The Complexity of Mapping Customer Needs...

(and the Myth of a Unanimous Customer)

Danielle M. Vossebeld[1,3,*], *Joseph T.* Foley[2], and *Erik* Puik[3]

[1] *University Medical Center Utrecht, Heidelberglaan 100, 3584 CX Utrecht, The Netherlands*
[2] *Reykjavik University, Menntavegur 1, Reykjavik 101, Iceland*
[3] *HU University of Applied Sciences Utrecht, Padualaan 99, 3584 CH Utrecht, The Netherlands*

Abstract. Axiomatic Design and Complexity Theory as described by Suh focus heavily on the coupling often found in functional requirements. This is so fundamental to the analysis of the design that it is the core of the Axiom of Independence which examines the coupling between functional requirements due to chosen design parameters. That said, the mapping between customer needs and functional requirements is often overlooked. In this paper we consider coupling, found due to this mapping, as a possible source of complexity in terms of a user interface to a designed product. We also re-examine the methodology of how customer needs are generated and translated into the other domains to understand how they can give further insight into the customer mindset. Based on this analysis, we believe customer domain complexity should always be examined in design that includes end-user interaction.

1 Introduction

At the initial phase of design, the voice of the customer (VOC) must be transformed into a list of specifications that can then be rendered into a final product. In Axiomatic Design, this is the process of generating customer needs (CN). CNs are mapped to functional requirements (FR), and usually ending with a list of design parameters (DP). This process is inherently complex[1], challenging, and multi-disciplinary, requiring experts in multiple fields to result in repeated success. A great deal of effort has been spent in the technical fields on the process of mapping FRs to DPs, analyzing their interactions, and rendering a solution. However, complexity in the customer needs themselves has not been closely examined. This is an oversight because any vagueness or contradiction inherent in the mapping to FRs is likely to result in challenges to the design.

Before complexity of customer needs and how this might result in complexity elsewhere is examined, the basics of Axiomatic Design and Product Design are explained. This paper is organized as follows: Section 2 outlines and summarizes the basic processes involved in Axiomatic Design, Complexity Theory, and generation of customer needs. Section 3 analyses the cases from the perspective of complexity. Finally, Section 4 discusses the findings and summarizes the findings.

---

*e-mail: D.M.Vossebeld@umcutrecht.nl

[1] In the traditional Suh [1] sense, which is the inverse probability of meeting the goal

2 Background

To better understand the interplay between customers and requirements, some relevant elements of Axiomatic Design theory are explained.

2.1 Axiomatic Design theory

Axiomatic Design theory is a general-purpose solution-agnostic design methodology developed by Nam P. Suh at MIT in the mid-1980's [3]. More recently, the theory has been generalized to incorporate more modern concepts of 'complexity' into its structure, leading to Suh to develop Complexity Theory [1]. In both cases, the basic premise for evaluating whether a design is good or not is according to two axioms [3, p.47]:

Independence Axiom: Maintain the independence of functional requirements (FRs).

Information Axiom: Minimize the information content of the design.

These axioms are in the order of importance: one should not consider the information content, i.e., the unreliability of meeting requirements, without first considering the coupling between the FRs. Traditionally, functional requirements and design parameters (DPs) are gathered in domains (See Figure 1). A domain is a set of elements bound by a specific characteristic, e.g., 'Functionality' for the FRs and 'Physical Properties' for the DPs. The domains are connected by a mapping process, relating FRs and DPs, structuring how functionality aspects are realized with physical design solutions. A third domain, called the
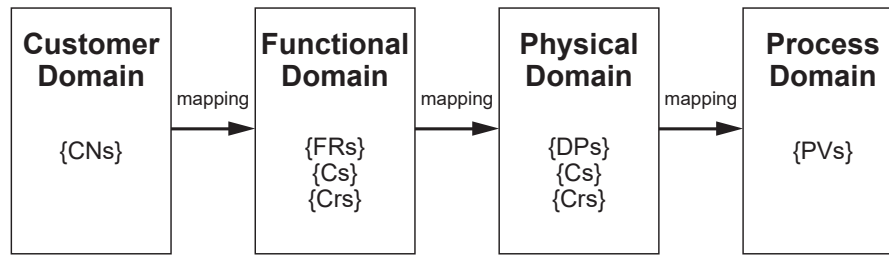
Figure 1: AD Domains and mapping. Items in braces are the lists generated during evaluation of the domain with zig-zagging. Based upon [2, p.11]

'process domain', comparably relates 'process variables' (PVs) to DPs. Domains are not automatically structured and may be in a state of illogical presentation. The understanding of the domains by the designer benefits from a clear organization. Therefore, domains are hierarchically decomposed into parallel lists following the process of zig-zagging.

Zig-zagging refers to the process of structured decomposition between parallel trees of the different domains [2]. The process usually starts with the functional domain. Then the elements in that domain are enumerated with a unique sequential identifier. For a Functional Requirement, this might be FR_1 or FR1. This list is then mapped to a list of elements in the next domain in the sequence. These elements are again enumerated with an identifier for the domain, e.g. DP_1 or DP1. Once this enumeration is complete, the process jumps to the functional domain to decompose each list element into smaller parts as needed. Girgenti [4] pleads for a slightly different procedure; instead of continuing the process of zig-zagging to the next level of decomposition, he first verifies his findings at the same level by reversing the zig-zagging process. He calls this process 'zig-zagging conversely'. The goal is "forcing to self-question whether the problem has been completely dissected or something better and more connected with real customer needs could be found". This verification is usually executed after decoupling of the system and is then called 'Reverse Zig-zagging' [5, 6]. It means that an iterative verification takes place to secure the findings before zig-zagging continues to the next-lower hierarchical level (by self-questioning).

The process of decomposition, with or without verification, should be Collectively Exhaustive and Mutually Exclusive (CEME), a mnemonic developed by Brown [7]. As such, decomposition advances from the higher to the lower level domains. All decomposed elements have IDs that explain the child-parent relationship: $FR_{1.1}$ or FR11 which is mapped to $DP_{1.1}$ or DP11. These are both children of FR_1 and DP_1, respectively. Following the process, the examination of the domains evokes the concept of a 'zig-zag'. The most common zig-zag decomposition is seen between the functional domain (FRs) and physical domain (DPs), eventually process domain (PV) is included.

Inter-domain coupling is evaluated by building a Design matrix: a Cartesian product-transfer-matrix of all the listed elements in the two domains associated with them [8, 9]. Traditionally, this is primarily done with FRs and DPs, but the same can be done with DPs and PVs.

After rearranging the lists to best fit a triangular matrix in traditional linear-algebraic transformations, the design can determine if the design is:

coupled (non-triangular matrix), in which case finding a generalized solution is difficult,

de-coupled or path-dependent (triangular matrix), for which the order in which elements are designed matters,

uncoupled (diagonal matrix) indicating that changes are localized between a single FR-DP pair so the design can be done in whatever order desired.

2.2 Complexity Theory

Axiomatic Design theory was further extended by Suh [1] in order to focus on 'relative' understanding of the system, analogue to the information concept brought up in the Information Axiom. He explores the meaning of complexity, finally settling on "Complexity is defined as a measure of uncertainty in achieving the specified FRs." [1, p.58] This definition is then subdivided into four different types:

time-independent real complexity which is simply the information content of a design: $C_R = I$, where $I_i = -\log_2 P_i$ and P_i is the probability of meeting satisfying FR_i.

time-independent imaginary complexity arises in coupled or path-dependent solutions where the order in which DPs should be addressed is unclear or improperly ordered.

time-dependent combinatorial complexity develops in systems in which operation has a higher probability of going out of specification due to "continued expansion in the number of possible combinations with time". In short, it is systems that progress toward chaotic states over some time period.

time-dependent periodic complexity is similar to combinatorial complexity except that a functional period has been identified over which the system can be reset before it enters an unpredictable state.

These four categories of complexity provide a generalization and quantification of the predicted reliability of a design based upon the operating specifications and analysis of coupling during the design phase. Suh also provides guidelines for reducing these types of complexity: see [1] for a in-depth discussion of this topic.

2.3 Customer Needs

The customer domain is characterized by the needs (or attributes) that the customer is looking for in a product or process or systems or materials. Examples are given for CAs[2] in various disciplines[2, p.12]:

Manufacturing Attributes that customers desire

Materials Desired performance

Software Attributes desired in the software

Organizations Customer satisfaction

Systems Attributes desired of the overall system

Business ROI

Suh [2, p.14] recognizes that finding and developing customer attributes is challenging. He recommends considering quality function deployment [10] and working with customers. He gives the tagline: "The rule is to ask the right questions to the right customers at the right time." Kurniawan et al. [11] also suggest employing: Voice of Customer, Conjoint Analysis, Kano Diagram, Kansei Engineering, and Web-based consumer elicitation as methods to bring out customer needs (or attributes). Once CA lists are developed, Pugh matrix [12], analytical hierarchy process (AHP) [13, 14], Taguchi loss functions [15], and the weighted sum of product attributes are methods for ranking and sub-selecting from the CAs.

Ulrich [16] gives a structured approach for developing these lists of customer needs: an entire chapter of the book focuses on this subject. The overall process is [16, p.75]:

1. Gather raw data from customers

2. Interpret the raw data in terms of customer needs. Organize the needs into a hierarchy of primary, secondary, and (if necessary) tertiary needs.

3. Establish the relative importance of the needs.

4. Reflect on the results and the process.

Gathering data is accomplished through interviews, focus groups, or observing the product in use. A key step during this is to identify and choose the correct customers to focus the more high-effort data collection activities. Helpful questions during interviewing potential end users are:

- When and why do you use this type of product?

- Walk us through a typical session using the product.

- What do you like about existing products?

- What do you dislike about the existing products?

- What issues do you consider when purchasing the product?

- What improvements would you make to the product?

In addition to normal needs, Ulrich [16, p.79–80] emphasizes the need to stay flexible and observe all aspects. Some needs can be identified through interaction and nonverbal expression. In particular, designers should be looking for 'latent needs' which are 'neither fulfilled nor commonly articulated and understood', which is also stressed by Hintersteiner [18]. If these needs can be met, they are excellent product differentiators from competitors in the market.

After collection, the CNs are interpreted by a marketer, designer or engineer. In an example of a screwdriver, Ulrich [16] shows a DP mentioned by a user, namely 'pistol grip', being interpreted as a CN 'comfortable to grip'. This is a combination of an FR of 'holding the screwdriver' and the criterion of 'comfort'. In this case, this criterion could be more extensive as comfort for holding the screwdriver also entails choosing the center of gravity of a tool. In this paper, the focus will be on the term customer needs (CN), as it focuses on the criticality of the request.

If the composition of all CNs in the customer domain is investigated, the essential elements of the system can be recognized, containing the whole range of combinations of FRs, DPs, PVs, nFRs, constraints, and criteria. Brown [19] also mentions the different kinds of elements in the CNs, leaving the sorting to the designer. Based on Ulrich [16], the CNs may be considered as a display of interpretations of typical characteristics of the system.

2.4 Summary

As previously mentioned, the 'customer domain' (Figure 1) is an unorganized source of knowledge about what requirements should be generated. The customer domain is characterized by the needs (or attributes) that the customer is looking for in a product, process, systems, or materials [2, p.10]. As such, the customer domain is a set of elements that the customer associates with a good design. Analogue to the other domains, the customer domain also may lack a clear structure, for the customer himself as well as the designer. Disorganization of the customer domain makes it more difficult to translate the customer's wishes into a clear list of FRs and leads to disorganization of the other domains. As such, any complexity in the customer domain may be forwarded to the other domains. This area that has not been studied in depth in the current literature and is of great interest to the authors.

To be explicit:

How is complexity, coming forth from disorganization in the CNs, forwarded to the functional-, physical-, and process-domains?
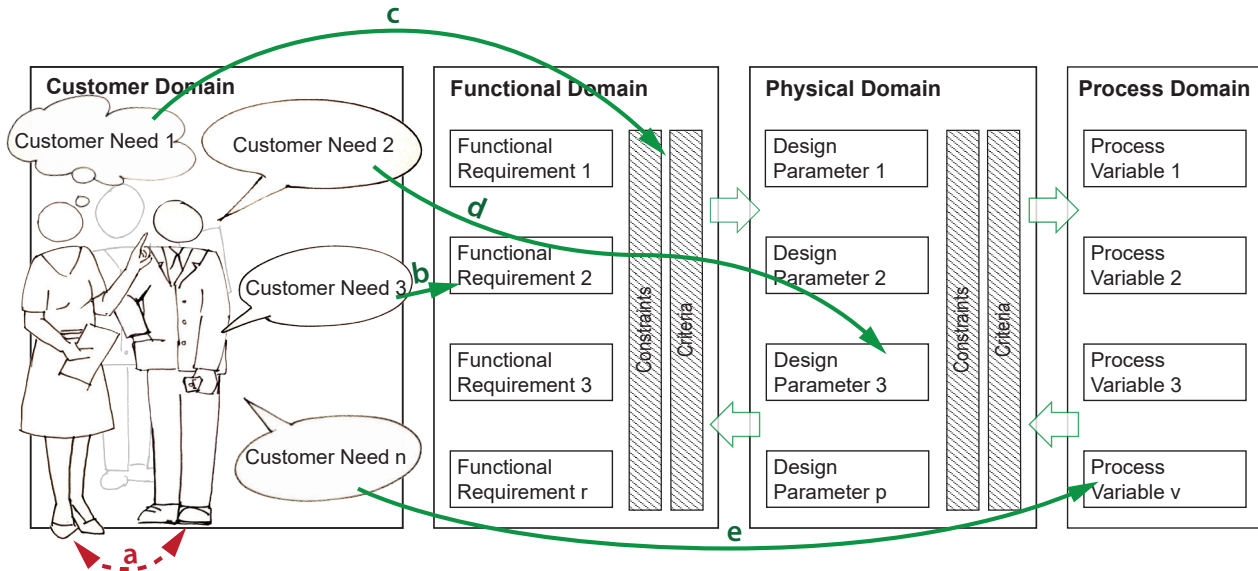
---

[2]Suh applies the term "customer attributes"

Figure 2: Interpreted CNs may add complexity other domains. (a) different stakeholders, (b) coupled FRs, (c) constraints, (d) coupled DPs, (e) limitations in PVs. Schematic based on [17].

.

3 Analysis of the complexity in the customer domain

To examine the effect of mapping CNs and the complexity, the nature of the customer domain and the CNs should be examined first. We explore existing literature on product or systems development using Axiomatic Design and also use examples of current research projects, to identify complexity and even the possibility of coupling.

Our presuppositions are that the complexity of effective mapping CNs is caused on five instances, which are annotated in Figure 2

a. Conflicting CNs of different stakeholders.

b. Coupled FRs are similar to (coupled) CNs.

c. Constraints and criteria in CNs cause coupling.

d. DPs in customer domain cause coupling.

e. Limitations due to PVs in customer domain.

These presuppositions will be applied as guidance to explore this subject.

3.1 Conflicting CNs of different stakeholders

The CNs in the customer domain contain (combinations of) FRs, DPs, PVs, nFRs, constraints, and criteria of different stakeholders. The term customer needs is applied for not just the needs of the customer who acquires a product, but also the needs of different parties within this customer as an organization or external parties influencing the needs of this customer. Thompson [20] refers to the CNs as SNs i.e. the stakeholder needs. The consequence of different parties or stakeholders is that they do not necessarily have

common needs and can have conflicting requirements according to Thompson [20] (see 'a' in Figure 2 and process in Figure 3). A customer who wants a product developed can have different goals and needs than a manufacturer that has to fabricate it.

Even within the organization that acts as a 'single customer', there can be conflicting needs. Weber et al. [21] refers to an automobile manufacturer with several internal groups with competing demands. He uses color coding in AD to identify the FRs of stakeholders or interest groups. Hintersteiner [18] mentions the well-known difference between the buyer and the user of a certain product. We also see this in our work. Within a hospital, for example, there are many involved parties for acquisition or design of (medical) equipment. Nurses want to register patient data at the bedside, to be able to spend more time on patient care and less on administrative tasks. A possible solution (DP), a computer on wheels (COW), can satisfy the need for registration at the bedside but was not
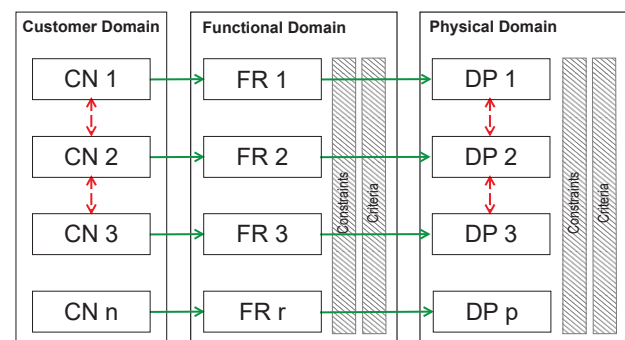

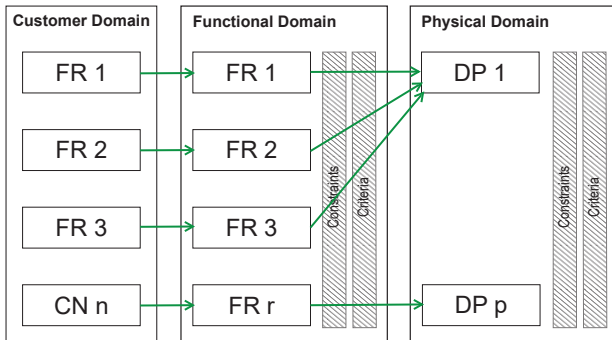Figure 3: Conflicting CNs of different stakeholders
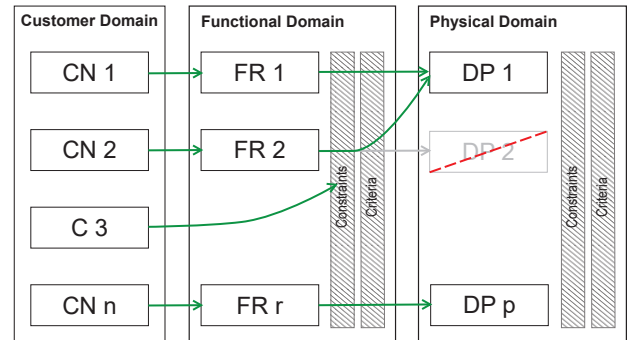
Figure 4: coupled FRs due to (coupled) CNs

Figure 5: coupling due to C or Cr in CN

allowed in the patient room due to hygiene rules imposed by the hygiene department who focused on infection control. Making the product easy to clean would allow the product in the room if is cleaned after each patient visit. This would take more time, which contradicts with one of the CNs of the nurses. The choice for COWs as a solution is among others based on cost (constraint or criterion) by management. The underlying CN for all parties, or in this case the hospital as one customer, is better patient care. The interpretation of this CN for the different internal stakeholders is conflicting. Due to these conflicts arising from effectively incompatible tolerances, it can be viewed as an instance of Real Complexity. If we also consider the possible changing state of stakeholders, this can also include some elements of Combinatorial Complexity.

3.2 Coupled FRs are similar to (coupled) CNs

Ulrich [16] mentions that CNs should be "largely independent from the product we might develop", thus independent from the DPs, by describing the CNs as what the product should do. This is very similar to Suh's guidance that "FRs should be solution independent" i.e. following the Independence Axiom. If these FRs are coupled with the choice of a single DP for two or more FRs [3], the CNs also may be considered to be coupled. As these CNs are directly mapped to the FRs, no coupling is shown between the customer domain and the functional domain (see 'b' in Figure 2 and process in Figure 4). The inherent coupling leads to Imaginary Complexity; there exists a lack of knowledge that prevents the CNs to be completely understood.

3.3 Constraints and criteria in CNs cause coupling

Suh [3] describes an example of the design of a refrigerator in his book. He emphasizes the choice of independent FRs, but also mentions how "constraints have a limiting effect on design". In the example, he uses the constraint of the maximum cost of a refrigerator door, which can cause coupling of FRs. Thompson [22] also mentions criteria, for ranking choices. A fixed maximum price can be a constraint, a criterion for choosing between alternative solutions can be 'lowest cost'. In both cases, not all DPs will

be acceptable as a solution (see 'c' in Figure 2 and process in Figure 5). Suh [3] mentions that the vertical hung refrigerator door "is *not* a good design" according to Axiom 1, due to the coupling of FRs caused by a constraint. Coupling due to more constraints is seen with the design of professional tools, like drills. The company Bosch [23] explains how drills should be made smaller and lighter to fit the build and strength of women. A constraint on the handle would be the size of hands or a maximum lifting power based on ergonomics. This would however also reduce the power of the tool, making the work more time-intensive to finish, which would not be acceptable for a professional tool. A maximum of time spent on a specific task could also be a constraint.

This is a mixture of Imaginary- and Real Complexity. The Imaginary Complexity is due to the confusion about what which needs are the ones to focus on and possible orderings that need to be considered; the FR 'powerful tool' causes coupling with the FR 'easy lifting/gripping'. The Real Complexity comes forth from mis-categorization during the translation process, allowing constraints to become FR and disrupting FR-DP relations.

3.4 DPs in customer domain cause coupling

Bragason et al. [24] describe one of the disadvantages of having an in-domain expert involved in the development process of CNs. Such experts already 'know how to solve the problem' without first going through the separation of function from implementation. The example given is for a non-explosive high-altitude rocket parachute release system developed by Bragason et al. [24] An interview with the rocket vendor interested in developing the project created a long list of CNs with a lot of technical specifications. As a thought experiment, these CNs were translated directly into FRs and an iteration of design completed (see 'd' in Figure 2 and process in Figure 6). It was quickly discovered that the significant coupling that arose in the FR-DP design matrix made finding a suitable design unfeasible. From considering the DP list, it was obvious that the CNs had effectively specified an FR-DP mapping without considering the possible coupling.

The team started over and applied proper FR and DP guidelines [2, 25]:

FRs begin with an action or transformational verb and are verifiable

DPs begin with a noun and can be quantified or compared

This new design matrix had significantly less coupling. A prototype was tested and was found to be unreliable. Further analysis realized this was due to coupling found in the thermally active material and the pressure release mechanism; heat softened the plug, but the release of the gas cooled the plug. This project has continued at Reykjavik University under the guidance of Professor Joseph T. Foley for a new iteration named Plasbar, which is expected to be published this year. In this new design, the coupling has been directly addressed by separating the plug mechanism from the thermally actuated element.

Girgenti [4] also cautions that the voice of the customer (VOC) can lead directly to DPs. He suggests finding the hidden CNs by zig-zagging conversely. In his example, he mentions the FR 'to provide classic Italian racing colors' and the mapped DP 'color red' for a car, with the hidden CN 'sporty feeling'. This is a non-functional requirement (nFR) [22], which applies to several parts of the car, like the body, interior and also the rims. Girgenti translates this to the FR 'to provide a racing experience'. Requirements of this sort are heavily driven by the subjective interpretation of the interaction between the user and the item being designed. This is a very hard requirement to evaluate, appearing in nature closer to the 'Feeling/Experience Requirement' ($\mathcal{FR}$) described in Creative Axiomatic Design by Foley and Harðardóttir[26]. Their strategy for dealing with such requirements is to condense them down to fundamental emotions to help simplify evaluation. To validate such requirements, the designer provides a test environment and surveys what emotions are registered by the participants. In many ways, this is very similar to Ulrich's [16] concept of a latent need: something that is not well expressed. From a practical implementation, nFRs such as these are best implemented as constraints or criteria that have a limited scope to only certain elements of the design. Iino and Nakao [27] address a similar challenge of limited-scope constraints in designs that need electricity for operation. They address it by use of specialized symbols on their design record graph visualization of an FR-DP decomposition; in the case of needing electricity, related FRs and DPs have a battery symbol next to them. This is an example of Imaginary Complexity due to increased coupling.

3.5 Limitations due to PVs in customer domain

Suh [28] mentions that the customer can also suggest PVs, e.g. when existing machines must be used (see e in Figure 2). This limits also the choice for DPs (see 'e' in Figure 2 and process in Figure 7). These CNs *or better* these PVs limit the choice for DPs and as such are considered to be constraints [28]. The underlying need can be a limitation on costs, which is a constraint in the functional domain and process domain. Replacing the PV with this constraint will increase the options for other, more suitable, DPs and PVs. It is an open question of what type of
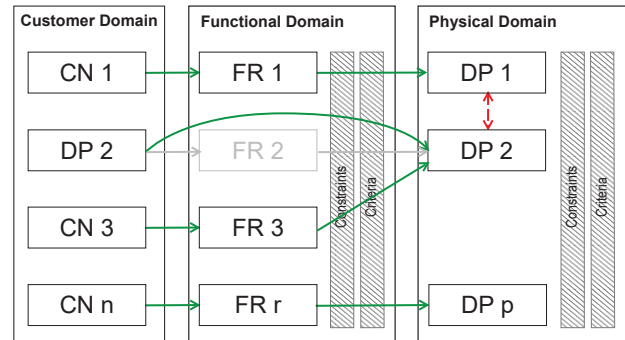

Figure 6: DP in CN causes complexity

complexity these limitations are, however, due to the conceptual impact it is at least expected to increase Imaginary Complexity.

4 Discussion

Now that the identified sources of complexity have been inventoried from the perspective of the customer domain, we will discuss to which insights these investigations have led.

4.1 Reducing complexity

The gathering of raw data from the customers or stakeholders is the first step of the designer to get to the essence of the true needs of the system to be developed. During this interpretation, the CNs can be carried over (mapped) or placed in the functional, physical, or process domains. This will give an overview of the project and mentioned needs. All CNs that are placed in the physical or process domain (see (d) and (e) in Figure 2) should initially be analyzed for the underlying CNs, FRs, constraints or criteria. In this interpretation DPs and possible PVs are translated to the underlying needs, or what Girgenti [4] mentions as zigzagging conversely (reverse zig-zagging). A reverse mapping between the domains could be more accurate as it server as an extra verification of potential unsubstantiated assumptions. Thus gained and verified knowledge reduces Imaginary Complexity during the design process and as such reduce the chance of coupling of FRs. Complete elimination of the interactions between the PVs or DPs is not always possible. As a result coupling of FRs may remain.

The reverse mapping from the physical domain to customer domain is similar to a reverse arrow (d) in Figure 2, not passing the functional domain. This is similar to using models and prototypes as DPs for feedback from the customer and also eliciting more (latent) needs.

4.2 Placing the customer in his domain

With reverse mapping, we place the interpreted needs based on given DPs or PVs back in the customer domain. From this mapping, we make an interpretation based on
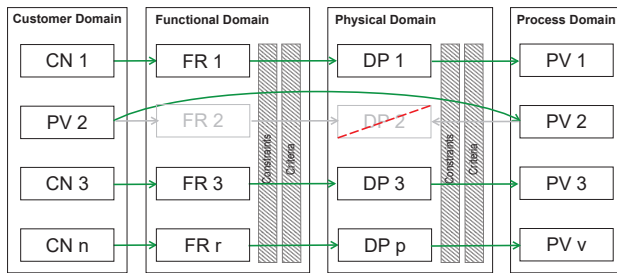
Figure 7: Limitation due to PV in CN

our knowledge of our customers and on the CNs in this domain. These interpreted needs, as a product of reverse mapping, should always be validated with the internal and external stakeholders, as stated needs and also by using prototypes. We want to observe the interaction with the 'living' customers as part of the customer domain.

4.3 Further research

The focus in Axiomatic Design lies on the mapping between FRs and DPs, and DPs and PVs. The customer domain and the mapping between CNs and FRs get less attention. The influence of customers and stakeholders on the development of products and systems can't be overlooked and should get a greater emphasis in Axiomatic Design, also mentioned by Hintersteiner [18]. Further exploration and discussion of how to involve the broad spectrum of (internal and external) stakeholders are needed. Integrating knowledge of other methods like "participatory design", where users are seen as partners in the product development process [29], can also emphasize the importance of the needs of our customers in Axiomatic Design.

4.4 Summary

Identification of CNs is a critical step in design of virtually anything, but the process of eliciting and interpreting them is still unrefined. The customer domain consists of (combinations of) FRs, nFrs, constraints, criteria, DPs, and PVs. These are a collectively exhaustive and mutually exclusive set of preconditions in which the CNs from multiple stakeholders are placed. The multiple stakeholders may have conflicting needs or opinions. The customer domain also contains invisible needs of which we and the customers are not aware of (e.g. hidden or latent needs).

As described in this article, there appear to be three steps:

1. There should be active reverse mapping from the process, physical and functional domain to the customer domain.

2. Customers should be part of the customer domain as persons, not just their interpreted needs.

3. The complexity of customer domain needs further exploration and discussing.

With products that involve many customers or stakeholders and that includes end-user interaction, customer domain complexity should always be analyzed in depth, to maximize independence, minimize information, and also optimize the fit with the customer.

References

[1] N.P. Suh, *Complexity* (Oxford University Press, 2005)

[2] N.P. Suh, *Axiomatic Design - Advances and Applications* (Oxford University Press, 2001)

[3] N.P. Suh, *The Principles of Design* (Oxford University Press, 1990)

[4] A. Girgenti, B. Pacifici, A. Ciappi, A. Giorgetti, *An Axiomatic Design Approach for Customer Satisfaction through a Lean Start-up Framework*, in [30], pp. 151–157, sep. 21–23

[5] E. Puik, E. Smulders, J. Gerritsen, B. van Huijgevoort, *A Method for Indexing Axiomatic Independence Applied to Reconfigurable Manufacturing Systems*, in *Proceedings of ICAD2013 The Seventh International Conference on Axiomatic Design Worcester – June 27-28, 2013* (2013)

[6] E. Puik, *Risk Adjusted, Concurrent Development of Microsystems and Reconfigurable Manufacturing Systems*, in *Doctoral dissertation, PhD thesis manuscript University of Warwick* (Coventry, United Kingdom, 2017)

[7] C.A. Brown, *Teaching axiomatic design to engineers — Theory, applications, and software*, Journal of Manufacturing Systems **24**, 186 (2005)

[8] D.S. Cochran, J.T. Foley, Z. Bi, *Use of the Manufacturing System Design Decomposition for Comparative Analysis and Effective Design of Production Systems*, International Journal of Production Research **55**, 870 (2016)

[9] E.M. Benavides, *Advanced engineering design - An integrated approach* (Woodhead Publishing, 2012)

[10] J.B. ReVelle, J.W. Moran, C.A. Cox, *The QFD Handbook* (J.C. Wiley, 1998)

[11] S.H. Kurniawan, M. Zhang, M.M. Tseng, *Connecting Customers in Axiomatic Design*, in *Proceedings of ICAD2004, The Third International Conference on Axiomatic Design, Seoul* (2004), pp. 1–7

[12] S. Pugh, *Total Design: Integrated Methods for Successful Product Engineering* (Addison-Wesley, 1991)

[13] T.L. Saaty, *The Analytic Hierarchy Process* (McGraw Hill, 1980)

[14] T.L. Saaty, *Decision making with the analytic hierarchy process*, International Journal of Services Sciences **1**, 83 (2008)

[15] G. Taguchi, E.A. Elsayed, T.C. Hsiang, *Quality Engineering in Production Systems*, McGraw Hill Series in Inductiral Engineering and Management Science (McGraw-Hill College, 1988)

[16] K.T. Ulrich, S.D. Eppinger, *Product Design and Development*, 5th edn. (McGraw-Hill, New York, 2012)

[17] D.M. Vossebeld, E. Puik, J.E. Jaspers, M.J. Schuurmans, *Development process of a mobile electronic medical record for nurses. A single case study.*, Manuscript submitted for publication. (2018)

[18] J.D. Hintersteiner, *Addressing changing customer needs by adapting design requirements*, in *Proceeding of ICAD2000, First International Conference on Axiomatic Design, Cambridge* (2000), pp. 290–299, `http://ns.axiomaticdesign.com/technology/icad/icad2000/icad2000{\_}063.pdf`

[19] C.A. Brown, *Elements of Axiomatic Design: a simple and practical approach to engineering design."*, Cazenovia, New York, NY (2006)

[20] M. Thompson, *Improving the requirements process in Axiomatic Design Theory*, CIRP Annals **62**, 115 (2013)

[21] J. Weber, D. Förster, J. Kößler, K. Paetzold, *Design of Changeable Production Units within the Automotive Sector with Axiomatic Design*, Procedia CIRP **34**, 93 (2015)

[22] M.K. Thompson, *A classification of procedural errors in the definition of functional requirements in Axiomatic Design theory*, in *7th International Conference on Axiomatic Design (ICAD 2013)*, edited by M.K. Thompson, CIRP (Axiomatic Design Solutions, Inc., Worchester, MA, 2013), Vol. 32, pp. 1–6

[23] D.M. Vossebeld, *In de prijzen*, Tijdschrift voor Ergonomie **34**, 44 (2009)

[24] G. Bragason, S. Þorsteinsson, R.I. Karlsson, N. Grosse, J.T. Foley, *Heat-activated Parachute Release System*, in *Proceedings of College International Pour La Recherche en Productique (CIRP), 9th International Conference on Axiomatic Design (ICAD)*, edited by M.K. Thompson, A. Giorgetti, P. Citti, D. Matt, N.P. Suh, Procedia CIRP (Elsevier ScienceDirect, Florence, Italy, 2015), Vol. 34, pp. 131–136, sep. 16–18, `http://www.sciencedirect.com/science/article/pii/S2212827115008203`

[25] D.S. Cochran, J. Li, K. Vanover, J.T. Foley, *A System Design of a Rural Hospital Operating Room*, in [31], pp. 597–603, jun. 15–17

[26] J.T. Foley, S. Harðardóttir, *Creative Axiomatic Design*, in [31], pp. 240–245, jun. 15–17

[27] K. Iino, M. Nakao, *Design Record Graph and Axiomatic Design for Creative Design Education*, in [30], pp. 173 – 178, sep. 21–23

[28] N.P. Suh, *Axiomatic design theory for systems*, Research in engineering design **10**, 189 (1998)

[29] E.B.N. Sanders, P.J. Stappers, *Convivial Toolbox: Generative Research for the Front End of Design* (BIS, Amsterdam, 2012)

[30] A. Liu, ed., *10th International Conference on Axiomatic Design (ICAD)*, Vol. 53, Procedia CIRP (Elsevier ScienceDirect, Xi'an, Shaanxi, China, 2016), sep. 21–23

[31] *26th CIRP Design Conference*, Procedia CIRP (Elsevier ScienceDirect, Stockholm, Sweden, 2016), jun. 15–17