Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 53 (2016) 101 – 106

www.elsevier.com/locate/procedia

The 10th International Conference on Axiomatic Design, ICAD 2016

Desirable complexity

Joseph Timothy Foley[a,*], Erik Puik[b], David S. Cochran[c]

[a]Reykjavik University, Menntavegur 1, Reykjavik 101, Iceland
[b]HU University of Applied Science Utrecht, Oudenoord 700, 3500 AD Utrecht, Netherlands
[c]Indiana University-Purdue University 2101 E. Coliseum Blvd., ETCS 229B, Fort Wayne, IN USA

[*] Corresponding author. Tel.: +354-599-6569; fax: +354-599-6201. *E-mail address:* foley@ru.is

Abstract

Design methodologies devote a great degree of effort on deciphering, decomposing, and simplifying problems. This approach is particularly true in Axiomatic Design to the point that inability to simplify and understand a situation is defined as complexity. The approach with Axiomatic Design is to avoid complexity because complexity is assumed to make a reliable solution intractable. What if an unreliable situation is needed?

This paper explores the concept of "desirable complexity", an application of Suh's complexity for fields which want to create problems or challenges rather than eliminating them: puzzles, sabotage, physical security, and unique identification. In these areas, inverting AD complexity theory gives suggestions to making duplication and solution discovery challenging by creating seemingly unsolvable problems.

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design

Keywords: Axiomatic Design; complexity; protection

1. Introduction

The core of any design is meeting the customer's needs. This is the traditional wisdom used in the majority of design methodologies in practice. The methodologies such as Axiomatic Design [1], TRIZ [2], and Design Thinking [3] all devote significant effort to deciphering, decomposing, and simplifying elements based upon customer needs. There is an underlying assumption here that is worth considering: all parties involved have a collaborative attitude. In this work, we discuss a number of cases which break this assumption: one of the parties has an antagonistic relationship with the designer. In this case, traditional design methodologies as written are often unable to provide concrete guidelines of how to proceed.

When existing tools are insufficient to meet a need, opportunities arise[1]. Slocum [6, page 3-16] suggests using a mental tool called *Critical Thinking: Maxwell's Reciprocity Theorem* in such a situation: If the current tool does the opposite of what you want, why not try reversing how you operate it? This suggestion is directly applicable to the invalid assumption that we *always* want a design to succeed. We propose the approach of following the basic approach of Axiomatic Design and Complexity theory [7], then actively investigating the opposite of what these methodologies suggest as "good design practice". Bragason et al. [8] previously explored what can be learned by exploring "improper" application of AD theory by translating expert Customer Needs (CN) directly into Functional Requirements (FR). The result was a coupled design in which so-called customer needs were stated, that were actually constraints, and then stated as FRs.

1.1. Axiomatic Design and Complexity

Axiomatic Design [1] was developed with the main purpose of understanding the relationship between conceptual requirements (Functional Requirements) and the details of implementation (Design Parameters). This idea is represented in the form of a transfer function in a matrix as shown in Equation 1.

$$\begin{Bmatrix} FR_1 \\ FR_2 \\ \vdots \\ FR_n \end{Bmatrix} = \begin{bmatrix} A_{11} & A_{12} & A_{13} & \vdots \\ A_{21} & A_{22} & A_{23} & \vdots \\ A_{31} & A_{32} & A_{33} & \vdots \\ \cdots & \cdots & \cdots & A_{mn} \end{bmatrix} \begin{Bmatrix} DP_1 \\ DP_2 \\ \vdots \\ DP_m \end{Bmatrix} \quad (1)$$

[1]"In confusion, there is profit!" *Milo Minderbinder (Jon Voight) in the movie adaption [4] of Catch-22 [5]*

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design
doi:10.1016/j.procir.2016.05.023

The first axiom leads to defining solutions that have only a one-to-one relationship between the functional domain and the physical domain of design A_{ij} as the best. The second axiom states that a design with the minimum information content has the highest probability of success of the system operating range achieving the design-specification FRs. Thus, a design is said to have the least information content and is the most robust when having the design range/capability of the design is completely within the system range specified by the designer. For example, a designer may specify that bar stock be cut to a tolerance of ± 0.05 mm. If a hack saw is used as the physical solution, its design range/capability is ± 1.5 mm, which results in very high information content. It is very unlikely (but not impossible) that the bar will be cut within the desired tolerance. Robustness can also be increased by minimizing the absolute value (gain) of each A_{ij} value as long as the gain stays above "noise." [7, page 37]. The most robust solution has the highest chance of success. Suh defines complexity as, "A measure of uncertainty in understanding what it is we want to know or in achieving a functional requirement (FR)" [7]. When information content cannot be kept small (or nonexistent), this condition is described as complexity [7].

Suh defines four categories of complexity:

Real is due to the inability of the chosen implementation to meet the requirements under all specified conditions.

Imaginary results from a path dependency of FR satisfaction that is not obvious to users, because the design of interest is partially coupled.

Combinatorial results when the system range changes with time because of time-dependent error inputs or system degradation.

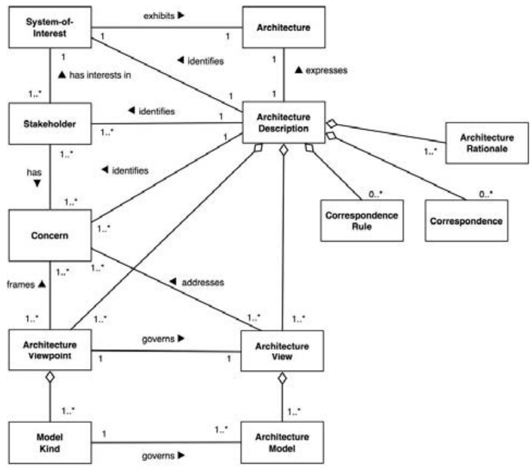
Periodic occurs when a system needs to be "reset" regularly in order to be able to meet its requirements.

Much of the recent Axiomatic Design literature focuses on how to reduce [9,10], manage [11], or measure [12] complexity in a way to compensate for it. Puik and Ceglarek [13] map complexity to knowledge in order to use the Cynefin Framework as guidance in how to explore a solution space in the correct category of unknown unknowns.

To understand why we may "want to fail" or to have a design that is complex leads to the need for solutions requiring self-organization in the face of complexity. Human beings may have been designed to address the seemingly abnormal relationship of "antagonism" in which unreliable solutions are preferred to reliable solutions.

Our knowledge about system design and systems engineering is evolving. Axiomatic design may be considered as one viewpoint within a system architecture description as defined by the relatively new (2011) ISO/IEC/IEEE 42010 standard [14] as shown in Fig. 1. One precept of Axiomatic design is to develop stable and reliable designs.

Yet, there are many other viewpoints that may be used to complete the picture of an architecture description. In computer science, Modular Dependency Diagrams and flowcharts are often used to describe the interaction of data and the underlying processing. These tools are used to find loops and sources of unreliability in the flow of data and of program execution. In all of the systems methods mentioned, reliability is enhanced



NOTE 1 The figure uses the conventions for class diagrams defined in [ISO/IEC 19501].

Fig. 1. Conceptual Model of an Architecture Description

by simplifying and minimizing the number of elements. If this is not what is desired, then any method such as inverting AD theory may be desirable because it will have the opposite effect.

2. Antagonistic Relationships

This initial exploration into assumptions can be found in [15] which discussed using Axiomatic Design to find assumptions to be exploited in security system bypassing. Designing security systems is quite challenging because security systems have Functional Requirements that focus on not having something occur: prevent theft, obscure private documents, contain suspect, etc. Such "negative FRs" are extremely hard to test; comprehensive analysis of all possible conditions is often not possible resulting in nearly guaranteed uncertainty. Security designers instead focus on limiting exposure of sensitive elements locally at the expense of the big picture. To best understand this mindset, we have to first consider the most basic of antagonistic relationships — puzzles.

2.1. Puzzles

Traditionally complexity is focused on tolerances in the physical realm, unless Suh's definition is applied. A typical application of "negative FRs" starts with the challenge of designing a puzzle.

The academic study of puzzles is defined as "enigmatology," an appropriate term coined by Will Shorts who received the first degree in the field [16]. The construction of a suitable puzzle, particularly crosswords, involves understanding the constraints of each possible answer. Consider the following cases for a word puzzle:

Unconstrained: Write a word here: