Proceedings of ICAD2014
The Eighth International Conference on Axiomatic Design
Campus de Caparica – September 24-26, 2014

ICAD-2014-19

A THEORY OF MATURITY

Erik Puik
erik.puik@hu.nl
HU University of Applied Sciences Utrecht
Oudenoord 700, 3513EX Utrecht
the Netherlands

Darek Ceglarek
D.J.Ceglarek@warwick.ac.uk
International Digital Laboratory
WMG, University of Warwick
Coventry, CV4 7AL, UK

ABSTRACT

The 'Axiomatic Design Methodology' uses 'Axioms' that cannot be proven nor derived from physical phenomena. The axioms serve as guidelines for the design process of products and systems. The latest contribution was the addition of the 'Complexity Axiom' in 1999. However, the underlying theory of complexity did not get much traction by designers and their managers yet. It emphasises difficulties in the design, not primarily focussing on solutions. The 'Theory of Complexity' is converted to a 'Theory of Maturity' in this paper. It is supported with a graphical way to plot maturity as it develops. It visualises the results in a way that can be understood by all entities in a company, engineers, managers, and executives. Understanding the maturity of a system enables selection of the right measures to control it. Visualisation enables communication between the interacting parties. If successful development trajectories are understood, eventually from earlier experience, even better corrective actions can be applied. The method appears an affirmative way to graphically represent progression in design, thus presenting advances in a positive context. Though positively presented, it is not the case that the method hides problems; presumed and legitimate project progression can be quite different, which challenges the designer to understand the process. In this way, the method sends out a continuous warning to stay critical on design choices made.

Keywords: Axiomatic Design, Information, Complexity, Independence, Maturity

1 INTRODUCTION

'On an Axiomatic Approach to Manufacturing and Manufacturing Systems' [Suh, 1978] was the first scientific paper to introduce the Axiomatic Design (AD) methodology. AD declares axioms that cannot be proven nor derived from physical phenomena. In that first paper, a number of seven axioms were defined. Only two of those seven axioms are still maintained over 35 years later, now known as the 'Independence-' and the 'Information Axiom'. The other axioms have been changed into corollaries or theorems, because they were incidentally proved or could be deduced from the remaining two Axioms. In 1999 a third Axiom was added, the 'Complexity Axiom', addressing different kinds of complexity that can reside in product- or system designs.

The activities to decompose axioms into corollaries and theorems have led to a substantial framework of design guidelines. Just under fifty have been defined in total, which tends to make the AD theory complex to novice users. The addition of the complexity axiom further extends AD, as well in opportunities as in multiplicity of AD itself. It is another reason that AD didn't get extensive traction yet.

In addition, a theory of complexity is not happily accepted by designers and their managers. It emphasises the difficulties in the design, not primarily focussing on solutions and the other positive aspects of a well functioning design. Therefore the 'Theory of Complexity' is converted in this paper to a 'Theory of Maturity'. It is supported with a graphical way to plot maturity of a progressing design and visualise the results in a way that it can be understood by all entities in a company. It combines the strengths of AD with a simple and clear way of communication: About 'doing the right things' followed by 'doing things right' [Puik, 2014].

This paper is organised as follows: In section 2, maturity is defined and explained in an axiomatic context. Section 3 explains the relation of maturity with the axioms. Section 4 introduces an axiomatic method to visualise maturity, by plotting presumed and legitimate maturity in a two dimensional space. Section 5 discusses the findings and summarises conclusions.

2 MATURITY OF A PRODUCT DESIGN

Maturity is defined as: 'The quality of reaching full development' [Oxford, 2014]. Products or systems that are fully developed, and thus mature, may be expected to function according to the user's expectations. It may be seen as a Utopian image that is difficult to attain but it would lead to an exceptional quality of products.

2.1 MATURITY TRACKING

Understanding the maturity of a system enables awareness how the system will behave under circumstances within or without the specified operational range. But it is even more important to monitor maturity as it evolves during the design process:

- It enables the possibility to follow the system- or product-design during the design stages to estimate progression;
- It increases understanding of errors made during the design process and how they were addressed in order to avoid them in the future;

- It helps selection of the right measures to control product development, like the use of appropriate methods for 'Systems Engineering' (or 'Engineering Design' as it is more commonly referred to in the US).

2.2 ABSOLUTE AND RELATIVE MATURITY

It is a legitimate question if it is possible to reach the state of full maturity. A product design can always be optimised further, which means that the future product will have a higher state of development and thus will be more mature. On the other hand, a product that completely fulfils the wishes of the customer, may be considered fully mature. Maturity has an absolute, as well as a relative characteristic. The on-going development of a car could be considered as an example: Good cars were produced already twenty-five years ago. E.g. Japanese cars in the late eighties already were generally reliable and well engineered. They might be considered mature. Cars today are not necessarily engineered to a higher level, but do have more options for passive and active safety. From this perspective, a modern car may be considered more mature than a late eighties car. This perspective recognises two kinds of maturity, 'Absolute Maturity' and 'Relative Maturity':

- Relative maturity indicates how well a design is engineered;
- Absolute maturity indicates to what extent a design is optimised, i.e., its design status compared to comparable products or systems in the past.

This paper focusses on relative maturity, how it can be decomposed, characterised and measured.

2.3 AXIOMATIC MATURITY

The AD methodology can be applied to characterise relative maturity. AD focusses on Functional Requirements (FRs) and how they are satisfied by Design Parameters (DPs) [Suh, 1990]. From this perspective, a mature design is capable of satisfying all FRs. Therefore Axiomatic Maturity (AM) is here defined as *'the quality of satisfying all FRs of a system'*. This makes AM inversely related to Axiomatic Complexity, which is defined as 'A measure of uncertainty in achieving specified FRs' [Suh, 2005]. However, complexity is in practice expressed in entropy, or information, according to the definition of Shannon for information technology [Shannon, 1948]. AM will be expressed as the vector sum of the joint probabilities of independence and robustness as

$$\overrightarrow{AM} = \vec{P}_j + \vec{P}_i \qquad (1)$$

where P_j is the probability coming forth from dependencies in the design or system and P_i is the probability due to limitations in robustness.

3 RELATION OF MATURITY AND THE AXIOMS

As AD uses axioms to advise on the actions to apply during the design of products and systems, it is a question how AM relates to the axioms. All axioms are information axioms according to the Shannon theory [Puik, 2014], so all axioms are associated in some manner to AM. Axiom 1, the independence axiom, is related to 'Unorganised Information', which is information that is present in a system that has not been organised yet and still has an incomplete or a coupled design matrix. Axiom 2, the information axiom, is related to 'Axiomatic Information', which may be present in a system, with a well-organised design matrix that is at least decoupled but can also be uncoupled. Axiom 3, the complexity axiom, is represented by both the information of axiom 1 and axiom 2 [Puik, 2014]. Since the two kinds of information are of a different kind, again the vector sum is taken. The complexity in the system, when the axioms have not been satisfied yet is given by

$$\vec{C}_{Total} = \vec{I}_{Ax} + \vec{I}_{UnOrg} \qquad (2)$$

$$I_{Ax} = C_{Re} = -log_b P_i \qquad (3)$$

$$I_{UnOrg} = C_{Im} = -log_b P_j \qquad (4)$$

where I_{Ax} and I_{UnOrg} respectively are the axiomatic and unorganised information and b is the again the base of the logarithm. The AM can be found by substitution of (3) and (4) into (1)

$$\overrightarrow{AM} = \vec{b}^{-I_{Ax}} + \vec{b}^{-I_{UnOrg}} \qquad (5)$$

$$\overrightarrow{AM} = \vec{b}^{-C_{Real}} + \vec{b}^{-C_{Im}} \qquad (6)$$

Two things should be noted when applying equation (5) or (6) to determine AM:

1) The equations are of a value that is constantly changing due to the evolving design and only become stable when the design matrix is known and at least decoupled. In this situation I_{UnOrg} (C_{Im} in (6)) becomes zero, the latter term will become one and will have no further effect. The reason for this is that the design matrix is a game changer to organised information. Changes of the matrix lead to the exclusion of existing DPs and include new DPs in the design, which implicitly means that the match of system range and design range becomes obsolete and needs to be reconsidered.

2) The latter term is of a much larger magnitude than the former. The reason is that the probability of unorganised information is caused by trial-and-error [Suh, 2005], for organised information it is the overlap of system and design range. The procedure of trial-and-error will cause considerably larger information content and therefore significantly smaller probabilities than the probability due to axiomatic information [Puik, 2014].

4 VISUALISATION OF MATURITY IN AN AXIOMATIC MATURITY DIAGRAM

Visualisation of the design process enables the product or system design to be graphically tracked as it develops. If ideal development trajectories are understood, and the graphical position can be determined, then corrective actions can be planned to bend the development in the most optimal direction for the specific situation, e.g.:

- Most efficient development path in terms of investment (SMEs);
- Optimised development path for project lead time (semiconductor industry);
- Lowest chance for development errors (safety systems, medical);
- Etc.

4.1 The Axiomatic Maturity Diagram

The Axiomatic Maturity Diagram (AMD) is based on the independence- and the information axioms. The diagram uses two axes, one for each axiom, plotting the joint probability that the axiom is fully satisfied. This is carried out applying the first and second term of (1). The vectors for P_j and P_i are perpendicular. P_j is the measure for independence on the horizontal axis. P_i is the measure for information on the vertical axis. The horizontal axis is the 'axis of ignorance' starting at full ignorance and ending with 'proof of concept'. This indicates that there is no ignorance left in the system. The vertical axis represents 'robustness of the DPs satisfying the FRs' from not- to fully robust. Robustness is applied according to Suh's real-complexity definition [Suh, 2005]. In this definition, the information axiom addresses only the axiomatic component of information and not the unorganised part, which is addressed by the independence axiom.

The lower left corner indicates a high level of ignorance. The designer has little knowledge how to satisfy FRs with his DPs and therefore the AM is low. The upper right corner shows low information content and maximum probability of FRs being satisfied. This is the area of high AM. The direction for development of products and systems is from the lower left to the upper right. Products are fully mature when they reach the upper right corner of the AMD, as marked with a dot. The AMD is plotted in figure 1.

Figure 1: The AMD. The horizontal axis plots the independence axiom, the vertical Axis plots the information axiom. The development path is arbitrary

It might be noticed that the axes of the AMD are swapped compared to the real and imaginary axes in Suh's complexity diagram [Suh, 2005:71]. This is correct. There are two reasons to deviate from that definition; 1) axioms 1 and 2 are simply plotted in that order, and 2) the level of independence, as set by axiom 1, never moves backwards (as long as no knowledge of the designer is lost, it will typically increase). This is not the case for the information axiom (optimisation of the design matrix can lead to different FR-DP relations and the satisfaction of the information axiom might initially decrease). By choosing this way of plotting, the maturity development path takes the form of a mathematical function. This makes reading the AMD more natural.

4.2 Presumed and Legitimate Position in the AMD

As a product design moves through the AMD, during the development process, its position as presumed by the designer may not be the same as the legitimate position of the design in the AMD. This difference is caused by a lack of knowledge to the designer. If the design error is discovered, a correction on the discrepancy can be carried through. If not, the discrepancy will be corrected at some point in the remaining part of the development process. The correction, in that case, will come as a surprise.

4.3 Determining the Legitimate Position

Since discrepancies between presumed and legitimate positions in the AMD are caused by a lack of knowledge to the designer, acquisition of the knowledge would lead to an instant disappearance of the discrepancies. Obviously, there is no method that comprehensively enables this; every design would indeed be a 'Good' design if such a method would exist. What can be done is to apply methodologies that objectively determine the position of a design in the AMD as it progresses. This forces the presumed position to be based on facts instead of gut feeling. It will contribute to a higher degree of realism of the designer. A few of such methods have been described in literature. To track the independence axiom, Suh has reported a sequence of steps to follow [Suh, 2004]. More recent work was done by Puik et al [2013-1]. It defines a framework of seven steps to follow the independence axiom during design progression, starting with decomposition of the design, finding the DPs, decoupling the matrix and, testing the system to make sure that all DPs really have been found. Following the information axiom is more straightforward as it does not blur the perception of the designer as much as the independence axiom (no further discrepancies). FMEA [Suh, 2004], [Puik, 2013-2] or qualitative analysis [Puik, 2013-3] could be applied to objectively monitor progression of the information axiom. By performing checks, based on these methods, on a regular basis, discrepancies between presumed and legitimate positions can be corrected before they grow to unwanted proportions. Heavy corrective actions may be prevented.

4.4 Ideal Development Path Through the AMD

Product development, as indicated above, will start somewhere at the lower left and will move diagonally upwards. The starting point will depend on the complexity of the project definition. A high-tech project that is new to the world might start with high amount of ignorance compared to a project that aims to develop according to the first-time-right philosophy.

The development will rarely follow a straight line. In the axiomatic literature [Suh, 1990], [Suh, 2001], [Suh, 2005], [El-Haik, 2005], it is found that axiom 1 and 2 are typically satisfied in that order. This is done by completion and decoupling of the design matrix before matching the system- and the design ranges of FRs and DPs. The result is 'doing the right things' first, followed by 'doing things right'. This leads to a preferred path that first moves to the right and then angles upwards. It is plotted in figure 2.

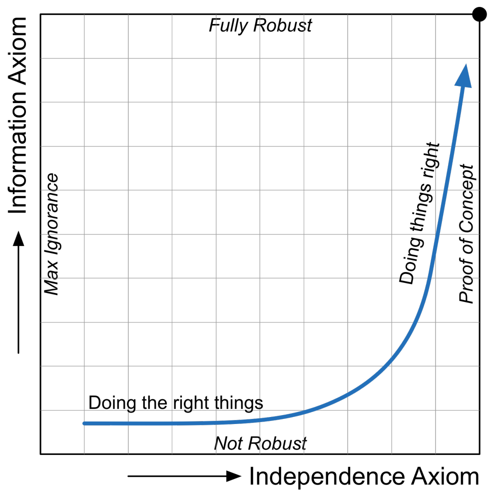

Figure 2: A preferred development path through the AMD, as indicated in literature, first moves to the right to satisfy Axiom 1. After this, Axiom 2 is satisfied in an upward direction

In practice, it is rarely the case that the ideal path is followed. It would mean that not any design flaws are made.

4.5 Consequences of Typical Errors

Design errors that occur can be plotted in the AMD. It will lead to a discrepancy of the presumed and legitimate position in the diagram. If a design error is found, by executing one of the methods as described in §4.4, the discrepancy decreases. If all errors in the design are known the presumed and legitimate positions will be the same. This does not necessarily mean that all knowledge is present but the designer does fully understand the status of the project.

Wrong DP

A wrongly chosen DP leads to the situation that the DP not or insufficiently satisfies the related FR. It will seem to the designer that the design matrix for this DP is known and decoupled, but in fact this is not the case. Time and effort are spent to match the system and design ranges of this DP, but since the DP has no or very little effect these efforts are spent in vain. Discovery of the error increases independence of the design but typically reduces organised information with less robustness as effect. The reason is that the design matrix will need corrections as a result to the faulty DP. A new DP will need to be installed to address the related FR. Discovery of the error leads to a loss of design efforts. The impact of the error will increase in proportion to the discovery time, as the negative effects tend to accumulate. Hence, early detection of errors leads to lesser loss of efforts and should be pursued. Figure 3 plots the discontinuity when discovering a faulty DP in the design matrix.

No decoupling

No decoupling means that all DPs are known but the design matrix is coupled. The design can be optimised by matching the system- and design ranges. However, unorganised information remains in the system. An example is the combination lock as described by Suh [2004].

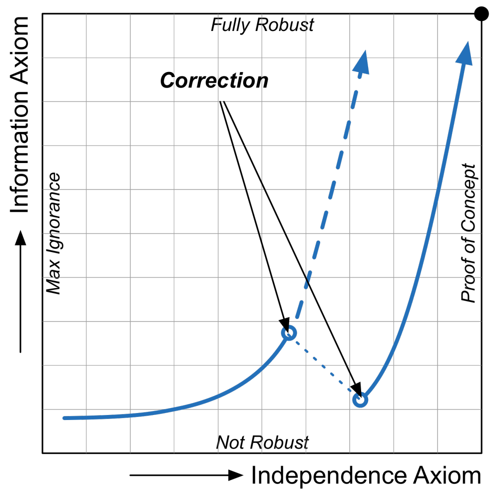

Figure 3: Discovery of a wrong DP leads to better independence but sets the Information Axiom back. The dotted line shows the development if the error would not have been discovered

If a combination lock is to be opened without knowing the code it is a matter of trial-and-error to open it. Even if the instruction manual is available it is not possible to open it without further knowledge (being the code). All organised information is removed from the system and axiom 2 is fully satisfied, but the independence axiom is not. The result, when the error is discovered, is dependent on the number of DPs that have to be replaced. The result of replacing DPs will lead to a fallback in satisfaction of axiom 2. If no DPs have to be changed, like the combination-lock example, the effects are minimal and the design will get rid of remaining dependencies. A fully mature state would instantly be the result. Both options are shown in figure 4.

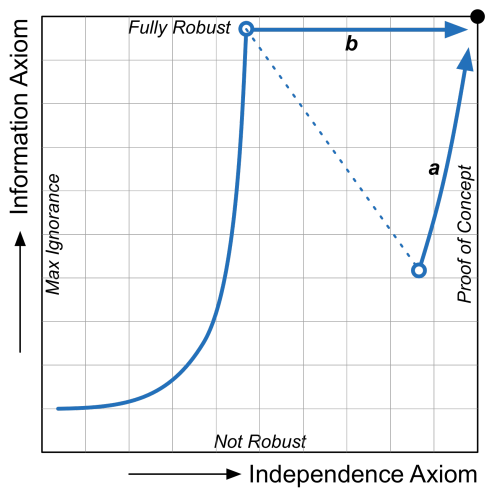

Figure 4: A coupled design matrix does not prevent satisfaction of axiom 2. However, if decoupling of the matrix needs replacement of DPs, axiom 2 is not automatically satisfied for the new DPs and a lot of efforts may be lost (option a). Option b shows a luckier

situation that the DPs can be maintained. In this case the impact on the design is minimal

Non matching system- and design ranges

A non-matching system and design range for one or more of the relations between FRs and DPs leads to the situation that the information axiom cannot be fully satisfied. Note that axiomatic information according to (3) is defined as joint probability (quantified product of all probabilities). Therefore the mature state is only reached if all system- and design-ranges are matched (figure 5).

Figure 5: A non-matching system- and design range prevents the mature state from being reached. The design will not become robust

In this case, the discrepancy between presumed and legitimate positions in the AMD has disappeared. The discrepancy is only present until the design becomes independent. Remaining organised information does not change it. Only unorganised information will lead to a discrepancy of presumed and legitimate position.

5 DISCUSSION AND CONCLUSIONS

The AMD is an unbiased way to graphically represent progression in design. It shows progression, and not a reduction of complexity, thus presenting advances in a positive context. However, it is not the case that the AMD shies away from problems in the design; by showing that the presumed and legitimate positions in the diagram can be quite different, it sends out a continuous warning to stay critical on design choices made and the related presumed status.

5.1 Ideal Development Path Through the AMD

Product and system design is about taking controlled risks. A certain amount of trial-and-error in finding a well functioning set of FRs and DPs is acceptable if not inevitable. The ideal path through the AMD, as indicated in literature, prioritises reduction of independence before reduction of axiomatic information. Though this seems a solid way to go, it might be acceptable in some cases to choose a different way e.g., a way that involves more risk but accelerates the design process at a cost of resources. It is not clear yet what the characteristics of such a path are and how it can be plotted through the AMD. It remains part of future research.

5.2 Discrepancies in the AMD

Studying the discrepancies between the presumed and the legitimate positions in the AMD is a good thing, but this does not help future design projects. The problem is that the discrepancies are caused by a lack of knowledge and, if the designers were aware of this, they would take care of the problem instantly. In the execution of design projects it is never completely clear if discrepancies are latent. Till now, the AMD cannot change this.

A positive contribution of the AMD is that faulty scenarios, eventually from the past, can be analysed and characterised. Causes and consequences become clear lessons for future design projects. In learning organisations, including universities, the AMD can serve as an effective tool to explain the origin of errors made in student-projects. It will significantly contribute to the learning experience.

Only unorganised information leads to discrepancies in the AMD. Axiomatic information does not contribute to it because errors in overlap of system and design range come to the surface automatically and will be instantly recognised by the designers.

Discovery of discrepancies in the AMD can reveal ineffective engineering efforts but this is depending on the kind of error that is made. Some errors have been described in this paper, many more scenarios could occur. Charting other scenarios is work for future research.

5.3 AMD as Means for Communication Between Management and Engineers

The AMD may elevate the level of communication in the organisation because the impact of errors that are found and the correction related discrepancies can be graphically communicated. It widens the scope of personnel to be addressed and being capable of understanding what went wrong, for engineers, managers and executives.

5.4 Further Remarks

The AMD could benefit from better methods to detect the discrepancies. However, this is not simple. These methods should be capable of detecting the ignorance of the designer. The current methods use benchmarks that could be organised in a balanced scorecard. No evaluation has been made yet to study how existing tools for systems engineering can contribute.

5.5 Conclusions

The Axiomatic Maturity Diagram is a way to plot the progression of a developing product or system design. There is a preferred path that a product could run through, coming forth from the AD literature that advised to first satisfy the independence axiom followed by satisfaction of the information axiom.

Any product has, in an arbitrary development stage, a legitimate position in the AMD, but that position may be different of the position as presumed by the designer. The discrepancy between these positions is a measure for the ignorance of the designer. It will probably lead to efforts being spent in vain.

The Axiomatic Maturity Diagram is especially suitable for a learning environment, because the feedback of development actions can be analysed afterwards to provide valuable feedback.

6 ACKNOWLEDGEMENTS

This research was funded by the Dutch Stichting Innovatie Alliantie and the HU University of Applied Sciences Utrecht.

7 LITERATURE

[1] El-Haik, B., Axiomatic quality, Wiley-Blackwell, 2005

[2] Oxford University Press, “Oxford English dictionary online,” 07-May-2014, www.oxforddictionaries.com, [Accessed: 07-May-2014]

[3] Puik, E. C. N., E. Smulders, J. Gerritsen, B. V. Huijgevoort, and D. Ceglarek, “A Method for Indexing Axiomatic Independence Applied To Reconfigurable Manufacturing Systems”, 7th International Conference on Axiomatic Design ICAD2013, Worcester, 1st ed., vol. 7, pp. 186–194, 2013-1

[4] Puik, E. C. N., D. Telgen, L. V. Moergestel, and D. Ceglarek, “Structured Analysis of Reconfigurable Manufacturing Systems”, International Conference Flexible Automation and Intelligent Manufacturing FAIM2013, 2013-2

[5] Puik, E. C. N., D. Telgen, L. V. Moergestel, and D. Ceglarek, “Qualitative Product/Process Modelling for Reconfigurable Manufacturing Systems”, International Symposium on Assembly and Manufacturing ISAM2013, 2013-3

[6] Puik E. C. N., and D. Ceglarek, “A Novel Perspective on the Information- and Complexity Axioms,” 8th International Conference on Axiomatic Design ICAD2014, 2014

[7] Shannon, C.E., “A Mathematical Theory of Communication,” p. 125, 1948

[8] Suh, N. P., The principles of design, Oxford University Press, 1990.

[9] Suh, N. P., A. Bell, and D. Gossard, “On an Axiomatic Approach to Manufacturing and Manufacturing Systems,” J. Eng. for Industry, 1978

[10] Suh, N. P., Axiomatic Design - Advances and Applications, Oxford University Press, 2001

[11] Suh, N. P., Exploration Systems Enterprise Request For Information, 2004

[12] Suh, N. P., Complexity, Oxford University Press, 2005