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.

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

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.

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

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.

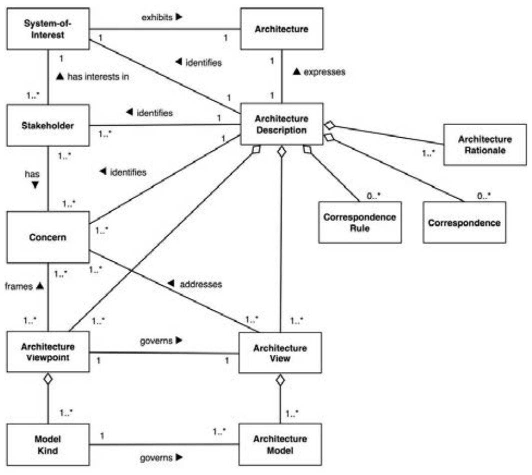
Periodic occurs when a system needs to be "reset" regularly in order to be able to meet its requirements.

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:

Lightly-constrained: Write a four-letter word in the box:

2			

Medium-constrained: Write a four-letter word ending "UCK":

3	U	C	K

Fully-constrained: Write a four-letter word of a waterfowl ending with "UCK":

3	U	C	K

Over-constrained: Write a four-letter English word that rhymes with "orange"

4			

Though allowing for great creativity, many people would find the Unconstrained case unsatisfying. Dissatisfaction is often due to the inability to verify the "correct" answer[2]. As the level of constraint becomes stronger, it becomes easier to validate the correct answer but harder to find it. Going beyond a certain point results in an unsolvable situation. Considering this case with AD terminology, the Unconstrained case has zero information content and should be the best. The Fully-constrained case has large information content and the Over-constrained case, infinite information content. What is going on here?

Another very popular puzzle is the Rubik's cube. This puzzle may be represented as a heavily coupled system. Each turn of a face results in changing the arrangement of 21 elements. There are 4.3252×10^{19} different states and with advanced computer-based strategies any combination can be solved in 18 moves [18]. This finding illustrates imaginary complexity as it is a system that is low in information content and extremely path dependent. Without fully understanding how the sides are connected and modified with each operation, it is extremely unlikely that the correct solution will be found. This puzzle has an obvious success case when the colors on the sides match. Again, AD says that this is a complex design, but it is extremely popular because of the unknown path dependency/imaginary complexity. Apparently, the user wants to be challenged by something that seems difficult. We can now treat the terms "difficult" and "complex" as synonyms in this light.

2.2. Resistance to change

Combinatorial complexity is sometimes desired in manufacturing environments where people are rewarded with overtime for missing a week's production. The people may change inputs to the system about number of products made, inventory on hand, etc. that cause the design to fail intentionally, leading to desirable complexity. This is directly in line with what AD suggests about making projects be on time: reduce the coupling in FRs [19]. This situation arises due to the simple nature of incompatible motivations: personal gain is in direct conflict with the efficiency of the company.

[2]Though in vary rare cases, dissatisfaction may be the desired result! Penn and Teller's Desert Bus video game was intended to be bad [17].

3. Applied Complexity

With the knowledge that complexity allows us to design tasks that are difficult or to seemingly seek failure, a new application arises, "how does one protect valuable assets and systems?" The owner of important assets would like to maintain that possession, something easily stated as an FR, but very hard to verify. Unfortunately, the "negative FR" needed to specify that someone should not be able to take the object away is easier to test, but is not good AD standard practice. Instead we need to focus on the difficulty of the system e.g. complexity of acquisition of assets

3.1. Physical Security

As with the previous puzzle example, we will consider an increasingly constrained system which will make it more complex and non-intuitively, seemingly better.

Unconstrained: asset sitting on the ground

Lightly constrained: asset under a cloth

Medium constrained: asset in a heavy safe

Heavily constrained: asset in a heavy safe, buried underground in a pyramid on a secret place in the world, all engineers and people with knowledge of asset buried in the pyramid.

Overconstrained: asset does not exist, owner pretends that it is inside the safe[3]

Considered from the perspective of complexity in AD, the difficulty to surpass a safety mechanism may be traced back to the knowledge that is required to understand the situation. This knowledge is addressed by the Axioms; the Independence Axiom addresses knowledge that is related to the conceptual status of the problem, and the Information Axiom addresses the knowledge to execute the tasks needed to satisfy the FRs under all circumstances. This codification may also be referred to respectively 'doing the right things' and 'doing things right' [21]. However, this case is aiming for the reversed effect; instead of solving problems we are looking for ways to make things difficult. Therefore, good AD practice is reversed into respectively: 'make it difficult to do the right things' and 'make it difficult to do things right'. Fig. 2 shows a 'Maturity Diagram' [22] that plots satisfaction of the axioms as a product design evolves. The ideal endpoint of the design would normally be the upper right corner in which a design may be considered mature. A mature product complies willingly with the requirements of any user as it is a very regulated situation. Safe and secure systems are adversely found at the lower left corner where it is very difficult to determine actions to circumvent the system and, if the actions were known, it is extremely difficult to execute them. This is clarified by an asset under a camouflage cloth; it is better protected than the asset under a red cloth because it is more difficult to spot. More extreme, opening a safe with a known construction would need some operational skills, but additional

[3]The character Xaro Xhoan Daxos did exactly this in the TV adaptation of Game of Thrones [20].

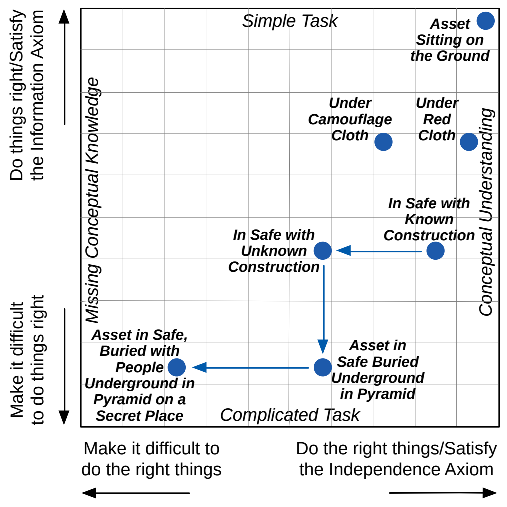

Do things right/Satisfy the Information Axiom
Make it difficult to do things right
Missing Conceptual Knowledge
Conceptual Understanding
Simple Task
Complicated Task
Asset Sitting on the Ground
Under Camouflage Cloth
Under Red Cloth
In Safe with Known Construction
In Safe with Unknown Construction
Asset in Safe Buried Underground in Pyramid
Asset in Safe, Buried with People Underground in Pyramid on a Secret Place
Make it difficult to do the right things
Do the right things/Satisfy the Independence Axiom

Fig. 2. Axiomatic analysis of Physical Security examples

conceptual knowledge is needed to open a safe with unknown construction. More operational excellence would be needed to dig a pyramid out of the ground and even more conceptual understanding is required to deduct its place somewhere in the world, especially with all people knowing about that place are buried in the pyramid.

This extreme example aside, just considering the safe's protection mechanism alone is still related to coupling and complexity. Knowledge in the form of conceptual understanding and operational excellence is needed to unlock it. A combination lock is a common interface to a safe. Standard safes (not embedded into a building) are rated by the Underwriter's Laboratory for their resistance to burglary ranging from TL-15 to TXTL-60. The highest rating (TXTL-60) must be able to withstand an attack with common mechanical and electrical tools, cutting torches, and high explosives for at least 60 minutes [23]. Depending upon the attacker, it may be acceptable to damage the lock and safe. In other cases, particularly with intelligence gathering, it is important that the protection mechanism not be tampered with.

Regardless of these considerations, a basic combination lock is by design a path-dependent mechanism to reduce the chances of success via Imaginary Complexity (Fig. 3, left). Knowledge of the right order of rotations to the correct locations instantly eliminates the complexity. Safe-crackers employ a number of tools in order to reduce this complexity. For older, purely-mechanical combination locks, the use of a ruler attached to the handle for disengaging the latch provides valuable insight into the inner workings of the safe during manipulation. By watching how the handle moves as the dial is spun, the exact location of the combination can be determined (Fig. 3, left).

Where the safe is allowed to be modified, a much simpler method has been employed by one of the author's colleagues: a carbide tipped drill. A hole was carefully drilled into the side of a safe near where the combination dial was located. By looking through the hole, it was quite clear when the slots in the disks behind the combination dial were moving. Again, complexity and information were reduced by gaining knowledge of the inner workings of the safe.

Higher-end safes have tried to eliminate traditional manipulation by ingeniously designed disks and clutch mechanisms. Newer electro-mechanical dial mechanisms have no kinetic connections to the lock mechanism at all. In an AD context, the design goal is to decouple the user from the mechanism so that there is no ability for knowledge to be transmitted through a coupled or partially coupled design.

A similar approach has been applied against the drill method; extremely high security safes contain a glass "locker-plate" which is connected to high-tension springs. If the locker-plate is broken through the use of a drill, an independent set of locking mechanisms engages. This approach reduces chance of success by preventing the the attacker from reducing information content via drilling (Fig. 3, left).

An additional barrier is added in embedded safes through the use of time locks. These locks have an internal clock which ensures that the safe can be only opened during those times. Not having the ability to try numerous codes when the the safe is blocked by the time lock increases the difficulty to gain knowledge of the safe which increases time-dependent complexity in gaining access clandestinely.

Finally, in the most extreme case of an attacker using shaped charged high explosives to open a safe, it is extremely hard to resist. In these cases, AD and best practice have the same suggestion: make the asset value coupled to the state of the container. The asset should be destroyed or equivalently unusable if significant force is used in retrieval.

3.2. Unique Identification

Similar to the these asset protection methods, there are many cases where a designer would like to identify a specific entity uniquely. This problem shows resemblance with the combination locks. Knowing a code satisfies the identification. By combining mechanical and electronic solutions, such as the one discussed below, the quality of identification may be further enhanced.

Modern manufacturing methods for Integrated Circuits (IC) have spent great efforts to generate consistent line widths on microchips. The basic limits of precision ensure that this is an effort that will never end until manufacturers are able and willing to do such mass manufacturing on an atom-by-atom basis. Until that point occurs, there will always be some variation in the resistance, capacitance, or gate-performance of IC-level transistors. If we are trying to get perfectly consistent performance, we are certainly in the area of complexity due to information.

As before, if we apply the general concept of "Desirable Complexity" we can take advantage of these inconsistencies to generate unique identification as in the Physically Unclonable Function (PUF) described in [24,25] (Fig. 3, left). Foley [26] describes a PUF-based method of developing an end-to-end RFID privacy and security solution. A PUF makes use of these manufacturing irregularities and amplifies them to create a unique ID that no other IC will have when made this way. Such a system is extremely resistant to duplication because it takes advantage of the limits of VLSI manufacturing technology's repeatability and turns them into a benefit. Replicating these extremely tiny geometries to replicate such a system is both financially and technologically unfeasible, while making them is not.

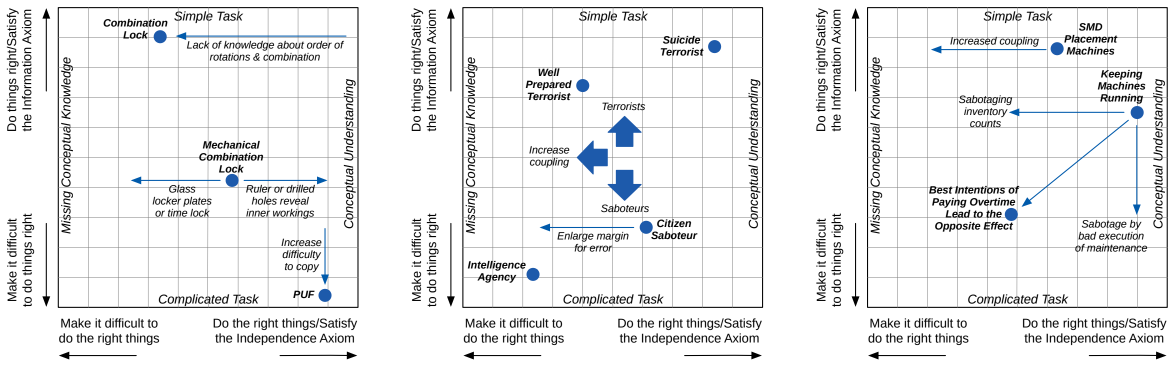

Simple Task
Complicated Task
Missing Conceptual Knowledge
Conceptual Understanding
Do things right/Satisfy the Information Axiom
Make it difficult to do things right
Make it difficult to do the right things
Do the right things/Satisfy the Independence Axiom
Combination Lock
Lack of knowledge about order of rotations & combination
Mechanical Combination Lock
Glass locker plates or time lock
Ruler or drilled holes reveal inner workings
Increase difficulty to copy
PUF
Suicide Terrorist
Well Prepared Terrorist
Terrorists
Increase coupling
Saboteurs
Citizen Saboteur
Enlarge margin for error
Intelligence Agency
SMD Placement Machines
Increased coupling
Keeping Machines Running
Sabotaging inventory counts
Best Intentions of Paying Overtime Lead to the Opposite Effect
Sabotage by bad execution of maintenance

Fig. 3. Axiomatic analysis of three typical cases of systems with desired complexity; Simple Sabotage, Unique identification, and Manufacturing examples

3.3. Sabotage and Terrorism

One highly motivated antagonistic group in the same vein as "Black Hat" hackers is intelligence agencies and saboteurs. These groups are looking for weaknesses in systems to exploit [15]. A key difference between what most people would regard as a "terrorist" rather than a "saboteur" is the concern of being caught: a terrorist may not care once the mission is complete (Fig. 3, middle). Intelligence agencies spend a great deal of effort creating human "assets" to support missions. It should be obvious that a reusable asset is generally worth more in the long run so it is of great interest that the asset not get caught nor killed. It is best if these "citizen-saboteurs" seem to just doing their normal work.

> Sabotage varies from highly technical *coup de main* acts that required detailed planning and the use of specially trained operatives, to innumerable simple acts which the ordinary individual citizen-saboteur can perform. This paper is primarily concerned with the latter type. Simple sabotage does not require specially prepared tools or equipment; it is executed by an ordinary citizen who may or may not act individually and without the necessity for active connection with an organized group; and it is carried out in such a way as to involve a minimum danger of injury, detection, and reprisal [27].

The need for sabotage during World War 2 motivated the Office of Strategic Services to produce a field guide for saboteurs [27]. One thing is immediately striking: many of the organizational sabotage suggestions would be considered almost standard practice in today's larger and overly-complicated business world [28]. As a practitioner in Axiomatic Design and Complexity Theory, something becomes even more striking: many suggestions are explicitly increasing coupling or information content:

> This type of activity, sometimes referred to as the "human element" is frequently responsible for accidents, delays, and general obstructions even under normal conditions. The potential saboteur should discover what types of faulty decisions and non-cooperation are *normally* found in his kind of work and should then devise his sabotage so as to enlarge that "margin for error." [27] (Fig. 3, middle).

Having made this intuitive leap, it becomes clear that looking for sources of coupling as in [15] or to create new sources of complexity is a viable strategy for sabotage. With respect to complexity and its impact, the guide is very clear.

3.4. Manufacturing

Desirable complexity occurs when an entity in a manufacturing system optimizes itself at the expense of the whole as described in Section 2.2. For example, a steering gear manufacturing plant could not produce the right quantity and mix of products on time for their customers. The result of this system failure was that production personnel were paid time-and-a-half overtime on the weekends to catch back up to schedule. Production personnel achieved their FR of increasing their personnel wealth at the expense of system complexity. To accomplish their objective, production personnel would make it difficult to do things right by sabotaging and/or not properly maintaining machines (Fig. 3, right). Likewise, production personnel would make it difficult for others to do the right thing by hiding inventory or providing inventory counts that would cause the master scheduler to produce the wrong product. Complexity was desired by production personnel because they were direct beneficiaries of the system failure and waste that they caused.

An Electronic Engine Controllers (EEC) plant's Operations research group, thought it desirable in order to maximize output to switch to another available machine rather than to fix a machine when it went down (Fig. 4). This policy about opera-

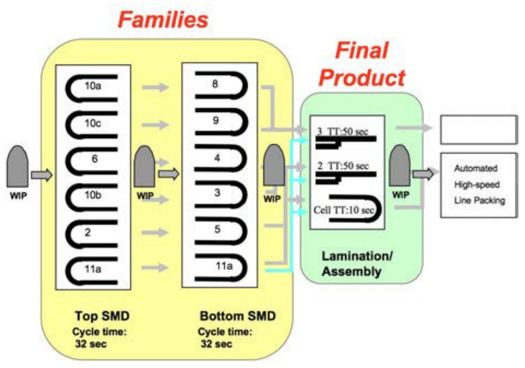


Fig. 4. Electronic Engine Controller manufacturing flow diagram

tions resulted in flow complexity. Any top side SMD placement machine could feed any bottom side SMD placement machine through any combination of flow paths. If top side SMD3 machine went down, for instance, then the operating policy was to run top side SMD4 machine instead of correcting the problem with SMD3. This approach institutionalized flow complexity by not taking the proper actions to repair a machine immediately in the event that it goes down.

Flow complexity is an issue because it is difficult to track root cause of a problem due to coupling of the processes. The irony is that the system design was based on an unrealistic constraint. . . to run the machines all of the time. Yet, the operating policy was to let a machine go down and to switch to another machine. . . which resulted in excessive downtime and uncontrolled inventory levels between machines represented by the Work In Process (WIP) tombstones illustrated in Fig. 4. This constraint is in contrast to a true FR of producing the customer-required quantity every shift. In the same way that the axioms both should be satisfied to realize a good design, satisfaction may be reversed to make something secure, and, while doing this, both axioms serve a different purpose.

3.5. Findings on the Reversed Application of Axiomatic Design

In the same way that AD is applied to investigate how FRs are satisfied by a decoupled set of DPs, its methods may also be applied to investigate how a system can be designed to refrain from satisfying its FRs. The Complexity Theory gives extremely helpful suggestions about how to secure, lock, identify, or sabotage systems and how to do this very subtly by merely increasing coupling or deceasing robustness. The maturity diagram may be helpful to visualize the information in the design and how the axioms are affected during this process.

4. Conclusion

It is not often the case that complexity is desirable. Most systems should be robust; designers want their products to succeed and be able to be manufactured in quantity. The need and desire for complexity comes when there is an antagonistic relationship between the entities involved in a system. In this case, "negative FRs" are effectively created which are most easily addressed by optimizing to increase coupling or information to discriminate against the antagonistic entity.

References

[1] Suh NP. Axiomatic Design - Advances and Applications. Oxford University Press; 2001.

[2] Altshuller G, Clarke DW. 40 Principles: TRIZ Keys to Technical Innovation. Technical Innovation Center, Inc.; 2005.

[3] Brown T. Design Thinking. Harvard Business Review. 2008 Jun;86(6):85–92.

[4] Henry B. Catch-22 Screenplay. Paramount Pictures; 1968.

[5] Heller J. Catch-22. New York, NY: Simon and Schuster; 1961.

[6] Slocum AH. FUNdaMENTALS of Design. MIT Precision Engineering Research Group; 2008. Available from: `http://pergatory.mit.edu/resources/FUNdaMENTALS.html`.

[7] Suh NP. Complexity. Oxford University Press; 2005.

[8] Bragason G, Þorsteinsson S, Karlsson RI, Grosse N, Foley JT. Heat-activated Parachute Release System. In: Thompson MK, Giorgetti A, Citti P, Matt D, Suh NP, editors. Proceedings of the 9th International Conference on Axiomatic Design (ICAD). vol. 34. Procedia CIRP. Florence, Italy: Elsevier ScienceDirect; 2015. p. 131–136.

[9] Matt DT. Reducing the Time Dependent Complexity in Organizational Systems Using the Concept of Functional Periodicity. In: Proceedings of the Fifth International Conference on Axiomatic Design (ICAD2009). vol. 5. Campus de Caparica, Lisbon, Portugal; 2009. p. 1–5. March 25–27.

[10] Cavique M, Mourão AF, Gonçalves-Coelho AM. Reducing Complexity in Outdoor Air Systems. In: Proceedings of the 4th International Conference on Axiomatic Design (ICAD2006). vol. 4. Firenze, Italy; 2006. p. 1–6. June 13–16.

[11] Li Z, Tate D. Managing Intra-class Complexity with Axiomatic Design and Design Structure Matrix Approaches. In: Proceedings of International Conference on Axiomatic Design ICAD2011. vol. 6. Daejeon, Korea; 2011. p. 142–151.

[12] Modrak V, Bednar S. Using Axiomatic Design and Entropy to Measure Complexity in Mass Customization. In: Thompson MK, Giorgetti A, Citti P, Matt D, Suh NP, editors. Proceedings of the 9th International Conference on Axiomatic Design (ICAD). vol. 34. Procedia CIRP. Florence, Italy: Elsevier ScienceDirect; 2015. p. 87–92.

[13] Puik E, Ceglarek D. The quality of a design will not exceed the knowledge of its designer; an analysis based on Axiomatic Information and the Cynefin Framework. In: Thompson MK, Giorgetti A, Citti P, Matt D, Suh NP, editors. Proceedings of the 9th International Conference on Axiomatic Design (ICAD). vol. 34. Procedia CIRP. Florence, Italy: Elsevier ScienceDirect; 2015. p. 19–24.

[14] ISO/IEC/IEEE. Systems and software engineering — Architecture Description. 3 Park Ave. New York, NY: IEEE; 2011. 42010.

[15] Brooks A, Schmiedl E, Foley JT. Exploitable Assumptions. In: DEFCON 2011. Las Vegas, NV, USA; 2011. p. 1–43.

[16] Shortz W; The New York Times. Talk to The Times: Crossword Editor Will Shortz. The New York Times. 2009;Published July 19.

[17] Parkin S. Desert Bus: The very worse video game ever created. The New Yorker [Online periodical]. 2013Available from: `http://www.newyorker.com/tech/elements/desert-bus-the-very-worst-video-game-ever-created`. July 9.

[18] Korf RE. Finding Optimal Solutions to Rubik's Cube Using Pattern Databases. In: Association for the Advancedment of Artificial Intelligence AAAI-97 Proceedings. vol. 14; 1997. p. 700–705.

[19] Suh NP. Challenges in dealing with large systems. In: Thompson MK, Giorgetti A, Citti P, Matt D, Suh NP, editors. Proceedings of the 9th International Conference on Axiomatic Design (ICAD). vol. 34. Procedia CIRP. Florence, Italy: Elsevier ScienceDirect; 2015. p. 1–15. Keynote.

[20] Martin GRR. A Clash of Kings. vol. 2 of A Song of Ice and Fire. New York City: Bantam Books; 1999.

[21] Puik ECN, Ceglarek D. A Review on Information in Design. In: Gonçalves-Coelho AM, editor. 8th International Conference on Axiomatic Design. vol. 8. Lisbon, Portugal: AxiomaticDesign.Com; 2014. p. 1–6. September 24–26.

[22] Puik ECN, Ceglarek D. A Theory of Maturity. In: Gonçalves-Coelho AM, editor. 8th International Conference on Axiomatic Design. vol. 8. Lisbon, Portugal: AxiomaticDesign.Com; 2014. p. 1–6. September 24–26.

[23] Phillips B. The Complete Book of Locks and Locksmithing. 6th ed. McGraw-Hill, Inc.; 2005.

[24] Chuang B, Xuecheng Z, Kui D. A Novel Thyristor-Based Silicon Physical Unclonable Function. IEEE Transactions on Very Large Scale Integration (VLSI) Systems. 2016;24(1):290 – 300.

[25] Akgün M, Çağlayan MU. Providing destructive privacy and scalability in RFID systems using PUFs. Ad Hoc Networks. 2015 Feb;32:32–42.

[26] Foley JT. Security Approaches for Radio Frequency Identification Systems [Ph.D. Dissertation]. Massachusetts Institute of Technology; 2007.

[27] United States Office of Strategic Services. Simple Sabotage Field Manual. Langley, Virginia, USA; 1944. 3.

[28] Feloni R. The 16 best ways to sabotage your organization's productivity, from a CIA manual published in 1944. Business Insider UK. 2015 Nov;.

[29] Thompson MK, Giorgetti A, Citti P, Matt D, Suh NP, editors. Proceedings of the 9th International Conference on Axiomatic Design (ICAD). vol. 34. Procedia CIRP. Florence, Italy: Elsevier ScienceDirect; 2015.

[30] Gonçalves-Coelho AM, editor. Proceedings of the 8th International Conference on Axiomatic Design. vol. 8. Lisbon, Portugal: AxiomaticDesign.Com; 2014. September 24–26.