Robust Product Architecture Development Combining Matrix-Based Approaches and Function-Based Failure Propagation Method – M-FBFP Framework

Krešimir Osman
Research and Teaching Assistant
University of Zagreb
Faculty of Mechanical Engineering and
Naval Architecture

Dragi Stamenković
Mechanical Engineer
Termoelektro Ltd., Belgrade

Mihailo Lazarević
Professor
University of Belgrade
Faculty of Mechanical Engineering

This paper proposes an M-FBFP framework with the objective to help designers tackle the problem of risk emerging from product architecture and the effects of uncertain operating conditions. The proposed framework combines matrix approaches (QFD and MDM) and the FBFP method. The QFD is an integrated set of tools for recording user requirements, engineering characteristics that satisfy these requirements, and any trade-offs that might be necessary between the engineering characteristics, while the MDM is applied to model structural arrangements and dependencies between the domains and within themselves. The FBFP method, on the other hand, is applied at the functional level, provides potential failure information based on product functions during conceptual design in product subsystems. As a result of the proposed framework, risk analysis of subsystems becomes possible and feedback on product architecture could be provided. To test validity of the proposed approach, here is presented a case study with climate chamber with heat regeneration.

Keywords: *complexity, product architecture, quality function deployment, multiple domain matrix, risk analysis, M-FBFP framework.*

1. INTRODUCTION

Safety integration during the design of industrial equipment has been studied intensively for the past 20 years. Many authors have proposed different methods to improve the effectiveness of this critical task. However, the redesign or modification of industrial equipment during its working life is a quite different situation. Design as an activity is based on the principle of generation and testing solution alternatives until they conform to designer's understanding of what has to be designed. At any abstraction level used during the design process two aspects interfere: establishment of the system architecture and evaluation of the system behaviour which emerged as a result of the proposed architectural structure [1,2]. Thus, the system's performance is dependent on designer's understanding of a design problem including personal beliefs and experience, and on the emergent behaviour of the system which was designed to perform within certain acceptable limits. Achieved by a designer, behaviour of a technical system may reflect only aspects and traits of behaviour modes which can emerge from established system's architecture introducing in such a way the risk in respect of the system functionality.

Change is a core to the development of complex products; successful implementation of change requests depends partly on the understanding of the design issues that are associated with a proposed change. Risk assessment is widely considered to be an integral part of good management practice [3] and it can also be used when assessing the consequences of design change [4]. Risk assessments are necessary to anticipate and prevent accidents from occurring or repeating. Since product safety and reliability are affected the most by decisions made during the early design phases, a risk assessment that can be performed with less mature data during these design phases is needed. Design engineers are faced with the problem that in order to carry out change processes effectively, they need to thoroughly analyse the possible effects of change. Depending on the approach taken for such assessments, a thorough evaluation process can be very time consuming. One possible reason is that each component interacts, sometimes in an unintended manner, with many other components within the product. The components which are affected by the initial change also have a tendency to further propagate the effects of change to other components extending the amount of assessment necessary for a single change. From a complex system's research area [5] it is known that even a small change into the system's architecture can lead to unexpected or even unstable behaviour of the whole system, or likewise that small perturbation of input conditions as unforeseen environmental conditions or modes of use yield in an undesired system's behaviour. We are in particular interested in the conceptual design stage that is considered to be the most critical step in product development. In this stage, an abstract description of the product is created that serves as the basis for subsequent design stages and decisions. To a large extent, the quality of the product concept determines the fate of the product. It is recognized that

---

Received: August 2011, Accepted: October 2011

Correspondence to: Krešimir Osman, Dipl.-Ing.
Faculty of Mechanical Engineering and Naval Architecture,
Ivana Lučića 5, 10000 Zagreb, Croatia
E-mail: kresimir.osman@fsb.hr

the initial stages of design are most influential in the quality and success of the product [6]. Yet, there is little support for design methods to assist designers in the initial stages of design. In contrast, detailed design enjoys enormous variety of tools that it even becomes an issue to select among them. The recent studies undertaken to analyze the accidents involving various types of industrial machinery, generally, reached the same conclusion: many tools, machines and industrial production systems are not well adapted to occupational safety and health considerations (OS&H) [7-9]. Now there is a growing recognition of the fact that this poor adaptation is the result of an inadequate design with regard to OS&H [10].

This work presents the matrix – function-based failure propagation method (M-FBFP) framework, which is aimed to provide a designer with a feedback about expected behaviour (properties) of a predefined subsystem architecture. It focuses specifically on the relationship between functions through the function-based failure propagation method and extend their influence on other domains (components and requirements) in the early design (conceptual design) phase by presenting a mapping as an iteration process to improve product architecture, which combines matrix-based approaches: quality function deployment (QFD) [11] and multiple domain matrices (MDM) [12] with the FBFP method [13]. The provided feedback should point out the elements within the system architecture, which are not able to operate within given parameters thus causing the unstable system behaviour. Although the long-term research goal is to establish automated feedback between structural and behavioural domains, at the current research stage the transformation between domains is performed manually, as well as resolving the implications of simulation results to product architecture.

2. STATE-OF-THE-ART

2.1 Matrix-based approaches

Starting in the early 1960s, several researchers proposed matrix-based methods for system modelling and analysis [14-17]. Steward used matrix-based techniques to analyze the structure of the system design process [15]. He represented information flows in binary form in matrices and developed algorithms for the realignment of elements in the matrix. These methods have increasingly attracted attention for managing the complexity of engineering systems and complex product design processes [18-21].

A large variety of matrix-based methods in product design applications can be classified by the quantity of the types of elements involved and executed computations. Some approaches focus on the representation and analysis between elements of the same type (e.g. dependencies between physical product components); others consider linkages between two different types (e.g. dependencies between customer requirements and product functions). If relations within elements belonging to the same type are examined, the related matrices can be defined as intra-domain. Matrices combining different elements that belong to different domains are referred to as inter-domain matrices. If in addition to the combined use of intra- and inter-domain matrices, computations of some subsets by information stored in other subsets are also included, such an approach is called the Multiple-Domain Matrix (MDM) [12].

The Dependency (called also Design) Structure Matrix (DSM) represents a commonly applied approach of an intra-domain matrix; the terms dependency structure matrix, dependency map, interaction matrix, incidence matrix, precedence matrix, and problem-solving matrix are also used in the literature [22]. Steward ultimately defined the term “DSM” in a publication in 1981, when he applied matrix-based techniques to analyze the structure of a system’s design [15].

According to a description given by Browning, a DSM fulfils the following technical criteria [22]: A DSM is a square matrix, i.e. a matrix with an equal number of rows and columns. It provides systematic mapping of elements and their relationships. These elements can be, for example, physical product components, performance attributes, engineering requirements, or process tasks. The element names are placed down the side of the matrix as row headings and across the top as column headings in the same order.

In general, intra-domain matrices are used to improve the design or the design process of a product. The aspect of knowledge capturing by a systematic process of matrix filling already contributes to this purpose. Future designers can benefit from the captured information about system connectivity, which will allow them to make designs faster and better. By appropriate realignment of the element rows and columns, DSMs support the identification of structural subsets, which can be interpreted and provide methods of structural improvement.

Whereas intra-domain matrices consider relations within the same domain, inter-domain matrices link elements of two different domains. These sorts of linkages are widely used in design methodology and labelled with different names, such as “Cause and Effect Matrix” or “Interface Structure Matrix” [23,24]. In 2001, Danilovic and Börjesson settled on the term “Domain Mapping Matrices” (DMMs) for a formal enhancement of the intra-domain (DSM) methodology to inter-domain matrices. They presented studies on linkages between product architecture and organization as well as between systems and organization [25,26]. In 2003, Danilovic and Sigemyr documented several methods that allowed the systematic analysis of DMMs [27]. Subsequently, Danilovic and Browning described the DMM as the complement to known DSM approaches [28].

Whereas single intra-domain matrices are commonly applied to different kinds of structures, Eppinger and Salminen describe the need for combining the three views of component, process, and organizational structures as a basis for successful product development [29]. They stress the importance of comparison across the pattern types, because the “development organization is executing the development process, which is implementing the product architecture”. The authors faced problems in the practical implementation of a systematic comparison, as appropriate analysis possibilities were not at hand. For this reason, they restricted their research to the comparison of one-to-one linkages between the three system views, i.e. they set up three inter-domain matrices [29]. A well-established application of combined intra- and inter-domain matrices is the House of Quality (HoQ), which comprises a part of the method of the Quality Function Deployment (QFD) [11,18]. Dong and Whitney introduce a systematic and mathematically founded approach to obtain a DSM from information in a Design Matrix (DM) [30]. Such DMs are applied in the Axiomatic Design approach and relate to the requirements of design parameters (i.e. such matrices represent DMMs) [31]. Dong and Whitney explain that a DSM is appropriate for capturing, understanding, and managing "the interactions occurring in the system of the product and the system of the design team" [30].

Multiple-Domain Matrices (MDMs) differ from the combined application of intra- and inter-domain matrices (mentioned in the previous section) by computations within the considered matrices. The term MDM has been mentioned by Maurer and Lindemann who point out that its creation is based on DSM and DMM approaches [12]. In design methodology a few practical applications exist, but they do not provide a generic description of the methodology and serve for specific applications only. Bongulielmi et al. introduce the "K- & V-Matrix" for managing different types of configuration knowledge [32]. Puls et al. describe a software implementation that allows for practical applications [33]. In addition, Puls describes the possibility of generating one of the three matrices by computing available information in the two other ones [34]. Danilovic and Börjesson apply a MDM to the management of a multi-project environment [25]. They generate separate DSMs on the tactical and operational level and provide links to transform strategic business decisions into a series of projects. Danilovic and Börjesson present several projects in DSMs, connect them by DMMs and mutually link project tasks [25]. One unique advantage of the approach is the possibility of dealing with an asymmetric DSM, i.e. a DSM with the same elements but aligned in different order on the two axes. Yassine et al. present a multiple domain approach called "connectivity maps" [35]. In regards to the decomposition approaches provided by the DSM methods, they state that "decomposition helps in containing the technical complexity of the design; however, it increases its managerial complexity. The synthesis of the different elements (or subsystems) into a final product (or system) requires the identification and understanding of the inter-relationships among the different elements".

So far, the approach is unique among known multiple-domain approaches, as it computes an inter-domain instead of an intra-domain matrix. This seems to be promising for future structure analysis. However, only a few analysis approaches are available so far, which allow the consideration of inter-domain matrices.

2.2 Risk analysis in engineering design

To prevent accidents from occurring, engineers have often used failure analysis tools to redesign products. While this often leads to advances in technology, traditional failure analysis techniques are not stand-alone tools. A risk assessment is also necessary to anticipate and prevent accidents from occurring or repeating rather than simply responding to failure events. This type of assessment involves the estimation or calculation of both the likelihood of potential failures and their consequences. The results of combining effective failure analysis and risk assessment tools include the improvement of the safety, reliability, and security of products.

Risk is the chance that an undesirable event will occur and the consequences of all its possible outcomes [36]. The early stages of the project represent the period when the opportunity for minimising the impact or working around a potential risk predominantly exists. Since the opportunity for minimising project risks occurs in the conceptual design phase, a tool that utilises failure analysis to estimate project risk in this design phase would be beneficial. Risk assessment has seen much attention from various disciplines and has accumulated a variety of techniques to assist engineers with their risk assessment and management.

The current state-of-the-art in quantitative risk analysis is probabilistic risk assessment (PRA) [37]. The PRA is a systematic and comprehensive methodology to evaluate the risk associated with every lifecycle aspect of a system in which the risk is assessed in quantitative terms. This type of assessment requires answering the following questions: What are the initiating events that lead to adverse consequences? How severe are these adverse consequences? How likely are these adverse consequences to occur? Some of the existing methodologies for answering each of the questions are as follows: failure modes and effects analysis (FMEA), event tree analysis, event sequence diagrams, and fault tree analysis. Combining these methods can create a PRA-based risk model. However, to be valid, such a model requires very detailed information and analyses, limiting the applicability of PRA during conceptual design [37].

To meet the need for a PRA that can be performed during the conceptual design phase, the function a product will perform is a useful engineering quality to discern, because it enables early analyses to be performed on the product before its form has been chosen. A functional model is a description of a product or process in terms of elementary operations or functions that are required to transform its input flows of energy, material, or signal into desired output flows. This model is a key step in the product design process. Functional models represent a form-independent blueprint of a product that can be derived early in the conceptual design phase from high-level customer needs [38].

Prior publications by the authors have presented the details of a mathematical mapping that links a product's functional model to potential failures, namely the function-failure design method (FFDM) [39,40]. The FFDM is a powerful tool that can present potential product failures during the conceptual design phase to help designers avoid these failures.

Probabilistic Risk Assessment (PRA) techniques such as Failure Modes and Effects Analysis (FMEA) [41], Event Tree Analysis [42], and Fault Tree Analysis [43] are useful tools to analyze risks of mature systems.

These techniques not only identify areas of potential failures (FMEA), but how those failures affect the rest of the system (ETA, FTA). This allows the design to be altered to account for these failures, either controlling or eliminating their danger to the system. Unfortunately, these techniques are not as successful at failure analysis during the conceptual design phase when the physical form of the system has yet to be determined. The method was then customized to analyze product requirements [44]. The product risk extension of FFDM, the risk in early design (RED) method [45], is significant because it will provide a classification of high-risk to low-risk function-failure combinations. The risks will not only contain failure information, they will provide a hierarchy of function-failure data relating both to the likelihood of failure and to the consequence of the failure. Risk in Early Design combines historical failure data with functional models to perform risk analysis as early as the conceptual design phase. The RED results include a listing of functions and their associated failure modes, likelihoods, and consequences. These results can then be plotted on a fever chart to better illustrate the risk level of the system.

However, these methods (FFDM and RED) of risk analysis are not complete, as these methods treat each function as an isolated event, not affected by any of the other functions in the design, they do not consider how failures affect the rest of the system.

Another method of risk analysis (Change Prediction Method – CPM) during design focuses on the effects of changing components in a currently existing design [46,47]. The underlying theory of this works that changing one component in a design effect other components as well. By using the opinions of a team of experts, data was collected on which components are dependent of each other, and with what likelihood a change in one component propagates to another. This data was used to create a model of the changes in the system. Each component also has a consequence of change, showing how much its change will affect the components dependent on it.

The function-based failure propagation method [13] is presented as a means to analyze chains of failures through the functions present in a system, making it applicable during early design stages before a product has assumed a physical form.

3. BACKGROUND

3.1 V procedural model in system architecture development

The V procedural model [48] (see Fig. 1) is a system's development model designed to simplify the understanding of the complexity associated with developing systems. In systems engineering it is used to define a uniform procedure for product or project development. The V-model is a graphical representation of the system's development lifecycle. It summarizes the main steps to be taken in conjunction with the corresponding deliverables within computerized system validation framework. The *VEE* represents the sequence of steps in a project life cycle development. It describes the activities and results that have to be produced during product development. The left side of the "V" represents the decomposition of requirements, and creation of system specifications. The right side of the *VEE* represents integration of parts and their verification. V model stands for "Verification and Validation".

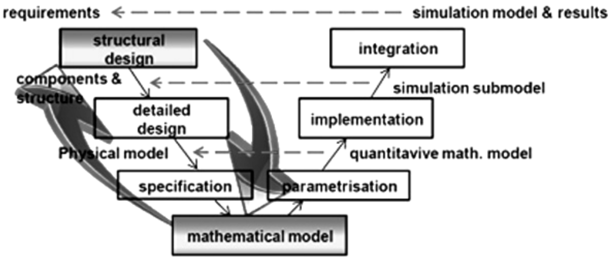


Figure 1. V procedural model [49]

The V model is also the proposed industrial solution for the development of mechatronic products, presented as the V model with the industrial guideline – VDI 2206 [50]. The ultimate goal is making the process, more concrete and forming solution variants into the principle. Since the ideas worked out for solution are usually not concrete enough to stipulate the final crossdomain concept, instead other issues have to be taken into account – e.g. fault susceptibility, weight, service life. The final assessment of end-solution variants are always subjected to technical and commercial criteria [51]. A complex mechatronic product is generally not produced within one-macro cycle, but within many macro cycles as a continuous macro cycle [51].

3.2 Quality Function Deployment (QFD)

Quality Function Deployment (QFD) was originated in the late 1960s to early 1970s, in Japan, by Professor Yoji Akao [11], and it is a formal technique commonly used in the early phase of the design process. It is an integrated set of tools for recording user requirements, engineering characteristics that satisfy these user requirements, and any trade-offs that might be necessary between the engineering characteristics. The QFD can help companies make the key trade-offs between what the customer wants and what the company can afford to build. By concentrating efforts on what will satisfy the customers and the company most, less time will be spent on redesign and modification of the product and process.

3.3 Multiple-Domain Matrix (MDM) approach

When applying the Multiple-Domain Matrix (MDM) (described in Section 2.1) to a complex system, the classification of implied domains and dependency types can help users keep track of the relevant system aspects and linkages. Users can then specify the most important domains. The alignment of a MDM automatically indicates all possible combinations of domains for subsequent specification of dependency types. Conversely, users can start with familiar dependency types and subsequently derive the corresponding domains of the complex system in question. In either case, the system of the MDM supports the complete capturing of all basic aspects of a complex system.

Once all domains are compiled (see Fig. 2), users can collect system elements within the domains separately. The right level of abstraction can be found when comparing the quantity of collected elements from different domains. This helps to assure that specific domains are not too detailed accidentally (e.g. because the user is better informed about a part of the system content).

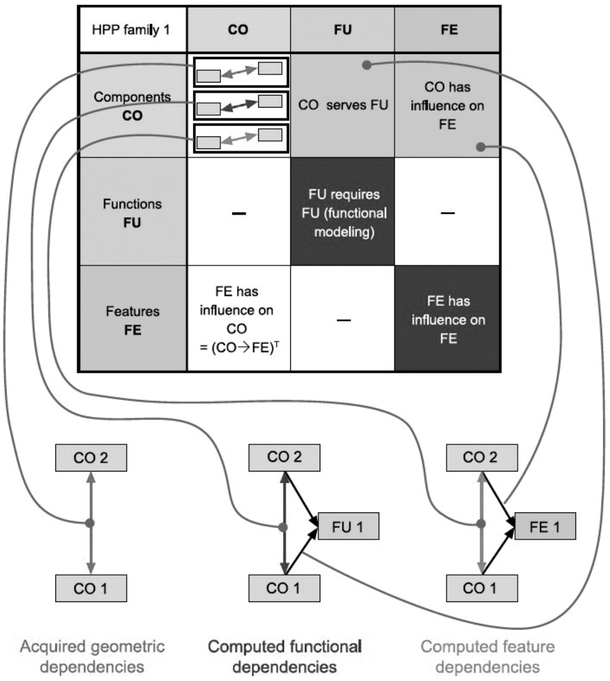


Figure 2. Deduction of different component dependency networks [12]

If within the DSM and DMM (described in Section 2.1), subsets more than one dependency type are identified, the MDM methodology allows these dependency types to be separated into different matrices and thus to keep track of complex linkages, which often appear even between the same elements.

If, in contrast, users can not identify a dependency type occurring in a specific DSM or DMM subset, then this part of the complex system can be excluded from subsequent acquisition of network dependencies.

Deduction of dependencies describes the procedure for computing dependencies between elements within one specific domain due to dependency chains leading through a further domain. This means that the deduction of dependencies can be used to compute the network of components which are linked because of existing dependencies between components and functions.

Figure 3 shows different dependency views (geometric, functional and feature) of the product structure (dependencies between components) of one specific family (example: high pressure pump).

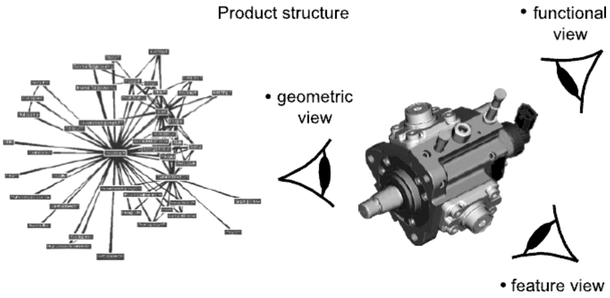


Figure 3. Different views of the product structure [12]

3.4 Theory of Technical Systems

According to the Theory of Technical Systems [52] which will serve as a theoretical foundation to this research, technical evolution, design and product development are explained as a response to those needs and requirements within human society for which, to be satisfied, an assistance of technical means was necessary. Such teleological view implies as a starting point to a development of a new product concept the definition of technical process as a process of technical system usage in which necessary effects must be delivered by technical product and human beings in order to enable purposeful transformation of operands. Built in the systemic reasoning, TTS models technical processes as transformation systems composed of a series of operations interrelated with operand flows and supported by necessary effects. Thus, the capability of delivering the necessary effects as the result of an internal transformation within technical system is considered as the function of technical system. Designers must consider different duties that human operator and technical system have to fulfil in order to enable transformation by reasoning about transformation variants within technical processes (see Fig. 4).

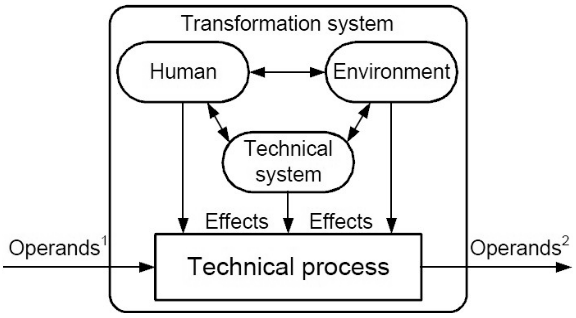


Figure 4. General model of transformation system according to TTS [52]

4. MOTIVATION

Considering the fact that the average product life cycle has shortened significantly within the past decade. As product life cycles are shortened, the product development life cycle becomes even more compressed. [53]. In order to survive in the steadily increasing market competition and, the basic threshold attributes of quality and function still have to be guaranteed. Since a hardly manageable number of individual products can result from the customizable attributes and their individual composition, ensuring the quality and function of first-time realized product families is a demanding task [54]. To focus on the design and development, most project budgets are committed during the design phase before the actual work tasks take place. Adequate planning is one of the dominators required to satisfy project quality, reduce financial and schedule risks, and help in the success of a project [55]. As a result, a systematic approach to product architecture development and evaluation is needed. How to make a robust plan for new product development has become an important concern for enterprises, especially the hi-tech industry.

The starting point of this paper was to study the concept choices, but in order to make good judgments and choose the best concept of product architecture, the concepts themselves need to be as good as possible and to have good concepts. The concept development process needs to be good with good methods and tools as well as good prerequisites. Conceptual involves translating customer needs into technical requirements and interface design, generating solutions and concepts, analyzing concept, evaluating concepts and eventually selecting a concept for further optimization.

Framework, presented here, focuses on improving the reliability of a product, which means making the system performance immune to variations, under uncertain operating conditions. Variations are everywhere, wanted and unwanted, but the unwanted variations can lessen the quality of the provided products and/or services. The aim with a robust design is not to try and eliminate the variations but rather to make the product insensitive to them.

5. PROPOSED FRAMEWORK

Proposed framework (see Fig. 5) is based on the classical V-model adapted from [49] (see Fig. 1).

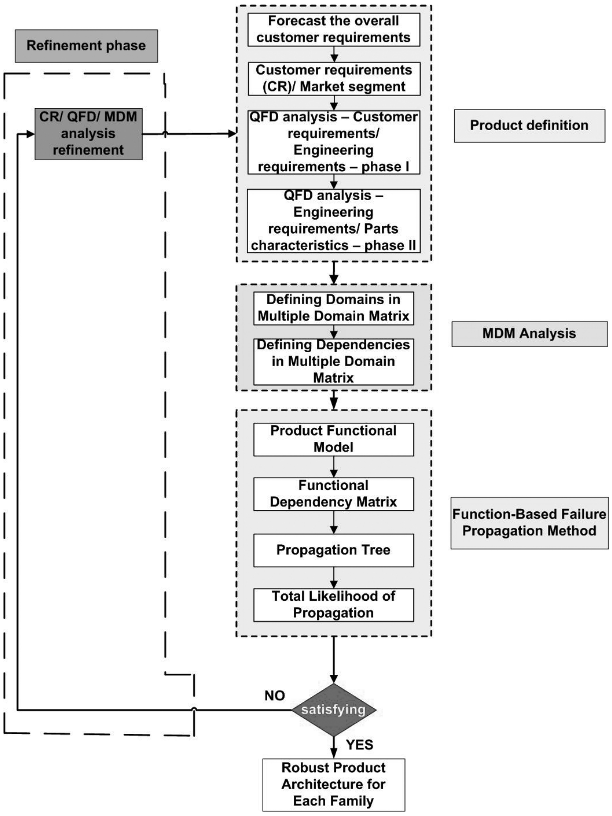


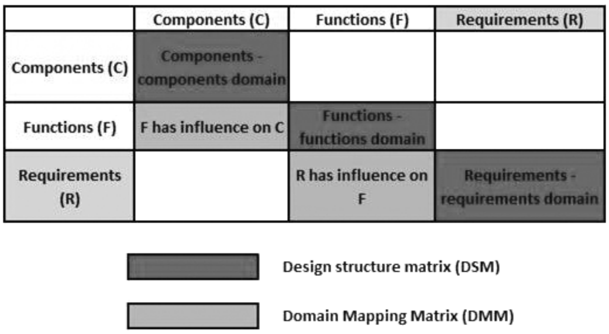
Figure 5. Proposed framework – M-FBFP

It can be seen that the feedback to the structural design is only based on the results. Besides the already mentioned disadvantages considering only this link, also a long period of time is passing between structural changing iterations.

The steps in the M-FPFM framework are introduced as follows:

1. Forecast the overall customer requirement. Map the overall design requirements regarding market segmentation grid. The market segmentation grid is an attention-directing tool providing a link between management, marketing, and engineering designers to help identify potential opportunities. Thus, the overall design requirement could be generated by integrating all of the market segmentation. In product definition phase, marketing and data collection work should be complete before modelling the procedure beginning.

2. Customer requirements/Market segment. We build up the overall customer requirements data including different requirements from different market segment grid. Market segmentation grid is created based on the size of the family, thus the market segment items are: small, medium and large. Importance data is given corresponding to the CRs and market segment as shown in Figure 6. Some importance is set to zero to represent that there is no requirement.



	Components (C)	Functions (F)	Requirements (R)
Components (C)	Components - components domain		
Functions (F)	F has influence on C	Functions - functions domain	
Requirements (R)		R has influence on F	Requirements - requirements domain

■ Design structure matrix (DSM)

▨ Domain Mapping Matrix (DMM)

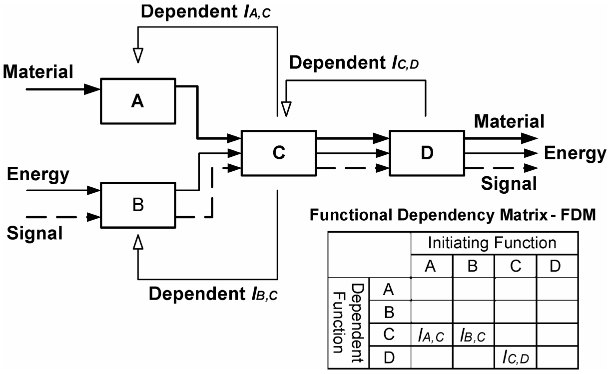
Figure 6. Domains and types of dependencies in the MDM

3. QFD analysis – phase I. In this step we import the overall customer requirements rating and customer requirement to house of quality (HoQ) to obtain the engineering characteristics. On the left side of QFD matrix, the importance value is presented by the overall rating from the previous step. The engineering requirements (ERs), which can satisfy CRs, are determined as shown on top and the relationships between CRs and ERs are given.

4. QFD analysis – phase II. Following the QFD procedure, input the ERs with weighting to the left side of phase II QFD, and the parts characteristics (PCs) are determined and the relationships between ERs and PCs are also obtained. Further, the interdependencies between PCs are represented on the roof of phase II QFD. The directions of interdependencies are illustrated in the grid.

5. MDM analysis. According to the procedure of structural complexity management introduced in Section 3.2, the system definition was carried out first, using the Multiple-Domain Matrix (MDM) [12]. Figure 6 shows the result of this first phase, specifically the considered domains and different types of dependencies between them. In order to identify the domains that play a decisive role in reaching the modularization goals mentioned, a deeper insight into the framework of a modular design has to be gained. A module is an assembly of several parts that can fulfil various functions. Characteristics of a module are the physical connections between its parts and, as a rule, the interchange ability of the unit. Modules can be developed, constructed, procured or produced independently of each other. The complexity becomes easier to control as there are fewer dependencies. Out of this, the key domains can be found, namely components, functions and requirements (according to TTS, see Section 3.3) [52]. In the next step, the types of dependencies between domains (inter-domain) were defined, as can be seen in the MDM. The specified MDM further shows (for our case study) that relevant dependency meanings have not been indicated for all possible domain combinations represented by the matrix subsets in Figure 6. The dependency meanings which are not shaded indicate that dependency information is available but is not required for further system investigation. Finally, meanings for the intra-domain dependencies of components, functions, and features were defined.

6. Product functional model. Functional modelling is a design tool that describes a product or system in terms of the functions it performs [56]. Since this model (see Fig. 7) is based on the function of a product rather than its components, this model can be generated before a physical artefact exists or components have been selected. The materials, signals, and energy are diagrammed as they flow from outside the system, through functions that act on those flows, and exit the system. These flows are determined from the high-level customer needs, and diagrammed as a black box model. This general function that makes up the black box is further defined into the functions that act on those flows, generating chains that show the process of one flow throughout the entire system. These chains are then combined to form the complete functional model of the system [56-58].



		Initiating Function			
		A	B	C	D
Dependent Function	A				
	B				
	C	$l_{A,C}$	$l_{B,C}$		
	D			$l_{C,D}$	

Figure 7. Product functional model and corresponding functional dependency matrix

7. Functional dependency matrix. To perform the function-based failure propagation method [13], a functional dependency matrix is generated from the functional model of the system using the flows as the common interface. Functions are directly dependant on the functions that are connected to them by one or more flows. For example, in Figure 7, Function D is dependent on Function C, and Function C is dependent on Functions A and B. Note that a function's dependency is independent of the type of flow and the number of flows from the previous functions. The functional dependency matrix is then populated with the likelihoods of a failure propagating to a particular function from one it is dependent on. The initiating functions are the functions that fail initially, and the dependent functions are those that the failure propagates to. In this example, the likelihood of propagation from C to D is $l_{C,D}$. For this method, the likelihood values are decimal values between zero and one, zero denoting no likelihood of propagation, and one representing certain propagation of the failure. This is done to allow the use of Boolean operators in the calculation of the total likelihood of propagation later on in the procedure. Likewise, each of the other functions' dependencies is used to populate the matrix. In places where there is no dependency, there is no likelihood of propagation, and thus the place filled in with a zero, (left black for figure clarity).

8. Propagation tree. Next, using the functional dependency matrix, propagation trees are built for each function in the model (see Fig. 8). These trees trace the path of a potential failure to each possible function that can propagate its failure to the end function. Each branch represents a different starting function, travelling to the same "root". In this example, functions A, B, and C can all propagate their failures to D. Function C propagates directly, and functions A and B propagate indirectly through C as seen in Figure 8. As shown in the figure, [A and C] or [B and C] or [C] can lead to failure of function D.

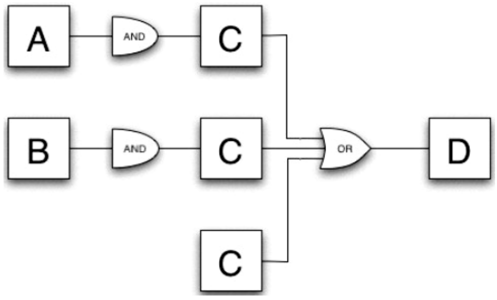


Figure 8. Propagation tree

9. Total likelihood of propagation. Finally, the total likelihood of propagation is calculated. Using the direct likelihoods from the functional dependency matrix and the trees generated, the total propagation likelihood is calculated using the Boolean operators "And" and "Or". Wherever there are multiple functions that failures can propagate from, the "Or" calculation is used. If a branch can only propagate a failure to a single function, the "And" calculation is used.

In order to properly use this method, historical data pertaining to failure propagation must exist. Finally, these failures were then tabulated into a matrix showing the number of times that each function pair had appeared. These numbers were then normalized, using the most frequently occurring failure propagation pair as the normalizing factor. In this way, each value collected becomes a decimal value between zero and one.

It is unlikely for each possible failure mode that a function might fail by has the same likelihood of propagation. Some failure modes might have higher or lower likelihoods of propagation than others.

However, for ease of calculation of those likelihoods, each failure mode for a function is assumed to have the same likelihood.

Using a modified form of the likelihood mapping from [59], the likelihood of each function pair was then calculated.

6. CASE STUDY – CLIMATE CHAMBER WITH HEAT REGENERATION

The purpose of the case study is to show how the M-FBFP Framework can support designers during conceptual design, using the example of the climate chamber with heat regeneration. Based on the product definition of the climate chamber, analysis was performed and according to the feedbacks in refinement phase and evaluation the final product architecture was proposed.

Climate chamber with heat regeneration is very often an integral part of HVAC for large objects (e.g. shopping malls, hotels or business objects). As within energy management (energy cost) the heat regeneration is a very desirable goal, fulfilment of the demand for shorter heating/cooling process time in respect to uncertain environmental conditions is very important. For our particular case study, initial working conditions for winter period are given as follows: outdoor (environment) temperature T_o = – 10 °C (which is average outdoor temperature for the town of Zagreb, Croatia) and air flow of q_{va} = 8.5 m^3/s. Our goal in the case study was to propose architecture for mentioned working conditions and check the stability of the proposed solution for given working conditions. Also, the proposed architecture should be tested to unexpected working conditions in order to simulate the chamber response to the temperature drop assumed at – 30 °C, simulating in such a way the uncertainty of the working conditions that are stochastically happening in the Zagreb area.

The first concept of the chamber architecture was developed based on the designer experience with similar systems, as presented in Figure 9. This scheme (for initial working conditions) is a starting point for understanding the relationship between the main subsystems.

The LOOMEO© (www.teseon.com) was used as a software tool for describing climate chamber subsystem structure for further analysis. Possibility for modularization of chamber's subsystems is determined by performing clustering operation over component domain in MDM. Figure 10 shows a detailed schema of climate chamber with heat regeneration, after a few iteration steps (refinement phases). Figure 11a shows a portion of the DSM matrix representation of the architecture after several steps of refinement including clustering has been conducted. Figure 11b shows the system in graph representation.

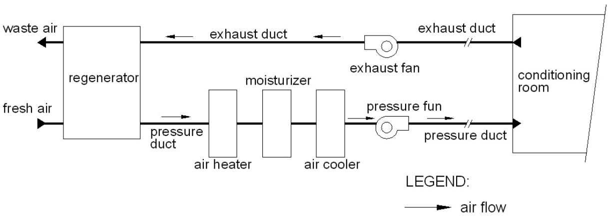


Figure 9. Simplified schema of starting conceptual design of climate chamber

The LOOMEO© (www.teseon.com) was used as a software tool for describing climate chamber subsystem structure for further analysis. Possibility for modularization of chamber's subsystems is determined by performing clustering operation over component domain in MDM. Figure 10 shows a detailed schema of climate chamber with heat regeneration, after a few iteration steps (refinement phases). Figure 11a shows a portion of the DSM matrix representation of the architecture after several steps of refinement including clustering has been conducted. Figure 11b shows the system in graph representation.

Based on the proposed module clustering [15], the behaviour modelling (risk) was conducted as follows. First, the initial proposal of the detailed schema for the entire system (climate chamber with heat regeneration) was developed (see Fig. 9). Three subsystems (two air heaters and moisturizer) were selected for performance of risk propagation testing in response to both initially imposed working conditions (T_o = – 10 °C) and uncertain working conditions (T_o = – 30 °C). The selected subsystems were chosen because they initiate the highest change of temperature (energy) in heat chamber (from environment in winter period to conditioning conditions).

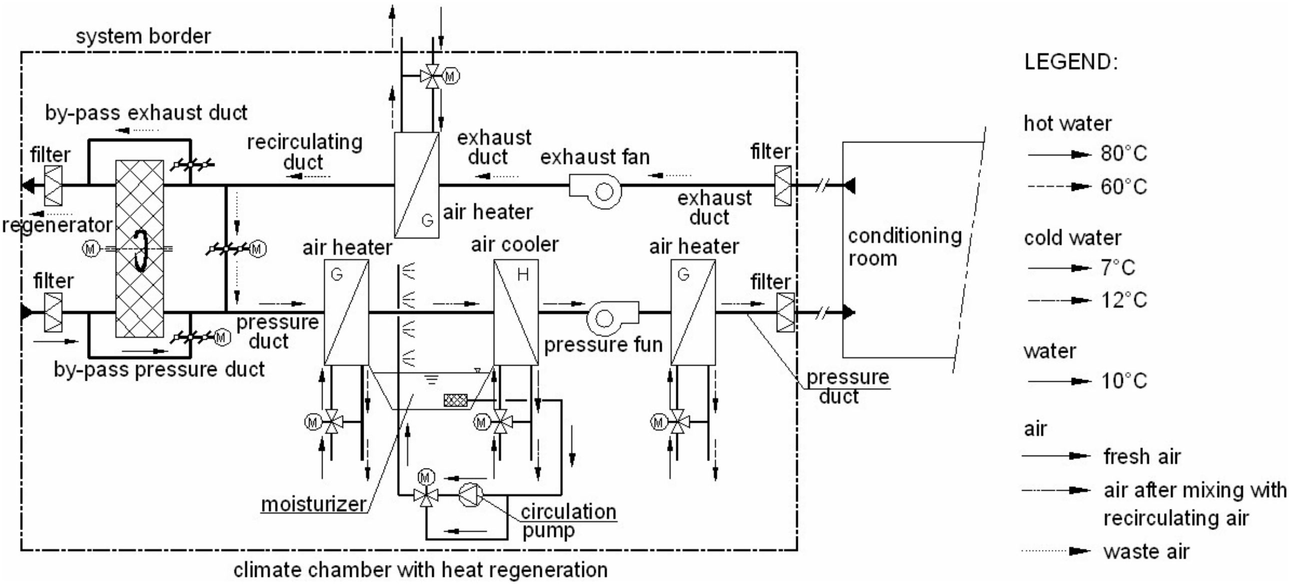


Figure 10. Detailed schema of proposed product architecture of climate chamber

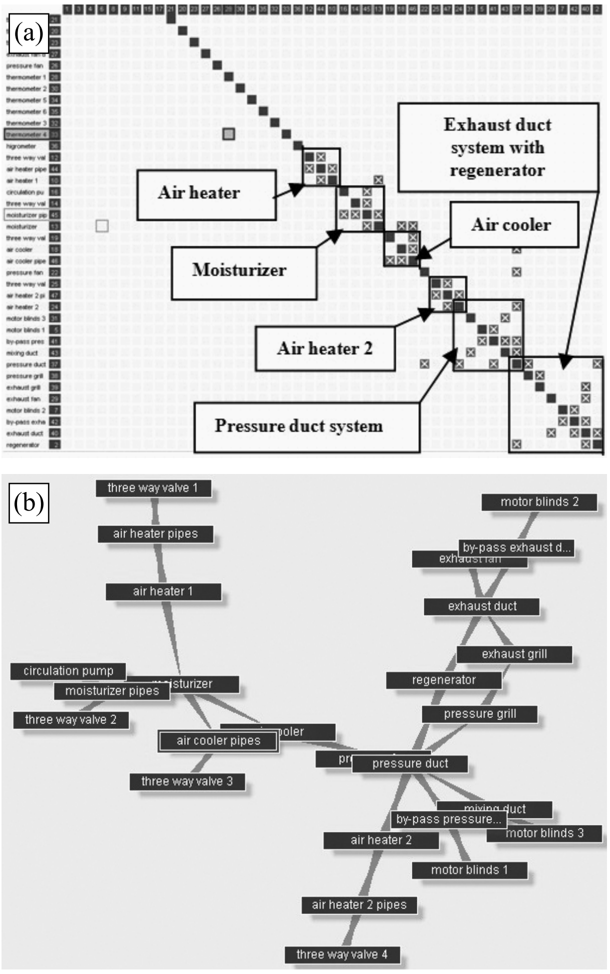


Figure 11. (a) Component domain representation with possible modules (subsystems) identified in proposed product architecture and (b) Graph representation of system (screenshots from LOOMEO®)

Following the proposed procedure through the M-FBFP framework and creating the entire "mix mixture" tree, the individual likelihoods of each branch and the most likely branch of the tree can be determined. Whenever many linear branches to a tree occur, the shortest branch will be the most likely occurrence. The total likelihood for this function is the likelihood that any of these branches will occur.

7. CONCLUSION AND FUTURE RESEARCH

This paper proposes an M-FBFP framework which could help designers with risk of product behaviour and prediction during product architecture development in early design stages. The proposed framework offers the opportunity to simplify, improve and accelerate development process for products that are facing uncertain conditions during operating phase, like the test on the case involving climate chamber with heat regeneration. Based on the proposed framework, it is possible to analyze different product architecture arrangements and subsystems or components interactions against the changes in architecture elements, through risk propagation. It also enables designers to make refinement on the existing subsystem structures, adding new features to them and predicting new behaviour based on the new features. It also enables the designers to see the impact on the other domains (components and requirements), and allow their refinement and change. However, it does not provide the consequences of risk for the functions in the model, requiring another method to provide the data it lacks.

Future research will be continued in the direction of the development of interface between structural and behaviour model, enabling in such a way the automatic indication of the problems occurring on subsystems level as the result of risk analysis. Also, this framework may be used in mechatronics systems development, where we can connect it with classical sequential product design procedures and domain isolated product development (s.c. over-the-wall syndrome) with substantial cost and time reduction.

ACKNOWLEDGMENT

This work is partially supported by the Ministry of Science of Serbia as Grant No. 35006.

REFERENCES

[1] Hubka, V. and Eder, W.E.: *Engineering Design: General Procedural Model of Engineering Design*, Springer-Verlag, Berlin, 1992.

[2] Hansen, C.T. and Andreasen, M.M.: Two approaches to synthesis based on the domain theory, in: Chakrabarti, A. (Ed.): *Engineering Design Synthesis*, Springer-Verlag, London, pp. 93-108, 2002.

[3] Cooper, D.F. and Chapman, C.B.: *Risk Analysis for Large Projects: Models, Methods and Cases*, John Wiley & Sons, Chichester, 1987.

[4] Clarkson, P.J., Simons, C. and Eckert, C.: Predicting change propagation in complex design, Journal of Mechanical Design, Vol. 126, No. 5, pp. 788-797, 2004.

[5] Mitchell, M.: *Complexity: A Guided Tour*, Oxford University Press, Oxford, 2009.

[6] Reich, Y. and Ziv Av, A.: Robust product concept generation, in: *Proceedings of the 15th International Conference on Engineering Design (ICED 2005)*, 15-18.08.2005, Melbourne, Australia, Paper 461.1.

[7] Mattila, M., Tallberg, T., Vannas, V. and Kivistö-Rahnasto, J.: Fatalities at advanced machines and dangerous incidents at FMS implementations, International Journal of Human Factors in Manufacturing, Vol. 5, No. 3, pp. 237-250, 1995.

[8] Pratt, S.G., Kisner, S.M. and Moore, P.H.: Machinery-related fatalities in the construction industry, American Journal of Industrial Medicine, Vol. 32, No. 1, pp. 42-50, 1997.

[9] Gardner, D., Cross, J.A., Fonteyn, P.N., Carlopio, J. and Shikdar, A.: Mechanical equipment injuries in small manufacturing businesses, in: *Proceedings of the 13th Triennial Congress of the International Ergonomics Association, Vol. 7*, 29.06-04.07.1997, Tampere, Finland, pp. 109-111.

[10] Gauthier, F. and Charron, F.: A structured procedure of risk analysis and control for safety

integration in machinery design, Journal of Engineering Design, Vol. 13, No. 2, pp. 77-99, 2002.

[11] Akao, Y.: *QFD – Quality Function Deployment*, Verlag Moderne Industrie, Landsberg, 1992.

[12] Lindemann, U., Maurer, M. and Braun, T.: *Structural Complexity Management: An Approach for the Field of Product Design*, Springer-Verlag, Berlin, 2009.

[13] Krus, D. and Grantham Lough, K.: Risk due to function failure propagation, in: *Proceedings of the 16th International Conference of Engineering Design (ICED 2007)*, 28-31.08.2007, Paris, France, Paper 513.

[14] Steward, D.V.: On an approach to techniques for the analysis of the structure of large systems of equations, SIAM Review, Vol. 4, No. 4, pp. 321-342, 1962.

[15] Steward, D.V.: The design structure system: A method for managing the design of complex systems, IEEE Transactions on Engineering Management, Vol. 28, No. 3, pp. 71-74, 1981.

[16] Kehat, E. and Shacham, M.: Chemical process simulation programs – 2: Partitioning and tearing of systems flowsheets, Process Technology International, Vol. 18, pp. 115-118, 1973.

[17] Warfield, J.N.: Binary matrices in system modelling, IEEE Transactions on Systems, Man and Cybernetics, Vol. 3, No. 5, pp. 441-449, 1973.

[18] Hauser, J.R. and Clausing, D.: The house of quality, Harvard Business Review, Vol. 66, No. 3, pp. 63-73, 1988.

[19] Kusiak, A. and Park, K.: Concurrent design: decomposition of design activities, International Journal of Production Research, Vol. 28, No. 10, pp. 1883-1900, 1990.

[20] Eppingera, S.D.: Model-based approaches to managing concurrent engineering, Journal of Engineering Design, Vol. 2, No. 4, pp. 283-290, 1991.

[21] Yassine, A., Falkenburg, D. and Chelst, K.: Engineering design management: An information structure approach, International Journal of Production Research, Vol. 37, No. 13, pp. 2957-2975, 1999.

[22] Browning, T.R.: Applying the design structure matrix to system decomposition and integration problems: A review and new directions, IEEE Transactions on Engineering Management, Vol. 48, No. 3, pp. 292-306, 2001.

[23] Allen, T.T.: *Introduction to Engineering Statistics and Six Sigma: Statistical Quality Control and Design of Experiments and Systems*, Springer-Verlag, London, 2006.

[24] Kusiak, A., Tang, C.-Y. and Song, Z.: *Identification of Modules With an Interface Structure Matrix*, Working Paper ISL_04/2006, Intelligent Systems Laboratory, The University of Iowa, Iowa City, 2006.

[25] Danilovic, M. and Börjesson, H.: Managing the multiproject environment, in: *Proceedings of the Third Dependence Structure Matrix (DSM) International Workshop*, 29-30.10.2001, Cambridge, USA.

[26] Danilovic, M. and Börjesson, H.: Participatory dependence structure matrix approach, in: *Proceedings of the Third Dependence Structure Matrix (DSM) International Workshop*, 29-30.10.2001, Cambridge, USA.

[27] Danilovic, M. and Sigemyr, T.: Multiplan – A new multi-dimensional DSM-tool, in: *Proceedings of the 5th Design Structure Matrix Workshop*, 22-23.10.2003, Cambridge, UK.

[28] Danilovic, M. and Browning, T.: A formal approach for domain mapping matrices (DMM) to complement design structuring matrices (DSM), in: *Proceedings of the 6th Design Structure Matrix (DSM) International Workshop*, 12-14.09.2004, Cambridge, UK.

[29] Eppinger, S. and Salminen, V.: Patterns of product development interactions, in: *Proceedings of the 13th International Conference on Engineering Design (ICED 2001)*, 21-23.08.2001, Glasgow, UK, pp. 283-290.

[30] Dong, Q. and Whitney, D.: Designing a requirement driven product development process, in: *Proceedings of the DETC 2001: 13th International Conference on Design Theory and Methodology*, 09-12.09.2001, Pittsburgh, USA, pp. 343-351.

[31] Suh, N.P.: *The Principles of Design*, Oxford University Press, Oxford, 1990.

[32] Bongulielmi, L., Henseler, P., Puls, C. and Meier, M.: The K- & V-matrix method – An approach in analysis and description of variant products, in: *Proceedings of the 13th International Conference on Engineering Design (ICED 2001)*, 21-23.08.2001, Glasgow, UK, pp. 571-578.

[33] Puls, Ch., Bongulielmi, L., Henseler, P. and Meier, M.: Management of different types of configuration knowledge with the K- & V-matrix and Wiki, in: *Proceedings of the International Design Conference – DESIGN 2002*, 14-17.05.2002, Dubrovnik, Croatia, pp. 403-410.

[34] Puls, C.: *Die Konfigurations- & Verträglichkeitsmatrix als Beitrag zum Management von Konfigurationswissen in KMU*, PhD thesis, Eidgenössischen Technischen Hochschule Zürich, Zurich, 2002.

[35] Yassine, A., Whitney, D., Daleiden, S. and Lavine, J.: Connectivity maps: Modeling and analysing relationships in product development processes, Journal of Engineering Design, Vol. 14, No. 3, pp. 377-394, 2003.

[36] Gray, C.F. and Larson, E.W.: *Project Management: The Managerial Process*, McGraw-Hill, New York, 2000.

[37] Kumamoto, H. and Henley, E.J.: *Probabilistic Risk Assessment and Management for Engineers and Scientists*, IEEE Press, New York, 1996.

[38] Hirtz, J., Stone, R.B., McAdams, D.A., Szykman, S. and Wood, K.L.: A functional basis for engineering design: Reconciling and evolving previous efforts, Research in Engineering Design, Vol. 13, No. 2, pp. 65-82, 2002.

[39] Tumer, I.Y. and Stone, R.B.: Mapping function to failure mode during component development, Research in Engineering Design, Vol. 14, No. 1, pp. 25-33, 2003.

[40] Stone, R.B., Tumer, I.Y. and Van Wie, M.: The function-failure design method, Journal of Mechanical Design, Vol. 127, No. 3, pp. 397-407, 2005.

[41] *MIL-STD-1629A: Procedures for Performing a Failure Mode, Effects and Criticality Analysis*, Department of Defense, Washington, 1980.

[42] Frank, M.V.: Re-entry safety: Probability of fuel release, in: *Proceedings of European Safety and Reliability Conference (ESREL '99)*, 13-17.09.1999, Munich, Germany, pp. 351-360.

[43] Vesely, W.E., Goldberg, F.F., Roberts, N.H. and Haasl, D.F.: *Fault Tree Handbook*, U.S. Nuclear Regulation Commission, NUREG 0492, Washington, 1981.

[44] Grantham Lough, K., Stone, R. and Tumer, I.Y.: The risk in early design method, Journal of Engineering Design, Vol. 20, No. 2, pp. 155-173, 2009.

[45] Grantham Lough, K.A.: *Risk in Early Design (RED)*, PhD thesis, Missouri University of Science and Technology, Rolla, 2005.

[46] Eckert, C., Clarkson, P.J. and Zanker, W.: Change and customisation in complex engineering domains, Research in Engineering Design, Vol. 15, No. 1, pp. 1-21, 2004.

[47] Jarratt, T.A.W., Clarkson, P.J., Parks, G. and Eckert, C.M.: Use of Monte Carlo methods in the prediction of change propagation, in: *Proceedings of the Engineering Design Conference (EDC 2002)*, 09-11.07.2002, London, UK, pp. 487-498.

[48] Forsberg, K., Mooz, H. and Cotterman, H.: *Visualizing Project Management*, John Wiley & Sons, New York, 2005.

[49] Nordwig, A.: Object-oriented development of simulation models for complex hybrid systems, in: Engell, S., Frehse, G. and Schnieder, E. (Eds.): *Modelling, Analysis, and Design of Hybrid Systems*, Springer-Verlag, Berlin, pp. 131-149, 2002.

[50] VDI 2206 Design methodology for mechatronic systems, 2004.

[51] Vasić, V.S. and Lazarević, M.P.: Standard industrial guideline for mechatronic product design, FME Transactions, Vol. 36, No. 3, pp. 103-108, 2008.

[52] Hubka, V. and Eder, W.E.: *Theory of Technical Systems: A Total Concept Theory for Engineering Design*, Springer-Verlag, Berlin, 1988.

[53] Williams, C.B.: *Platform Design for Customizable Products and Processes with Non-Uniform Demand*, MSc thesis, George W. Woodruff School of Mechanical Engineering, Georgia Institute of Technology, Atlanta, 2003.

[54] Lindemann, U. and Maurer, M.: Early evaluation of product properties for individualised products, International Journal of Mass Customisation, Vol. 1, No. 2-3, pp. 299-314, 2006.

[55] Shtub, A., Bard, J.F. and Globerson, S.: *Project Management: Processes, Methodologies, and Economics*, Prentice-Hall, London, 2004.

[56] Otto, K. and Wood, K.: *Product Design: Techniques in Reverse Engineering and New Product Development*, Prentice Hall, New York, 2001.

[57] Dym, C.L. and Little, P.: *Engineering Design: A Project-Based Introduction*, John Wiley & Sons, Honoken, 2004.

[58] Cross, N.: *Engineering Design Methods: Strategies for Product Design*, John Wiley & Sons, Chichester, 2000.

[59] Grantham Lough, K., Stone, R.B. and Tumer, I.: The risk in early design (RED) method: Likelihood and consequence formulations, in: *Proceedings of the ASME 2006 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (IDETC/CIE2006)*, 10-13.09.2006, Philadelphia, USA, pp. 1119-1129.

РОБУСНИ РАЗВОЈ АРХИТЕКТУРЕ ПРОИЗВОДА КОМБИНАЦИЈОМ МАТРИЧНО БАЗИРАНИХ ПРИСТУПА И ФУНКЦИЈСКИ БАЗИРАНОЈ МЕТОДИ ПРОПАГАЦИЈЕ НЕДОСТАТКА – М-ФБФП ОКРУЖЕЊЕ

Крешимир Осман, Драги Стаменковић, Михаило Лазаревић

Овај рад предлаже окружење М-ФБФП са циљем да се помогне конструкторима приликом решавања проблема са ризиком, који се појављује у архитектури производа, те са ефектима при раду у неизвесним радним условима. Предложено окружење представља комбинацију матрично базираних приступа (*QFD* и *MDM*) и ФБФП методе. *QFD* приступ је интегрисани скуп алата за прикупљање захтева корисника, инжењерских карактеристика, које задовољавају те захтеве, те осталих веза између инжењерских карактеристика, док се *MDM* приступ примењује за моделирање структура и зависности између домена, те унутар самих њих. ФБФП метода се примењује на функционалном нивоу, те нам даје потенцијалну информацију о недостатку или грешци у самим функцијама производа и његовим подсистемима током фазе пројектовања. Као резултат овог окружења, могуће је спровести анализу ризика у подсистемима производа и самим тим добити повратну информацију, да ли је нешто у предложеној архитектури потребно додати или променити. У оквиру овог рада приказан је пример клима-коморе са регенерацијом топлоте, чиме је приказан принцип рада предложеног окружења.