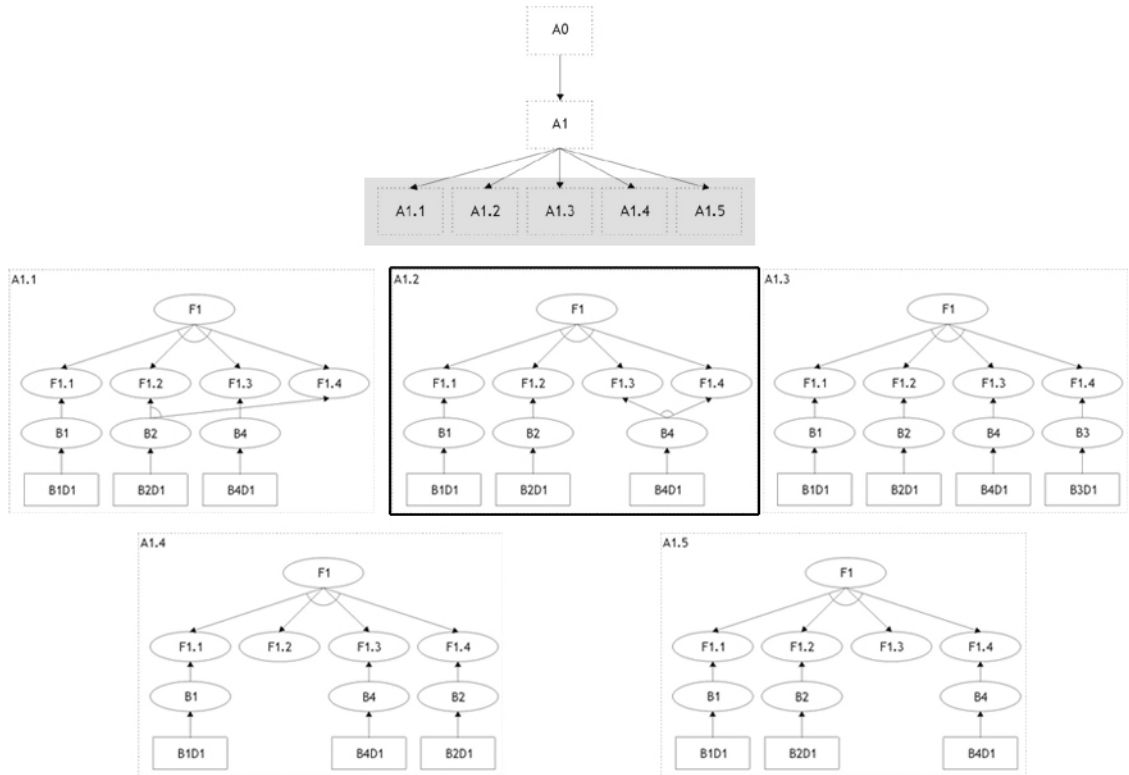


Fig. 5 - Alternative solved

4.3 Heuristic estimative for alternatives

Through the estimative calculations, the alternatives were ordered and the best one, the one with the lowest value was selected. In Table 2 below, the linguistic (Fuzzy) values, k (unexplored functions) and \hat{h} are shown as the scores calculated using the precepts of the heuristic state-space approach.

		Alt A1.2		B1D1			B2D1			B4D1		
i		w	r	\bar{r}	r	\bar{r}	r	\bar{r}	r	\bar{r}	k	\hat{h}
1	Manufacturability	fairly high	0.65	high	0.8	0.52	low	0.2	0.13	low	0.2	0.13
2	Assemblability	high	0.8	very high	0.925	0.74	fairly low	0.35	0.28	fairly low	0.35	0.28
3	Maintainability	very high	0.925	medium	0.5	0.4625	low	0.2	0.185	low	0.2	0.185
4	Reliability	high	0.8	medium	0.5	0.4	very low	0.075	0.06	very low	0.075	0.06
5	Efficiency	fairly low	0.35	high	0.8	0.28	very low	0.075	0.02625	very low	0.075	0.02625
							k	0			\hat{h}	11.82

Table 2 - Heuristic calculation

The selected alternative was a table whose functions and behaviour values are shown in Fig. 6.



Fig. 6 - Alternative generated

5 DISCUSSION

After the reasoning process, a more valuable alternative could be composed in the shortest time and process. This was possible due the iterative actions of exploration and combination, avoiding the “too fine” decomposition of function, and a combinatorial explosion [6]. The final product is a drawing table with few components, and these components represent a reuse of behaviours that could satisfy all the functional system.

Throughout the reasoning procedure some discussion arose on the following areas: constraints, behaviours, evaluation and algorithms.

5.1 Constraints

In Zhang’s research, limiting the search by constraints is difficult to reproduce due to the lack of information available. The author explains how it is used in a subjective way, which could not be applied to the current solution state-space. It was observed that this important delimiter is essential to define the set of solutions.

To achieve better results in a computable approach, two distinct kinds of delimiters should be used: qualitative or quantitative information. The first is based on a search of linguistic terms, separating types of behaviours by means of a database of words, or related words (e.g., “pressure”, when using hydraulic or pneumatic solutions). The second limits the variables by using numbers in order to further restrict the search space (e.g., Occupied space, between 90-110 cm).

5.2 Behaviours

With regard to behaviour, the heuristic state-space approach seeks information in a knowledge base. Nevertheless, the database presented by Zhang is generic and far from designers’ reality. In this case, solutions could be feasible but are computationally expensive. We think that, by restricting the search space, designers can guide solutions that fit a specific kind of problem (e.g. Mechanical and Electrical solutions). This can be done using groups of behaviours that participates in one or more domains.

5.3 Evaluation

Evaluation is an important phase of a solution. This can verify the real efficiency or feasibility of the results. Most products, before going on to a prototype or model stage, might have their functions solved in number and length components that could satisfy the requirements and constraints defined in the initial stages. The reasoning presented here does not make use of this structured process of restudy. This gap of information can provide problems of space used, the combination of components or resources, and unexpected use during the life-cycle. For this reason, we propose that an evaluation process or algorithm should be incorporated in a computable form to make this set of alternatives more valuable for the solution to the problem.

5.4 Algorithms and system

In spite of these assumptions, several modifications can be applied to the algorithm which would provide more valuable solution spaces in a relevant number of alternatives, and more innovative design solutions would be produced during the process of synthesis.

As an alternative to the algorithm presented in this paper in a situation of growing complexity, another algorithm called Recursive-Best-First-Search [7] can be used. In this case, the algorithm shows concern for memory consumption and maintains the last set of alternatives that could allow a new exploration to be carried out in the case of an unsatisfied decomposition or an unfeasible solution.

We intend to use the present research in a project involving a new software platform that could help designers throughout the synthesis process, that is, from the most abstract to the physical representation. Fig. 7 depicts the structure of this future system, which will be based on the 3 layer concept. The first layer will be responsible for maintaining the knowledge needed by the reasoning process. The second layer will refer to the application level which supplies functional data to the third one. This last one represents the availability of desktops and/or CAD systems. Thus, we expect to obtain an open platform to communicate with other tools.

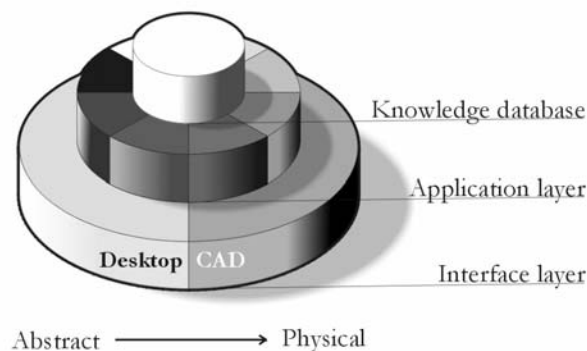


Fig. 7 – Proposed system model

6 CONCLUSIONS

We conclude that the algorithm under study here is efficient for problems involving few function requirements, as shown in the course of this article. It has been proved that its use in a computer platform can benefit designers in the creativity process, and it could also help generate a set of more valuable and optimised solutions.

By introducing the modifications presented in this paper and with a more detailed computational model, a system can be structured for implementation. This can make application of the algorithm feasible in a real situation involving a synthesis process. In the future we expect to be able to show an evolution of the architecture presented here in a coded system.

ACKNOWLEDGEMENTS

This research was supported by the Alban Programme, the European Union Programme of High Level Scholarships for Latin America, scholarship no. E04D030611BR, and by Bancaja project E-2004-22.

This research is part of projects DPI2002-04357-C03-01, supported by the *Ministerio de Ciencia y Tecnología* and co-funded by FEDER funds, and “Modelado y gestión del conocimientos para la ayuda al diseño de productos en las Pequeña y Mediana Empresa” supported by *Fundació Caixa Castelló Bancaixa-Universitat Jaume I*.

7 REFERENCES

- [1] W.Y. Zhang, S.B. Tor and G.A. Britton A Heuristic State-Space Approach to the Functional Design of Mechanical Systems, *The International Journal of Advanced Manufacturing Technology* 19 (2002) 235-244.
- [2] J. Hirtz, R. Stone, D. McAdams, S. Szykman and K. Wood A functional basis for engineering design: Reconciling and evolving previous efforts, *Research in Engineering Design* 13 (2002) 65 - 82.
- [3] R. Vidal, E. Mulet and E. Gómez-Senent Effectiveness of the Means of Expression in Creative Problem Solving in Design Groups, *Journal of Engineering Design* 15 (2004) 285-297.
- [4] Y. Liu, A. Chakrabarty and T. Bligh A computational framework for concept generation and exploration in mechanical design further developments of function, *Artificial Intelligence in Design '00* (2000) 499 - 519.
- [5] W.Y. Zhang, S.B. Tor and G.A. Britton A Prototype Knowledge-Based System for Conceptual Synthesis of the Design Process, *The International Journal of Advanced Manufacturing Technology* 17 (2001) 549-557.
- [6] W.Y. Zhang, S.B. Tor and G.A. Britton FuncDesigner a functional design software system, *Int. J. Adv. Manuf. Technol.* 22 (2003) 295-305.

- [7] S.J. Russell and P. Norvig Artificial intelligence: a modern approach, Upper Saddle River, N.J. Prentice Hall cop., 2003.