Using VR for Complex Product Design

Loukas Rentzos, Charalampos Vourtsis,
Dimitris Mavrikios, and George Chryssolouris

Laboratory for Manufacturing Systems and Automation,
Dept. of Mechanical Engineering and Aeronautics, University of Patras, Greece
xrisol@lms.mech.upatras.gr

Abstract. Virtual reality is a key technology for the designing of products through complex human-product interactions. This paper deals with the development of a product design method for complex human-product interactions, using the virtual reality (VR) technology. This VR method uses the graph theory in order for the complexity of the designed product to be measured on the basis of human task analysis. The latter is for the purpose of recording and analyzing the human-product interactions within an immersive simulation session. The proposed method undergoes tests in a realistic aerospace case.

Keywords: Product Design, Product Complexity, Immersive Environment, Virtual Prototyping.

1 Introduction

Modern design versions of traditional products (e.g. aircrafts) have become more and more complex due to the constantly growing demand for regulations and standards, imposed by the globalized nature of their associated markets. Most of the products that are being manufactured today have some kind of interaction with humans. It can be considered that the human is the end-user of the product (e.g. airline passenger), or the operator of the product (e.g. aircraft pilot), or the worker involved in its manufacturing (e.g. human worker in aircraft assembly line), or the technician/engineer concerned with the maintenance of the product (e.g. aircraft maintenance tasks). All different aspects of human-product interaction define a vast number of factors that need to be taken into account during product design. Furthermore, some products go through a heavy "automatization" (e.g. commercial aircraft) that further increases the complexity of human-product interaction. Virtual reality is a key enabling technology for designing products with complex human-product interactions. The study presented in this paper aims at developing a product design method for complex human-product interactions through the virtual reality (VR) technology. The latter enables the simulation of human factors during product design, in their full context, with high flexibility and reusability [1], [2]. Furthermore, it provides high levels of flexibility and cost efficiency during the early phases of product design. Since VR enables the simulation of human tasks in full context, it can provide the ideal platform for the measuring of product complexity by analyzing the human tasks, during the usage of products.

R. Shumaker and S. Lackey (Eds.): VAMR 2014, Part II, LNCS 8526, pp. 455–464, 2014.

Product design with the CAD systems available, offers a perception of a 3D model's parameters such as shape, color, kinematics etc., nevertheless, the need for real time human interaction is not satisfied. The VR technology allows engineers/designers to interact, to a great extent, with the 3D model in an immersive environment and enables the testing, experimentation and evaluation of the product in full context. This technology can be considered as an extension to the conventional CAD tools by means of further extending the human integration with the product in its environment. Therefore, the VR technology offers a great added value for use in the early design phases of complex products by means of testing and simulating them. However, a question that arises is whether or not besides testing and simulating a product in a virtual environment, VR can also measure its complexity and provide a useable metric that could support the engineers and designers to improve their design. According to [3], a good design is the one that satisfies all functional requirements with a minimum number of components and relations. In addition, a simple design is preferable to a complex design [4]. Therefore, there is a need that this complexity be minimized during design. A collection of different views has been made to increase the value of perception over the definition of complexity. In product design, from an assembly aspect, the predominant definition of complexity is the interconnection of parts. [5]. The information aspect of complexity suggests that complexity is a measure of the minimum amount of information required to describe the given representation [3, 6]. Complexity could also be stated as a measure of entropy randomness in a design [7] and as a measure of the number of basic operations, required for the solution of a problem [8]. A more generic perspective is that complexity can be defined as an intersection between elements and attributes that complicates the object in general [7]. In [9], a complex system is defined as that comprising a large number of parts interrelating in a non-simple manner. Approaches to reducing complexity can also be found in the literature out of methodologies for the reduction of assembly complexity [5] to approaches leading to product simplification [10].

Complexity measures could be categorized on the basis of what is evaluated, the basis of the measure, the method, as well as the type of measure. Considering the existing complexity measures, the most common types are size, coupling and solvability complexity [11]. Size complexity measures focus on several product elements, including the number of design variables, functional requirements, constraints applied and subassemblies. Size complexity measures are usually developed based on the information that primarily derives from entropic measures of a representation. . The complexity of a design could be measured as the cluster of reduced entropy at each step of the design process, thus a more complex design requires more reduction in entropy. Coupling complexity measures refer to the strength of interconnection among the elements of a design product, problem or process. The representation of the elements measured, needs to be in graph format. Coupling complexity, in most cases, is measured by the decomposability of every graph's representation. Finally, solvability complexity measures indicate whether the product design may be predicted to satisfy the design problem. It is also referred as the difficulty of the design process to result in the final design. Measuring the difficulty, could be stated as the time required for the designing of a product or the number of steps to be followed for its completion.

In [11], a comparison of the complexity measures is presented based on the existing literature. The main variables for this comparison are the focus of complexity evaluation (i.e. design process, design problem, design product), the basis (computational/algorithmic analysis, information based, and traditional design), the focus of measurement (size, coupling and solvability), the interpretation (objective, subjective) and finally, whether an absolute or relative metric has been used.

This paper presents a VR method, developed for complex product design that records and analyzes the human-product interactions within an immersive simulation session and evaluates the product's coupling complexity. The VR framework is based on graph theory methods for the measuring of a product's coupling complexity. The latter is generated automatically, whilst the function structure and bipartite graphs of human-product interactions are analyzed.

2 Complexity Evaluation Method

The coupling complexity measure of a product could be defined as the measurement of interconnections between a product's variables at any level. The coupling measure chosen to be used has been thoroughly described by [12]. The process requires that the design be represented in a graph format, where the tasks are depicted with nodes and are connected with simple lines in order to form dependencies. The method tries to decompose the product's graph representation and thus, the working principle is that any relationships be removed until the graph could be separated into other graph formats in order for the coupling in each of them to be measured. The algorithm (see Fig. 1) aims to decompose the graph to the utmost extent. The graph is being decomposed every time by questioning its connectivity feature. The algorithm for the graph analysis begins with the removal of unary relations and continues with the recording of the remaining variables. After this point, the algorithm keeps applying to all the sub graphs produced from the initial one. The arithmetic record is being kept so as for the interconnectedness of the graph to be measured and finally, conclude to an arithmetic value of the product's coupling complexity.

The current study aims at applying this algorithm inside a virtual environment. The calculation is made on the basis of graphs generated by the interactions performed with the product inside the virtual environment by the human user. The representation method chosen in this case is the function structure graph that seems to be most appropriate for the engineering systems. The function structure graph is a block based diagram, used for the analysis of engineered systems by representing the relations among the different functions of a product (see Fig. 2). The relations to be created for the representation of a problem are described by three basic types, namely Function-Function (F-F), Input-Function (I-F), Function-Output (F-O). These are also referred to as primitive relations (operators) and the building blocks of the graph primitive modules (operands). Complexity is usually correlated with the type of representation. The coupling complexity in a function structure graph is visualized by the interconnectedness of functions in a product.

```
Eliminate unary relations
Level = 1, Total = 0
FOR each (sub)-graph:
    Size = 1
    FOR all combinations of relations in Size
        Remove Size relations
        Check for separation
        IF Separation = TRUE
            THEN Mark relation as removed
    IF no relation removed
        THEN Size = Size +1 AND Go to step 3b
    FOR all relations sets marked
        Find combination of Sets → Remove "MAX(relations)" AND "Duplicate = FALSE"
    Total = Total + Level*Size*Sets
    Level = Level +1
```

Fig. 1. Pseudo-code for bi-partite graph decomposition [11]

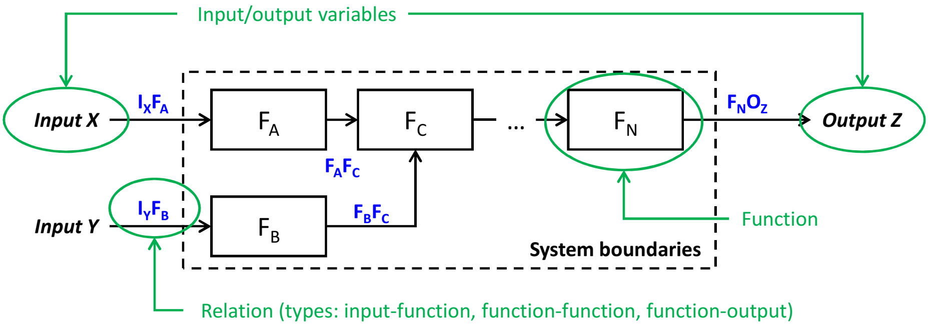


Fig. 2. Function structure example representation [12]

Following the definition of the function structure graph, a bi-partite graph (see Fig. 3) is used as the basis for decomposition. This graph is composed by left and right hand nodes, which are the entities and constraints respectively. The connection lines between them are the relations derived from the function structure graph. In order for the final coupling complexity score to be reached, the bi-partite graph is decomposed, to its fullest extent, into several sub-graphs. Record of the complexity score is kept through the iterative decomposition of equation (1). The index number of the iteration step is the level (l), the minimum number of relations to be removed for a separated sub-graph to be had is the size (s) and the actual number of removed entities is the number (b).

$$\sum_{l=1}^{n}\sum_{l=1}^{n-1} l_n * s_n * b_n \tag{1}$$

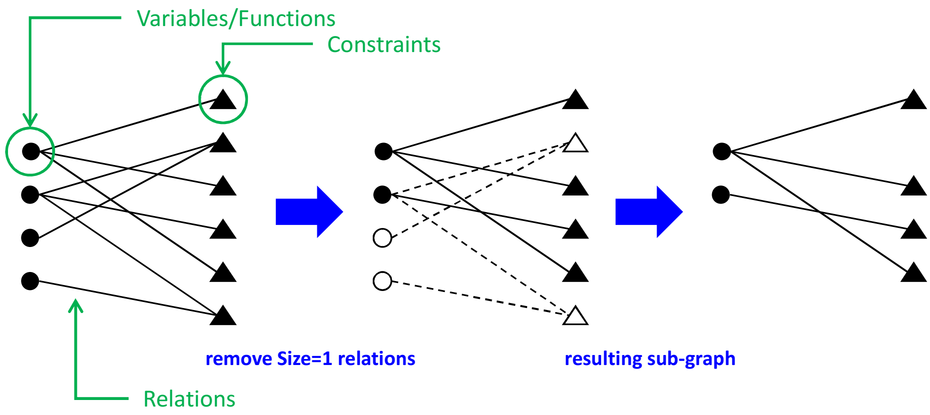


Fig. 3. Bi-partite decomposition method [12]

The coupling complexity of a product design is an aspect of complexity in a design. There are different representations available in literature, besides the various algorithms that can be used for the performance of graph analysis. The method selected for this study relies on the functions involved in the user-product interaction. Function decomposition does not take into account the relationship between the functions, through and input and output associativity, but provides a realistic evaluation of complexity, while remaining less representation-dependent, compared with other methods (e.g. size complexity).

3 VR Design Method

The VR method developed aims at measuring a product's coupling complexity by monitoring the human-product interaction within an immersive virtual environment. The main philosophy of this development in VR is to enable the human user to perform all natural operations and procedures with a product and at the same time to generate the function structure graphs to be used for the evaluation of the complexity of the product at hand. The VR method proposed uses an algorithm developed for the generation of the function structure graph, based on the human user's motion and his interactions (i.e. collisions) with several elements of the virtual product. As depicted in Fig. 4, the architecture of the proposed VR method, implemented for the use-case, described in section 4 of this paper, uses a repository of the product elements in the virtual environment and of the tasks carried out by the human. These repositories are currently used for the generation of the function structure graph and can be replaced by semantic ontologies that will allow for further reasoning to be used and more complex function structure graphs to be generated in a future study. Human motion tracking is performed with 3D objects in the user's virtual hand for the detection of collision with various elements of the product (cockpit).

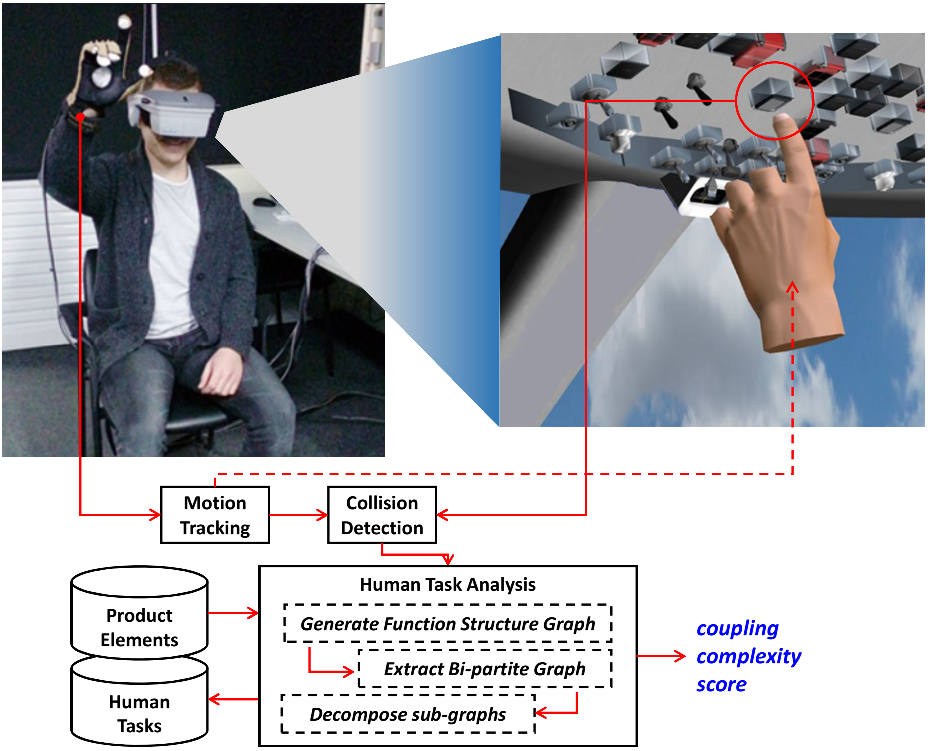


Fig. 4. Architecture of the proposed VR method

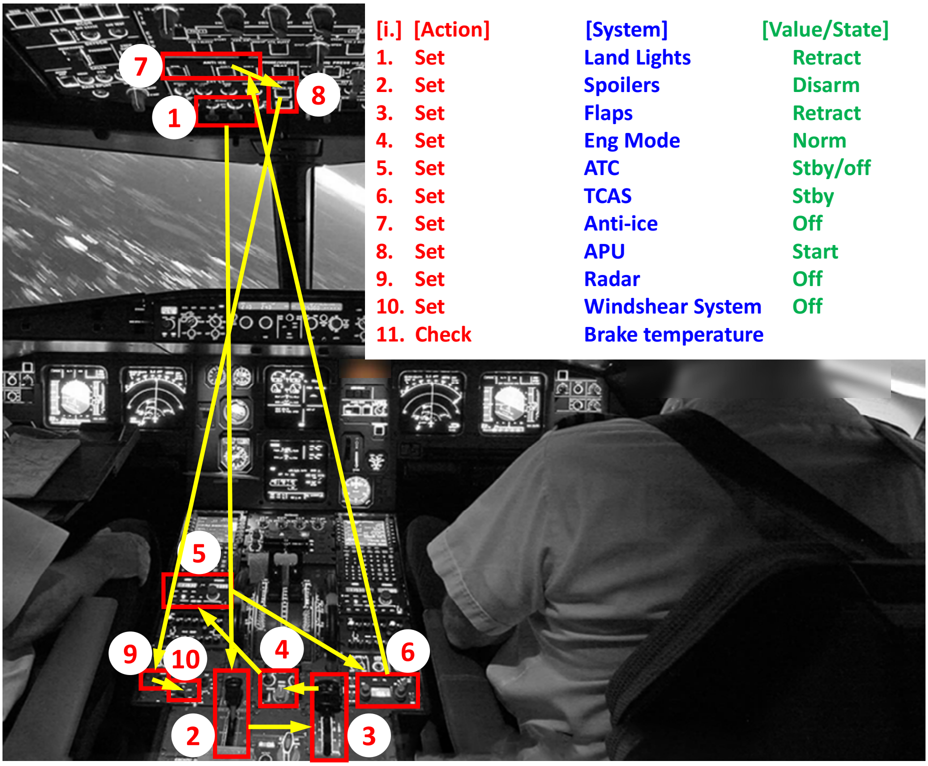
The human task analysis (HTA) capabilities are brought about primarily with the user's hierarchical categorization (pilot) inside the virtual environment (cockpit). The input is the task to be performed by the user (e.g. flight procedure performed by the pilot). The HTA module of the VR method starts by generating the function structure graph, based on the elements, which are stored in a repository in the form of an array and the user interacts with. Each component of the product corresponds to a certain functionality.

As far as the function structure graph implementation in VR is concerned, the first thing to be stated is the number and type of every variable to be included in the graph. The function structure graph has three types of variables namely, input, function, and output. Every value considering the graph generation is stored and handled in an array. The array has three corresponding rows, which the variables are stored in. Considering the interactions that the user performs with the virtual environment, the relations are stored in the array. Specifically, according to the users' type of interaction, (hand, eye, camera tracking) the algorithm stores the appropriate types in the input row. The engine recognizes the users' interaction (Boolean check, collision detection TRUE/FALSE, ray cast TRUE/FALSE) with the virtual environment and due to the fact that every element's function is stored in a product element database, the engine stores the elements outcome in the output row. The connections are stored in a similar manner. For example, after collision detection is made with an element, the engine registers the human hand in the first input cell, the human motion in the first function cell and the relation between them, in a format (cell, cell, 1) where the number 1, will be held for the relation statement. The number increases after the first element of the connection is used again. The engine's configuration is to avoid duplication of the input variable. For example, the users' hand variable and the human motion should exist only once, and only after the interaction type is stated as collision detection performed with some kind of human motion. At the same time, the HTA module updates the human tasks repository that is based on the tasks/actions performed by the user in the environment. The human tasks are stored in an array and act as the relations between the human user and the product. A key logger function is able to distinguish and keep track of the user's every element of interaction, in the virtual environment. In addition, the human task repository is also updated for further task evaluation. An array keeps the stored product elements that the user must interact with in order to perform a distinct task hierarchically. In case of error, the user is virtually notified, of the right element to interact with, or in what manner, considering the value set or the kinematic of the element.

After the function structure graph has been generated, the algorithm extracts the bipartite diagram and starts its decomposition. The function structure graph and the bipartite graph are used as the basis for the decomposition algorithm implementation. The first thing to be examined is the way that the function structure graph is generated and in what degree of detail. In order for the complexity results to be accurately compared, the function structure graphs need to be identical.. After the relations between the variables have been stored in the relation row, the next step is to translate the coupling complexity measure algorithm, proposed for graph decomposition in an array handling engine. The algorithm is transformed accordingly so as to handle the arrayed data. Firstly, the third row of the existing table should be reformed into one unique table for the better handling of its elements. The new table comprising three rows should have the address of the first cell in the first row, in the second one, that of the second cell and in the third row the connection number. The implementation described above follows the pseudo-code, presented in section 2 of this paper.

4 Aerospace Use-Case: Aircraft Cockpit

A realistic use-case aerospace industry, specifically that of an aircraft cockpit design, is used for the demonstration of the applicability and value of the VR method developed. Aircraft cockpits are highly complex products with a huge degree of human interaction during all operating conditions. The proposed VR method offers an easy to use way of evaluating the complexity of a cockpit design by performing flight procedures in a virtual environment. The use-case presented is based on a simple procedure so as to extract the necessary data for the evaluation of complexity. The user, in the virtual environment, interacts with the cockpit in order to perform the procedure set. While performing each task, the user is monitored by the VR algorithm, described in the previous section.



[i.]	[Action]	[System]	[Value/State]
1.	Set	Land Lights	Retract
2.	Set	Spoilers	Disarm
3.	Set	Flaps	Retract
4.	Set	Eng Mode	Norm
5.	Set	ATC	Stby/off
6.	Set	TCAS	Stby
7.	Set	Anti-ice	Off
8.	Set	APU	Start
9.	Set	Radar	Off
10.	Set	Windshear System	Off
11.	Check	Brake temperature	

Fig. 5. Actions performed in cockpit during the “After landing” procedure [13]

The procedure selected for this use-case is an “AFTER LANDING” procedure from the Flight Crew Operations Manual of a commercial airliner. The “AFTER LANDING” procedure is an eleven-task (11) procedure included in the standard operating procedures and is immediately performed after landing (see Fig. 5). It should be mentioned that the procedure was selected among others, due to the high number of the pilots’ interactions with several physical objects and the low need for their communicating with air traffic control (ATC). The user during the procedure needs to interact with two levers, three toggle switches, three rotation knobs and one display. The user is expected to interact with the elements in the predefined order and set the necessary values or states. In cases indicated by the task that the user has to interact with a display or checklist, it is considered as the human user is the output variable.

After the procedure has been carried out, the VR method generates the function structure and the corresponding bipartite graph as depicted in Fig. 6. For this particular procedure, the graph consists of two input variables, twelve function variables and eleven output variables. The coupling complexity algorithm yields a score of 46 for this use-case (level=1, size=1, number=2 and level=2, size=2, number 11).

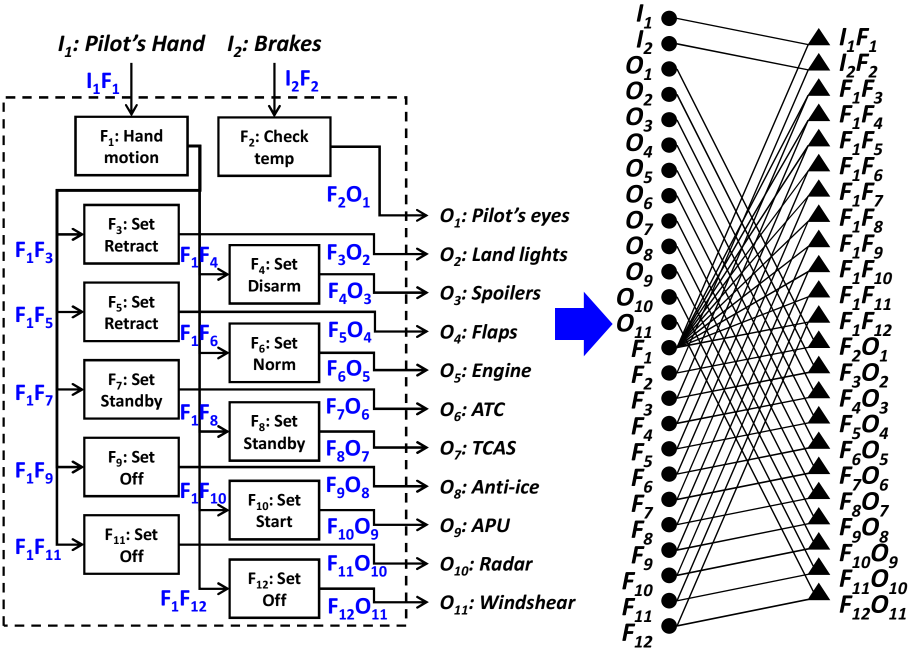


Fig. 6. Function structure graph (left) and bi-partite graph (right) for "after landing" procedure

5 Conclusions

In this paper a VR method for evaluating complex product design is proposed. It enables the evaluation of the complexity by executing the tasks in a natural way. This paper aims at discussing the development and usage of a complexity calculator, in a virtual environment, in order to support the fast and efficient development of the early design product phases. The proposed VR method is implemented on a realistic aerospace use-case. The human user performs a normal flight procedure, in the virtual environment, whilst the tool can calculate the coupling complexity of this particular procedure.

Future study and further enhancement of the VR method presented will consider any additional complexity measurement techniques, used inside a virtual environment, in order for more aspects of product complexity to be evaluated under a single metric. In addition, a semantic implementation of this VR method will allow for advanced reasoning capabilities, during the human task analysis, and will provide the means for increasing the level of detail for the evaluation of complexity.

Acknowledgements. This study was partially supported by the project i-VISION (AAT-2013-605550), funded by the European Commission under the 7th Framework Programme.

References

1. Chryssolouris, G.: Manufacturing Systems: Theory and Practice, 2nd edn., p. 606. Springer, New York (2006)
2. Makris, S., Rentzos, L., Pintzos, G., Mavrikios, D., Chryssolouris, G.: Semantic-based taxonomy for immersive product design using VR techniques. CIRP Annals - Manufacturing Technology 61(1), 147–150 (2012)
3. Suh, N.P.: Theory of complexity, periodicity and the design axioms. Research in Engineering Design - Theory, Applications, and Concurrent Engineering 11(2), 116–131 (1999)
4. Pahl, G., Beitz, W.: Engineering Design: A Systematic Approach. Springer, New York (1996)
5. Boothroyd, G., Dewhurst, P., Knight, W.: Product Design for Manufacture and Assembly. Dekker, New York (2002)
6. Braha, D., Maimon, O.: The measurement of a design structural and functional complexity. IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans 28(4), 527–535 (1998)
7. El-Haik, B., Yang, K.: The components of complexity in engineering design. IIE Transactions (Institute of Industrial Engineers) 31(10), 925–934 (1999)
8. Ahn, J., Crawford, R.: Complexity analysis of computational engineering design processes. In: Proceedings of the 1994 ASME Design Technical Conferences, Minneapolis, MN, USA, vol. 68, pp. 205–220. American Society of Mechanical Engineers, Design Engineering Division (1994)
9. Simon, H.: The sciences of the artificial. MIT Press, Cambridge (1998)
10. Balazs, M.E., Brown, D.: Design Simplification by Analogical Reasoning. In: Cugini, Wozny (eds.) From Knowledge Intensive CAD to Knowledge Intensive Engineering (2002)
11. Summers, J.D., Shah, J.J.: Mechanical Engineering Design Complexity Metrics: Size, coupling, and solvability. Journal of Mechanical Design, Transactions of the ASME 132(2), 0210041–02100411 (2010)
12. Ameri, F., Summers, J.D., Mocko, G.M., Porter, M.: Engineering design complexity: An investigation of methods and measures. Research in Engineering Design 19(2-3), 161–179 (2008)
13. Airbus Flight Crew Operating Manual A319/A320/A321 Flight Operations 3