Proceedings of the 2015 International Mechanical Engineering Congress & Exposition
IMECE2015
November 13-19, 2015, Houston, Texas, USA

IMECE2015-50993

AXIOMATIC PRODUCT DESIGN IN THREE STAGES; A CONSTITUENT ROADMAP THAT VISUALISES THE STATUS OF THE DESIGN PROCESS BY TRACKING THE KNOWLEDGE OF THE DESIGNER

Erik Puik
HU University of Applied Science Utrecht
3500AN, Utrecht, the Netherlands

Darek Ceglarek
WMG International Digital Laboratory University of Warwick, CV4 7AL, Coventry, UK

ABSTRACT

Increasing global competition in manufacturing technology puts pressure on lead times for product design and production engineering. By the application of effective methods for systems engineering (engineering design), the development risks can be addressed in a structured manner to minimise chances of delay and guarantee timely market introduction. Concurrent design has proven to be effective in markets for high tech systems; the product and its manufacturing means are simultaneously developed starting at the product definition. Unfortunately, not many systems engineering methodologies do support development well in the early stage of the project where proof of concept is still under investigation. The number of practically applicable tools in this stage is even worse. Industry could use a systems engineering method that combines a structured risk approach, concurrent development, and especially enables application in the early stage of product and equipment design. The belief is that Axiomatic Design can provide with a solid foundation for this need. This paper proposes a 'Constituent Roadmap of Product Design', based on the axiomatic design methodology. It offers easy access to a broad range of users, experienced and inexperienced. First, it has the ability to evaluate if knowledge application to a design is relevant and complete. Secondly, it offers more detail within the satisfaction interval of the independence axiom. The constituent roadmap is based on recent work that discloses an analysis on information in axiomatic design. The analysis enables better differentiation on project progression in the conceptual stage of design. The constituent roadmap integrates axiomatic design and the methods that harmonise with it. Hence, it does not jeopardise the effectiveness of the methodology. An important feature is the check matrix, a low threshold interface that unlocks the methodology to a larger audience.

1. INTRODUCTION

Increasing global competition in manufacturing technology puts pressure on lead times for product design and production engineering [1]. On one hand, time to market pushes revenues backwards and erodes market penetration. However, on the other hand, when the development risks of a product have not yet been eliminated, early market introduction leads to bad performance of products and dissatisfied users [2,3]. The development risks can be addressed in a structured manner by the application of effective methods for systems engineering (engineering design).

1.1 Application of Methods for Systems Engineering

By optimisation of products and processes in the virtual realm, before committing resources to physical production, the combination of qualitatively acceptable products and timely market introduction can be significantly improved for at least three reasons:

1) Shorter times to market lead to growing importance of concurrent execution of product design and production engineering, e.g. by application of 'Design for Manufacturability' (DFM) [4,5]. Simultaneous modelling of the product design process and product manufacturing engineering increases understanding of the relation between these activities and thus brings incompatibilities to the surface. Problems can be addressed in an earlier stage before they escalate to larger proportions and performed work proves in vain. Therefore, the integrally modelling approach is more effective. Hence, cost of development may be reduced and/or time to market can be shortened. For designs with a short product life cycle, like the market for high-tech systems, gains can be substantial.

2) As indicated in figure 1, the impact of making decisions early in the product life cycle is high, and declines steeply as a product matures. Conversely, while there are many systems engineering modelling tools to help manufacturers make good decisions about

products late in the process, there are very few available early in the process, where they are needed the most [6,7]. Tools that are applicable in the conceptual design stage may improve this situation.

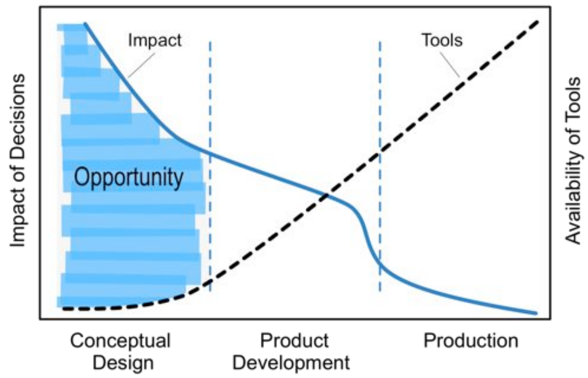

Figure 1: The availability of tools is low, just as it could lead to maximum result

3) Methods for systems engineering can be applied as a means for communication to close the gap between managers and technicians, or even between customers and suppliers [8].

In all three cases, the central theme is knowledge about the design and its periphery; 1) to gain integral knowledge between the domains of design and manufacturing, 2) to gain knowledge in the conceptual stage of design, and 3) to share knowledge between stakeholders.

1.2. Application of Axiomatic Design

In the world of systems engineering tools, Axiomatic Design (AD) has a unique capability; it does not only monitor knowledge as applied to the status of the design but, by means of the axiomatic design matrix, also knowledge of the designer [9]. Many other models, e.g. waterfall based models like the V-model or PRINCE2, focus mainly on the status of the product design. This particular strength makes AD popular with product designers. Product designers typically are strongly technology-oriented and therefore guided by knowledge-based reasoning. Because the quality of a product design typically does not exceed the knowledge of the designer [10], this knowledge is the basis of a reliable design provided that it is relevant and well applied. Therefore, relevant and complete knowledge is a binding condition to enable a satisfactory design process, and it is accurately monitored within the AD methodology.

A second capability of AD is its concurrent approach. The four axiomatic domains are crossroads for the customer, marketer, product designer and the manufacturing engineer. A process called 'Zigzagging' is applied to repeatedly check the alignment of the domains to see if knowledge is missing [11]. Zigzagging makes AD intrinsically 'concurrent' over its domains.

Obviously, there are also drawbacks. AD suffers from its academic nature. Users will need some time to get comfortable with the methodology. Design matrices and the concept of information have a strong scientific origin and tend to alienate mathematically obtuse users. This learning path appears shorter for technical- than for non-technical oriented staff members. This problem is approached in this paper by extending focus on visual representation of the status of product development. In this way, the methodology can easier be used as a means for communication.

Another drawback of AD is that many problems, during the early stage of the project, tend to funnel back to the first axiom; 'the Independence Axiom'. The implementation of this axiom is usually relatively one-sided because it just focuses on making the design independent by decoupling the design matrix [12]. This is an activity that takes place at the finalisation phase of the 'Conceptual Design' (figure 1). This limitation does not come forth from a lack of intrinsic capability of the AD methodology; further decomposition of the independence axiom has been investigated in recent literature and demonstrates room for more detail in the conceptual stage [12]. However, these insights have not yet been implemented in the axiomatic design methodology for product design. The possibilities for implementation will be explored in this paper.

1.3. Research Questions

As the number of practically applicable tools in the early stage of design is low, industry could use a systems engineering method that combines a structured risk approach, concurrent development, and especially enables application in the early stage of product and equipment design. The hypothesis is that axiomatic design can provide with a solid foundation for this need. Therefore, the research questions of this paper are:

How can the AD methodology be applied to provide a generic and industrially applicable framework for systems engineering of the product development process in which:

1. *The applicability threshold is improved so inexperienced users are enabled to evaluate if knowledge application is relevant and complete;*
2. *The method can be applied from the early stage of conceptual design to the finalisation of the design.*

Though the framework in principle is based on AD, it is a secondary goal to integrate other methods for systems engineering rather than developing new methods. Therefore, the focus of this paper is on smart integration of methods with AD as a starting point.

1.4. Structure of this paper

This paper is organised as follows; section 2 inventories the background on industry popular methods for systems engineering for product design and on developments in conceptual design. Section 3 investigates how AD currently matches the demand for the envisioned functionality. Section 4 proposes an AD-based framework for visualisation of the progression of concurrent development. In section 5 is explained how the framework is to be applied. Finally, section 6 discusses the findings and draws conclusions.

2. BACKGROUND

Modelling of the product design process is gaining momentum since the early 1980s when researchers began to realise the impact of design processes on downstream activities [7]. As a result, different methodologies such as design for assembly, design for manufacturing and concurrent engineering have been proposed, developed, and improved.

2.1. Mature Work on Design Science

Early work on design science is mainly from German schools with Hansen dating back to the fifties as reported in the book 'Konstruktionssystematik' [13] and the widely spread book of Pahl & Beitz [5] in the late sixties. This last work was well maintained with rewrites up to 2007. The 'Theorie der Maschinensysteme' is a design science framework presented in 1973 by Hubka [14]. This work introduces design axioms and vector representation of domains. From here the established theories were developed inter alia by Suh with the Axiomatic Design Methodology [15] and Andresen with the 'Domain theory' [16].

2.2. Models that are Widely Used in Industry

Effective systems engineering methods for organisational control, as applied in modern industry, are mainly 'Waterfall' based models. The waterfall model describes design processes sequentially. It was originally developed for software engineering, in which progress is seen as steady flow downwards through a number of stages in the design. Royce was the first to report about the waterfall model [17]. The popular V-Model is also based on the waterfall model. It was simultaneously developed by VDI/VDE and Rook in 1986 [18,19]. In the 1991 proceedings for the National Council on Systems Engineering (NCOSE, today INCOSE), the V-Model was adopted in the US. The V-Model is a planning tool that adds solid testing functionality at different hierarchically decomposed levels to overcome some of the shortcomings of the initial waterfall model. The 'PRINCE' method, as introduced in 1989 (PRojects IN Controlled Environments), is another widely used model that was derived from the waterfall model. PRINCE2 was a method with broader possibilities for application than just software developments and is still maintained to date by the British semi governmental organization 'Office of Government Commerce' [20].

Waterfall models, also the V-Model and PRINCE2, are criticised for the problem of missing iteration in design [17,21]. This is not as much a problem to accountants and project managers, as for developers and testers; a most damaging aspect is the effect that these models discourage user involvement in evaluating the design before arriving at the formal testing stages. By then it is too late to make significant changes to the design. Nevertheless, the V-Model, and in somewhat lesser extent the traditional waterfall model, today are popular systems engineering methods in industry since they provide in needs for management.

2.3. Models for Conceptual Design

Though there are not many generally applicable models for the conceptual design phase, a lot has been reported in literature. Though dated, a good overview of work till 2002 is given by Wang & Hu [22]. A little more recent is the work of Ayag in 2005 [23] and the work of Li et al. [24] (2010). After closer investigation, quite some conceptual design methods appear to have been presented over the last thirty years. These models can be classified into three categories based on focal points and tools used: 1) design models according to the design criterion of products, 2) design models based on the design strategies of products, and 3) design models adopting artificial intelligence. As this paper focuses on the first category, where the designer has the traditional role of being in charge of the design process, category 2) and 3) are not further investigated.

More recent models of the first category include the work of Li et al. [24] where a method is presented based on AD with alternative domains. The conceptual design process is defined as an integrated system with five stages and four mappings and mathematical descriptions are applied as input for an expert system. A similar approach is applied by Tay & Gu [25]. AD is applied to derive the hierarchical topology of the design from the functional and physical domains. The thus obtained primitives are inputted into a relational data model. The work of Chen et al. [26] expands this method with a production framework. The method stays in the conceptual stage. Deng et al. [27] also have a very similar approach as Tay & Gu. However, this work does not use the AD methodology but instead of this, a self-defined framework called 'functional design model' is applied. The architecture framework for manufacturing system design of Benkamoun et al. [28] also uses the axiomatic domains and the hierarchical structure. The framework applies IDEF0 to define relations between the domains. IDEF0 is very suitable for manufacturing because of its capability of modelling sequential processes. Knowledge about process and configuration is stored in the framework and can be reapplied when the system needs to be reconfigured. Zhang et al. [29] have developed an interesting approach for the design of product and maintenance by combining AD, Quality Function Deployment (QFD) and Failure Mode Effect Analysis (FMEA). Knowledge and applied-knowledge are combined in a single model that gives a complete overview of their relations to indicate if parts are missing. Unfortunately, the model is only applied during the conceptual design phase and would need to be expanded for Product Development and production. Ulrich & Eppinger [30] have broken down the process of concept development in seven stages. These stages are each again broken down in 4-7 steps further, which provides an extensive amount of fairly simple steps to follow. However, this apparent simplification does not guarantee that this solves the complexity of the conceptual design stage. As reference, to check that the designer does not forget important issues, it can be useful. Komoto & Tomiyama [31] describe a product modelling framework called System Architecting CAD. SA-CAD tracks system decomposition, it models parameter relations, and performs consistency management of the parameters. An interesting aspect is that SA-CAD could eventually store design knowledge used in system architecting independently from specific engineering disciplines such as physical contacts to constrain the topology of a set of entities. Puik & Ceglarek [12] have recently performed a study on the concept of information in AD. Based on Suh's complexity theory, information in design is decomposed into smaller parts that can be directly related to the purpose of the axioms, including in particular the independence axiom. This enables expansion of AD to two conceptual sub-stages of product design.

The work of Benkamoun, Ulrich & Eppinger, and Komoto is particularly valuable for this research since they all add the capability of actively securing the knowledge content in the model itself or in the periphery of the model. Zhang's model does the same but additionally links this knowledge to the applied knowledge; the current appearance of the design itself, though it should be converted from the realm of maintenance to that of product design.

2.4. Phasing of the Design Process

Most models divide the total product design process in two basic stages. The initial stage is the conceptual stage that ends with a proof of concept and the second stage is a product development stage that deals with realisation and test. The V-Model visualises this with it two legs; the left leg handles conceptual design and the right leg handles integration and testing. Other conceptual design methods that follow the standard V-Model, e.g. Komoto & Tomiyama [31] do the same. In AD, the independence axiom focuses on conceptual design and the information axiom focuses on robustness of the design. This is also referred to as respectively 'Doing the right things' and 'Doing things right' [12]. The standard work of Pahl & Beitz [5] divides the design process in four stages: 'Definition', 'Conceptual', 'Embodiment', and 'Carryout'. Note that the conceptual stage ends with a number of alternative options and the embodiment stage ends with proof of principle of the design, so basically the conceptual stage is split into two stages.

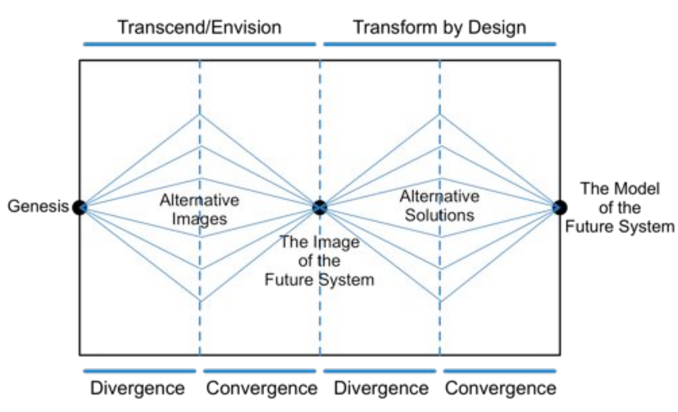

Figure 2: The 'Theory Dynamics of Divergence & Convergence' splits the conceptual stage in two stages

Banathy [32] describes in his theory 'Dynamics of Divergence and Convergence', shown in figure 2, an iterative approach of diverging and converging cycles, respectively focusing on the 'Image of the future system' and the 'Model of the future system'. This is comparable to the approach of Pahl & Beitz, dividing the conceptual stage in two parts. Wang et al. [22] also split the conceptual phase in two stages by defining a mapping stage for fuzzy customer requirements to functional specifications, and a development stage for multiple design solutions. This is not conforming the AD methodology where this could be done simultaneously by the application of concurrent design. As explained in §2.3, the approach of Ulrich & Eppinger [30] is quite detailed. However, a more profound look learns that this approach also sets target specifications, analogue to Banathy's image of the future system, and subsequently it reduces the number of alternatives to a final concept. This makes the approach on a par with Pahl & Beitz and Banathy. Höhn & Höppner [33] describe the V-Modell XT, which is the German variant of the V-model. The model is expanded with a start-up phase before the model goes into the left leg. This phase analyses the project procurement process. It has similarities with the 'Definition' stage of Pahl & Beitz. However, the model does not split the conceptual phase in two stages like Pahl & Beitz and Banathy. The INCOSE Systems Engineering handbook [34] distinguishes for the conceptual stage: identification of stakeholder's needs, exploration, and the proposal of viable solutions.

3. RELEVANT AND COMPLETE KNOWLEDGE IN AD

This section focuses on the state of the art of AD. Section 3.1 describes where knowledge and applied knowledge are stored in AD. Section 3.2 shows recent developments on information in AD. Section 3.3 summarises the developments on conceptual design and their relation to AD.

3.1. Knowledge in Axiomatic Design

AD uses domains to gather the state of the product design. These domains are Customer Attributes (CA), Functional Requirements (FR), Design Parameters (DP), and Process Variables (PV). Figure 3 shows these domains and how they are related.

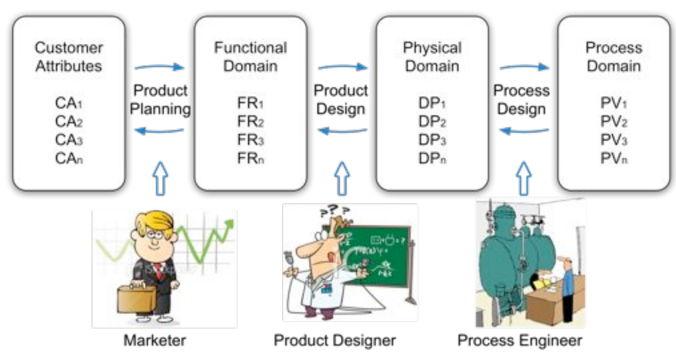

Figure 3: Axiomatic domains and how they are connected

The domains are connected by typical expert roles. As such, the marketer watches over the relation between the customer domain and the functional domain. The product designer monitors the relation between the functional domain and the physical domain and lastly, the process engineer does that for the physical domain and the process domain. In order to make a good design, knowledge must be applied to the relations of the domains. But by doing this, the knowledge is not transferred to the domains; it stays with the experts. Instead, applied knowledge is transferred to the domains, not the knowledge itself [10].

Typically, the knowledge and how it is applied will be in design reports. The marketer, product designer and process engineer will author respectively a market implementation report, a functional description and a process description report (figure 3). Even the way in which knowledge is applied is not transferred to the design; if a DP satisfies a certain FR, it can be hard to recover the relation by reverse engineering. AD uses 'Design Matrices' to connect the domains, starting with the design equations according to good AD practice. The design matrices show how knowledge has been implemented correctly, which proves that the designer has gathered the required knowledge. Summarising:

1. Applied knowledge is transferred to the design;
2. The way knowledge is implemented is in the design matrices (also secured in reports);
3. Knowledge stays with the designers and may be secured in reports.

3.2. Monitoring Progression Based on Information in Design

Information in Axiomatic Design is derived from the information technology using a logarithmic measure of Boltzmann entropy according to Hartley [35] and Shannon & Weaver [36]. According to this theory, information is inversely related to the probability of success of DPs causing their FRs to be within tolerances. The information axiom dictates that information content should be minimised, and thus maximising the probability of FRs to be satisfied. Suh describes two main types of information, 'Useful' and 'Superfluous' information [37]. Useful information is information that affects FRs. Superfluous information does not affect FRs. Therefore, superfluous information is no information from the axiomatic perspective. Suh also defines information according to the Information Axiom, here to be referred to as Axiomatic Information. Puik & Ceglarek recently published a review on information in design [12], indicating that if useful information and axiomatic information are not the same, which it is not according to AD practice, there should be at least one other kind of information. This information is called 'Unorganised Information' due to the fact that this is caused by the absence of a decoupled design matrix. Figure 4 shows a breakdown of that analysis. Unorganised information is the information that is related to the independence axiom, axiomatic information is related to the information axiom. It is clear that, though by the axiomatic definition not defined in that way, the independence axiom in fact is also an information axiom. This axiom has been decomposed further in two kinds of information, 'Unrecognised' and 'Recognised' information, depending if it is recognised by the designer. If it is recognised, the designer understands that the information is relevant for the design but the design matrices are not yet decoupled. If it is unrecognised, it is not yet clear which DPs satisfy what FRs, and which PVs satisfy the DPs.

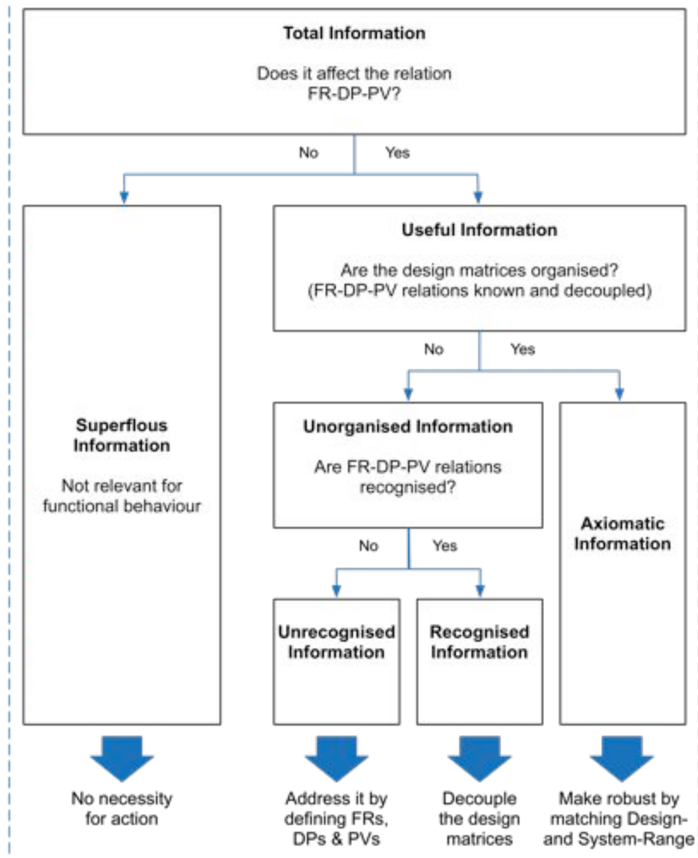

Figure 4: Breakdown of information in design. Superfluous information is, according to the axiomatic definition, no information in the context of AD

3.3. Product Design in Three Stages

The breakdown of information in AD of figure 4 decomposes all axiomatic kinds of information. Useful information is the highest kind of fully relevant information for the design. Useful information is equal to the ignorance of the designer, and was defined as the reciprocal of axiomatic knowledge [10]. If useful information is eliminated from the design, which is the case if all the knowledge in the design is implemented, all FRs will be satisfied. The AD methodology proposes a strict order for the treatment of the axioms, starting with axiom 1. This means that unorganised information should first be eliminated from the design followed by the axiomatic information. Within the unorganised information, unrecognised information is addressed first. This means that information is eliminated from left to right in figure 4. Superfluous information is not addressed since this information does not affect the satisfaction of any FR.

As unorganised information is related to the conceptual stage of design, this means that the conceptual design stage is split into two parts. Axiomatic information is related to robustness and is the final design stage. Summarised, three successive stages for the design are recognised:

1. A stage to create coherence in the product design process, in which the FRs, DPs, and PVs, are gathered;
2. A stage that validates the concept by completion and decoupling of the design matrix;
3. A robustness stage in which the DPs and the FRs are tuned.

In the next section, these three stages will be applied in a general roadmap for concurrent product design.

4. ROADMAP FOR PRODUCT DESIGNERS

The ingredients of a generic roadmap for product design have been inventoried. Summarised, the roadmap would combine: 1) Specification of requirements over the domains, 2) A distinction between knowledge and applied knowledge, and 3) combine the three sequential stages, search for coherence, conceptual validation, and robustness.

4.1. Constituent Roadmap of Product Design

The constituent roadmap of product design uses the AD methodology as a starting point. Its particular feature is that it tracks the implementation of knowledge but, at the same time, also tracks the knowledge related processes to visualise how knowledge is applied to the product design. It uses three sequential stages in design analogue to the standard of Pahl & Beitz, Banathy, and Puik & Ceglarek. It is intended to be modular in itself and integrates with most existing models that focus on a particular topic e.g. Quality Function Deployment (QFD), Qualitative Analysis (QA), Failure Mode Effect Analysis (FMEA), Morphological Matrix, and Structured Analysis Design Technique (SADT) [8,11,38].

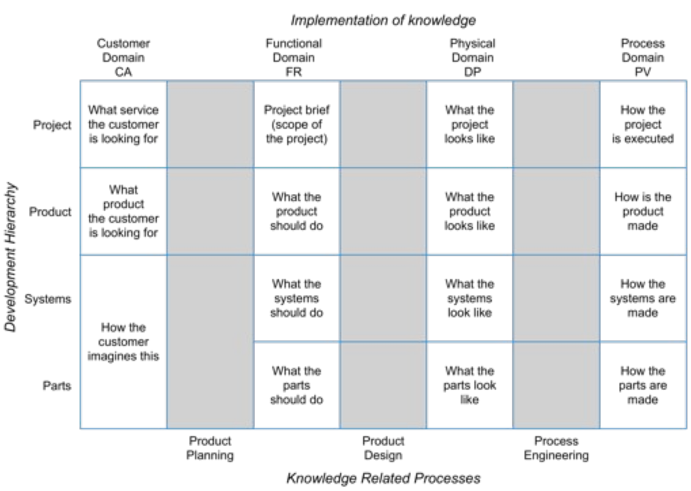

Figure 5: Part 1 of the constituent roadmap focuses on the application of knowledge

The constituent roadmap is a merged matrix of 7 x 4 cells that gathers applied knowledge in the odd rows and gathers knowledge in the even rows. Figure 5 shows the contents of the odd rows. Obviously, there is strong resemblance with the axiomatic domains and their hierarchy as defined by Suh [37]. Note that the customer domain is included as well [11]. This domain is often excluded since it is difficult to determine a design matrix between the vectors {CA} and {FR}. This is caused by the softer way customer attributes are specified. The lowest rows for systems and parts are less relevant to the roadmap because usually customers have no attributes for systems or parts in particular; sometimes customers envision a certain technology. Decomposition is placed on the vertical axis. This is analogue to the decomposition as applied in the V-Model. In this case the levels are reused from the German V-Modell XT as it starts at the project level, down to products, systems and parts [33].

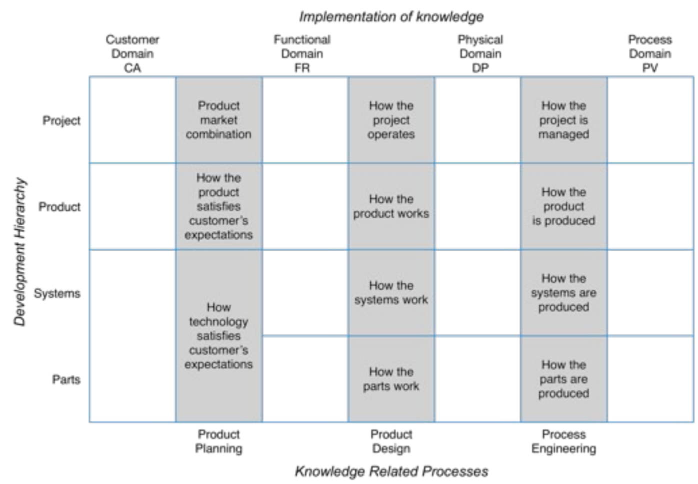

Figure 6: Part 2 of the constituent roadmap focuses on the knowledge that is needed to get a good design

Figure 6 shows the knowledge related processes in the even columns. AD offers the decoupling strategy to finalise the conceptual stage. The knowledge may be secured by applying suitable tools, e.g. QFD for product planning, the morphological matrix for product design or SADT for process engineering, but other methods may be applied as well. The application of these tools takes care of 1) securing knowledge about the design in the project documentation, 2) the alternatives for the design that were not applied and what the reason was to decide so.

Finally, the two matrices will be combined to relate FRs, DPs, and PVs statistically by applying robust design techniques. Tools that can be applied in this stage are Six Sigma [39] or the Taguchi method [40].

4.2. The Constituent Check Matrix

The ‘Check Matrix’ is applied to track the progression of the constituent roadmap. Its structure is based on part 2 of the roadmap as shown in figure 6. The status of the according relation is represented with a number from 0–3, directly referencing to the best-completed development stage of that relation, this is shown in figure 7. The content of the check matrix represents the status of the knowledge relation of the domains. It is a measure to track progression of elemental parts of the product development process.

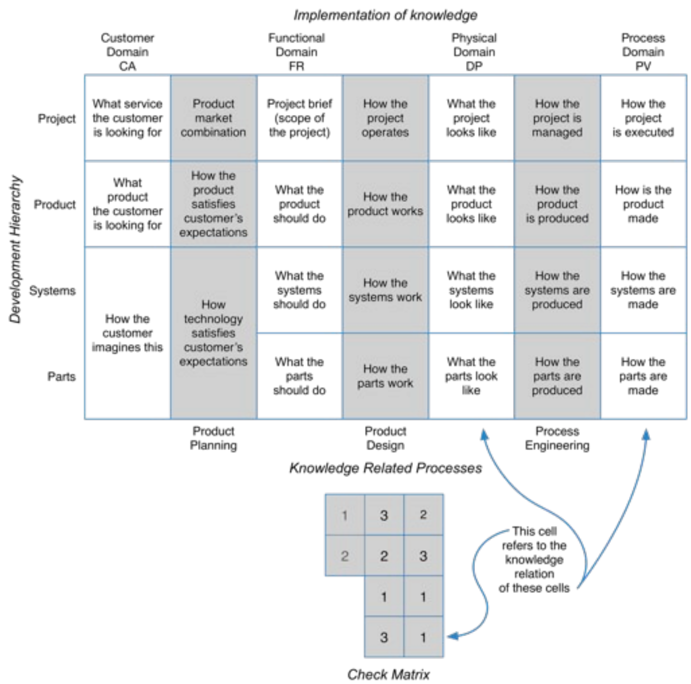

Figure 7: The check matrix gives an overview of the knowledge status of the product

5. PROCEDURE OF APPLICATION

Suh describes a zigzagging process for decomposition of the domains. Zigzagging decomposes the product layer by layer until the design reaches the final stage; a design that can be implemented [11]. However, in this case for the constituent roadmap, zigzagging is only applied to validate the conceptual design. Zigzagging is a process to validate if the design is in harmony. It checks if the relations between FRs, DPs, and PVs are present and decoupled. It checks if knowledge was correctly applied to the design. Zigzagging is a thorough process with a strictly defined path through the domains moving top down in the hierarchy. If an error in the design is found, zigzagging stops and starts again from the top. Because of this thoroughness, zigzagging also is a slow process. When large amounts of quick tests need to be done to check if knowledge on a certain topic is available, as is the case in the explorative stage to look for coherence, a more agile process is needed.

To enable this, a process called 'Yo-yoing' is applied. Yo-yoing describes a motion through the matrix that is more erratic than the process of zigzagging. Yo-yoing goes up and down and if needed left or right. It starts at the top where the overview on the project is maximal, but jumps from there to the place where the development risks are highest. It performs an explorative search in order to check if knowledge is present to successfully connect the FR with a DP, or a DP with a PV. From there it bounces back to the highest level to check for the next largest risk. In this way it pokes around in the matrix to address the largest development risks one by one. This is shown in figure 8.

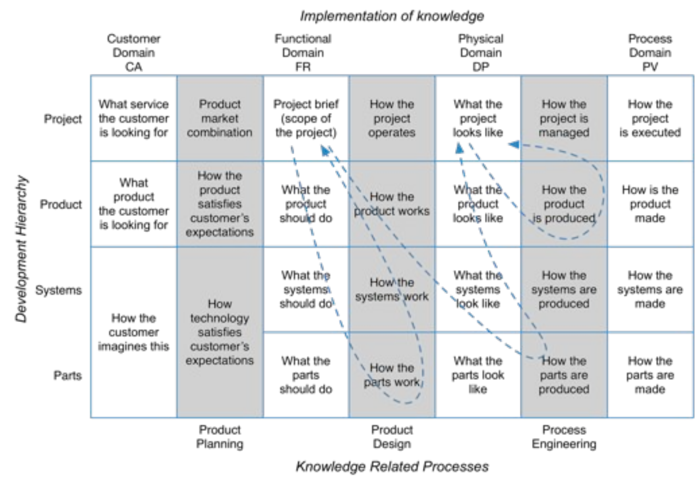

Figure 8: The full constituent roadmap and the process of yo-yoing

5.1. Stage 1: Search for Coherence

The first stage forms the exploratory part of the conceptual stage. The goal is to determine preliminary design relations and which knowledge is relevant for the association of CAs, FRs, DPs, and PVs. The check matrix starts with zeros for all relations.

At first, the mission of the project (e.g. the project brief) is decomposed to form an image of the product. A wide scope of the designer is required. Yo-yoing is applied to address immature relations in the product design using a risk-driven way of working; the order of addressing the relations in the check matrix is determined by severity of the risk. Coding may be applied in the context of qualitative risk analysis to find the appropriate order [38,41].

During on-going decomposition, FRs, DPs, and PVs are determined. All FR–DP (or DP–PV) relations must be accompanied with a notion how that DP is going to satisfy the FR. The notion is based on knowledge of the designer. The check matrix is applied to gather the results; if the FR and the respective DP and the notion are present, the respective cell of the check matrix is set to '1'. If all zeros have disappeared from the check matrix, stage one is completed. Stage one may be repeated to find alternative product solutions.

5.2. Stage 2: Conceptual Validation

The second stage leads to proof of principle. It is based on decoupling of the matrix. Typically QFD would be applied to establish the relation between CAs and FRs. morphological matrices between FRs and DPs, and IDEF0 (or the almost similar SADT) may be applied to define the relations of DPs and PVs (figure 9).

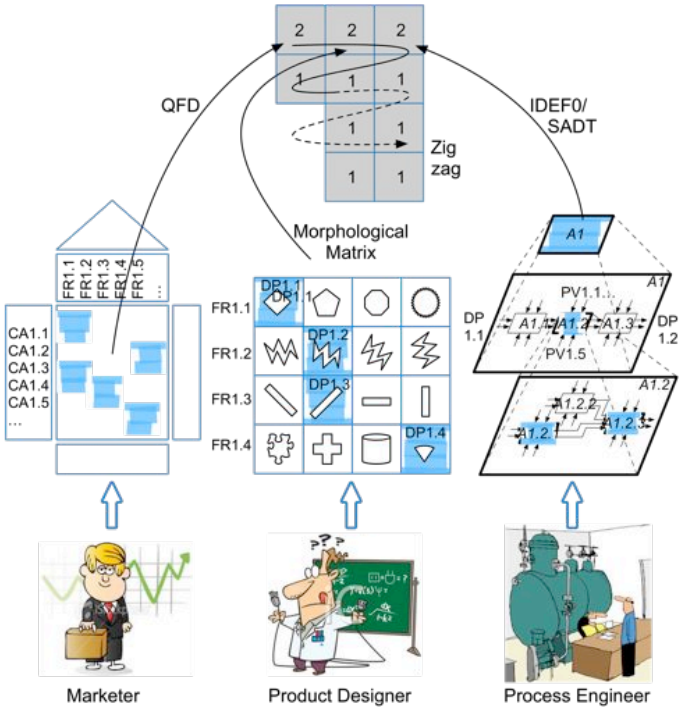

Figure 9: Combining known methods for systems engineering with the constituent roadmap during the stage of conceptual validation

Zigzagging is applied top-down in hierarchical direction to confirm the relations and verify that the design matrices are decoupled. It is possible to start the process with more than a single product concept. However, at the end a single, most promising solution is selected for continuation to the next stage. This solution gets the status of proof of concept when the design matrices are fully decoupled.

For every process relation, the value in the check matrix is increased to '2' to indicate that the relations are understood. When all cells of the check matrix have advanced to '2', the system is conceptually understood, which means that it is fully modelled from the functional and manufacturing perspective.

5.3. Stage 3: Gaining Robustness

The final stage is executed conform good axiomatic practice. The product is made robust by matching the design range and the system range within the common statistical frameworks of Six Sigma [42] or Robust Design [40]. It ensures reliable satisfaction of FRs with DPs incorporating their tolerances (analogously PVs satisfy DPs). When relations are proven robust, the check matrix is upgraded to '3'. The product is robust and fully engineered when the complete check matrix is set to '3'. This final stage completes the implementation of all relevant knowledge in the design.

5.3 Recording of design information

During all three stages, the check matrix has a purpose in recording design information. At the end of each stage, information that was applied to upgrade the check values is stored in the project database (e.g. output of the QFD, Morphological Matrix and IDEF0/SADT conform figure 9).

6. DISCUSSION & CONCLUSIONS

In essence, the constituent roadmap is a knowledge-tracking model. In the first stage, it monitors the presence of necessary knowledge during the search for coherence. In the second stage, it monitors if the knowledge was applied to choose the right solutions during conceptual validation. Finally, in the third stage, it checks if knowledge was applied to implement solutions according to good practice. The check matrix visualises progression in a consistent way throughout the development process. This enables easy access for less-experienced users. The roadmap is based on information in axiomatic design, in which the unorganised component, related to the independence axiom, was split into an unrecognised and a recognised part. Users are stimulated to designate the relevant knowledge in the design, before making final conceptual choices when the milestone 'proof of concept' approaches. This increases the resolution of the method. Therefore, the research questions of this paper can be acknowledged; 1) users are enabled to check if knowledge application is relevant and complete, and 2) the method can be applied throughout the development process from the early conceptual stage to engineering.

6.1. Particular Strengths of the Constituent Roadmap

The way of knowledge stocktaking for the product and its domains, in relation to the hierarchy of the V-Model (V-Modell-XT) is new. Also new is the way this knowledge is visualised in the check matrix that focuses on all knowledge related processes. Together it combines accessibility of the scientifically forceful method of axiomatic design, yet dealing with the plenitude of the product design in a structured way. Due to the visual nature of the model, it is suitable as a universal language to improve understanding between al stakeholders in the organisation, whether they are managers, staff or technicians. Especially since the roadmap is simple to apply, its use is attractive to all parties.

The method retains its solid academic basis due to the application of the axiomatic design methodology. Many ways to combine AD with other systems engineering methodologies have been reported in the past. The constituent roadmap may benefit from these combinations comparably. However, the application of a conceptual stage, with divided attention for exploration and validation, analogue to the work of Banathy [32], Pahl & Beitz [5], Ulrich & Eppinger [30], and Wang [22], increases resolution in the early stage of design. The dynamics of convergence and divergence according to Banathy (figure 2) can be seamlessly mapped to the constituent roadmap but is expanded to the rear. The stage 'search for coherence' will lead to 'the image of the future system, 'conceptual validation' leads to 'the model of the future system', and the stage 'gain robustness' will lead to a new goal: 'the implemented system'.

6.2. Weaknesses and Limitations of the Constituent Roadmap

The constituent roadmap also has shortcomings. Complete underpinning of the check matrix can be laborious, especially when descending in the hierarchical tree when it tends to gain in width. Quite a number of design relations have to be scanned before complete understanding of knowledge application can be guaranteed. Moreover, dutiful application of the roadmap does not relieve the designer of the need to collect relevant design knowledge, and thorough understanding of the design remains essential. The constituent roadmap may be easy to apply but cannot unburden the designer of gathering design knowledge.

The number of stages of the constituent roadmap is limited to a number of three. No sub-stages have been defined. The V-Model with its two main stages has a number of eight sub-stages in total. This feature is much appreciated by managers due to its accurate gating function during project management. Sub-stages may need to be defined in the constituent roadmap as well.

The hierarchical action, that characterises the V-Model, is not graphically represented in the constituent roadmap. In the first two stages this is not so much a problem since yo-yoing provides with the same functionality in the first stage and zigzagging does this for the second stage. In the third stage, testing is not as clearly implemented compared to the V-Model. Therefore the user should bear in mind that testing is an essential activity during the engineering process.

As said, communication within the company could improve when the constituent roadmap is used. However, a limiting factor is that this only works when some instruction about the method is given to its users.

6.3. Scope for Extension of the Method

A valuable expansion of the constituent roadmap might be to upgrade the number of gates within the three stages as defined in this paper. This could be realised by applying elements from the more nuanced procedure as described by Ulrich & Eppinger [30]. In the last stage, testing might be implemented equivalent to the V-Model.

A selection of accessible and matching system engineering methods that integrate well with the constituent roadmap would be welcome.

7. CONCLUSIONS

The constituent roadmap may serve as a model to track product development from the earliest stage to market introduction. Two new features are characteristic for the constituent roadmap. First, the way of knowledge stocktaking for the product and its domains may be considered novel. Secondly, the way this knowledge is visualised in the check matrix focuses on all knowledge related processes. Together it combines accessibility of the scientifically vigorous method of axiomatic design, yet dealing with the plenitude of the product design in a structured and clear way.

8. ACKNOWLEDGMENTS

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

9. REFERENCES

[1] Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., and Van Brussel, H., 1999, "Reconfigurable Manufacturing Systems," CIRP Annals - Manufacturing Technology, 48(2), pp. 527–540.

[2] Puik, E. C. N., Rijfers, A., and Tillie, L., 2002, "A lower threshold to the world, by combining a modular approach and a matching production framework," Mancef, ed., Proceedings COMS, Ann Arbor, p. 3.

[3] Puik, E. C. N., and Moergestel, L. V., 2010, "Agile Multi-Parallel Micro Manufacturing using a Grid of Equiplets," S. Ratchev, ed., Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 271–282.

[4] Boothroyd, G., 2005, Assembly automation and product design, CRC.

[5] Pahl, G., and Beitz, W., 2013, Engineering Design, Springer Science & Business Media, London.

[6] IMTR, 2000, Integrated Manufacturing Technology Roadmapping Project.

[7] Hsu, W., and Liu, B., 2000, "Conceptual design: issues and challenges," Computer-Aided Design, 32(14), pp. 849–850.

[8] Puik, E. C. N., Smulders, E., Gerritsen, J., Huijgevoort, B. V., and Ceglarek, D., 2013, "A Method for Indexing Axiomatic Independence Applied To Reconfigurable Manufacturing Systems," M.K. Thompson, ed., Worcester, pp. 186–194.

[9] Suh, N. P., 2005, Complexity: Theory and Applications.

[10] Puik, E. C. N., and Ceglarek, D., "The Quality of a Design Does Not Exceed the Knowledge of the Designer; an analysis based on Axiomatic Information and the Cynefin Framework," M.K. Thompson, ed., Florance.

[11] Suh, N. P., 2001, "Axiomatic Design - Advances and Applications," advances and applications, Oxford University Press, New York.Oxford, 2001, Chapter 5, pp. 239-298.

[12] Puik, E. C. N., and Ceglarek, D., 2014, "A Review on Information in Design," International Conference on Axiomatic Design, CAD2014, Lisboa.

[13] Hansen, F., 1968, Konstruktionssystematik.

[14] Hubka, V., 1973, Theorie der Maschinensysteme, Springer-Verlag, Berlin, Heidelberg.

[15] Suh, N. P., Bell, A., and Gossard, D., 1978, On an axiomatic approach to manufacturing and manufacturing systems, Journal of Engineering for Industry.

[16] Andresen, M. M., "Machine design methods based on a systemic approach."

[17] Royce, W. W., 1970, "Managing the development of large software systems."

[18] Rook, P., 1986, “Controlling software projects,” Software Engineering Journal.
[19] Friedrich, J., Hammerschall, U., Kuhrmann, M., and Sihling, M., 2009, Das V-Modell XT, Springer.
[20] Office of Government Commerce, 2008, Managing successful projects with PRINCE2.
[21] Christie, J., 2008, “The Seductive and Dangerous V-Model,” Testing Experience.
[22] Wang, L., Shen, W., Xie, H., Neelamkavil, J., and Pardasani, A., 2002, “Collaborative conceptual design—state of the art and future trends,” Computer-Aided Design, 34(13), pp. 981–996.
[23] Ayag, Z., 2007, “An integrated approach to evaluating conceptual design alternatives in a new product development environment,” International Journal of Production Research, 43(4), pp. 687–713.
[24] Li, W., Li, Y., Wang, J., and Liu, X., 2010, “The process model to aid innovation of products conceptual design,” Expert Systems with Applications, 37(5), pp. 3574–3587.
[25] Tay, F. E. H., and Gu, J., 2002, “Product modeling for conceptual design support,” Computers in Industry, 48(2), pp. 143–155.
[26] Chen, X., Gao, S., Yang, Y., and Zhang, S., 2012, “Multi-level assembly model for top-down design of mechanical products,” Computer-Aided Design, 44(10), pp. 1033–1048.
[27] Deng, Y. M., Britton, G. A., and Tor, S. B., 2000, “Constraint-based functional design verification for conceptual design,” Computer-Aided Design, 32(14), pp. 889–899.
[28] Benkamoun, N., ElMaraghy, W., Huyet, A.-L., and Kouiss, K., 2014, “Architecture Framework for Manufacturing System Design,” Procedia CIRP, 17, pp. 88–93.
[29] Zhang, Z., and Chu, X., 2010, “A new approach for conceptual design of product and maintenance,” International Journal of Computer Integrated Manufacturing, 23(7), pp. 603–618.
[30] Ulrich, K. T., and Eppinger, S. D., 2012, Product Design and Development, McGraw-Hill Europe.
[31] Komoto, H., and Tomiyama, T., 2012, “A framework for computer-aided conceptual design and its application to system architecting of mechatronics products,” Computer-Aided Design, 44(10), pp. 931–946.
[32] Banathy, B. H., 2013, Designing Social Systems in a Changing World, Springer Science & Business Media, Boston, MA.
[33] Höhn, R., Rausch, A., Höppner, S., Broy, M., Bergner, K., Petrasch, R., Biffl, S., Wagner, R., and Hesse, W., 2008, Das V-Modell XT.
[34] Haskins, C., Incose Systems Engineering Handbook V3.
[35] Hartley, R. V. L., 1928, “Transmission of information,” Bell System Technical Journal, The, 7(3), pp. 535–563.
[36] Shannon, C. E., and Weaver, W., 1949, The Mathematical Theory of Communication.
[37] Suh, N. P., 1990, The Principles of Design (Oxford Series on Advanced Manufacturing), Oxford University Press on Demand.
[38] Puik, E. C. N., Telgen, D., Moergestel, L. V., and Ceglarek, D., 2013, “Qualitative Product/Process Modelling for Reconfigurable Manufacturing Systems.”
[39] Yang, K., 2003, Design for six sigma.
[40] Taguchi, G., Chowdhury, S., and Wu, Y., 2005, Taguchi's Quality Engineering Handbook, Wiley-Interscience.
[41] Saldaña, J., 2012, “The Coding Manual for Qualitative Researchers.”
[42] Yang, K., 2009, Design for Six Sigma: A roadmap for product development, McGraw-Hill.

