



HAL
open science

Physical interface ontology for management of conflicts and risks in complex systems

Vincent Holley, Marija Jankovic, Bernard Yannou

► To cite this version:

Vincent Holley, Marija Jankovic, Bernard Yannou. Physical interface ontology for management of conflicts and risks in complex systems. *Concurrent Engineering: Research and Applications*, 2014, 22 (2), pp.148-161. 10.1177/1063293X14520760 . hal-01216800v2

HAL Id: hal-01216800

<https://centralesupelec.hal.science/hal-01216800v2>

Submitted on 20 Sep 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Physical Interface Ontology for Management of Conflicts and Risks in Complex Systems

Abstract

Interface definitions are important in complex system design, as they contribute to the definition and satisfaction of functional flows. Design Structure Matrix approaches have only recently focused on integrating possible interface types in conceptual design. In order to address the issue of interface definition, a physical interface data model is proposed in order to define the interface design space. Moreover, this data model is integrated into complex system architecture definitions in order to support collaborative conflict detection. The data in the physical interface data model is used as a basis for risk identification in the collaborative design space and associated shared design parameters.

Key words: Design Structure Matrix, Interface ontology, Collaboration management, Conflict management

1. Introduction

Matrix-based approaches are known to be particularly suited to concept generation and evaluation in complex system design (1-3). In order to explore and evaluate possible concepts, matrix-based approaches generally use system decomposition to identify possible connections between subsystems (1, 2). Only recently, some studies have focused on different types of connections in order to integrate them into design space exploration (2, 3). The data used in these approaches consists mostly of quantitative evaluations of connection strength by different experts. The problem is that the design team is not able to easily identify shared design parameters that define this interface design across disciplines.

In this work, we propose to integrate the ontologies of different physical connection types into a Design Structure Matrix (DSM) so as to preserve the option to computationally explore all possible configurations and to identify shared design parameters across disciplines. In combination with other types of matrices, these new data help identify risks linked to the collaboration design space and the design parameters that define it. The Multiple-Domain Design Scorecards (MDDS) (4) approach is proposed as a methodological support mechanism for early design concept exploration. This matrix-based methodology is based on gathering data within three matrices: a Functional Flow - Design Mapping Matrix (FF-DMM), a Physical Connections - Design Structure Matrix (PC-DSM) and a Voice of Design Department Matrix (VoDD).

In section 2, an overview of existing methods and approaches integrating different interface-related elements is presented and discussed. In section 3 and 4, we introduce Multi-Design Domain Scorecards (MDDS) and define a physical interface data model within the PC-DSM matrix. Section 5 presents a case study conducted in an oil company; advantages and limitations in the deployment are discussed. Finally, section 6 concludes with our contribution to interface data integration for concept generation.

2. Background: Integrating Product Connections into Concept Generation and Evaluation

The Design Structure Matrix Method has been known to support several operating principles for complex system design (5)(6-12). Lindemann et al. (13) classify matrices: intra-domain, inter-domain or multiple-domain. Inter-domain matrices or DSMs map relationships between like elements (product components, performance attributes, engineering requirements). Several scholars (2, 14) have provided very good overviews of different dependency types for these matrices. Inter-domain matrices (also called Domain Mapping Matrices, or DMMs) (15), are matrices that consider dependencies between different domains (e.g., between functional requirements and product components). Finally, Multiple-Domain Matrices (MDMs) represent dependencies between elements of several domains. Browning (16) identified two types: static

and time-based. Static matrices represent elements that coexist, while time-based matrices order rows and columns to indicate flows in time.

Most approaches consider existing interfaces during conceptual design, but only some define these interconnections. Hellenbrand and Lindemann (17) used DSM proposed a compatibility matrix that captures possible compatibilities between two different product components and their weights. In the proposed consistency algorithm, all possible product concepts are generated when an interface is identified. Pimmler and Eppinger (2) integrated four types of interaction in DSMs: spatial, energy, information and material. A weighted scale is used to represent a coupling coefficient (from -2 to +2). Wyatt et al. (1) proposed using matrix based approaches to define the rules governing concept generation. This approach proposes a component DSM in order to compare different design concepts. In an effort to generate concepts, the authors considered different component and connection types. Concept generation rules are expressed as a mapping component and component types, and component types and connection types, as they define concept generation constraints. The authors also proposed a network-based approach (3) that integrates different types of connections in order to support product architecture design: structural, behavioral, assignment and geometrical. These connection types are integrated into a network representation of the product architecture that is used for constraint-based classification and visualization of possible architectures. However, information about constraints is limited, and impact on the global architecture performance cannot be deduced and analyzed. Sharman and Yassine (9) integrated three levels of system representation using DSM approaches: global design rules, interface rules and intramodule design rules. The existence of product interfaces is an input for determining architecture visibility and dependency. Albers et al. (18) proposed using the Contact and Channel Approach (C&C-A) for system architecture generation. C&C models represent the interactions between systems, subsystems and parts through working surface pairs, or WSP (geometric interfaces between artifacts or between an artifact and its environment) and channel and support structures, or CSS (physical components or volumes of liquids, gases or fields directly connecting two WPS). Information related to interfaces is not integrated into architecture-level behavior assessment. Moullec et al. (19) developed a Bayesian network-based model for architecture generation. The possible connections between product elements are modeled as compatibility constraints that can be crisp or uncertain. Integrating uncertainties into product connection supports uncertainty estimation at the global architecture level. As in previous examples, the interfaces are characterized as nodes in Bayesian networks and do not integrate parameter definitions that are important for a particular interface. Rahmani and Thomson (20) propose an interface model using the Model Based System Approach. The aim of this model is to represent the interface space using UML language and permitting connection between different sub-systems.

In the majority of previously discussed approaches, different types of interfaces are proposed, but the interface design space is not defined. As interfaces also contribute to functional flows, it is necessary to characterize this design space in order to identify the impact of one interface on overall system functioning. In order to bridge this gap, the MDDS approach integrates three types of matrices and combines them with data related to the definition of physical interfaces. This data is used in identifying conflicting design parameters and managing them in an adequate Collaborative Failure Modes, Effects and Criticality Analysis (Collaborative FMECA) (21).

3. Multiple-Domain Design Scorecards: Integrating Interface Knowledge

Multiple-Domain Design Scorecard MDDS method is methodology for architecture generation and evaluation method (4) based on the semantic enrichment of three matrices related to functional flows (FF-DMM: functional flow – design mapping matrix), system structures (PC-DSM: physical connections – design structure matrix) and performance evaluations (VoDD: voice of design department matrix). Within this method a particular focus is made on identifying of potential difficulties and conflicts between different design departments related to interfaces and this is an object of this paper. As interfaces define functional flows, the aim of this approach

is to capture expert knowledge on the interface design space, specifically on design trade-offs and risk management related to the collaboration. The FF-DMM matrix is designed to capture data generated during functional analysis (22). Information related to concept brainstorming, and possible modules, technical solutions and structural interfaces is captured within the PC-DSM matrix. Finally, the technical performance brainstorming step allows experts to discuss possible performance outcomes for the modules and the ability to achieve performance goals at the system level. Based on these three steps, the MDDS method proposes six Design Assessment Cards (DACs) (4). DACs are special connectivity maps (23) used to support design process and identify possible trade-offs.

3.1 Defining the Functional Flows

The functional flow – design mapping matrix (FF-DMM) is a cross-functional flow and concept structure mapping matrix populated with data captured during the functional analysis step (22). It is an enriched DMM matrix representing different functional flows as relationships between functions and technical solutions. Technical solutions in this case represent different possible design solutions for modules (i.e., product subsystems). The resulting FF-DMM matrix is a numerical DMM matrix where numbers (Figure 1) represent the functional flow directions of one function through technical solutions. Determining the direction of functional flows helps designers identify product interfaces.

	Chassis		Box		Connector	Collar	Substrate	Electronics	Wiring
	I	Delta	HPHT	Pivot					
Generate power supplies					3		2	1	4
Dissipate heat	4	4	3	3			2	1	
	4	4	3	3	5	2	1		

Figure 1: Functional Flow – Design Mapping Matrix (FF-DMM)

For instance, the flow of the function *dissipate heat* starts from electronics (1), and flows to the substrate (2), box (3) and chassis (4). Technical solutions *I* and *Delta* represent technical solutions for the module *chassis*. Moreover, FF-DMM also integrates concurrent functional flows. *Dissipate heat* can be supported by two concurrent systems: (1) *electronics*, *substrate*, *box* and *chassis*; or (2) *electronics*, *substrate*, *box*, *chassis* and *collar*. An FF-DMM data model showing different data types and their relationships is shown in Figure 2.

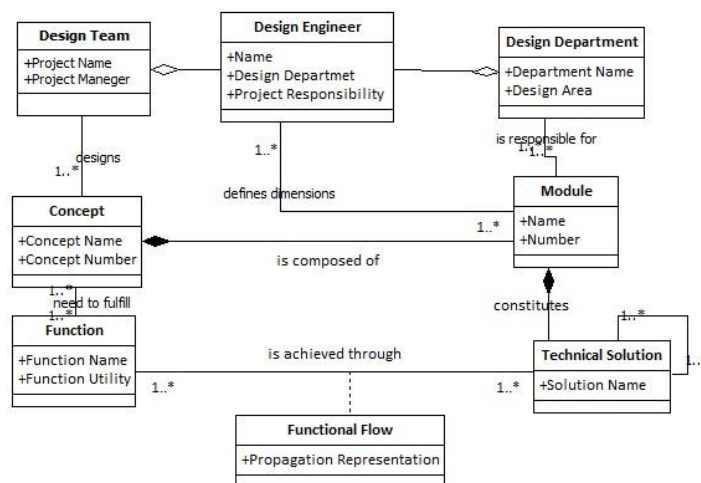


Figure 2: FF-DMM Model Ontology

The *design team* is the team in charge of product design. It is identified by its *project name* and its *project manager name*. Therefore, the *design team class* has an aggregation relationship with the *design engineer class*. The design engineer class is defined by *name*, *department* and *project responsibility*. Design engineers have specific areas of expertise and are members of design departments. The *design department class* is defined by *name* and *design area*. In general, design projects are multidisciplinary, and each design department is responsible for designing a specific module. The *module class* represents different subsystems. It is defined by *module name* and *module number*. For each module, in order to satisfy different technology and performance criteria, several design alternatives are examined and proposed, *technical solutions*. Since a design team is in charge of product design, the design team class has a composition relationship with the *concept class*. A concept has a *concept name* and a *concept number*. The concept class is comprised of several modules. Each *concept* is defined by a system's objectives and function. Therefore, for each concept there are several functions that describe the system's mission. *Function* is defined by a *function name* and *function utility* describing the goal of the function. For each function, several technical solutions are required. The function class is therefore associated with technical solutions. This association also is defined by the *functional flow class*. The functional flow class defines the sequence of technical solutions through system architecture.

3.2 High-Level Architecture Description as Result of Design Team Brainstorming

The PC-DSM matrix summarizes possible concept configurations. This matrix is a semantically-enriched square matrix that integrates physical interface typology (Figure 3). Based on their typologies, these physical interfaces are linked to their corresponding data model (Figure 4). Both rows and columns list modules and their technical solutions (concept breakdowns as well as other possible technical solutions). Some of the types of physical interfaces that are identified include: *E* for elastomer, *V* for screw, *S* for silicone, *G* for glue, and *F* for fitting. For example, the physical interface between a *Pivot* box and a *Delta* chassis can be either screws (*V*) or silicone (*S*).

		Chassis		Box (Packaging)		Connector	Collar	Substrate	Electronics	Wiring
		I	Delta	HPHT	Pivot					
Chassis	I	Not applicable		E or V	V	No assembly	V or E	No assembly	No assembly	No assembly
	Delta			E or (V&S)	V or S	No assembly	V or E	No assembly	No assembly	No assembly
Box (packaging)	HPHT		Not applicable		V	No assembly	G	No assembly	No assembly	
	Pivot				V	No assembly	G	No assembly	No assembly	
Connector					Not applicable	No assembly	B	B	P	
Collar						Not applicable	No assembly	No assembly	No assembly	
Substrate							Not applicable	CMS	No assembly	
Electronics								Not applicable	No assembly	
Wiring									Not applicable	

Figure 3: PC-DSM Matrix

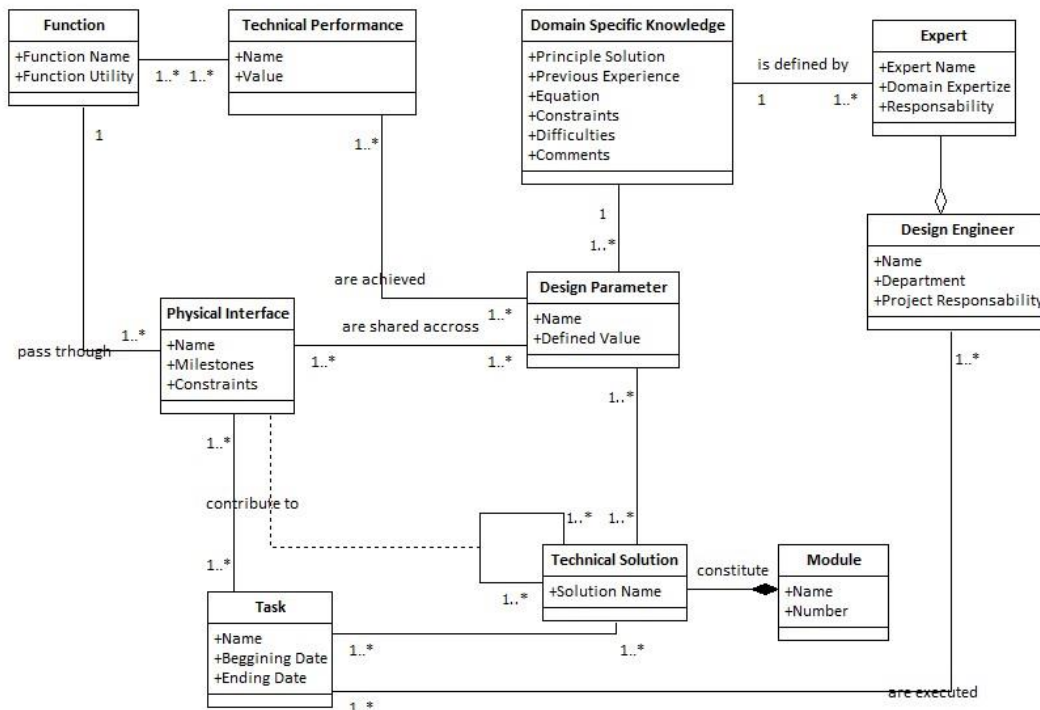


Figure 4: Physical Interface Data Model

A *physical interface class* represents possible physical interfaces between technical solutions in PC-DSM (Figure 4). It is defined by its *name*, *milestones* and *constraints*. Linked to it are *design parameters* that are shared across physical interfaces. For each design parameter, *name* and *value* are defined. Design parameters are linked with *domain specific knowledge*, which is defined by *principle solution*, *previous experience*, *equation*, *constraints*, *difficulties* and *comments*. This class contains a designer's (expert's) knowledge about a given design parameter. The *expert class* is an aggregated class of design engineers, defined by *expert name*, *domain expertise* and *responsibility*. Design parameters contribute to the achievement of technical performance, which are performance criteria defined at the system level. Therefore, the *technical performance class* is associated with the *function class*. Of course, the function class is also related to physical interface class. One function may have several physical interfaces. For example, if glue (*G*) is used to connect the substrate to the box, the physical interface data show that three design parameters are shared across the interfaces: surface area, viscosity and thickness. Moreover, previous experience and testing data can be found in guideline number 100236239 (company reference document).

3.3 Integrating Performance Evaluation and Design Department Constraints

The Voice of Design Department (VoDD) matrix is a QFD-like matrix defined in order to capture designers' assessments of alternate technical solutions (Figure 5). It is also used to evaluate the potential of each technical solution to meet required performance criteria.

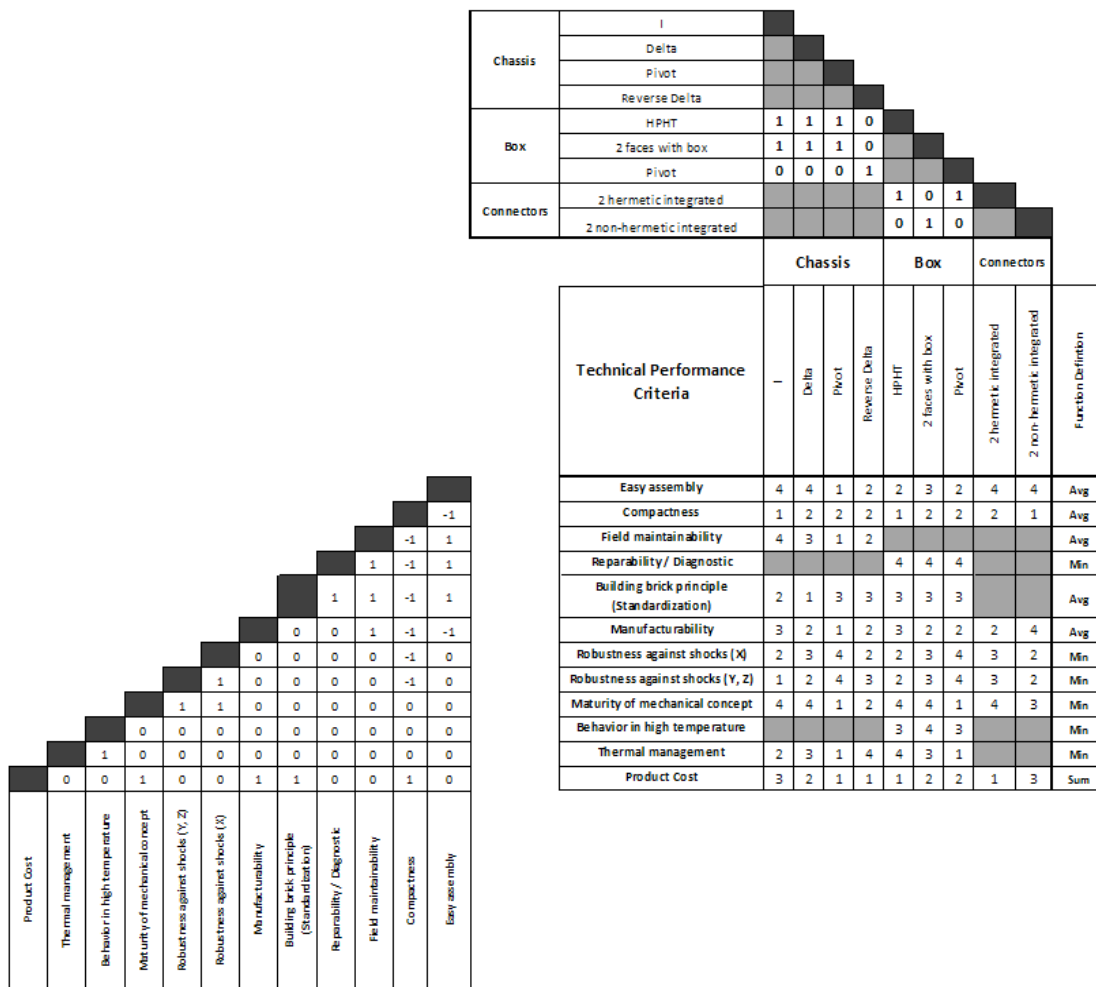


Figure 5: Voice of Design Department VoDD Matrix

Columns correspond to modules and their technical solutions. Rows list the expected technical performance in order to evaluate functions as well as to differentiate each technical solution. The contribution of technical solutions to the fulfillment of technical performance criteria represents the body of the matrix in the form of scale-based evaluations.

Table 1: Maturity scale for the estimation of technical performance satisfaction.

Maturity Scale		
General Guide		Adaptation to “shock acceleration” performance of PEC case study
4	Never done in the world	Over 50G of shock resistance
3	Done by business competitor	Between 30G and 50G
2	Already done in our industry	Between 10G and 30G
1	Can be done without any problem	Below 10G of shock resistance

As seen in Table 1, there are four levels in this scale: 1- performance objectives are achievable without any major problems; 2 - performance objectives are known to have been achieved in

previous projects; 3 - performance objectives have been achieved by competitors, but not by the company; and 4 - performance objectives have never been achieved by any company worldwide. The left side of the VoDD matrix defines the correlation between technical performance criteria, either positive or negative. On the right side of the matrix specifies the manner of evaluating a function, based on technical performance evaluations of the associated concepts. Each technical performance is associated with one of the following:

- *Min* indicates that the technical performance of the given concept is defined by the minimum value of the contribution to its technical solutions,
- *Max* indicates that performance is defined by the maximum value of its contribution to technical solutions,
- *Avg* indicates that performance is defined by the average value of its contribution to technical solutions, and
- *Sum* indicates its total contribution to technical solutions.

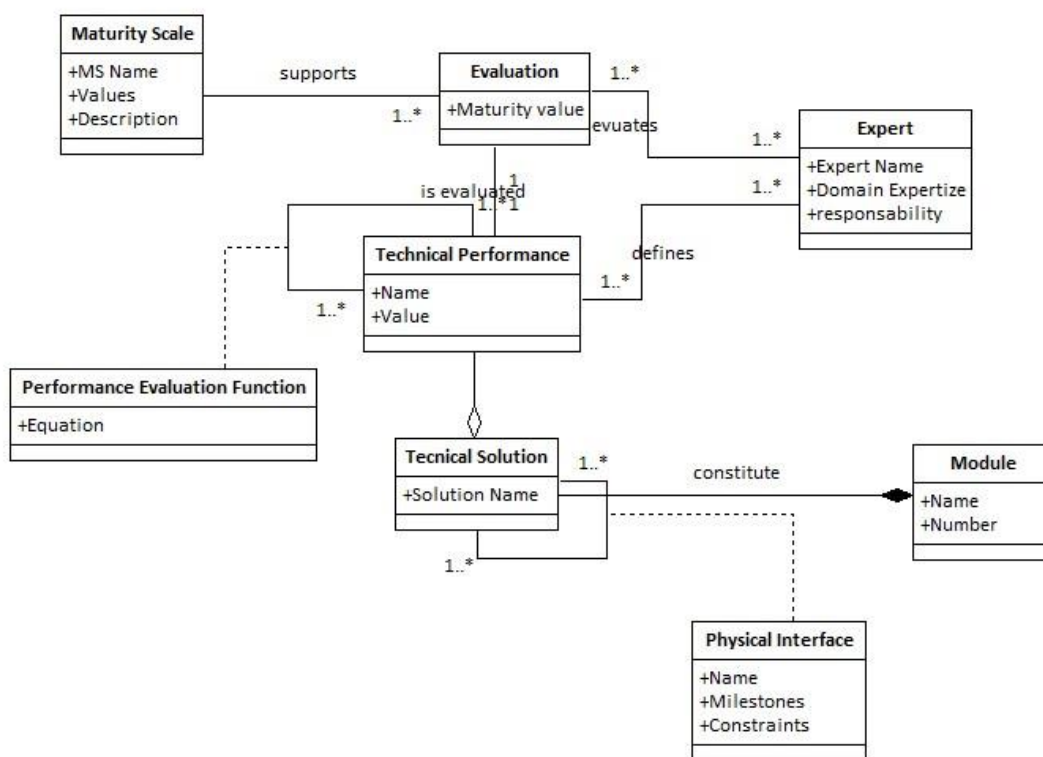


Figure 6: Voice of Design Departments Matrix

The data captured through VoDD is represented by the ontology in Figure 6. Each technical performance is related to different technical solutions. The technical solution class is defined by its name. Different evaluations are based on maturity scales and are represented by the maturity value for each technical performance. Evaluations are expert-based; hence the *expert* class is associated with the evaluation class. The expert class is defined by *expert name*, *domain of expertise* and *responsibility* that an expert has within a company and a project.

4. Collaborative Conflict Detection and Risk Management

The interfaces that are taken into account are physical interfaces representing mostly spatial and material interfaces; and not necessarily energy or information (2). A design interaction DAC, representing a particular connectivity map (23) identifies performance targets for interfaces and identifies conflicts between different design departments that require further expert discussion. Conflicts that are often found in the design are for example the height of condenser that will impact the number of electronic card and therefore the manufacturing costs or the interface

between chassis and box that cannot dissipate Jules per hour if a particular a mechanical interfaces is chosen mostly for its robustness against shocks capacities. PC-DSM data related to the interface design space are then used to frame discussions about potential risks and mitigation plans. All information concerning collaboration conflicts, associated risks and mitigation plans are captured in a Collaborative-FMECA (C-FMECA).

A design interaction objectives DAC is used to identify physical interfaces that might cause problems (Figure 7). A design interaction objectives DAC relates experts' evaluations with regard to required performance criteria for a single concept. In the first step, physical interfaces are extracted from the PC-DSM matrix (Figure 7) in the form of a list of possible interfaces. This list of interfaces is compared with the FF-DMM matrix in order to identify interfaces that contribute to the overall system function. This is done in order to distinguish interfaces that impact high-level functioning of the system. This comparison yields a list of physical interfaces associated with different technical solutions and design departments.

For each physical interface, a VoDD matrix is used to extract data on related performance criteria and expert evaluations for each constituting technical solution. The target of this matrix is set using a performance evaluation function in VoDD (Figure 5). Therefore, for each physical interface, we identify related performance criteria, the ability to achieve performance predicted by experts, and targets for a given physical interface with regard to design requirements (see Design Interaction Design Assessment Card, i.e., DAC, in Figure 7).

Figure 7: SADT showing overall conflict identification process

The advantage of this DAC is that it supports the identification of technical maturity conflicts between different design departments. Therefore, for the same interface, the gap between what is achievable with one technical solution in one design department and that of a technical solution proposed by another design department is identified. Moreover, the design interaction DAC identifies the limitations of different design departments. For example, for the same interface, one design department might be less technically mature and considerable effort may be required in order to reach the same performance target that can be achieved more easily by another department. The evaluation of required effort and related risks can induce the redefinition of performance targets.

The conflict list is then discussed among members of the affected design departments. Information about conflicts is complemented with interface typology data that are stored within the PC-DSM matrix and is represented by the physical interface data model (Figure 4). This data model is designed to support the definition process of design parameters across interfaces.

This information related to physical interfaces and related design parameters then helps us to identify design parameters and domain specific knowledge in order to better manage related risks. During collaborative risk meetings, this information is available to the design team. For the company case study, usually it takes half a day to discuss these collaborative conflicts. During such discussions, designers can estimate the degree of difficulty related to one interface, renegotiate performance targets or set mitigation plans in order to ensure that interfaces will not decrease overall system performance.

The information resulting from this discussion is contained within Collaborative Failure Mode, Effects and Criticality Analysis (C-FMECA). FMECA (21) constitutes a method describing possible system failures, their effects, and assessments of their criticality. It is a widely used reliability engineering tool in industry. In most cases, FMECA addresses global system risks and different measures to mitigate those risks. In this work, we propose Collaborative FMECA (C-FMECA), which addresses only collaborative risks based on data related to physical interfaces. The advantage of C-FMECA is that it essentially addresses collaborative risks. However, this would not be possible if adequate tools for identifying conflicts related to physical interfaces

were not used. Moreover, the interface data model, which includes data related to physical interface typology, is indispensable for this approach because it is used as the basis for data gathering. These data embody interface characteristics as well as previous experience related to similar interface typology, hence enhancing the design team's knowledge of the design problem.

5. Experimental Study

The MDDS approach was used and tested in the oil industry. This case study addresses the design of a power electric controller (PEC), a regulator board that is used as a power supply for the motor control and main controller boards. The main function of this subsystem is to generate the +3.3V and +1.9V power supplies. The PEC system is comprised of a substrate, electronic components, a box, a chassis, a collar, substrate, connectors with other systems and related wiring. The internal functional analysis (22) is provided in Figure 8. For confidentiality reasons, the PEC is only partially represented. There were three design departments involved in the design project: mechanics designed the chassis, packaging designed the box, and electronics designed the other modules.

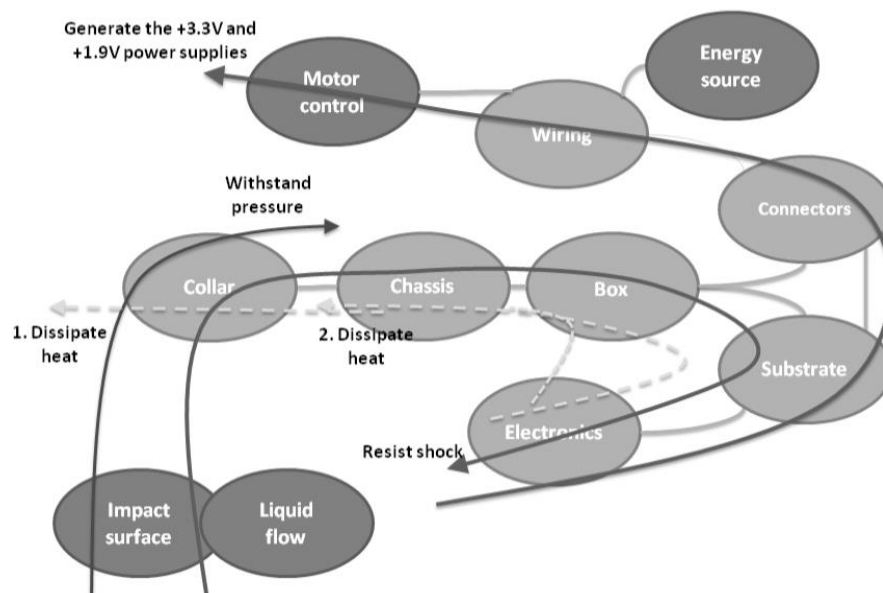


Figure 8: Internal Functional Analysis of PEC

(Light grey lines between components represent physical connections. Functional flows are represented by arrows indicating the direction of the function flow itself. Dashed light grey lines show several possible flows for the heat dissipation function.)

There were four main functions for the PEC: (1) dissipate heat with two possible functional flows; (2) resist shock; (3) withstand pressure; and (4) generate the +3.3V and +1.9V power supplies. The directions of functional flows are represented by arrows. In this case, only the function *dissipate heat*, required to cool electronic components, had two possible functional flows (see dashed lines with arrows in Figure 8); depending on the chosen solution for the chassis, heat could be dissipated through the chassis, or pass through the chassis and be dissipated through the collar. In both heat dissipation functional flows, air flow against the box, chassis or collar could facilitate the process.

MDDS is generally deployed during early conceptual design brainstorming workshops. In this case, the workshop was divided into three days: (1) functional analysis, (2) brainstorming about different modules (sub-systems), possible technical solutions, interfaces, and system performance, and (3) expert performance evaluations for each solution, DAC computation and system selection. In order to illustrate some of the possible concepts yielded from this discussion, the PEC reference "HPHT" (High Pressure High Temperature) concept is shown in

Figure 9. The left side shows an open casing with the substrate and electronic components inside; the right side shows the box, wiring and connectors integrated onto the chassis (shown in yellow). Concept 7 was chosen to illustrate interface detection and collaborative risk management.

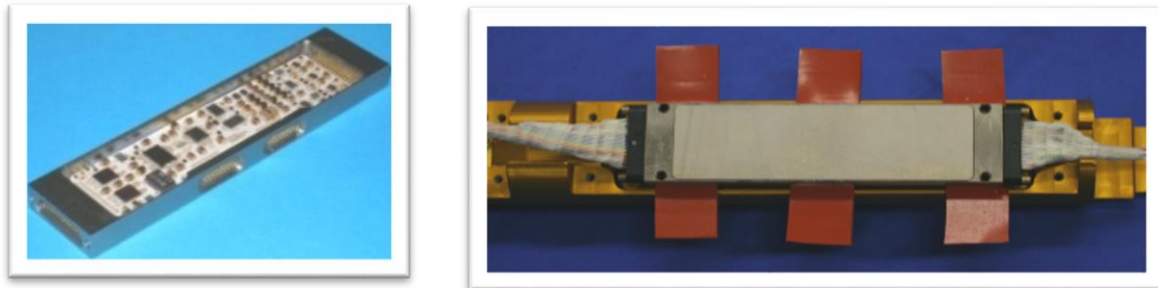


Figure 9: The PEC "HPHT" Concept

After data collection in FF-DMM, PC-DSM and VoDD matrices, and Design Interaction Objectives DAC is calculated (see figure 10). This extract shows different system performances linked to one interface (matrix rows). Columns show the targets defined for each department (i.e., shared modules and performance targets). Here we show only one interface, but others were generated for each cross-domain interface. For example, *robustness against shocks (X)* was set as a minimum (Min function in VoDD, see Figure 5). This means that the global robustness values for each concept were defined by the minimum values of their technical performance evaluations. Here, the design of the interface *chassis-box* could not be more than 2, the value for *robustness against shocks (X)*. If the robustness of the interface had been 3, it would have been considered overkill for achieving the final robustness goal for the concept. The Design Interaction Objectives DAC and physical interface data (Figure 4) are used to frame discussions on concept feasibility and related risk management during an expert review meeting. These data were then captured and shared in a form of a Collaborative-FMECA (Figure 11).

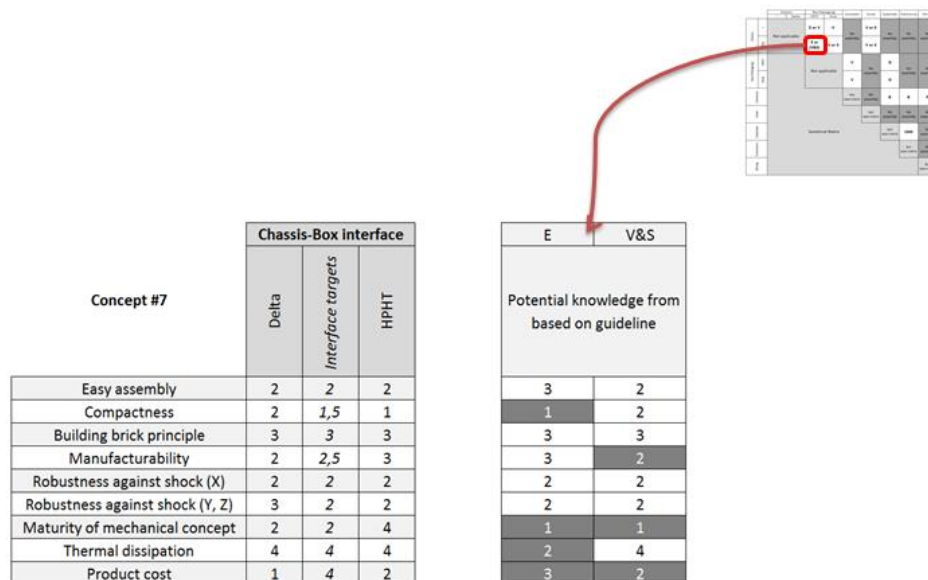


Figure 10: Reverse Delta-HPHT Interface for Concept 7

(Left side: An excerpt from the VoDD matrix. Right side: Design Interaction Objectives DAC)

Figure 11 shows an excerpt from the C-FMECA resulting from the MDSS approach. The C-FMECA is not designed as a replacement but rather as a complimentary approach. This data is integrated in the overall risk management process and FMECA's related to different phases and

departments in the design. Data related to the C-FMECA come from physical interface data model and data that are considered to be standard in FMECA. Identified interfaces are then correlated to possible solutions. It shows an example of the *glue* physical interface between the *box* (packaging) and the *substrate*. In this example, one can see that shear and pull-off strength were critical in the design of a *glue* physical interface. During interface dimensioning, designers need to consider the *surface area* of contact between technical solutions, the *viscosity coefficient*, and the *thickness* of the glue between contacts, as defined in physical interface data model. The estimated probability is 5 because in previous experiences (guideline number 100236239) described in the physical interface model, every time a design team dimensioned a glue interface, it failed during initial testing. The impact is also considerable because the interface broke without prior reduced functionality (i.e., no forewarning). Both Likelihood and Severity are defined on a scale from 1 to 5 as it was already the case in the company. Design recommendations are afterwards summarized, and one week is deemed necessary to optimize design parameters with the support of a specialist. The designer in charge of dimensioning this interface had to take consider that glue would have to resist shocks up to level 2 (i.e., the performance target already attained in previous designs).

Unique Risk ID	Concept Number	Interface	Discipline	Function	Technical Performance	Target (Extracted from DAC)	Physical Connection	Key Design Parameters	Description of Risk	Likelihood	Severity	Original Exposure	Mitigation Plan	Ressource	Duration
56	7	Chassis/Box	Packaging & Mechanics	Heat dissipation	Easy assembly	2	Glue	Surface area, Viscosity and Thickness	In a glue physical connection, shearing strenght and pull-off strength vary a lot following the management of these design parameters: Surface, viscosity coefficient and thickness of glue. See design guideline number: 1000236239 for further data.	5	5	25	Look equations and target values extracted from empirical approach on the design guideline number 100236239.	Contact Houston and the glue specialist.	About 1week
					Compactness	2									
					Building brick principle	3									
					Manufacturability	2									
					Robustness against shocks (X axis)	2									
					Robustness against shocks (Y, Z axis)	3									
					Maturity of mechanical concept	2									
					Thermal dissipation	4									
Product Cost	1														

Figure 11: Excerpt from C-FMECA

In the next step, design convergence was achieved by integrating the data in the Collaborative-FMECA and analyzing if all technical performance targets could be achieved through physical interfaces. In the *glue* physical interface example (see 1), the design team had to decide whether glue could achieve the expected technical performance for the *chassis-box* interface design. In this case, the *thermal dissipation* performance target was set to 4. Using the previously mentioned guidelines, this target was not feasible based on the results of previous testing. This level of thermal dissipation could not be achieved with a glue physical interface. Therefore, the data implied that a glue interface, and thus the analyzed concept, was unfeasible.

This method was applied on two examples in industry. It has been applied in early stages but following one project using this approach thought the entire system development has yet to be done. However, we have organized at the end of the workshop a group feedback on utilization and relevance of the method. As conflict identification is only a part of the methods this is not the only point discussed during the feedback. The major advantage that has been pointed out is the possibility to capture design knowledge, both risk and interface related, using this approach and to identify collaborative conflicts early in the design. Designer have also expressed that they thought that managing collaborative conflicts and shared parameters can possible accelerate the “convergence” in the design process. For example, in the C-FMECA condenser height has been identified as a problem. An engineer has pointed out that this problem has been seen previously

on one project and it was considered to have cost almost 1,5 year of work as it was needed to be redesigned. However, it has to be pointed out that methods for collaborative conflict management were nonexistent in the company, and that a detailed field research has showed these conflicts induced up 1/3 of the overall project time overdue. During the workshop it has also been observed that designers were obliged to discuss the values or interval ranges in the interface definitions. They have pointed out that this might be an interesting way to support design negotiation further in the design process. However, the degree of innovation within a system can influence the difficulty in deploying this method. This approach is based upon previous experience and therefore, for radically new projects, where both the system architecture and the system itself are radically new, it would be hard to support it with previous knowledge. Further evaluations of the methods are needed. It is necessary to follow one project from the beginning to the end. It would be necessary to have a detailed investigation of the efficiency of the method and in depth comparison. For example, identify how many risks have been identified early in the design process? How many risks have been known previously or are new? It would be also interesting to investigate saved design time due to this risk management. However, this is only possible if the method is deployed on several industrial cases and onto the overall project.

6. Conclusion and Perspectives

Interfaces define and intersect functional flows, and therefore define overall system behavior. Only recently have studies integrated interface design into design methods in order to propagate and predict their impact. In this study, we proposed to integrate physical interface ontology into the Design Structure Matrix, thereby enabling experts to harness their experience related to design parameters, previous projects, constraints, and principal solutions. The proposed approach, Multiple-Domain Design Scorecards (MDDS), enables the identification of collaborative risks among different design departments.

The proposed approach was illustrated through a real case study in the oil industry. The approach enabled the identification not only of conflicting interfaces, but also of related risks and mitigation plans. These data were stored within a Collaborative-FMECA that addressed only collaborative conflicts. It is reasonable to expect that all companies use FMECA as an essential approach to managing risk, but it is necessary to investigate if the data considered in physical interface data models can be used for other types of design projects.

Additional difficulties stem from a need to integrate software tools with this approach. For the time being, these data represent a mock-up. If there is a need for larger capitalization of this data, several issues must be investigated. One of the related issues is actually defining the related database, its architecture and potential usability within the design process. It is also obvious that the existence of this database influences the deployment method. We have already seen that in order for a method to be a success, an expert in the method must participate. In this case, the question is whether additional experts are needed in the process of collaborative risk database exploitation.

Acknowledgments

This research work was done within the CIFRE convention and in collaboration with Schlumberger. Hereby, we would like to thank them for their support and very constructive insights during the development and testing.

7. References

1. Wyatt D, Wynn D, Clarkson J, editors. Synthesis of Product Architecture using a DSM/DMM Based Approach. 10th International Design Structure Matrix Conference; 2008 November 11-12; Stockholm, Sweden.
2. Pimpler TU, Eppinger SD, editors. Integration Analysis of Product Decompositions. ASME Design Theory and Methodology Conference.; 1994; Minneapolis, USA:.

3. Wyatt D, Wynn D, Jarrett J, Clarkson P. Supporting product architecture design using computational design synthesis with network structure constraints. *Research in Engineering Design*. 2012;23(1):17-52.
4. Jankovic M, Holley V, Yannou B. Multiple-domain design scorecards: a method for architecture generation and evaluation through interface characterisation. *Journal of Engineering Design*. 2012;23(10-11):743-63.
5. Eppinger SD. Model-based Approaches to Managing Concurrent Engineering. *Journal of Engineering Design*. 1991 1991/01/01;2(4):283-90.
6. Yassine AA, Braha D. Complex Concurrent Engineering and the Design Structure Matrix Method. *Concurrent Engineering: Research and Applications*. 2003;11(3):165-76.
7. Eppinger SD, Browning TR. *Design Structure Matrix Methods and Applications*. Cambridge: MIT Press; 2012.
8. Ulrich KT, Eppinger SD. *Product Design and Development*. New York: Irwin McGraw-Hill; 1995.
9. Sharman DM, Yassine AA. Characterizing complex product architectures. *Systems Engineering*. 2004;7(1):35-60.
10. Steward D. On an Approach to the Analysis of the Structure of Large Systems of Equations. *SIAM Review*. 1962;4(4):321-42.
11. Steward D. The Design Structure System: A Method for Managing the Design of Complex Systems. *IEEE Transaction on Engineering Management*. 1981;28(3):79-83.
12. Maurer MS. *Structural Awareness in Complex Product Design*. Munich: Technical University of Munich; 2007.
13. Lindemann U, Maurer MS, Braun T. *Structural Complexity Management*. Berlin Heidelberg: Springer Verlag; 2009.
14. Jarrat T. *A Model-Based Approach to Support the Management of Engineering Change*. Cambridge: Cambridge university; 2004.
15. Danilovic M, Börjesson H, editors. *Managing the Multiproject Environment*. 3rd Dependence Structure Matrix (DSM) International Workshop; 2001 October 29-30; Cambridge, USA: Massachusetts Institute of Technology.
16. Browning TR. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. *Engineering Management, IEEE Transactions on*. 2009;48(3):292-306.
17. Hellenbrand D, Lindemann U, editors. *Using the DSM to Support the Selection of Product Concepts*. 10th International Design Structure Matrix Conference; 2008 November 11-12; Stockholm, Sweden.
18. Albers A, Braun A, Sadowski E, Wynn DC, Wyatt DF, Clarkson PJ. System Architecture Modeling in a Software Tool Based on the Contact and Channel Approach (C&C-A). *Journal of Mechanical Design*. 2011;133(10):101006-8.
19. Moullec M-L, Bouissou M, Jankovic M, Bocquet J-C, Réquillard F, Maas O, et al. Toward System Architecture Generation and Performances Assessment Under Uncertainty Using Bayesian Networks. *Journal of Mechanical Design*. 2013;135(4):041002-1.
20. Rahmani K, Thomson V. Managing subsystem interfaces of complex products. *International Journal of Product Lifecycle Management (IJPLM)*. 2011;5(1):73-83.
21. IEC IEC. *Analysis techniques for system reliability - procedure for failure mode and effects analysis (FMEA)*. Geneva, Switzerland 2006.
22. Aoussat A, Christofol H, Le Coq M. The new product design - a transverse approach. *Journal of Engineering Design*. 2000 2000/12/01;11(4):399-417.
23. Yassine A, Whitney D, Daleiden S, Lavine J. Connectivity maps: Modeling and analysing relationships in product development processes. *Journal of Engineering Design*. 2003 2011/11/14;14(3):377-94.

Annexes

List of acronyms

MDDS	Multiple-Domain Design Scorecards
FF-DMM	functional flow – design mapping matrix
PC-DSM	Product Component-Design Structure Matrix
VoDD	Voice of Design Department
C-FMECA	Collaborative Failure Mode, Effects and Criticality Analysis
DSM	Design Structure Matric
DMM	Design Mapping Matrix
MDM	Multiple-Domain Matrices
DAC	Design Assessment Card