Title: Decision Making and Software Tools for Product Development Based on Axiomatic Design Theory

Authors: Vigain Harutunian, Mats Nordlund, Derrick Tate, and Nam P. Suh (1)

Mr. Harutunian, Mr. Tate, and Dr. Suh are affiliated with the Department of Mechanical Engineering at the Massachusetts Institute of Technology (MIT) in Cambridge, MA (US).

Mr. Nordlund is affiliated with the Department of Manufacturing Systems at the Royal Institute of Technology in Stockholm, Sweden.

Submitted to: The 1996 CIRP General Assembly in Como, Italy, August 25-31, 1996. (*CIRP Annals*, Vol. 45/1)

Abstract

A software tool based on axiomatic design is being developed. Axiomatic Design (AD) provides a framework to describe design objects and a set of axioms to evaluate relations between intended functions (FR's) and means by which they are achieved (DPs). AD analysis can be performed for engineering change orders (ECO) and field support systems with the capability for organizational learning. The software effort attempts to enhance the engineering CAD environment through the documentation of design rational based on AD and the implementation of AD matrices to evaluate design decisions and provide the proper development sequence.

Key words: Design, Axioms, Software

1. Introduction

Axiomatic design has been used to design various products and processes - better refrigerators to reliable military devices. However, the abstract theoretical concepts have been a stumbling block in teaching the design theories of axiomatic design to an average designer/engineer. Axiomatic design is also a powerful tool in designing large systems (Suh [5]) and in concurrent engineering (Suh [4]). To deal with these issues, we have embarked on the development of a software shell that can be used by designers in developing a product, a large system, concurrent engineering, and other applications. This paper describes the software, including the background information.

The goals of product development are three-fold: reduce development time, increase customer-perceived value, and reduce life-cycle cost for products designed and developed. To achieve these goals, researchers in design theory have sought to understand the relations between the areas of fundamental knowledge in design. Like axiomatic design itself, this paper deals with two of these areas: the design process and the design object.

The purpose of this paper is to describe how the design theories of axiomatic design can be used to facilitate decision making in design with the goal of improving performance relative to these three areas. Specifically this paper is concerned with the generation and use of knowledge about design objects. This knowledge of design objects is structured in a form prescribed by axiomatic design theory, and its use will be described in two cases: the decision to issue an engineering change order and the troubleshooting done during maintenance.

These ideas can be applied more efficiently and more effectively to large-scale systems design through software tools. The ideas described in this paper are being applied to a graphical software tool currently being developed. The motivation for this software, its benefits, and its use will be discussed.

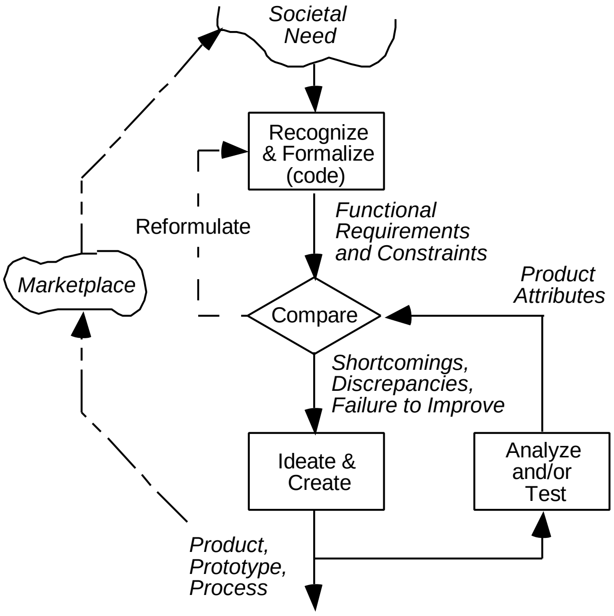

Figure 1. The design process according to Wilson [7]

2. Axiomatic design concepts

In this section the concepts of axiomatic design theory (AD) are described, so that the reader will be able to follow the remaining sections. This section focuses on the process of performing axiomatic design and on the information about the design object which is produced during this process. For a more thorough explanation of axiomatic design see [3].

2.1 The design process and design objects

The design process is the development and selection of a means to satisfy objectives, subject to constraints. It is a series of steps, or activities, by which inputs (including the customer's perception of a problem and an understanding of the resources available for its solution) are transformed to an output (the design object, a solution to this problem). This transformation occurs by means of the designer assisted by design tools/methods and a knowledge base of additional information. The design object may be a physical object, a process, an organization—whatever the customer is willing to accept. Figure 1 shows a simple model of the design process as described by Wilson [7].

2.2 Domains and mapping

During the design process, the problem which is being addressed can be divided into four domains. The number of domains remains constant at four, but the nature of the design elements in each domain changes depending on the field of the problem. (Gebala and Suh [1] list examples of the breakdown of problems.) The four domains may be generalized as the customer domain, the functional domain, the physical domain, and the process domain. Associated with each domain are the design elements it contains. In the order listed, the elements associated with each domain are customer needs (CNs), functional requirements (FRs), design parameters (DPs), and process variables (PVs). Functional requirements are defined as the minimum set of requirements which completely characterize the design objectives for a specific need [3 p. 38]. These FRs must be specified in a "solution-neutral environment", i.e., in terms of the functions to be achieved, not particular solutions

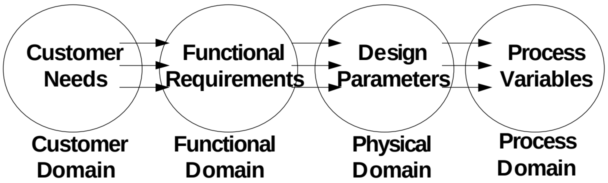

Figure 2. Design domains

These four domains are shown in figure 2. There are guidelines provided by axiomatic design theory (consisting of axioms, theorems, and corollaries) about the relations which should exist between FRs and DPs, that is, these guidelines answer the question—will a set of DPs satisfy the FRs in an acceptable manner? These relations should also hold between DPs and PVs. The relations between CNs and FRs, however, are more loosely structured.

2.3 Zigzagging

The design process progresses from a system level to levels of more detail (from systems to subsystems to assemblies to parts to part features). This may be represented in terms of a design hierarchy, and hierarchies exist for any design object in each of the domains: functional, physical, and process. The decisions which are made at higher levels affect the statement of the problem at lower levels. That is, the designer goes through a process whereby he/she zigzags between domains – functional, physical, and process – in decomposing the design problem.

For example, in designing a transportation system for a city, if the top-level solution is the use of personal automobiles, then the further decomposition of the problem will be very different from the case where a mass-transit system is selected.

At a given level of the design object, there exists a set of functional requirements. Before these FRs can be decomposed, the corresponding design parameters must be selected. Once a functional requirement can be satisfied by a corresponding design parameter, that FR can be decomposed into a set of subrequirements, and the process is repeated. Zigzagging between the functional and the physical domains is illustrated in figure 3.

The designer should realize what choices he/she makes, options should be identified, and a good solution selected. The criterion for evaluation should be—is this option the one most likely to provide a satisfactory result? The design axioms, combined with the designer's knowledge, are a way to answer this question at early (even conceptual) stages of the design process. The designer follows this zigzag approach, checking the correctness of the design at each level, until he/she has decomposed the problem to a point where the solution to the remaining subproblems is known.

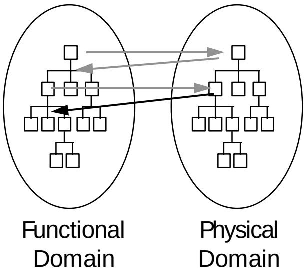

Figure 3. Decomposition by zigzagging

2.4 The design axioms

As described above, the designer follows a design process in which decisions are made about a design object starting with high-level, system decisions and progressing to levels of increasing detail. In following this process—at each level of detail—the steps through which the designer progresses can be described as in the Wilson model (see section 3.1): problem formulation, synthesis, and analysis.

The design axioms provide a tool for analysis, particularly during conceptual design. The two design axioms may be stated as follows [Suh, 1990]:

- *The Independence Axiom (First Axiom)*:
  Maintain the independence of functional requirements.
- *The Information Axiom (Second Axiom)*:
  Minimize the information content [of the design].

Once a set of FRs has been formulated and possible sets of DPs have been synthesized, the two design axioms are applied to evaluate the proposed designs. The application of the Independence Axiom will be described in terms of the design matrix [3].

2.5 Evaluating the design matrix

The design matrix (DM) shows the relations between the FRs and DPs at a given level of the design hierarchy. The design matrix–[A]–results from a design equation of the form shown:

$$\begin{Bmatrix} FR_1 \\ \vdots \\ FR_n \end{Bmatrix} = [A] \begin{Bmatrix} DP_1 \\ \vdots \\ DP_n \end{Bmatrix}. \qquad (1)$$

The elements of the design matrix are determined from the set of equations [3 p. 122]:

$$\begin{aligned} \Delta FR_1 &= \frac{\partial FR_1}{\partial DP_1}\Delta DP_1 + \cdots + \frac{\partial FR_1}{\partial DP_n}\Delta DP_n \\ &\vdots \\ \Delta FR_n &= \frac{\partial FR_n}{\partial DP_1}\Delta DP_1 + \cdots + \frac{\partial FR_n}{\partial DP_n}\Delta DP_n \end{aligned} \qquad (2)$$

where each

$$A_{ij} = \frac{\partial FR_i}{\partial DP_j}. \qquad (3)$$

There are three possibilities for the nature of the design matrix. It can be a matrix populated both above and below the diagonal, a triangular matrix, or a diagonal matrix. These are shown in figure 4. (In this figure an X represents a strong effect by a DP on an FR, and an O indicates a weak effect, relative to the tolerance associated with the FR.) A triangular matrix (B) is known as a <u>decoupled design</u>. A diagonal matrix (C) is an <u>uncoupled design</u>. Any other matrix (A) is known as a <u>coupled design</u>. In an uncoupled design the FRs can be independently satisfied by means of the corresponding DPs. In a decoupled design the FRs can be satisfied if the DPs are varied in the right sequence. A coupled design has no guaranteed point where the FRs can be satisfied. A similar matrix is associated with the mapping between DPs and PVs.

A)	DP1	DP2
FR1	X	X
FR2	X	X

B)	DP1	DP2
FR1	X	O
FR2	X	X

C)	DP1	DP2
FR1	X	O
FR2	O	X

Figure 4. Design matrices: a) coupled b) decoupled c) uncoupled

Axiomatic design provides a framework for describing design objects. This framework is consistent for all types of problems and at all levels of detail. Thus different designers can quickly understand the relations between the intended functions of an object and the means by which they are achieved.

Furthermore, axiomatic design theory encompasses a design process which has several benefits for the creation of designs and information structures to describe them. First, the design axioms provide an analysis tool which is probably unique among design theories. The design axioms provide a rational means for evaluating the quality of proposed designs so that design decisions may be made on a rational basis, supported by easily understood analytical results. Secondly, the design process used guides designers to consider alternatives at all levels of detail and makes choices between these alternatives more explicit.

3. Design information and axiomatic design

A designer following the axiomatic design process produces a detailed description of what functions the object is to perform, a description of the object that will realize those functions, and a description of how this object will be produced. These descriptions are found in the functional domain, the physical domain, and the process domain respectively. Furthermore, there is information about which part(s) of the object perform or affect which functions, as well as information about what manufacturing process variable(s) affect which physical parts in the object. This information is captured in the A (FR to DP) and B (DP to PV) design matrices respectively.

3.1 Information structures

The information captured within each domain is organized in information structures as shown in figure 5. These structures contain information about vertical and horizontal relations within one domain. Information at the highest level of abstractionis at the top of the information structure. The highest level abstractions are decomposed during zigzagging, as described in section 2.3. From this information, it is possible for the designer to deduce, for example, the context of a function or a physical parameter.

Information on the same branch of an information structure, is always directly related. For example, if function C (in figure 5) is not realized, then function B and A are not realized either. The opposite is not necessarily true, i.e., function C may be realized without function B and A being realized. This could happen when function D is not realized.

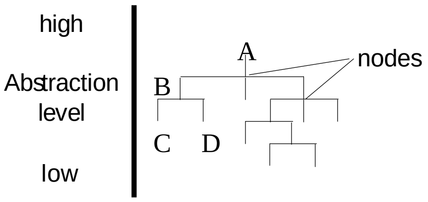

Figure 5. Information structure within one design domain

3.2 Design matrices

The design matrices capture information about the horizontal relations between adjacent domains. There is one design matrix for each node of the abstraction structure. Figure 6 and equation 4 describe the same horizontal relations between two adjacent design domains (functional and physical).

$$\begin{Bmatrix} \mathrm{FR1} \\ \mathrm{FR2} \\ \mathrm{FR3} \end{Bmatrix} = \begin{bmatrix} X & O & X \\ O & X & O \\ O & O & X \end{bmatrix} \bullet \begin{Bmatrix} \mathrm{DP1} \\ \mathrm{DP2} \\ \mathrm{DP3} \end{Bmatrix} \qquad (4)$$

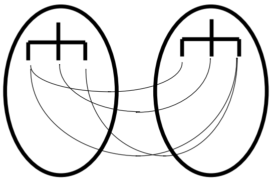

Figure 6. Relationship between information structures in two design domains

3.3 Application of theory to ECO and field services

Based on a knowledge of the information which is captured in the design domains and in the design matrices during the design process, it is possible to expand on the theory of axiomatic design in order to draw conclusions about how this information can be used in different activities related to the design process. The remainder of section 3 presents theory development and conclusions about how a rational analysis can be performed to conduct an engineering change order (ECO) and how this information can be used to create a field support system with the capability for organizational learning (Nordlund [2]).

3.4 Supporting the change order process

When considering the change of a design, consequences must be identified so a rational economic decision about proceeding with the change can be made. The information described above, captured in the design hierarchies and the design matrices, is all the information necessary to analyze the consequences of a change order. To illustrate this, consider the following example: The designers want to change $DP11_1$ for $DP11_2$. What are the consequences? What will be the total cost? The information structures and design matrices for this example are provided in figure 7 and equations 5 through 8.

Changing $DP11_1$ to $DP11_2$ should not affect FR11, as $DP11_2$ should be adjusted so that FR11 still is satisfied. However, equation 7 shows that $DP11_2$ will affect FR12; this has to be corrected by adjusting DP12. DP12 is not a leaf node, it is decomposed into DP121 and DP122. Thus, DP121 and 122 must be changed. Since equation 8 shows another decoupled design, DP122 must be changed before DP121 to minimize the number of iterations. Finally at the higher abstraction levels, equation 6 is an uncoupled design. therefore the changes done at the lower abstraction levels (DP11, DP121, and DP122) will not affect other branches of the information structure (DP2, DP3). If equation 6 had shown a decoupled design where DP1 affected more than FR1, it would have been necessary to adjust either DP2 or DP3, or both.

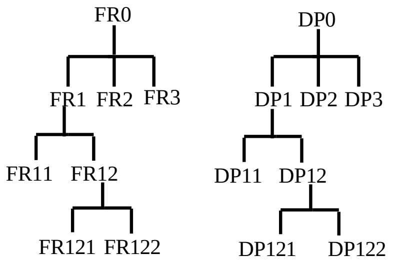

Figure 7. Information structures

$$\{\mathrm{FR0}\} = [X] \bullet \{\mathrm{DP0}\} \qquad (5)$$

$$\begin{Bmatrix} \mathrm{FR1} \\ \mathrm{FR2} \\ \mathrm{FR3} \end{Bmatrix} = \begin{bmatrix} X & O & O \\ O & X & O \\ O & O & X \end{bmatrix} \bullet \begin{Bmatrix} \mathrm{DP1} \\ \mathrm{DP2} \\ \mathrm{DP3} \end{Bmatrix} \qquad (6)$$

$$\begin{Bmatrix} \mathrm{FR11} \\ \mathrm{FR12} \end{Bmatrix} = \begin{bmatrix} X & O \\ X & X \end{bmatrix} \bullet \begin{Bmatrix} \mathrm{DP11} \\ \mathrm{DP12} \end{Bmatrix} \qquad (7)$$

$$\begin{Bmatrix} \mathrm{FR121} \\ \mathrm{FR122} \end{Bmatrix} = \begin{bmatrix} X & X \\ O & X \end{bmatrix} \bullet \begin{Bmatrix} \mathrm{DP121} \\ \mathrm{DP122} \end{Bmatrix} \qquad (8)$$

Thus, by examining the information structure, it can been shown that changing $DP11_1$ to $DP11_2$ results in a need to adjust both DP121 and DP122. Therefore, the total cost of this change is the cost of changing $DP11_1$ for $DP11_2$ plus the cost of adjusting DP121 and DP122.

From this example, it is clear that uncoupled designs provides maximum flexibility to implement ECOs because the consequences of such a change do not propagate throughout the design. Also, changes in decoupled designs are likely to force more than one change in the design because of consequences elsewhere in the design.

3.5 Supporting field service.

When sending out technicians to service equipment on site, it is possible to develop a fault localization system based on the information captured in the design domains and the design matrices. The customer will have noticed that the performance of some function at a high abstraction level is not satisfactory. Since the high level functions are decomposed into several lower level functions, the defective part of the system is most quickly found by tracing down through the abstraction levels of the functional information structure until a faulty leaf node is found. Then, the design matrix at that level of abstraction can be used to determine which subsystem or components realize this function (figure 8).

This fault localization system can also be used to record statistics of what problems occur in the field. By providing this information feedback to the design department, the company is equipped with a valuable organizational learning resource which will enable designers to avoid error-prone solutions in their future designs.

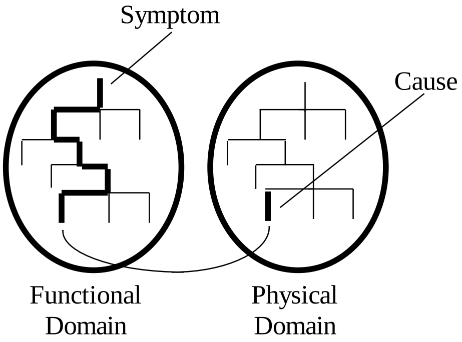

Figure 8. Tracing an error through the information structures from symptom to cause

4. Axiomatic design and software tools

Ullman lists 13 issues dealing with the development of systems to capture and use design history, design rationale, and design intent. Design rationale is defined as an information structure which can be used to answer questions about why an artifact is designed as it is. Specifically it relates the design implementation to the satisfaction of specifications. Design history is a recording of “the evolution of constraints and the effect of decisions on the artifacts being developed” [6 p. 250]. Both design history and design rational are addressed by combining axiomatic design and software tools.

4.1 Development of axiomatic design software

An axiomatic design software tool is being developed using a graphics user interface developer's tool that is both portable and easy to program. The Tcl/Tk development package, written by John Ousterhout, was chosen because of its portability and high-level programming. Thus far, Tcl/Tk is executable on UNIX-compatible machines and is soon to be executable on Microsoft Windows.

Currently, the AD software incorporates the fundamentals of AD. These include the FR, DP, and PV domains, decomposition (with a numbering scheme), and the generation of design matrices (see figures 9 and 10). Once the DMs are entered, the program computes the set of least coupled sequences for an entered matrix. The elements resulting from coupled or decoupled matrices are tagged to prompt the user that any subsequent changes may propagate down to their children. The software also includes general functions such as error messages and save and print functions.

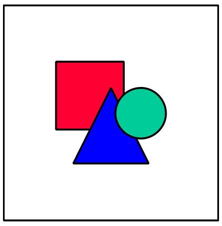

Figure 9. Functional and physical domains

4.2 Software benefits for axiomatic design

Although axiomatic design may be carried out on paper, it is very cumbersome to decompose large development efforts with only a designer’s notebook. An efficient and easy-to-use designer's database can facilitate the management of the data within the AD domains. In addition, a well structured computer-based tool will expedite both the entering and editing of data into the AD numerically labeled hierarchical data structure. Computational power allows the user to automatically sift through the permutations of the entered design matrices to find the least coupled sequence for selecting design parameters. This relieves the designer from the tedious task of rearranging the sequence of the elements within the design matrices manually. Error messages and help files also serve as teaching tools for novice users. The software guides the user through the design process by displaying messages if proper elements are not entered.

4.3 Axiomatic design benefits for software

Modern engineering CAD environments are very powerful and include many analysis tools and data managers. The AD software effort attempts to incorporate a formal design method into the engineering CAD environment. The AD software provides a documentation system that records the design rational and decision making at every step within the design process based on the AD method. The development and propagation of constraints can be traced from the conceptual to the detailed levels of design. Thus, the impact of engineering change orders can be deduced by analyzing the design matrices and hierarchies pertinent to the change. The AD software also provides a common language to communicate design goals and objectives among design team members. All initial constraints and subsequent design goals and solutions are laid out in a logical and consistent manner to be used for reference throughout the design process and forthcoming projects.

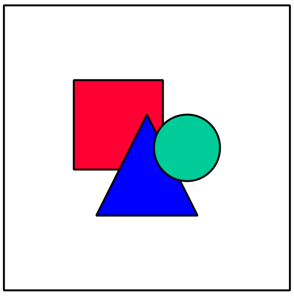

Figure 10. FR-DP design matrix

Future research areas in the implementation of AD with the software tool include a DP database and constraint handling system. A DP database will store critical data and schematics about DPs that may be accessed through the file system by a search engine. Other research issues include constraint handling system which records the generation of constraints resulting from higher level DP selections.

5. Conclusions

Axiomatic design theory provides a fundamental understanding of what information is being used in design. Based on this understanding,, it is possible to utilize the information captured during the design process in activities that take place after the initial design is completed, such as engineering changes and field service. Furthermore, developing an integrated information management system provides an opportunity to feed field information back to the design department.

In order to utilize the information management capabilities of Axiomatic design in large scale system design, it is necessary to implement these capabilities on a computer. Such a demonstration software is currently being developed at MIT. In addition to managing large amounts of design information, the software will be an important tool in teaching the design process to students and practicing engineers.

6. References

[1] Gebala, D.A.; Suh, N.P.; 1992, “An Application of Axiomatic Design”, *Research in Engineering Design*, Vol. 3, pp. 149-162.

[2] Nordlund, M; 1996, *Ph.D. Thesis* (forthcoming), Department of Manufacturing Systems, Royal Institute of Technology (KTH), Stockholm, Sweden.

[3] Suh, N.P; 1990, *The Principles of Design*, Oxford University Press, New York.

[4] Suh, N.P.; 1991, “Development of Science Base for the Manufacturing Discipline”, Proceeding of SME Conference on Manufacturing Education, Washington, D.C.

[5] Suh, N.P.; 1995, “Design and Simulation of Large Systems”, Journal of Manufacturing Systems.

[6] Ullman, D.G.; 1994, “Issues Critical to the Development of Design History, Design Rationale, and Decision Intent Systems”, *DE-Vol. 68 (DTM ‘94)*, ASME, pp. 249-258.

[7] Wilson, D.R.; 1980, *An Exploratory Study of Complexity in Axiomatic Design*, Ph.D. Thesis, Department of Mechanical Engineering, MIT, Cambridge, MA.

.