Transdisciplinary Engineering: Crossing Boundaries
M. Borsato et al. (Eds.)

doi:10.3233/978-1-61499-703-0-117

Modularization in Concept Development Using Functional Modeling

Christoffer LEVANDOWSKI[1], Jakob R. MÜLLER and Ola ISAKSSON
Department of Product and Production Development
Chalmers University of Technology

Abstract. This paper presents a set-based approach to use functional models for platform concept development to identify feasible modules at early design stages. The concepts are defined using functional requirements, design solutions and their interconnections. These models are then encapsulated into functional modules through clustering of Design Structure Matrixes (DSM). A metric is introduced to quantify the ability to modularize a certain concept, which may be used to assess and eliminate inferior concepts. The approach is illustrated using a case study from the aerospace industry. The result shows that modules can be identified by clustering of the functional structure. This has an integral effect on early division of work, possibility to design reuse, etc. The ability to modularize a specific concept The case study also shows that, despite of the traditionally integrated character of the product studied, it is possible to identify functional modules for reuse in a platform.

Keywords. Product platforms, functional modeling, modularization, design structure matrix, set-based concurrent engineering

Introduction

Modern products are significantly more complex than their predecessors. As this complexity is introduced, the developing companies are challenged to manage it in their design processes and organization. This is often solved by dividing the product architecture into manageable chunks, or modules, and dedicate specialized competence to each module. These modules can be developed in concurrency, as long as the interfaces between them are maintained. In product platforms, modules are used and reused as exchangeable design blocks to satisfy a range of customer needs, while simultaneously reaping the benefits of scale in production [1] [2].

However, current practice in modular platforms is based on the physical architecture of the product. This means that modularization as a means of supporting development is infeasible for companies with physically highly integrated products. Also, dividing a product into modules based on physical interfaces fails to provide support for early phases of concept development when the physical form is unknown, and where several conceptual solutions for the platform are developed, analyzed and evaluated.

Research has proven Functional Modeling a feasible method for concept development as a way to model conceptual solutions without requiring physical models

[1] Corresponding Author, E-Mail: levandow@chalmers.se

[3]. The functional models enable integration of technologies from multiple domains, as parallel or complementary solutions. Though there is theory describing the need for, and the concept of, early modularization, no stringent approach is available [4].

This paper presents an approach to use functional models for platform concept development to identify feasible modules at early design stages. It applies an approach with several alternative design concepts, which are assessed in their ability to be modularized. The platform concepts are defined using functional requirements, design solutions and their interconnections. These models are then encapsulated into functional modules to reap some of the benefits from modularization, already at early concept phases.

1. Background and Scope

1.1. Modularization and Design modeling

The encapsulation into modules allows engineers to study the systems and their interfaces on a high level without considering every detail [5]. There are accounts of several different facets of modularization. For example, Mattson and Magleby [7] divide modularity into design, manufacturing and customer modularity. The definition of design in this case aims towards detailed design, rather than concept development.

When applying modularization in the design phase, it allows for concurrent engineering by dividing the effort and knowledge need into manageable tasks. In conjunction, it provides a structure to coordinate tasks and decisions [9]. According to Simpson [10], modular platforms enable horizontal leveraging, i.e. to serve a range of customer segments by providing different functionality. Knowing early what capability and and bandwidth a platform concept has may be useful in selection of concept.

The objectives for modularization determine what constitutes a good module [8]. Also, Shoval [11] concludes that "systems may require different modularization architectures in different lifecycle phases". Consequently, if a company aims to support design decisions in early phases, the modular structure should be based on functional models, rather than physical embodiment.

1.2. Set-based concurrent engineering

Sobek, et al. [12] summarize SBCE as engineers and product designers "reasoning, developing and communicating about sets of solutions in parallel and relatively independently." They further define three principles that apply to a set-based design process: *map the design space, integrate by intersection and establish feasibility.* Through these principles, SBCE addresses issues with regular product development by considering a broad range of alternative design solutions that are systematically narrowed down by eliminating undesirable solutions [13]. An integral part in SBCE is to create tangible information on which to base design decisions [14].

As modularization is an integral part in facilitating concurrency in design, it is also an enabler for set-based concurrent engineering. In addition, the ability to modularize a concept may very well be reason for keeping in, or eliminating it from, the set.

Though often associated with detailed design, SBCE is applicable for concept development. Raudberget, et al. [15] illustrate how interconnectivity in a functional model can be used as a means for eliminating inferior design concepts in early design stages.

1.3. Design modeling in concept development

The prevailing paradigm in many engineering companies is a design support structure constituted of tightly connected CAD and CAE systems. While this provides excellent capabilities for analysis and synthesis based on geometric representations of the design, it fails to support phases where ideas and concepts are explored without physical embodiment [17]. In these early phases of ideation and technology consideration, formal support is rare in practice.

Functional models, if used right, support concept development with their inherent capability to express structures without explicit physical attachment. The EF-M model, developed by Schachinger and Johannesson [18], provides a structure of functional requirements (FR) that express system needs and design solutions (DS) for these. There are several relationships between the objects in the EF-M tree. The *interacts_with* (iw) relationship between DS is of great importance since it expresses functional interaction. The iw are asymmetric.

1.4. Design Structure Matrixes in engineering design

A DSM is can be used to represent elements and their connections in a standardized way. In a DSM relations between different elements are represented in a matrix, showing all elements in rows as well as columns. Interactions are marked at in the cells, with the direction row to column.
To cluster a DSM, the elements are rearranged so that the number of connections in between the different clusters is reduced to a minimum. As a direct consequence of this, relations of elements are accumulated inside the modules. As a conclusion of this, an appropriate clustering algorithm optimizes for maximum internal connectivity together with minimal external connectivity.

Hölttä-Otto [4] suggests a clustering algorithm for finding modules in functional and physical structures. The algorithm is based on the Idicula-Gutierrez-Thebeau Algorithm (IGTA) for clustering Component-DSM which assigns elements to new clusters, evaluates the new positioning and chooses the best possible allocation. IGTA is explained in detail in [19].

1.5. Scope and research method

In the phase, the design work is not yet constrained by assumptions of the detailed design or a physical form but allows for free ideation. These ideas are formally modeled as functional architectures. On that note, this paper aims to elaborate on two questions: (1) *How can modularization be supported in the concept development phase?* and (2) *How can* ability to modularize *be used as a criterion to compare different design alternatives?*

The research in this paper was carried out as a part of the VITUM research project. The project aims to develop virtual demonstrators for aircraft engine component design. The partner company is GKN Aerospace Engine Systems Sweden, which designs and manufactures engine parts for the aerospace industry. The project grants the researchers access to real design data, for verification and development of methods. The illustrating use case in Chapter 3 was developed together with GKN Aerospace. The models were developed by the researchers and validated through project meetings and workshops.

2. Modularization in Concept Development Using Functional Modeling

This section accounts for the approach suggested to define and use modules in concept development. The approach specifically addresses development and assessment of trans-disciplinary design decisions. Defining modules in early phases enables modularity for designs that are traditionally not modularized (i.e. physically integrated designs).

A key objective is to identify what knowledge is needed to realize different parts of the design. To facilitate concurrency, a second objective is to minimize the number of interfaces between design teams that need to be maintained, and clarify the character of those interfaces. How well an Architectural Option (AO) performs in terms of modularity is used as a criterion for pursuing or eliminating that design.

The approach builds on SBCE as a framework for expanding the design space, assessing it, and narrowing it down using modularity as an assessment criterion. The AO are modelled as EF-M. Figure 1 illustrates the suggested process for *modularization in concept development using functional modeling*.

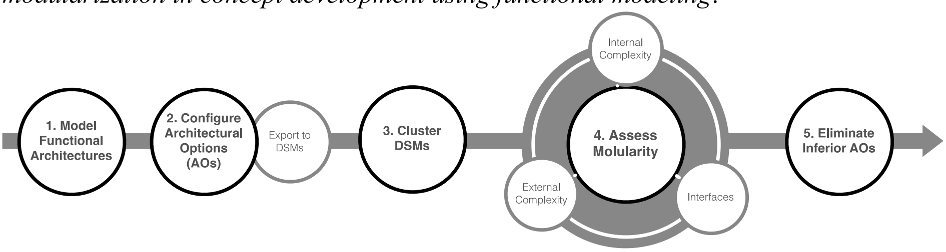

Figure 1. The suggested work process for modularization in concept development using functional modeling.

2.1. Modelling Architectural Options

As a first step, the product is modelled using EF-M trees. For novel designs, the EF-M is created from scratch, or in case of incremental design it is expanded. To *expand the design space*, several alternative DSs may be considered for each FR. The variety may be introduced on any level in the EF-M tree, which when done will have several alternative branches and leafs. The *interacts_with* relationships are modelled between DSs and express the functional connections across branches in the tree. The *iw*s are modelled on a conceptual level, as the information about the designs in early phases is limited. By combinatory operations, the DSs are compiled into a discrete set of architectural options, including the iw-relationships. These are transformed into a DSM with DSs as rows and columns, and iws marked in the cells of the matrix. Each AO is represented by one DSM.

2.2. Modularizing functional models

The DSM created from the functional structure is clustered using an IGTA implementation in CAM. Modules that share very complex interfaces can be accumulated into super-modules. The clusters can be collapsed to show only the assembly structure, as shown in Figure 3. From the initial clustering, shown in blue boxes, a new cluster, highlighted in green, is formed due to the high connectivity between the two modules.

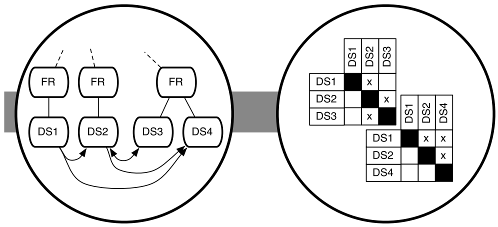

Figure 2. Step 1 and Step 2 in the process. The EF-M model represents two possible architectural options which are converted to two separate DSMs.

Shown with orange boxes are the interfaces between the different modules. In the collapsed state, the interfaces are kept and reduced to a single interaction between the submodule and the element, as seen with the connections DS6-DS1 and DS6-DS2 which are then collapsed to DS6-Cluster 1A.

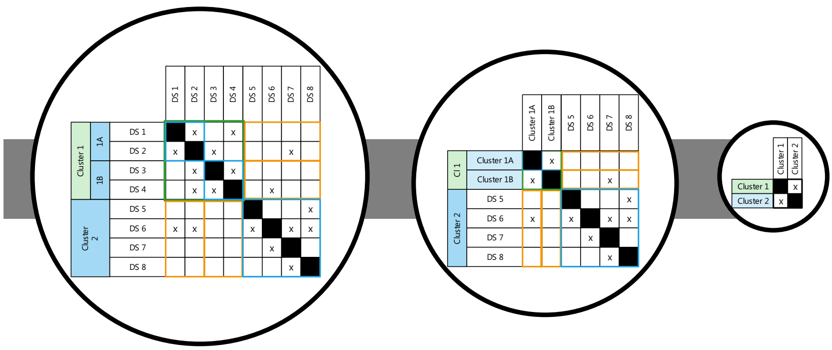

Figure 3. Example of multi-level clustering and collapsing of a clustered DSM.

Once the DSM is clustered into modules, the functional structure can be evaluated for its quality in terms of modularity. This assumes that the clustering algorithm applied has found the best way to cluster each functional structure. The metrics are internal connectivity, external connectivity and interface complexity.

Internal connectivity is measured through how many of the possible connections in a cluster are active, which is calculated by $c_{int} = \frac{x_{iw}}{n^2-n}$, where x_{iw} is the total number of internal connections and n the number of elements in the cluster. The value for c_{int} reaches from 0 to 1, with 1 as the most desirable value representing the highest possible internal connectivity.

External connectivity is measured by two factors, the average number of interfaces per module and the average number of connections per module. For each module only the interfaces to highest level modules were counted, and for high-level interfaces the connections of the internal modules were counted in collapsed state.

Interface complexity is evaluated by the average number of connections per interface, however detailed analysis of the distribution among the interfaces has to be observed. For modules sharing complex interfaces a regrouping into a super module might be beneficial.

2.3. Eliminating inferior design alternatives

Since external connectivity is expressed by two factors, optimizing for a modular design can follow two strategies. Either to aim for a minimal complexity of the interfaces (low number of iw between two modules) or for reducing the number of interfaces per module.

Architectural options (AO) with few and simple interfaces are preferable from a modularity point of view. AOs with complex and many interfaces on the other hand are considered less feasible, and may be reason to dismiss the AO.

An AO with few but complex interfaces between modules may be candidate for merging clusters into multi-level modules. If all complex interfaces can be integrated into clusters, the AO as a whole can improve its modularity, and be moved from the top left to one of the lower squares in the 2x2 matrix shown in Figure 4. The many-simple case indicates several interfaces that in themselves are not very complex. It may not be reason for eliminating the AO, but is should be considered less fit for realizing the benefits of modularization in early stages.

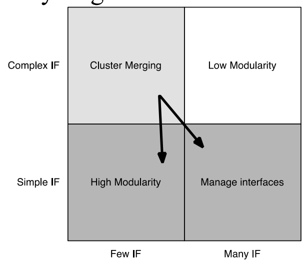

Figure 4. 2x2 matrix to characterize interfaces between clusters.

3. Functional modularization of a turbine rear structure (TRS)

To illustrate the above mentioned approach to clustering, a functional model of a TRS was created. The model is illustrated as an E-FM tree, shown in Figure 5. As a foundation for the clustering, the different leaf-DS are connected with iw.

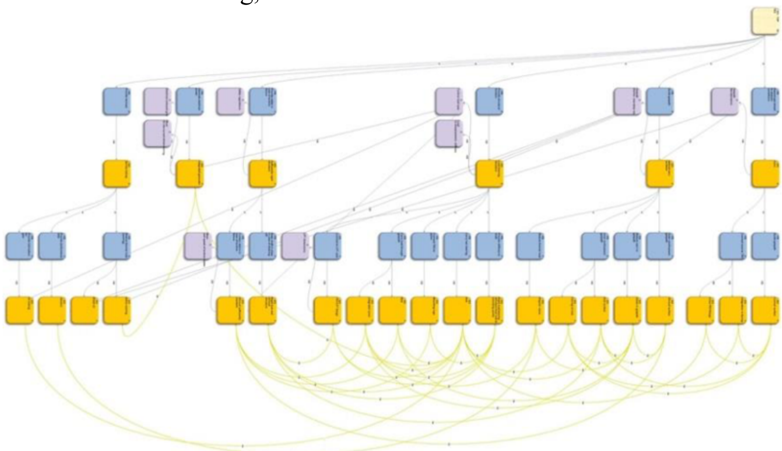

Figure 5. The EF-M tree of the Turbine Rear Structure, including several alternative DSs (yellow) for some FRs (blue).

From the functional model, which included several alternative design solutions, two instantiations were analysed. They differ mainly in the way the vane/strut structure is build up, with design A having the load bearing structure integrated in the vane shape, whereas design B seperates the aero- and mechanical functionality with a rod for structural loads and a faring for aerodynamic performance.

The iw from the EF-M tree were converted into a DSM, listing all DS and showing their respective relations.

Each module, which is seen as a greyed out square in the DSM, is constructed from DS elements and sub-modules. Their respective connections are managed inside the module.

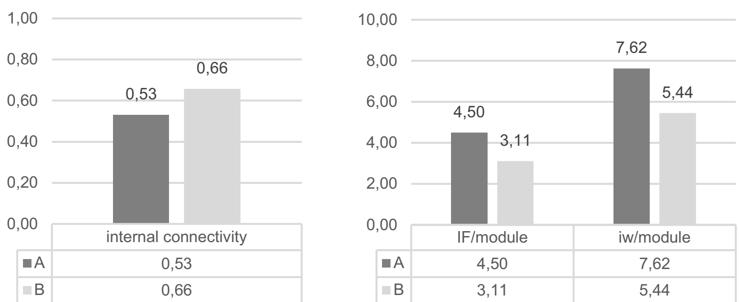

Figure 6. Internal (left) and external (right) connectivity metrics of the two example architectural options.

	Conical surface	Cone TRS flange	Oil tubing	Integrated generator	Active cooling	Bleed air	TRS LPT flange	Rigid hub and shroud structure	Hub	Hub surface	Sensors	Angular vanes	Vanes in gaspath	Integrated vanes	Shroud sufficiently resistant	Shroud surface	Mounting lugs	to mounting points
Conical surface	x									1								
Cone TRS flange	1	x							1									
Oil tubing			x															
Integrated generator				x	1													
Active cooling				1	x													
Bleed air						x												
TRS LPT flange							x	1	1									1
Rigid hub and shroud structure							1	x	1	1								1
Hub		1	1	1			1		x	1				1				1
Hub surface	1							1	1	x			1			1		
Sensors											x	1						
Angular vanes											1	x		1				
Vanes in gaspath												1	x	1		1		
Integrated vanes									1			1	1	x	1			1
Shroud sufficiently resistant														1	x	1	1	
Shroud surface										1			1		1	x		
Mounting lugs																	x	1
Rigid load bearing structure from hub to mounting points							1	1						1			1	x

Figure 7. Clustered DSM of architectural option A.

External relations are collected in interfaces. Inside the interface object all connections and the information concerning them to one module are collected. This approach to structuring should allow the modelling of the modular structure in an object oriented approach.

As shown in Figure 6, the clustering of design B is more consistent, in the way that it has a higher average internal connectivity of all modules, and also a lower external connectivity shown in a lower average number of iw per DS, and also lower average number of interfaces per module.

	Integration in fairing	Fairing on hub	Conical surface	Sensors	Angular vanes	Oil tubing	Integrated generator	Active cooling	Bleed air	TRS LPT flange	Rigid hub and shroud structure	Hub	Hub surface	Mounting lugs	from hub to mounting points	Vanes in gaspath	Rod	Shroud sufficiently resistant	Shroud surface
Integration in fairing	x	1	1																
Fairing on hub	1	x	1													1			
Conical surface	1	1	x										1						
Sensors				x	1														
Angular vanes				1	x														
Oil tubing						x													
Integrated generator							x	1											
Active cooling							1	x											
Bleed air									x										
TRS LPT flange										x	1	1			1				
Rigid hub and shroud structure										1	x	1	1		1				
Hub		1				1	1			1		x	1		1		1		
Hub surface			1								1	1	x			1			1
Mounting lugs														x	1				
Rigid load bearing structure from hub to mounting points										1	1			1	x				
Vanes in gaspath		1			1											x	1		1
Rod												1				1	x	1	
Shroud sufficiently resistant														1			1	x	1
Shroud surface													1			1		1	x

Figure 8. Clustered DSM of architectural option B.

4. Discussion

The model used in the approach aims to capture architectural options in early design phase in an effort to explore a large number of designs. Although based in immature knowledge about the design, it adds the benefits of modularization in an early phase of development.

The approach builds on existing theory and established design tools, such as EF-M modeling and DSM clustering. The suggested tools provide steps towards design automation of specific tasks in the concept development phase.

The example compares two alternative solutions against each other, and therefore only requires relative comparison of metrics. To automatically eliminate solutions based on their possibility to modularize would require threshold values for all metrics. However, the automation of such decisions is not the end goal. It is rather to automatically generate enough data so that the architect can make informed design decisions.

Using EF-M modeling to express concepts is quick and is easy. However, the knowledge required to accurately draw the iw-relationships should not be underestimated. If done accurately, the model supports modularization of the concepts, but the outcome of the analysis is sensitive to the modelling quality. All information has to come from the engineers' experience and ability to estimate system behaviour. This is especially the case for the development of new products, and less so for incremental development.

This paper does not suggest a new way of clustering, it rather makes use of the DSM clustering as a standardized format of assessing elements and interfaces. Yet, the quality of the clustering depends on the quality and purpose of the clustering algorithm. Thus a pivotal part is to identify a clustering algorithm that matches the purpose of the approach.

Though modularization theory presents a range of benefits in development, only some of those are targeted in this paper. There are possibly more benefits than the ones addressed here, and there are most likely benefits that cannot be achieved until the physical embodiment exists.

The example in this paper also neglects to take account of extreme results of clustering, such as on single module as output or multiple elements that could not be assigned to any module and are "free". This might be either managed by adapting the clustering algorithm, or by respective guidelines in the further design work.

It is also not yet considered how the next step in the design process is executed, and how the transition from functional to physical clustering is handled. This may be the topic of a future paper.

5. Conclusions

Modularization in concept phases as presented in this paper provides benefits to companies that may otherwise not be able to use modules in their development. It does so by applying functional modeling to model designs in early stages, and act as basis for clustering into functional modules, rather than physical modules.
The composition of the modules determines the knowledge need, task decomposition and design team composition for each module, whereas the interfaces between the modules identify communication needs. The clustering focuses on minimizing interfaces between modules, to minimize the need for cross-modular decisions.

The illustrating case shows how the approach is applied to a turbine part design process. The resulting clustering generates a number of modules within the product that can be used to introduce concurrency in the design process.

To assess the ability to modularize different concepts, a connectivity metric is proposed. It relates to the complexity of those interfaces that cannot be allocated to a specific cluster. These can be few or many, and simple or complex. The preferred architectural options should have few and simple interfaces, whereas simple-many and few-complex can be managed. The architectural options with many complex inter-modular interfaces may be discarded in favor of other architectural options.

The main benefit of the approach is that it provides structured assessment and exploration of different design concepts in phases where the design freedom is high, but the knowledge about the design is uncertain. It enables informed design decisions about functional modularity. This design phase is however confined by the low maturity of the models, which is why careful consideration needs to be made of how the models can be used, without introducing uncertainty that undermines any decisions built on the models. To create structures for managing knowledge in these phases, functional modularization may provide the necessary means to dividing the design work, and devising a feasible system breakdown.

Acknowledgements

This research was funded by the Swedish innovation agency VINNOVA. The support is greatly appreciated.

References

[1] B.J. Pine, Mass customization: the new frontier in business competition, *Harvard Business Press*, 1999.
[2] M.H. Meyer and A.P. Lehnerd, *The Power of Product Platforms: Building Value and Cost Leadership*, Free Press, New York, 1997.
[3] C. Levandowski, D. Raudberget and H. Johannesson, Set-Based Concurrent Engineering for Early Phases in Platform Development. In: J. Cha et al. (eds.) *The 21st ISPE International Conference on Concurrent Engineering - CE2014*, Beijing, IOS PRess, Amsterdam, 2014, pp. 564-576.
[4] K. Hölttä-Otto, *Modular product platform design*, PhD Thesis, Helsinki University of Technology, Espoo, 2005.
[5] D.K. Hitchins, *Advanced systems thinking, engineering, and management*, Artech House, Boston, 2003.
[6] J.K. Gershenson, G.J. Prasad and Y. Zhang, Product Modularity: Definitions and Benefits, *Journal of Engineering Design*, Vol. 14, 2003, No. 3, pp. 295-313.
[7] C.A. Mattson, and S.P. Magleby, The influence of product modularity during concept selection of consumer products, *Proceedings of the ASME DETC Design Theory and Methodology Conference*, Pittsburgh, PA, Sept, 2001, pp. 9-12
[8] P. Gu and S. Sosale, Product modularization for life cycle engineering, *Robotics and Computer-Integrated Manufacturing*, Vol. 15, 1999, No. 5, pp. 387-401.
[9] C.Y. Baldwin and K.B. Clark, *Design rules: The power of modularity*, MIT press, Cambridge, 2000.
[10] T.W. Simpson, Product platform design and customization: Status and promise, *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, Vol. 18, 2004, No. 1, pp. 3-20.
[11] S. Shoval, Dynamic Modularization throughout System Lifecycle Using Multilayer Design Structure Matrices, *Procedia CIRP*, Vol. 40, 2016, pp. 85-90.
[12] D. K. Sobek, A. C. Ward and J. K. Liker, Toyota's Principles of Set-Based Concurrent Engineering: *Sloan management review*, Vol. 40, 1999, pp. 67-84.
[13] R.J. Malak Jr, J.M. Aughenbaugh and C.J. Paredis, Multi-attribute utility analysis in set-based conceptual design, *Computer-Aided Design*, 2009, 41(3), 214-227.
[14] L. Holmdahl, *Lean Product Development på Svenska*, Lars Holmdahl, Gothenburg, 2010.
[15] D. Raudberget, C. Levandowski, O. Isaksson, T. Kipouros, H. Johannesson, and J. Clarkson, Modelling and Assessing Platform Architectures in Pre-Embodiment Phases through Set-Based Evaluation and Change Propagation, *Journal of Aerospace Operations*, Vol. 3 (2015), pp. 203–221.
[16] A.H.B. Duffy and M.M. Andreasen, Enhancing the Evolution of Design Science, *International Conference on Engineering Design - ICED '95*, Prague, Czech Republic, 1995.
[17] S. Gedell, *Efficient Means for Platform-Based Development - Emphasizing Integrated, Information-Rich System Models. Department of Product and Production Development*, Vol. PhD Thesis. Chalmers University of Technology, Gothenburg, 2011.
[18] P. Schachinger and H. Johannesson, Computer Modelling of Design Specifications, *Journal of Engineering Design*, Vol. 11, 2000, No. 4, pp. 333-353.
[19] R.E. Thebeau, *Knowledge management of system interfaces and interactions from product development processes*, MSc Thesis, Massachusetts Institute of Technology, Cambridge, 2001.