Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 53 (2016) 70 – 77

www.elsevier.com/locate/procedia

The 10th International Conference on Axiomatic Design, ICAD 2016

Ignorance is bliss: sudden appearance of previously unrecognized information and its effect

Erik Puik[a,*], Joseph Timothy Foley[b], Darek Ceglarek[c]

[a]*HU University of Applied Science Utrecht, Oudenoord 700, 3500 AD Utrecht , Netherlands*
[b]*Reykjavik University, Menntavegur 1, Reykjavik 101, Iceland*
[c]*International Digital Laboratory, WMG, University of Warwick, Coventry, CV4 7AL, UK*

[*] Corresponding author. Tel.: +31 6 51554041; *E-mail address:* erik.puik@hu.nl

---

Abstract

The Information Axiom in axiomatic design states that minimising information is always desirable. Information in design may be considered to be a form of chaos and therefore is unwanted. Chaos leads to a lack of regularities in the design and unregulated issues tend to behave stochastically. Obviously, it is hard to satisfy the FRs of a design when it behaves stochastically. Following a recently presented and somewhat broader categorization of information, it appears to cause the most complication when information moves from the unrecognised to the recognised. The paper investigates how unrecognised information may be found and if it is found, how it can be addressed. Best practices for these investigations are derived from the Cynefin methodology. The Axiomatic Maturity Diagram is applied to address unrecognised information and to investigate how order can be restored. Two cases are applied as examples to explain the vexatious behaviour of unrecognised information.

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design
Keywords: axiomatic design; information categorization; Cynefin

---

1. Introduction

Since the introduction of Axiomatic Design (AD) in 1978 [1], the Axioms are applied to determine the technological soundness of a design. The initial number of seven axioms relatively soon [2] were brought back to a number of two, the Independence Axiom and the Information Axiom. These two axioms have been very successful; they are broadly applied for over 35 years since.

The Independence Axiom and the Information Axiom may be considered independent from each other [3], however, this is only the case within the definition of information as it applies for AD. ‘Information’ or ‘Entropy’ may be considered as chaos in design. Information in AD is derived from the information technology using a logarithmic measure of Boltzmann’s entropy according to Hartley [4] and Shannon & Weaver [5]. It states that information is inversely related to the probability of success. Probability is the central theme of AD around which the axioms are carefully wrapped. Knowledge is applied, in good accordance with the nature of the axioms, to maximise the probability of Design Parameters (DPs) satisfying Functional Requirements (FRs). Knowledge is therefore the most important enabler to address information and consecutively increase the probability of a design to function as expected.

Recently, a broader decomposition of information was introduced for AD, that starts with the Information as defined by Boltzmann, Hartley, and Shannon. The decomposition ends at the bottom with three kinds of information that directly influences a product or system design. One of them is ‘Axiomatic Information’, directly related to the Information Axiom defined by Suh [6]. Two other kinds of information are ‘Recognised’ and ‘Unrecognised information. Recognised information may be addressed by making the design independent. Unrecognised information on the other hand has more mystical traits; it is not known by the designer and as such it is difficult to address. This paper investigates how to deal with unrecognised information: 1. how it may be found and 2. when found, how to address it. Note, when unrecognised information is found, it instantly changes to recognised information. The research questions for this paper are:

- How can unrecognised information be found?
- How can recognised information be addressed?

This paper is organised as follows. Section 2 focuses on the background of information and complexity. Section 3 considers the concept of unrecognised information, how it can be found and how it may be addressed. Section 4 explains a number of

Peer-review under responsibility of the scientific committee of The 10th International Conference on Axiomatic Design
doi:10.1016/j.procir.2016.05.047

cases that deal with unrecognised information, and elaborates on the concept. Finally, Section 5 discusses the findings and conclusions are found in Section 6.

2. Background

2.1. Background on Information or Entropy in a Design

Information in Axiomatic Design is derived from the information theory using a measure of Boltzmann entropy according to Shannon & Weaver [5–7]. It uses the logarithmic representation as introduced by Hartley to make information additive instead of multiplicative [4]. According to the information theory, information is inversely related to the probability of success of a goal being met.

Suh describes three types of information in AD, 'Total' information, which consist of 'Useful' and 'Superfluous' information [6]. Useful information is information that affects FRs and their relations to the other domains. Superfluous information does not affect the relation of FRs and the other domains. Therefore, superfluous information is no information from the design perspective. Puik & Ceglarek decomposed information in the axiomatic context as shown in Figure 1 [3]:

- Total information; the total information content or full entropy of the design as defined by Suh [6];
- Useful information; the part of total information that affects the relation between FRs and DPs [6];
- Superfluous information; information that does not affect the relation between FRs and DPs [6];
- Axiomatic information; useful information due to a discrepancy in design ranges and system ranges as will lead to 'Real' complexity[8];
- Unorganised information; useful information that is not recognised as such due to ignorance of the designer [3];
- Unrecognised information; information of which the designer is not aware of yet [9];
- Recognised information; information of which the designer is aware of but is not addressed yet [9];

2.2. Background on Complexity in Axiomatic Design

Complexity in AD is defined as 'A measure of uncertainty in achieving the specified FRs' [10]. The Complexity Axiom advises to 'Reduce the complexity of a system'. The theory defines two kinds of complexity, 'Time-Independent' and 'Time-Dependent Complexity'. In the case of time-independent complexity, the behavior is governed by the given set of FR and DP relationships. Time-dependent complexity depends upon the initial condition with FR and DP relationships, but unless the system goes back to the same set of initial conditions periodically, the distant future behavior is totally unpredictable as the system tends to escalate [11]. Time-dependent complexity is not further investigated in this chapter.

Time-independent complexity consists of two components: 'Real' and 'Imaginary' time-independent complexity, further to be referred to as real complexity and imaginary complexity (C_R and C_{Im}). Real complexity is inversely related to the probability of success that the associated FRs are satisfied according to one of the following relations

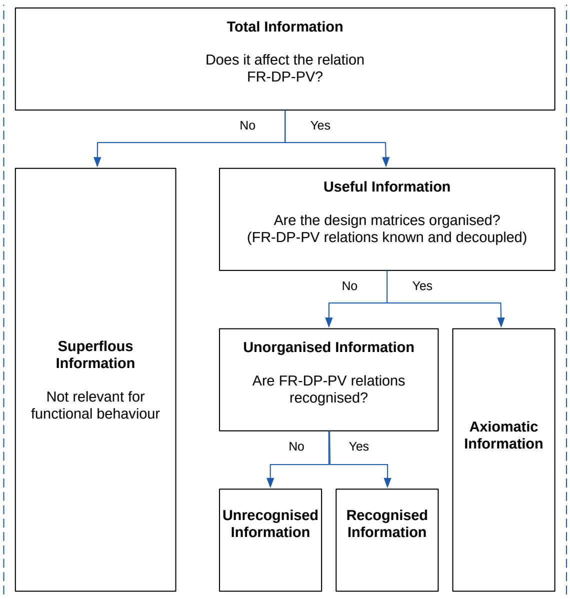


Fig. 1. Overview with types of information and their relations

$$C_R = -\sum_{i=1}^{m} \log_b P_i \tag{1}$$

$$C_R = -\sum_{i=1}^{m} \log_b P_{i|\{j\}} \qquad \text{for } \{j\} = \{1, 2, \ldots, i-1\} \tag{2}$$

depending if the system is uncoupled (Equation 1) or decoupled (Equation 2). Relation 1 is under the reservation that the total probability P_i is the 'joint probability of processes that are statistically independent'. Relation 2, for decoupled systems, is modified to correct for dependencies in the probabilistic function [10]. b Is in both cases the base of the logarithm, usually in bits or nats depending of the preferred definition. Given 1 and 2, real complexity can be related to the information content in AD, which was defined in terms of the probability of success of achieving the desired set of FRs[6], as

$$C_R = I \tag{3}$$

in which C_R is real complexity and I is information as defined in Section 2.1. Imaginary complexity is defined as complexity that exists due to 'a lack of understanding about the system design, system architecture or system behavior' [11]. It is caused by the absence of essential knowledge of the system. The designer cannot solve the problems in a structured manner and therefore is forced to apply trial-and-error. Imaginary complexity exists until understanding of the problem is acquired; it instantly and permanently disappears when the knowledge becomes present. Though the source of imaginary complexity, a trial-and-error process, can be stochastic, Suh never relates trial-and-error probabilities to information of any kind. The motive for this choice was not found in Suh's work; it is basically a matter of definition.

2.3. The Complexity Approach of the Cynefin Framework

Cynefin is a decision making framework that can be applied on organisations, systems, or even complex social environments [12]. It was applied, evaluated and refined at the IBM Institute of Knowledge Management [13] and later expanded to be used as a leadership model [14]. Cynefin has not yet gained much drag within the AD community or even product development in general, but with the view on information in AD as reported by Puik & Ceglarek [3,15], both methodologies appear to connect and harmonise well together.

The framework consists of three basic types of systems; 'Ordered' systems, 'Complex' systems and 'Chaotic' systems. Ordered systems are divided in to two types: 'Simple' ordered systems and 'Complicated' ordered systems. In the centre of the four contexts is a fifth field added: 'Disorder'. Together this leads to the Cynefin framework as shown in Figure 2.

- In the simple context, cause and effect relationships are clear, predictable, repeatable, and generally linear. The systems in this context are self-evident to every reasonable person. The decision model of the simple context is sense-categorise-respond. Good response in these situations would be to watch what is coming in, match it to previously determined categories and decide what to do. The simple context is the context of 'best practice';
- In the complicated context, there is a logical relation between cause and effect, but it is not self-evident and therefore requires expertise. An analytical method is needed to solve problems, or an expert could be called in. The decision model therefore is sense-analyse-respond. The complicated context is the context of 'good practice';
- A complex system is a system without causality. Cause and effect are only obvious in hindsight, with unpredictable emergent outcomes. The decision model is probe-sense-respond. Carrying out experiments is a key characteristic; a successful outcome is enhanced, a bad outcome is suppressed. Actions lead to a novel way of doing things. The complex context is the context of 'emergent practice';
- A chaotic system shows no relation between cause and effect. The goal should be to restore order. The decision model therefore is to act-sense-respond. Actions will be new and unconventional. This is the context of 'novel practice';
- Disorder is the space when it is not clear to which context a situation should be appointed.

The boundaries between the contexts are transitions that can be taken without specific effects, except for the boundary between the simple context and the chaotic context. This boundary is referred to as the 'Complacent Zone' or the 'Cliff'. The danger is that once a system is in the simple context, people start to believe that things are simple by nature. It may lead to the belief that things are always ordered and that success from the past is proof that systems cannot fail. The result is that the actual position moves to the border and at a given moment falls over the cliff into a crisis.

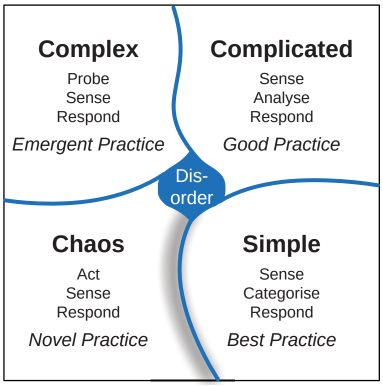


Fig. 2. The four contexts of the Cynefin framework. When in disorder, the actual context is not known

2.4. Synergy between Axiomatic Design and the Cynefin Framework

Although originally from a completely different background, AD and the Cynefin framework have a number of similarities. First, for both methodologies, knowledge is enabling for the determination of the status of a system. Secondly, both methods deal with the level of organisation in systems or contexts. AD, as was shown in figure 1, has unorganised information and axiomatic information, the latter dealing with an organised design matrix and therefore also to be considered as 'organised' information [3].

3. Considerations on Unrecognised Information

3.1. Characteristics of Unrecognised Information

Unrecognised information is a state of chaos in the design that remains unnoticed by the designer. Though the designer is unaware of the presence of information in the design, his design contains true information and the FRs may not be satisfied. Unrecognised information is usually the result of a lack of knowledge of the designer. Typically, this is caused by a DP that is not recognised. All recognised DPs, that have an effect on a relating FR, are normally 'Fixed' by the designer; the DP is set at a known value so it will not influence the FR during development and use of the product. As a result of the unfamiliarity with a DP, also called a hidden DP, it will not be fixed by the designer. When its value changes in a later stage, it may dissatisfy the FR and a problem occurs. Though it is possible that unrecognised information remains in the design because the designer was negligent to properly investigate the system, this is usually not the case. Unrecognised information stays hidden when the hidden DP stays within acceptable margins and thereby does not (or moderately) interfere with the functionality of the design. Therefore, a design may function quite

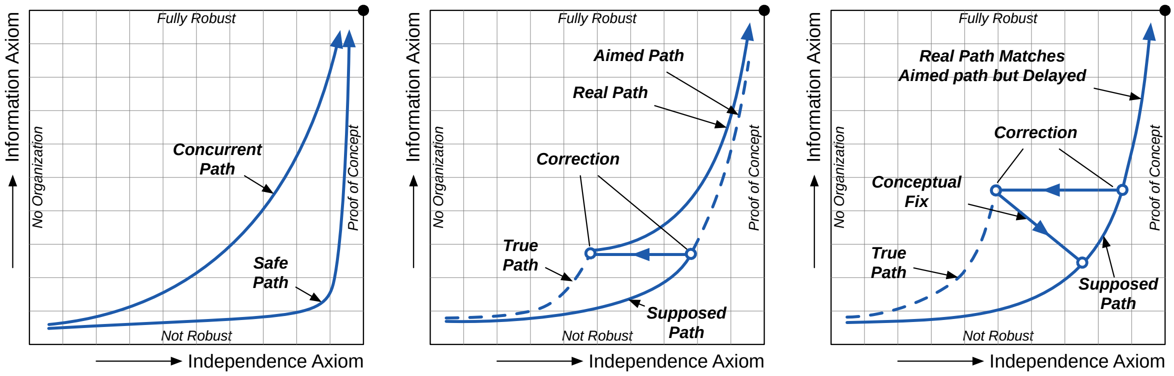


Fig. 3. The Axiomatic Maturity Diagram plots the way a product matures

well with unrecognised information. At the moment that the DP starts changing, due to some influence that puts the change into effect, the functionality of the system will be compromised. This will normally come as a complete surprise to the designer as he has no notice at all about the cause of the problem.

What happens when unrecognised information reveals itself can be visualised by the Axiomatic Maturity Diagram (AMD) [16]. The AMD plots the axioms on both axes of a diagram to monitor progression as the design matures. The Independence Axiom is plotted on the horizontal axis and the Information Axiom is plotted on the vertical axis. The development line follows the path according to the satisfaction of the axioms in the design. A safe development path through the AMD would first go to the right side of the diagram and then bend upwards and move to the dot that indicates a mature design (Figure 3 left). A concurrent way of development would cut the lower right corner and could gain development time at the cost of increased risk [9]. When unrecognised information is found, it will lead to a discontinuity in the development path because a correction from the path as supposed by the designer to the true development path is carried through. The satisfaction of the Independence Axiom could be set back when this happens (Figure 3 center), or both axioms could be set back when a conceptual fix needs to be carried through that affects robustness (Figure 3 right). In the first situation the project will not continue as intended but will in reality follow a new path to the upper right corner of the AMD. In the second situation the conceptual fix brings the situation back to the initially aimed development path.

Finally, a remarkable characteristic of unrecognised information is that it only exits in its hidden state. It instantly changes to recognised information when it is discovered. In the new presence of recognised information, the designer can address it by completing and decoupling the design matrix.

3.2. How to Find Unrecognised Information

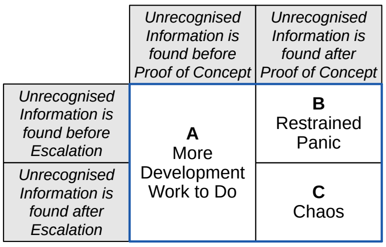
Finding unrecognised information is a key challenge for product designers and there is no method that comprehensively enables this. The product design may be seen as a complex system which indicates it is not completely understood. According to Shannon and Weaver, a system that is not understood may introduce features with a stochastic nature. The stochastic nature is caused by missing structure of the design that is a requisite to satisfy the independence Axiom. Gell-Mann & Lloyd call this missing structure a ‘lack of regularities in the system’. The lack of regularities increase entropy in the system and ‘the smaller the entropy, the less spread there will be among the entities that follow these regularities’ [17, p. 50]. A lack of regularities in the design will increase its chaotic behavior and thus increase information. The definition of well-chosen FRs, the process of selecting matching DPs, decoupling the relations between FRs and DPs, making sure that all DPs are relevant, and ensuring that all relevant DPs are known, are all regularities that contribute to a more predictable behavior of the design and hence they eliminate information from the design.

As stated at the beginning of this paragraph, there is no method that comprehensively enables or guarantees the detection of missing regularities in the design. However, there are two ways to increase the chance that missing regularities are found. A product or system will not behave stochastically if all regularities needed to satisfy the FRs are known:

- The first way to deal with missing regularities in design is found in the application of the Axioms, in particular the Independence Axiom. If the set of FRs is truly complete (the FRs are ‘CEME’; collectively exhaustive and mutually exclusive [18]) and an exhaustive search is done for all DPs that may influence the FRs, the chances that the unrecognized DPs are found is maximal;

- The second way to increase the chance to find missing regularities is described by Kurtz and Snowden [13]. The decision model they propose is to create ‘probes’ to make potential patterns more visible before taking action. These patterns then can be sensed and responded to e.g., by stabilizing the patterns that are desirable and by destabilizing those that are not. Positive patterns may be seeded to make them more likely to emerge. In the world of the designer, this approach is very related to physically testing the system. Testing is extremely successful in finding missing regularities because ‘people get fooled, but not nature’ [19]

Summarising, it is not possible to guarantee that all regularities of a system are found but the chance of finding missing irregularities is maximised when 1. an exhaustive search for FRs and DPs is performed, and/or 2. the system is exhaustively tested.



	Unrecognised Information is found before Proof of Concept	*Unrecognised Information is found after Proof of Concept*
Unrecognised Information is found before Escalation	**A** More Development Work to Do	**B** Restrained Panic
Unrecognised Information is found after Escalation		**C** Chaos

Fig. 4. The options how to act when unrecognised information is found

3.3. What to Do if Unrecognised Information is Found

As explained in §3.1, unrecognised information may remain hidden in the design till the point that some property puts the change into effect. From that moment the problem may start escalating. It is difficult to respond adequately to an escalating problem because no solution was foreseen to address it. Other solutions that concern the same FR may work partially or may not work at all. Another essential point of disengagement to learn about unrecognised information in a design is the stadium 'Proof of Concept'. If unrecognised information is found before this point, before the design matrix is officially found to be decoupled, the impact can be overseen. At this stage, it is 'just' another problem to address. The design team is at full strength to deal with this kind of difficulties. Different is the situation where the stadium proof of concept is passed. Detection of unrecognised information sets the project back in the conceptual phase, no matter if it is detected in the robustness stage or after the release of the product in the field. The product has a conceptual weakness, some of its design features are not fully regulated and will behave stochastically. It will lead to inexplicable behaviour of the product or system. Figure 4 shows the options that apply when unrecognised information is found.

When unrecognized information is found within the conceptual development stage, represented by situation 'A', it means more work but this is unlikely to be disastrous; the design team is equipped with knowledge and tools to solve problems of this kind and there are still many problems to solve. From the perspective of the Cynefin framework of Figure 2, this means that the transition from Complex to the Complicated context will require extra efforts. Delay of the project is possible if the unrecognised information is of serious order.

When unrecognised information is found after the proof of concept, the impact is more severe. If it is shortly after the transition to proof of concept, the project status may be set back to the conceptual stage. If the project already was released, the problems are substantially bigger. Now, two options apply: 'B' shows situation where unrecognised information is found, but it is not escalating because the hidden DP is still within reasonable margin of its setpoint. There will be time to develop a solution. A kind of 'Restrained Panic' will emerge because the problem may escalate any time. The development of a solution will need the reassembly of a design team that has the ability and is given the time to develop a solution. From the perspective of the Cynefin framework of Figure 2 it means that a jump is made from the Simple domain to the Complex domain. When a solution is found, the design moves clockwise through the Complicated context back to the Simple context. The other option is marked with 'C'. The hidden DP has changed and cannot be restored. Now the problems get troublesome because functionality of the product is compromised and there is no quick fix. From the perspective of the Cynefin framework, a conceptual weakness without the knowledge to solve the problem pushes a project into the complacent zone and over the cliff, straight into the chaotic context. The only way to get out is to make the full circle clockwise through all contexts; restore order, eventually with drastic measures, and find cause and effect by reassembly of the design team. From this point, make things analysable and categorisable again.

4. Two Cases that Deal with Unrecognised Information

Two cases are descried to support the previous theory. Situation 'A', in which the project is still in the conceptual phase will not be explained in more detail since presence and sudden appearance of unrecognised information is usual fare in this phase. The first case describes a situation in which the problem does not directly escalate. The second case is about a problem where unrecognised information escalated right away when it revealed itself.

4.1. An On-Line Payment Application

4.1.1. General

The first case concerns a Dutch company that delivers solutions for on-line payments. This store management system combines online payments, store payments, and integrates stock keeping of stores and warehouses in a single solution. It is a complex multi-mainframe system with many interfaces. The system may be seen as the core around which the operations of many stores are organised. If the system goes down, no payments can be done in both physical and on-line stores.

Because of the importance for the operations to the customers, the company gives an up-time guarantee with penalty clause. Maintenance of the system is done at certain nights of the week when all stores are closed. Regular updates take place to add features and to correct malfunctions. Backups are made to secure data. All this is done over the internet from a single location in the Netherlands over several thousands of cash registers in Europe. Security is a significant issue; many attempts to hack the system take place. The company also hires professional hackers to test the system for vulnerabilities. Since all systems are connected to the internet there are many interfaces and even more ports to approach the system.

4.1.2. The problem

At a given day, the ICT manager of the company is hinted by the one of the professional hackers that a certain interface gives access to the system because a port is opened. The manager has this problem examined by the team and they confirm that the port is opened. This is a necessity to ad certain functionality to the system. However, the team is convinced that this vulnerability is not of a worrisome nature. Some days later, the manager is not quite comfortable with the situation and in the evening he tries to get access to the system from his home location. To his surprise, he is able to gain access without any password and he is also able to start and stop processes on the mainframes and even worse, he is able to execute ghost payments.

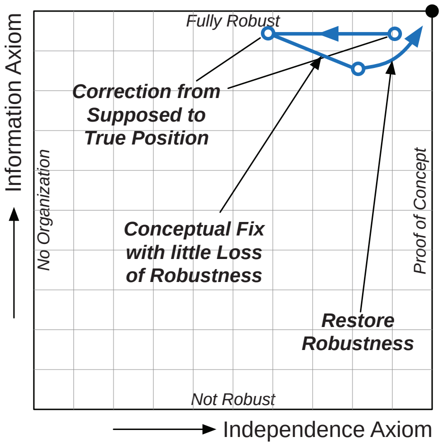


Fig. 5. When unrecognised information is found, a correction from the supposed to the true position becomes clear to the designer

4.1.3. The Consequence

The same evening, the manager reports the problem to the general manager. That same night they try to reconstruct the origin of the problem. They conclude that the vulnerability has been there for over six months. Next morning, a crash team is composed. A risk analysis indicates a severe problem. A few thousand systems are in the field with the same vulnerability.

4.1.4. The Solution

The problem can be fixed; the team has to reroute a number of communication channels to restore the vulnerability. After some long days, a fix is completed. It is implemented on a test system and tested for a week. After this it is rolled out to a limited number of systems before it is rolled out completely.

4.1.5. Elaboration from the Perspective of Information

In the beginning of this evaluation the vulnerability already is in the system. There is peace in the company because no one is aware of the problem. But this calm is unfounded; The system may be terribly hacked any moment with the result that the system can be halted or fake bank transfers take place. All this is caused by the presence of unrecognised information in the system.

Once discovered, the calmness in the company gives way to the ‘Restrained Panic’ of the knowledge that anything may go wrong any moment. This is visualised in Figure 5 with a drop in the AMD. After this, the engineers concurrently develop a conceptual fix. The fix needs changes in the software design which may reduce the robustness of the system. Robustness is regained by testing the system again. The end position in the AMD is comparable to the supposed end position before learning about the unrecognised information but it is more mature than the true starting point.

In the Cynefin framework, the situation moves from the Simple context directly to the Complex context. There is no state of chaos in the company, but all engineers feel the pressure to understand the situation and come up with the solution. Since it is a complex system, they need time to find that solution. Once rolling out the system starts, the company comes at ease and moves via the Complicated context back to Simple.

4.2. Case 2 De Havilland Comet

4.2.1. General

The second case is a case from the history books. It concerns the De Havilland Comet. Extensive research has been done to find the cause and effect of this case [20,21]. The Comet was the world’s first production commercial jetliner developed and manufactured by de Havilland. It featured an aerodynamically clean design with four turbojet engines buried in the wings, a pressurised fuselage, and large square windows. The plane was a gigantic step forward in avionics, with cruising speeds up to 800 km h^{-1} and cruising altitudes of over 13 000 m.

4.2.2. The problem

In 1954 two de Havilland Comets broke up in flight with no apparent reason known at that time. Because of this, the plane was grounded.

4.2.3. The Consequence

Investigations were needed to find the problem that caused the two crashes in 1954 and this appeared not easy. The planes were put in a water basin to test the integrity of the fuselage by pressurising it. After a number of load changes it ruptured. Further investigation learned that fatigue cracks starting at the pivots of the square windows and hatches led to accelerated growth of cracks. When the cracks became too large, the fuselage ruptured, starting at the forward escape hatch and the top hatches (Figure 6).

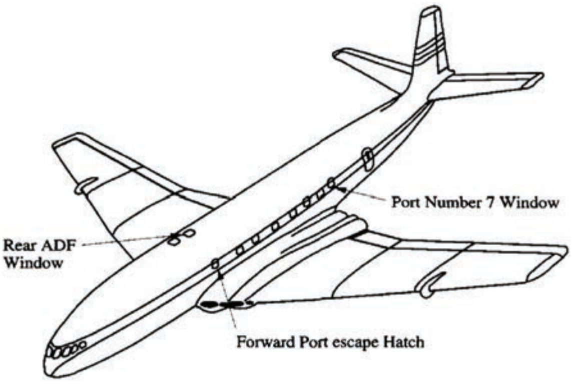


Fig. 6. Cracks started at the escape hatches and the windows that both had a square shape and were pivoted [20]

4.2.4. The Solution

The problem was solved by improving the pivots and the shape of the hatches and the windows, but it took till 1958 before commercial flights resumed.

4.2.5. Elaboration from the Perspective of Information

At the beginning of the design, this plane already suffered from the weakness that the shape of the square hatches led to tension concentrations in the metal. Pivots weakened the fuselage further at the locations with high tension. The wall of the fuselage was relatively thin to save weight and the combination of these factors led to the presence of unrecognised information. When this information came to the surface the problems were

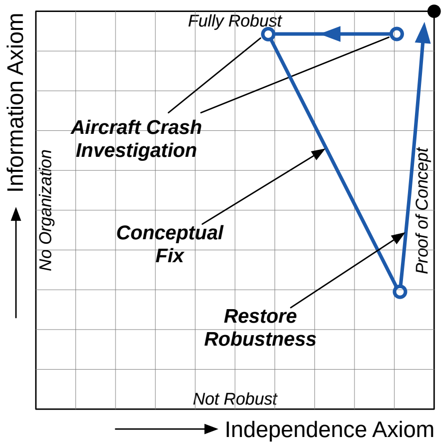


Fig. 7. In this situation a safe development path is followed, characterised by a steep incline during the restoration of robustness

difficult to oversee; not only many lives were lost but also the confidence in the safety of the plane disappeared. Enthusiasm about a great plane made place for total chaos.

The aircraft crash investigation that followed revealed the true position in the AMD (Figure 7). The fatigue problem shortened the lifespan of the plane, and that FR was no longer satisfied. Substantial conceptual improvements were needed which resulted in a significant drop in robustness. The improvements restored the independence of the system up to a high extent. The repair cycle ended with restoring the robustness by repeating the many tests that are required to get the necessary permissions to resume service.

In the Cynefin framework, the situation moves from the Simple context to the complacent zone and falls over the cliff straight into chaos. When the fuselage tests were completed the relation between cause and effect was restored. Based on that understanding a new start could be made by the De Havilland Comet. That restart was successful from the technological perspective as the fuselages remained intact from that moment.

5. Discussion

Ignorance indeed is bliss, in any case till unrecognised information presents itself. Finding unrecognised information in a design, before it is released to the market, is probably the biggest challenge for the designer. Especially if a design has many totally new design solutions, it is difficult to be sure that all unrecognised information is found. Two methods are given in this paper to address unrecognised information; exhaustive modelling the design is the first and extensive testing is the other. Methods to address information have been presented when unrecognised information is found. The Axiomatic Maturity Diagram can be applied to determine the right response when a designer is confronted with unrecognised information.

5.1. Strengths of this Approach to Deal with Unrecognised Information

The two methods to find unrecognised information are not new. Exhaustive understanding is the basis of AD. The CEME method for defining FRs, collectively exhaustive and mutually exclusive, is well known and generally applied within the AD community. However, this investigation learns that it is also needed to exhaustively chart all possible DPs that are associated to the FRs in case. When DPs are not properly fixed, they may start drifting at some point and cause problems in time.

The Axiomatic Maturity Diagram is still relatively new but it turns out to be a practical tool for the analysis of problems in the conceptual phase, because it also shows the impact on the robustness phase and warns the designer for an eventual loss of design efforts. Its visual character enables sharing thoughts between designers of different disciplines or managers and technicians. The analysis makes insistently clear what the results are when unrecognised information is ignored or sloppy efforts of the designer make him overlook it.

5.2. Weaknesses and Limitations of this Approach to Deal with Unrecognised Information

The two methods to find unrecognised information, even when applied in the most conscientious manner, give the designer no certainty that his search was exhaustive. Both methods are costly; exhaustive modelling is time consuming for skilled engineers and testing requires realisation of prototypes or test setups that claims resources and investments. The Axiomatic Maturity Diagram is not yet generally known and could benefit from more exposure to give it a low threshold for application.

A limitation is that a designer is never certain that unrecognised information is properly addressed. Some may reside in the product design and it cannot be predicted if, and when unrecognised information rears its ugly head.

5.3. Opportunities for Further Research

An opportunity could be to apply an adapted version of the method of inventive problem solving (TRIZ) [22]. As is, the method is able to contribute to the synthesis of solution concepts. It could be modified that it finds potential risks when inventive principles are applied. A quick literature scan learns that some work has been done in this direction e.g., Regazzoni & Russo present an improved risk management model for product and system design to reduce failure occurrence based on TRIZ and FMEA [23]. Teoh & Case published a knowledge modelling procedure based on FMEA that is particularly suitable for automation [24].

6. Conclusions

Unrecognised information is a intrinsic problem to product development. This paper presents two methods to find unrecognised information during the product design process. Secondly, the paper presents a way to visualise the effects when unrecognised information is found. Basically, the appropriate way to deal with unrecognised information is to go back to the conceptual phase of the product design, because this is where this kind of information intervenes with the product design. As a consequence, it may be the case that the robustness of the design is affected when unrecognised information is addressed.

Acknowledgements

This research was supported by the HU University of Applied Sciences Utrecht, and the project 'AEROBIC', funded by Stichting Innovatie Alliantie (SIA) in the Netherlands.

References

[1] Suh NP, Bell A, Gossard D. On an axiomatic approach to manufacturing and manufacturing systems. Journal of engineering for Industry. 1978;100(2):127–130.

[2] Suh NP, Bell AC, Wilson DR, Rinderle J. Exploratory study of constraints on design by functional requirements and manufacturing. NASA STI/Recon Technical Report N. 1981;82:16304.

[3] Puik ECN, Ceglarek D. A Review on Information in Design. In: Proceedings of College International Pour La Recherche en Productique (CIRP), 8th International Conference on Axiomatic Design (ICAD). vol. 33. Lisbon, Portugal: Elsevier ScienceDirect; 2014. p. 1–6.

[4] Hartley RVL. Transmission of Information. Bell Labs Technical Journal. 1928;7(3):535–563.

[5] Shannon CE, Weaver W. The mathematical theory of communication. University of Illinois press; 1949.

[6] Suh NP. The Principles of Design. Oxford University Press; 1990.

[7] Brillouin L, Gottschalk CM. Science and Information Theory. Physics Today. 1962;15(11):68.

[8] Suh NP. Complexity in engineering. CIRP Annals—Manufacturing Technology. 2005;54(2):46–63.

[9] Puik ECN, Ceglarek D. Axiomatic Design in Large Systems: Complex Products, Buildings Manufacturing Systems. 2016;1:Chapter 4. Edited by Suh, NP. and Farid, A., Springer Verlag.

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

[11] Suh NP. A Theory of Complexity, Periodicity and the Design Axioms. Research in Engineering Design. 1999;11(2):116–132.

[12] Snowden D. Cynefin: a sense of time and space, the social ecology of knowledge management; 2000.

[13] Kurtz CF, Snowden DJ. The new dynamics of strategy: Sense-making in a complex and complicated world. IBM systems journal. 2003;42(3):462–483.

[14] Snowden DJ, Boone ME. A leader's framework for decision making. Harvard Business Review. 2007;85(11):68.

[15] Puik ECN, 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 College International Pour La Recherche en Productique (CIRP), 9th International Conference on Axiomatic Design (ICAD). vol. 34. Procedia CIRP. Florence, Italy: Elsevier ScienceDirect; 2015. p. 19–24.

[16] Puik ECN, Ceglarek D. A Theory of Maturity. In: Proceedings of College International Pour La Recherche en Productique (CIRP), 8th International Conference on Axiomatic Design (ICAD). vol. 33. Lisbon, Portugal: Elsevier ScienceDirect; 2014. p. 1–6.

[17] Gell-Mann M, Lloyd S. Information measures, effective complexity, and total information. Complexity. 1996;2(1):44–52.

[18] Brown CA. Teaching axiomatic design to engineers—Theory, applications, and software. Journal of Manufacturing Systems. 2005;24(3):186–195.

[19] Puik ECN. The Future is Random, but it May Be Conditioned: Managing Risks During Product Development and Manufacturing Engineering of More than Moore Systems [Ph.D. Dissertation]. University of Warwick. UK; 2016.

[20] Withey P. Fatigue failure of the de Havilland Comet I. Engineering failure analysis. 1997;4(2):147–154.

[21] Wanhill RJH, Molent L, Barter S. Milestone Case Histories in Aircraft Structural Integrity. In: Reference Module in Materials Science and Materials Engineering. vol. 1. Elsevier; 2016. p. 1–9. Prepared as a chapter in Volume 1: Structural Integrity Assessment - Examples and Case Studies, of the Elsevier Science treatise "Comprehensive Structural Integrity".

[22] Barry K, Domb E, Slocum MS. What is TRIZ? The TRIZ Journal. 2016;Available from: `http://www.triz-journal.com/triz-what-is-triz/`.

[23] Regazzoni D, Russo D. TRIZ tools to enhance risk management. In: Cavallucci D, Vaneker T, Cascini G, Duflou J, editors. Proceeding of the ETRIA World TRIZ Future Conference. vol. 9; 2011. p. 40–51.

[24] Teoh PC, Case K. Failure modes and effects analysis through knowledge modelling. Journal of Materials Processing Technology. 2004;153:253–260.

[25] Proceedings of College International Pour La Recherche en Productique (CIRP), 8th International Conference on Axiomatic Design (ICAD). vol. 33. Lisbon, Portugal: Elsevier ScienceDirect; 2015.