Proceedings of ICAD2013
The Seventh International Conference on Axiomatic Design
Worcester – June 27-28, 2013
ICAD-2013-03

VALUE-BASED AXIOMATIC DECOMPOSITION (PART I): THEORY AND DEVELOPMENT OF THE PROPOSED METHOD

Pedro Alexandre Marques
pamarques@isq.pt
UNIDEMI,
Department of Strategy and Special Projects
ISQ – Welding and Quality Institute
Av. Prof. Dr. Cavaco Silva, N.º 33, Edif. H
Taguspark-Oeiras, 2740-120 Porto Salvo, Portugal

Pedro Manuel Saraiva
pas@eq.uc.pt
Department of Chemical Engineering
Faculty of Science and Technology
University of Coimbra
Pólo II – Pinhal de Marrocos, 3030-290 Coimbra, Portugal

José Gomes Requeijo
jfgr@fct.unl.pt
UNIDEMI,
Department of Mechanical and Industrial Engineering
Faculty of Science and Technology
Nova University of Lisbon
Campus de Caparica, 2829-516 Caparica, Portugal

Francisco Frazão Guerreiro
fjguerreiro@isq.pt
Department of Strategy and Special Projects
ISQ – Welding and Quality Institute
Av. Prof. Dr. Cavaco Silva, N.º 33, Edif. H
Taguspark-Oeiras, 2740-120 Porto Salvo, Portugal

ABSTRACT

Decomposition is very useful in simplifying design problems, by breaking down a set of goals, constraints, requirements, behaviours and structures, into less complex and more manageable ways. In Axiomatic Design Theory, this top-down approach takes place by zigzagging back and forth between at least two adjacent design domains. Nevertheless, the decomposition activities pose some challenges, such as assuring the consistency of the design decisions made between levels, generating proper functional requirements (FRs) and constraints (Cs) at the lower levels of abstraction, defining adequate design parameters (DPs) and integrating them into physical and/or logical parts, in order to achieve the required functions and the desired life cycle properties for the system.

In this paper we propose a new decomposition method that integrates Axiomatic Design with FAST (Function Analysis System Technique). The use of FAST diagrams and Value Engineering principles, during the zigzag path, are combined with the concepts and guidelines from Axiomatic Design Theory. This systematic articulation increases the ability to define a sufficient number of FRs at each layer of the design hierarchy as well as the coherence between sub-FRs.

In part II of this paper, a practical example describing the applicability of the proposed decomposition approach is provided.

Keywords: design decomposition, consistency, Axiomatic Design, Functional Analysis System Technique (FAST).

1 INTRODUCTION

Decomposition can be described as an iterative process where the high-level required functions of a technical system being designed are broken down into subfunctions, and, at the same time, the corresponding top-level design solutions are detailed, embodied and integrated into specific physical and/or logical elements.

A wide variety of strategies are available for accomplishing design decomposition [Koopman Jr, 1995]. In Axiomatic Design Theory (ADT), decomposition is achieved according to a zigzagging procedure between the system functional requirements (FRs) and the developed design solutions (design parameters, DPs) to achieve those requirements. As this top-down zigzagging proceeds, the details of the technical system emerge and a clear design hierarchy of FR-DP pairs is obtained, until such system can be implemented.

The decisions that are made at higher levels affect the statement of the design definition at the lower levels of the hierarchy [El-Haik, 2005]. During the decomposition process, lower-level design decisions, in terms of sub-FRs, sub-DPs, and their relationships (indicated by the corresponding design matrices), need to be consistent with the highest-level FR-DP pairs that represent the design intent. A consistent decomposition is defined as one in which, at every layer of the design hierarchy, the lower level design decisions match those that were made at the higher level [Tate, 1999].

Maintaining the consistency of the decisions between all levels of the design hierarchy is not just a crucial but also a difficult task faced by design teams. One difficulty concerns the lack of effective methods that can be used to develop good hierarchical decompositions [Brown, 2011].

This paper proposes a decomposition method based on Axiomatic Design Theory that incorporates the functional analysis principles from the Value Engineering discipline, in particular by taking advantage of the "How-Why" logic among functions provided by the FAST technique. Our intent in developing this value-based decomposition method is to contribute to guide designers in dealing with some of the most difficult issues that arise during the design decomposition activities, especially in the following:

- To ensure that a minimum and sufficient set of FRs have been established at all levels of the design hierarchy.
- To allocate all potential sub-FRs to the proper level of the design hierarchy.

- To verify if the sub-FRs provide the functionality described by their parent FR-DP pair.
- To determine what sub-FRs are actually required to perform the parent FR.
- To identify which FR-DP pairs do not need to be further decomposed.

In addition, because the method establishes a functional classification for the FRs located at all levels of the hierarchy, design decisions that comply with axioms can also be made on a value analysis basis.

We start by reviewing the state of the art regarding design decomposition principles and methods, in particular within the context of Axiomatic Design Theory and of the Functional Analysis System Technique (FAST). Then, in section 3, we present and discuss the proposed value-based decomposition method.

2 STATE OF THE ART

2.1 DESIGN DECOMPOSITION

The process of creating a design architecture often follows a process of decomposition, in which a top-level concept of the system's required functions is broken down into sub-functions, and at the same time the most abstract version of its physical form is broken down into subsystems capable of performing the subfunctions [Crawley *et al.*, 2004]. From this definition, and according to Ullman [2002], decomposition can be viewed from two perspectives:

- As the deployment and refinement of the high-level functions performed by the technical system. This is called functional decomposition.
- As the break-down of the means, or design solutions, for providing the functions. This is often called physical decomposition.

Every function that must be done by the system needs to be identified and defined in terms of allocated functional performance, and other limiting requirements [INCOSE, 2004]. This means that for each function that is partitioned into subfunctions, the requirements allocated to that function need to be decomposed with it.

In addition to the system's functions, their corresponding requirements and the defined conceptual design solutions, it is important to ensure the decomposition of other design goals [Koopman Jr, 1995], such as critical performance targets, aesthetics, limits in weight, and desired life cycle properties, among others. These goals are often known as design constraints.

Despite being widely employed in practice, there are several approaches used for performing design decomposition. Yu *et al.* [1998] and Mullens *et al.* [2005] review a wide set of decomposition techniques. Yu *et al.* [1998] propose a taxonomy structure to classify the different design decomposition approaches (Figure 1). This paper focuses on the hierarchical decomposition methods.

Many decomposition models, such as the ones proposed by Pahl and Beitz [1996], Ullman [2002] and Ulrich and Eppinger [2004], first make a full functional decomposition and only when all subfunctions are completely described does the search for design concepts/solutions initiate.

According to Meijer *et al.* [2003] and Gonçalves-Coelho *et al.* [2005], the functional decomposition should be done attending to the design decisions made in the physical domain. The zigzag decomposition adopted by Axiomatic Design Theory [Suh, 1990] and the decomposition reasoning used in the Critical Parameter Management (CPM) model [Creveling *et al.*, 2003] are two approaches that take this into account, meaning that both functional and physical decompositions occur in parallel.

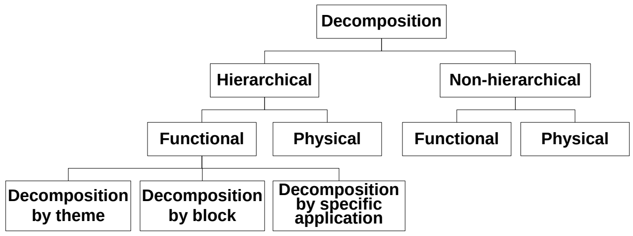

Figure 1. Taxonomy of decomposition methods (adapted from: Yu *et al.* [1998]).

The CPM model, proposed by Creveling *et al.* [2003], is based on the Systems Engineering discipline and it is often employed during a product design or technology development project. In this model, the House of Quality from the Quality Function Deployment (QFD) is used to capture, relate and flow-down all critical requirements and functions. At each level of the hierarchy, and before proceeding to the next lower level, design concepts/solutions are developed in order to perform the intended functions and satisfy the corresponding requirements in a capable way.

2.2 DECOMPOSITION IN AXIOMATIC DESIGN

According to Suh [1990], the world of design consists of four domains: 1) Customer domain; 2) Functional domain; 3) Physical domain; 4) Process domain. Associated with each domain are the design elements it contains [Tate, 1999]. In addition to these elements, a set of constraints (Cs) imposing limits or bounds to the design task can also exist.

Apart from the customer domain wherein the decomposition process is usually not considered, the remaining domains may have several levels of abstraction that jointly describe the technical system architecture [Marques *et al.*, 2009]. As depicted in Figure 2, the decomposition process in Axiomatic Design is achieved by zigzagging back and forth between at least two adjacent design domains, depending on the scope of the design process [Gonçalves-Coelho *et al.*, 2005]. By use of this zigzagging method, hierarchies for FRs, DPs, and PVs are created in each design domain [Suh, 2005]. In some designs the process domain will be fully developed so that the PVs relate to the DPs like the DPs relate to the FRs [Brown, 2006]. The lowest levels in each branch of the hierarchy are often called "leaf-levels". Like FRs and DPs, constraints can be refined and clarified as decomposition progresses [Hintersteiner, 1999].

The zigzagging decomposition process is explained in detail in Suh [1990; 2001]. Some researchers proposed some advances to this traditional decomposition process. Authors like Guenov and Barker [2004], Tang *et al.* [2009] and Hong and Park [2009] developed enhanced decomposition methods by integrating Axiomatic Design with the Design Structure

Matrix (DSM), in order to capture the interactions amongst the DPs and to facilitate the design decisions in the physical domain. Mullens *et al.* [2005] present an axiomatic decomposition method that combines Alexander's network partitioning formulation of the design problem with the Independence Axiom, and uses a cross-domain approach in a House of Quality context to estimate the interactions among the functional requirements. Kim and Cochran [2000] suggest the use of the Su-Field model from TRIZ to complement the decomposition process of Axiomatic Design.

In his PhD thesis, Tate [1999] developed a roadmap with the design activities that should performed during the decomposition process and their sequence. A set of useful guidelines and tools to assist designers in their decisions, in order to maintain the consistency of the decomposition, are also described in Tate's research work. Hintersteiner and Friedman [1999] and Gumus [2005] provide standard templates for supporting and documenting, in a systematic and consistent manner, the design decisions made at every level of the design hierarchy. The coherent construction of a system's architecture also relies on a proper classification of the functions, constraints, and design parameters. With this in mind, Tate [1999] proposes a classification for functions and constraints, while Gumus *et al.* [2008] define five types of design parameters, depending on their relative position in the design hierarchy.

2.3 Decomposition Using the FAST Approach

The Function Analysis System Technique (FAST) was proposed in the 1960s by Charles W. Bytheway as an extension of the Value Engineering approach. An important contribution of FAST is its synergistic way of developing, decomposing, and understanding the functions of any product, process, service, or organization [Wixson, 1999]. It is a useful method for identifying and classifying the functional relationships during a design effort.

By making use of the intuitive "How-Why" logic, FAST is a prime tool for functional mapping and analysis, enabling designers to relate functions located at different levels of detail. When questioning "how" a given function is performed, new function(s) is(are) brought into existence, while when asking "why" a certain function exists, it is possible to identify the function that caused that particular function to come into existence [Bytheway, 2007]. The example of Figure 2 illustrates the reasoning behind the "How-Why" logic.

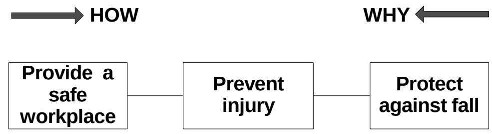

Figure 2. Example of the "How-Why" logic.

When repeated, this procedure allows the construction of a FAST diagram, whose classical model is depicted in Figure 3. Although there are different types and versions of FAST diagrams [Dallas, 2006], the "How-Why" logic is at the heart of them all. The main steps to construct a FAST diagram are the following:

1) Determine the scope of the conceptual process, which includes the definition of the technical system to be designed.
2) Identify the basic function(s) of the technical system. A basic function describes a fundamental task that must be performed by the system, thus representing the required reason for its existence.
3) Decompose the basic function(s) by applying the logical questions: How is the function accomplished? Why is the function performed?

All the functions on the right side of the basic function(s) describe the "concept" (i.e. design solutions) chosen to perform that basic function(s) [Yang, 2005]. The "objectives or specifications", which correspond to quantitative critical performance requirements that need to be met to satisfy the highest-order function, can also be indicated in the diagram. The FAST diagram also includes the logic operators "AND" and "OR": the first means that two or more functions need to be performed simultaneously, while the second signifies that two or more alternative dependent functions are available.

Support functions and activities are placed above and below the primary path, respectively. A "support function", also known as independent function, does not comply with with the "How-Why" logic, but it supplements the basic function(s) placed on the same level of abstraction. An "activity" is the method selected to perform a function. The FAST method is explained in detail by Bytheway [2007].

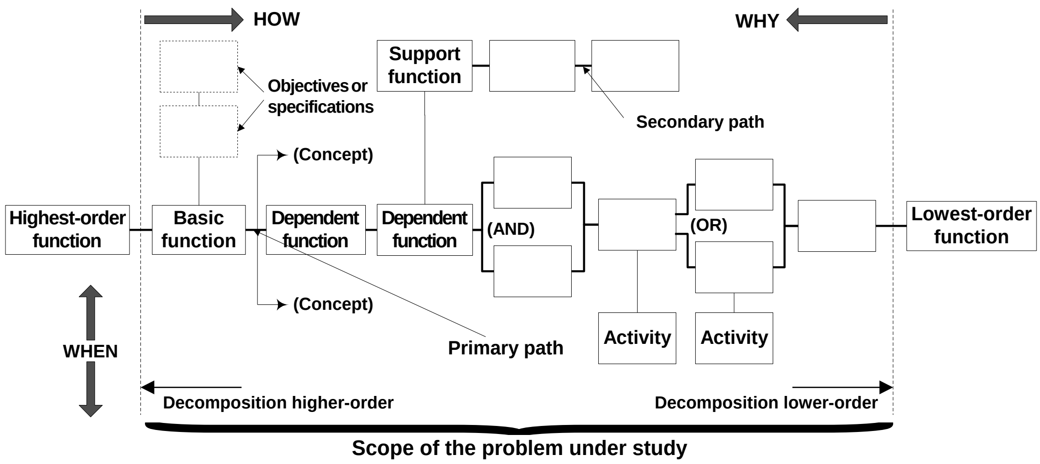

Figure 3. Classical FAST diagram (adapted from: Yang [2005]).

3 VALUE-BASED DECOMPOSITION METHOD

The proposed value-based axiomatic decomposition method is depicted in Figure 4. The activities where FAST plays an important role are identified. The recursive nature of the decomposition process is clear in the method, since the same set and sequence of activities are performed, layer-by-layer, until the architecture of the designed system is completed.

The proposed method is intended to address the following difficulties that often arise at the beginning or during the traditional zigzagging decomposition process:

- Systematize the development of a necessary and sufficient set of functional requirements at every level of the design hierarchy.
- Distinguish the FR-DP pairs that require further decomposition from those that have reached the leaf-level.
- Properly define sub-FRs, by ensuring they provide the functionality described by their corresponding FR-DP pair.
- Ensure that all sub-FRs are correctly allocated to the different levels of the design hierarchy.

In addition, the purpose of the value-based decomposition method is to enable designers to make use of the principles from Value Engineering, while applying Axiomatic Design.

In the next sections the activities included in the value-based decomposition model will be discussed in detail.

3.1 DEFINITION OF THE DESIGN ELEMENTS AT THE TOP-LEVEL OF THE HIERARCHY

The pre-decomposition activities are very important since they have a great impact on the design decision to be made during the decomposition process. Figure 5 exhibits the suggested procedure to establish the top-level Cs, the initial set of FRs and DPs, as well as their corresponding design matrix.

The first challenge is to define the initial set of functional requirements and the top-level constraints. Corollary 2 of Axiomatic Design states that the number of FRs and Cs should be minimized; nevertheless, they should be sufficient to fully represent the customer domain. In addition, it is important to clearly distinguish the FRs from the Cs [Brown, 2006].

The procedure considers that both top-level Cs and the initial set of FRs derive from the following elements of the customer domain: (1) customer needs (CNs); (2) design requirements (DRs). The CNs represent the "voice of the customer" and are translated into specific DRs using the House of Quality framework. The description of each design requirement is accompanied by its corresponding operational definition, which is clear, unambiguous, and observable standard of acceptance. The House of Quality is also used to identify the most important DRs, which is an important step towards the determination of the critical performance specifications type of constraints. Later on, during the decomposition process, all the critical performance specifications are to be refined into sub-FRs, as recommended by Tate [1999].

The procedure to define the initial set of FRs relies on the generic template for listing FRs of Hintersteiner and Friedman [1999] and on the functional classification from Value Engineering. It is recommended that the initial set of FRs, in order to be minimum but sufficient in number, should be associated with the basic functions of the technical system. The basic functions can be regarded as the process functions referred by Hintersteiner and Friedman [1999]. As in a FAST diagram, the FRs that are associated with the basic functions should be located at the top-level of the design hierarchy.

To minimize the number of FRs, the definition of FRs associated with secondary functions that complement the basic functions should be avoided, except when a command and control function and/or a support and integration function need(s) to be established. Hintersteiner [1999] and Tate [1999] discuss both the command and control and the support and integration functions.

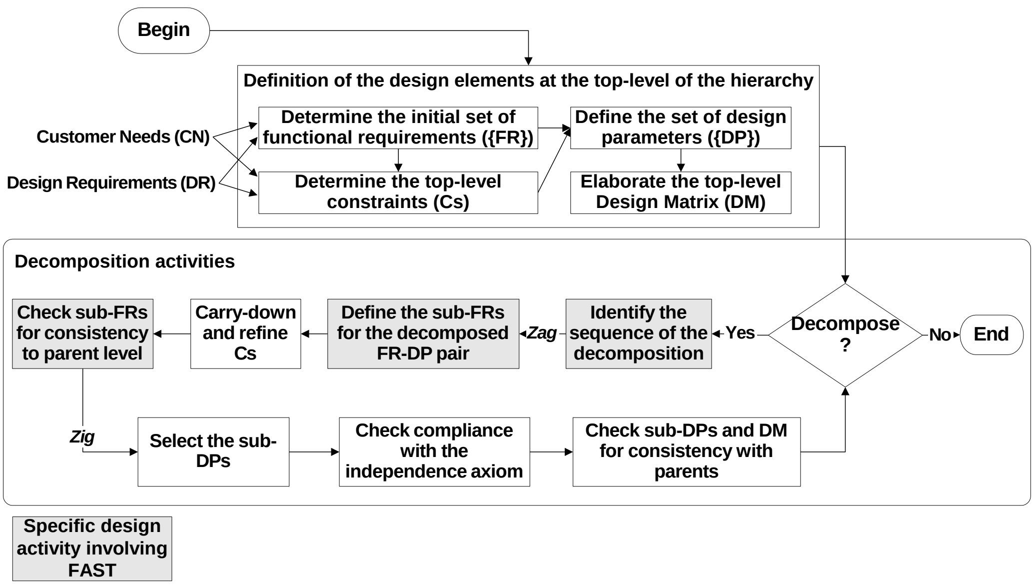

Figure 4. Value-based axiomatic decomposition method.

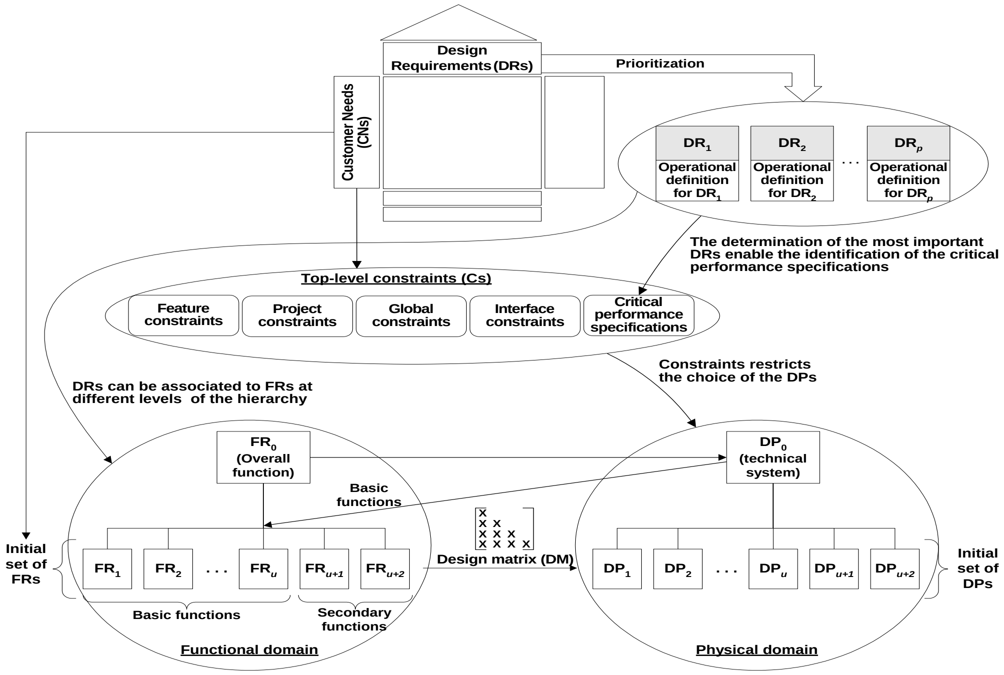

Figure 5. Procedure to define the initial set of FRs and DPs, and the top-level constraints (Cs).

The five categories of constraints indicated in Figure 5 were proposed by Tate [1999] and are herein adopted. The most important DRs usually give origin to the critical performance specifications type of constraints.

It is important to check if the specified initial set of FRs and top-level Cs actually are representative of the CNs. The template suggested by Gumus [2005], for relating CNs with FRs and Cs, may be useful for this purpose.

The initial set of DPs represents the design intent. These DPs are chosen with the aim of ensuring that their respective FRs can be independently achieved, by at the same time satisfying the bounds and restrictions imposed by the constraints on the possible design solutions. The top-level design matrix (DM) relates the initial sets of FRs and DPs, and its analysis enable to conclude if the intended design conceptrepresents a decoupled, uncoupled or coupled design.

3.2 Identification of the Decomposition Sequence

This step involves the identification of the: (1) FR-DP pair(s) requiring further decomposition; (2) most appropriate sequence in which that decomposition should be conducted.

If any of the initial FR-DP pairs needs to be further detailed, the decomposition process begins. To identify which of the initial FR-DP pair(s) require further decomposition, the following guideline is formulated, based on the FAST model:

- The designers have the option to consider an initial FR-DP pair as a leaf when the function associated with the FR is classified as a secondary function according to the Value Engineering principles, since secondary functions do not belong to the primary path.

As the decomposition process proceeds, designers still need to identify, at each level of the hierarchy, which FR-DP pairs have reached the leaf-level and those that should be further decomposed. To help designers in this task, the previous guideline can be generalized:

- At a certain level of the design hierarchy. designers have the option to consider a certain FR-DP pair as a leaf when the function associated with that FR is classified as a support function, since it does not comply with the "How-Why" logic with the corresponding parent function (i.e. a support function does not make part of the primary path).

When, at a certain level of the design hierarchy, two or more FR-DP pairs have to be decomposed, one needs to determine the most suitable sequence to be followed. For the case of a decoupled design, the value-based decomposition method recommends that the following guidelines, provided by Tate [1999], should be employed:

- To identify the next FR-DP pair to decompose, at each level, define sub-FRs in the order described by the design matrices.
- To identify the next FR-DP pair to decompose, there is no penalty in terms of time/iteration for decomposing one branch of the design hierarchy more deeply than another, provided that the order follows that given in the design matrices.

3.3 Definition of Sub-FRs

For a certain FR-DP pair to be decomposed, a sufficient and necessary set of sub-FRs has to be specified. To achieve this goal, all potential sources of sub-FRs should be considered [Tate, 1999], in particular the following: parent DP; parent FR; parent-level Cs; parent DM; set of CNs. These potential sources are indicated by order of importance.

The FAST model and the Value Engineering principles for functional analysis can aid the development of sub-FRs with origin on the parent DP and parent FR, by following the reasoning depicted in Figure 6. Consider that a certain FR_i-DP_i pair needs to be decomposed. If FR_i is part of the initial set of FRs, then it is associated with a basic function; if it is not part of the initial set, then FR_i is associated with a dependent function.

The application of Value Engineering principles to the analysis of the parent DP (DP_i) helps designers to determine its basic functions, enabling them to identify the sub-FRs that describe DP_i. By its turn, the development of sub-FRs based on the knowledge of the parent FR (FR_i) can be done using the "How-Why" logic of the FAST model, particularly by answering the following question: "how is FR_i performed?"

The development of sub-FRs based on the knowledge of the parent Cs and DM, as well as on the set of CNs is discussed in detail by Tate [1999], who provides a set of useful guidelines on the subject.

The application of Value Engineering principles to the analysis of the parent DP (DP_i) helps designers to determine its basic functions, enabling them to identify the sub-FRs that describe DP_i. By its turn, the development of sub-FRs based on the knowledge of the parent FR (FR_i) can be done using the "How-Why" logic of the FAST model, particularly by answering the following question: "how is FR_i performed?"

The development of sub-FRs based on the knowledge of the parent Cs and DM, as well as on the set of CNs is discussed in detail by Tate [1999], who provides a set of useful guidelines on the subject.

All the sub-FRs that actually answer "how the parent FR is performed?", including those that describe the parent DP, are classified as dependent functions. The sub-FRs that do not answer this question are classified as support functions. The functional classification of the sub-FRs is important for designers to detect potential FRs at the leaf-level, as described in section 3.2, but all the sub-FRs have the same importance as required in Axiomatic Design Theory.

During this step, for large or flexible system design [Suh, 1995], the employment of the logic operator "OR" adopted in the FAST technique can be useful to define different or alternative sets of FRs that the system may need to perform during its life time.

3.4 Carrying-Down and Refining Cs

This activity is entirely performed attending to the guidelines provided by Tate [1999] about carrying down and refining Cs. Critical performance specifications and interface constraints are refined into sub-FRs, while global and project constraints are refined but remain as constraints at the lower levels of the hierarchy.

3.5 Checking Sub-FRs for Consistency

The good practices for generating sub-FRs, described in section 3.3, provide the conditions needed for consistency. The sub-FRs are consistent if they are descriptive (i.e. they describe consistency with respect to the parent DP), sufficient and necessary (i.e. they describe consistency with respect to the parent FR). Again, the "How-Why" logic of the FAST model can be used to check the consistency between the sub-FRs associated with a dependent function and the parent FR.

3.6 Selection of Sub-DPs and Checking Compliance with the Independence Axiom

Once the set of sub-FRs are established, it is time to find the corresponding sub-DPs located in the physical domain. It is important to consider and assess alternative candidates for each of the sub-DPs, before selecting the final set of sub-DPs. Value Engineering principles, in terms of cost-benefit analysis, can be employed to evaluate alternative sets of sub-DPs.

The potential sets of sub-DPs, to be viable, should ensure the functional independency of their corresponding sub-FRs, while satisfying the imposing constraints. When possible, the Information Axiom should be applied to select the best set of sub-DPs complying with the Independence Axiom.

3.7 Checking Sub-DPs and the DM for Consistency with Parents

In this step, the consistency of the design decisions in the selected sub-DPs and in the elements of the design matrix, that relates sub-FRs and sub-DPs, need to be confirmed. To check the consistency of the sub-DPs, it is necessary to verify if they:

- Provide enough capability in satisfying the parent FR.
- Satisfy the Cs applied to the parent DP.
- Have been integrated into physical and/or logical element(s) in a way that does not violate the functional independence indicated in the parent level.

The consistency of the design matrix elements of all lower level design decisions can be checked by constructing the full design matrix [Suh, 2005]. In addition to the construction and analysis of the full design matrix, the guidelines provided

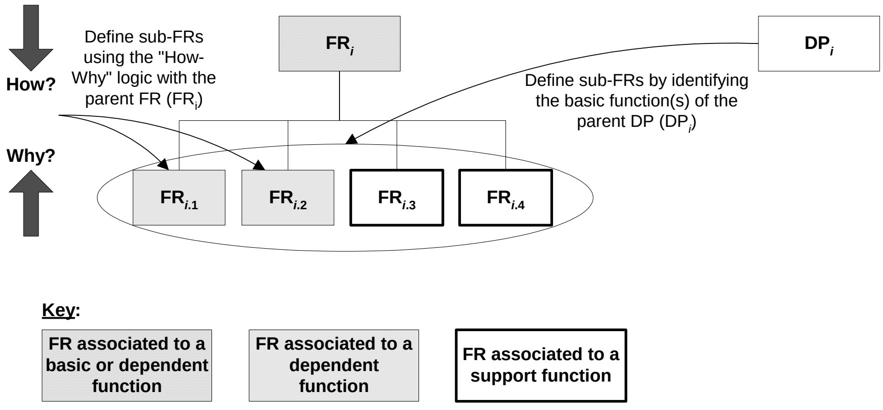

Figure 6. The role of the FAST model in the definition of sub-FRs.

by Tate [1999] enable designers to check consistency of the DM elements in every level of the hierarchy.

3.8 Finalising the Decomposition Process

The activities of the value-based decomposition method, described in greater detail from section 3.2 to section 3.7, are performed until the technical system is detailed enough to be fully implemented. At the end of the decomposition process the system architecture is thus completed.

Figure 7 illustrates a typical hierarchical structure of the design that is obtained after the value-based decomposition method is employed. It presents the case of a technical system that performs "*u*" basic functions and one secondary function. It means that at the highest-level of the design hierarchy there are an initial set of "*u+1*" FRs and an equal number of corresponding sub-DPs. As depicted, only the basic functions and their dependent functions were decomposed.

The "How-Why" logic and the functional classification provided by the FAST model contribute to systematise and enhance the consistency of the decomposition process.

4 CONCLUSIONS

A decomposition method integrating Axiomatic Design Theory with Value Engineering, in particular with the Function Analysis System Technique (FAST) approach, was proposed and described in the first part of this paper.

This value-based axiomatic decomposition method was developed with the aim of helping designers, with a logic framework and a set of new guidelines, to perform the decomposition activities in a way that the design decisions are coherently made in all the layers of the design hierarchy. More specifically, the main contributions of the proposed decomposition method, to the advance of this important subject, are the following:

- Increase the coherence of the functional decomposition by adding the functional mapping and the "How-Why" intuitive logic, both provided by the FAST model, to the traditional decomposition process followed in Axiomatic Design.
- Provide a systematic procedure to define a sufficient and necessary set of FRs in all levels of the design hierarchy, ensuring, at the same time, that the sub-FRs are allocate to the proper level of detail.
- Enhance the ability to determine which FR-DP pairs, along the design hierarchy, should be considered as being at the leaf-level, and those pairs that can be further decomposed.

In the second part of this paper, a practical application of the proposed method, developed at a Portuguese company, will be presented.

In future studies, it is our objective to make use of this value-based decomposition method in the context of the Design for Six Sigma (DFSS) methodology.

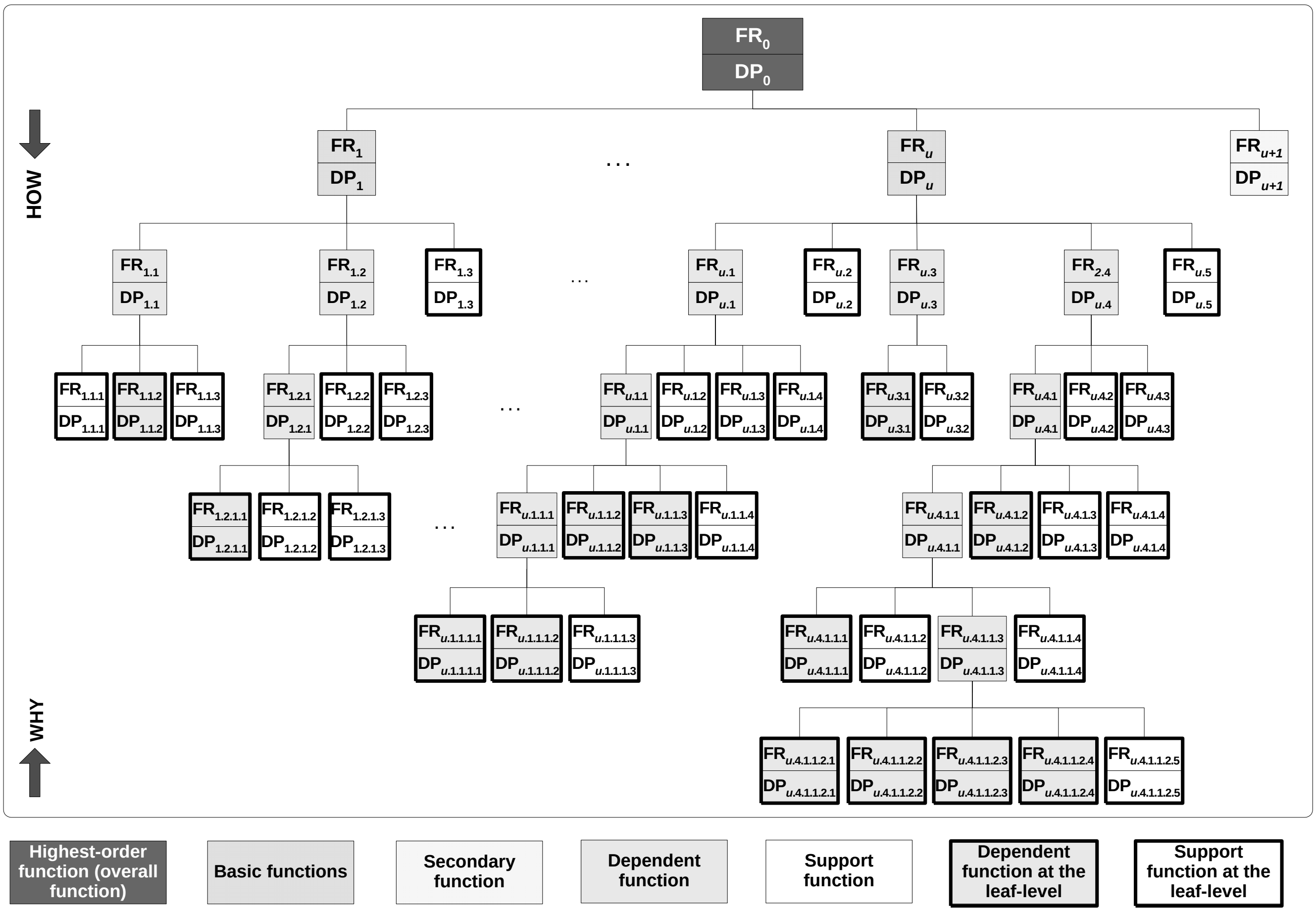

Figure 7. Typical design hierarchy structure that results from the use the value-based decomposition method.

5 REFERENCES

[1] Brown C.A., *Elements of Axiomatic Design: a Simple and Practical Approach in Engineering Design*, Draft: 29 March, New York: Cazenovia, 2006.

[2] Brown C.A.," Decomposition and Prioritization in Engineering Design", *Proceedings of the Sixth International Conference on Axiomatic Design, ICAD2011*, Daejeon, South Korea, March 30-31, 2011.

[3] Bytheway C.W., *FAST Creativity & Innovation*, Florida: J. Ross Publishing, 2007. ISBN 1-932159-66-0

[4] Crawley E., De Weck O., Eppinger S., Magee C., Moses J., Seering W., Schindall, J., Wallace D., Whitney D., "The Influence of Architecture in Engineering Systems", *Engineering Systems Monograph*, The ESD Architecture Committee, Massachusetts Institute of Technology (MIT), Cambridge, MA, USA, 2004.

[5] Creveling C.M., Slutsky, J.L., Antis, D., *Design for Six Sigma in Technology and Product Development*, New Jersey: Prentice-Hall, 2003. ISBN 0-13-009233-1

[6] Dallas, M.F., *Value & Risk Management: A Guide to Best Practice*, Montana: Blackwell Publishing, 2006. ISBN 1-4051-2069-X

[7] El-Haik B.S., *Axiomatic Quality*, New Jersey: John Wiley & Sons, 2005. ISBN 0-471-68273-X

[8] Gonçalves-Coelho A.M., Mourão A.J.F., Pereira Z.L., "Improving the use of QFD with Axiomatic Design", *Concurrent Engineering: Research and Applications*, Vol. 13, No. 3, pp. 233-239, 2005.

[9] Guenov M.D., Barker, S.G., "Application of Axiomatic Design and Design Structure Matrix to the Decomposition of Engineering Systems", *Systems Engineering*, Vol. 8, No. 1, pp. 29-40, 2004.

[10] Gumus B., "Axiomatic Product Development Lifecycle", *Doctoral of Philosophy Thesis*, Texas Tech University, Lubbock, TX, USA, 2005.

[11] Gumus B., Ertas A., Tate D., Cicek I., "The Transdisciplinary Product Development Lifecycle Model", *Journal of Engineering Design*, Vol. 19, No. 3, pp. 185-200, 2008.

[12] Hintersteiner J.D. "A Fractal Representation for Systems", *Proceedings of the 1999 International CIRP Design Seminar*, Enschede, Holland, March 24-26, 1999.

[13] Hintersteiner J.D., Friedman G. "System Architecture Template", *MIT White Paper*, Axiomatic Design Group, 1999.

[14] Hong E, Park, G., "Decomposition Process of Engineering Systems Using Axiomatic Design and Design Structure Matrix", *Proceedings of the Fifth International Conference on Axiomatic Design, ICAD2009*, Campus de Caparica, March 25-27, 2009.

[15] INCOSE, *Systems Engineering Handbook – A "What To" Guide for All SE Practitioners*, Seattle: International Council on Systems Engineering, 2004.

[16] Kim Y.S., Cochran D.S., "Reviewing TRIZ from the Perspective of Axiomatic Design", *Journal of Engineering Design*, Vol. 11, No.1, pp. 79-94, 2000.

[17] Koopman Jr, P.J. "A Taxonomy of Decomposition Strategies Based On Structures, Behaviours, and Goals", *Proceedings of ASME 1995 Design Engineering Technical Conferences*, Vol. 83, pp. 611-618, 1995.

[18] Lindholm D., Tate D., Harutunian V., "Consequences of Design Decisions in Axiomatic Design", *Transactions of the Society for Design and Process Science*, 3(4), pp. 1-12, 1999.

[19] Marques P.A., Alink T, Saraiva P.M., Albers A., Requeijo J.G., Guerreiro FF, "Integration of the Contact and Channel Model with Axiomatic Design", *Proceedings of the Fifth International Conference on Axiomatic Design, ICAD2009*, Campus de Caparica, March 25-27, 2009.

[20] Meijer B.R, Tomiyama T., Van der Holst B.H.A., Van der Warff K., "Knowledge Structuring for Function Design", *CIRP Annals – Manufacturing Technology*, Vol. 52, No. 1, pp. 89-92, 2003.

[21] Miles L.D., *Techniques of Value Analysis and Engineering*, 3rd Edition, New York: Lawrence D. Miles Value Foundation, 1989. ISBN 978-0070419261

[22] Mullens M.A., Arif M., Armacost R.L., Gawlik TA, Hoekstra R.L., "Axiomatic Based Decomposition for Conceptual Product Design", *Production and Operations Management*, Vol. 14, No. 3, pp. 286-300., 2005.

[23] Pahl G., Beitz W., *Engineering Design: A Systematic Approach*, 2nd Edition, London: Springer-Verlag, 1996. ISBN 0-387-50442-7

[24] Suh N.P., *The Principles of Design*, New York: Oxford University Press, 1990. ISBN 0-19-504345-6

[25] Suh N.P., *Axiomatic Design: Advances and Applications*, New York: Oxford University Press, 2001. ISBN 0-19-513466-4

[26] Suh N.P., "Design and Operation of Large Systems" *Journal of Manufacturing Systems,* 14(3), pp. 203-213, 1995.

[27] Suh N.P., *Complexity: Theory and Applications*, New York: Oxford University Press, 2005. ISBN 0-19-517876-0

[28] Tang D., Zhu R., Dai S., Zhang G., "Enhancing Axiomatic Design with Design Structure Matrix", *Concurrent Engineering*, Vol. 17, No. 2, pp. 129-137, 2009.

[29] Tate D., "A Roadmap for Decomposition: Activities, Theories, and Tools for System Design", *Doctoral Thesis*, Department of Mechanical Engineering, Massachusetts Institute of Technology (MIT), Cambridge, MA, 1999.

[30] Ullman D.G., *The Mechanical Design Process*, 3rd Edition, New York: McGraw-Hill, 2002. ISBN 0-07-237338-5

[31] Ulrich K.T, Eppinger, S.D., *Product Design and Development*, 3rd Edition, New York: McGraw-Hill, 2004. ISBN 0-07-247142-5

[32] Wixson, J.R., "Function Analysis and Decomposition Using Function Analysis Systems Technique", *International Council on Systems Engineering Annual Conference, INCOSE'99*, Brighton, England, June 6-10, 1999.

[33] Yang K., *Design for Six Sigma for Service*, New York: McGraw-Hill, 2005. ISBN 0-07-144555-2.

[34] Yu H., Shu L.H., Venter R., "An Enhanced Axiomatic Design Process", *Proceedings of the 1998 ASME Design Engineering Technical Conferences, DETC'98*, Atlanta, Georgia September 13-16, 1998.