9TH INTERNATIONAL DESIGN STRUCTURE MATRIX CONFERENCE, DSM'07
16 – 18 OCTOBER 2007, MUNICH, GERMANY

SOCIAL NETWORK TECHNIQUES APPLIED TO DESIGN STRUCTURE MATRIX ANALYSIS. THE CASE OF A NEW ENGINE DEVELOPMENT AT FERRARI SPA

Michele Liberati[1], Federico Munari[1], Paolo Racchetti[2], Tazio Splendiani[2]
[1]Department of Management, University of Bologna
[2]Ferrari Spa

Keywords: DSM, Social Network Analysis, Product Architecture, Organizational Design.

1 INTRODUCTION

Product architecture design encompasses the definition of product functional elements, the allocation of these elements to physical components and the definition of the interfaces among components [1]. Ulrich and Eppinger's view on product architecture is based on the scheme by which the product's subsystems and components interact [2]. These issues are particularly relevant for complex products. In fact, included under this label are all the products "made up of a large number of parts that interact in a non-simple way, in which the whole is more than the sum of the parts" [3]. Understanding functioning and performances of products like automobiles and aircraft, therefore, means understanding the interaction between chunks at different levels: the interaction between the product's main subsystems, the interaction between the components within and between these subsystems, etc. Moreover, the design and development of these products require and involve a large number of technicians and engineers, with strongly interrelated tasks. The appropriate definition and coordination of the organizational units are therefore as critical as the product functions and architecture definition, in these contexts. Sosa et al. [4] argued that aligning product architecture and organizational design is a necessary condition for successful product development and highlighted "the importance of identifying design interfaces during the project planning stage so that corresponding design team are managed efficiently during project execution" (p. 240).

Starting from these considerations and based on an extensive review of the literature on this topic, this work aims at developing a structured approach to link architectural analysis and development team organization for complex products. The intended contribution of this study is to formalize a methodology for the analysis and alignment of product and organizational architectures, by relying on the approaches already developed by the literature, introducing new concepts and new instruments, based on Social Network Analysis (SNA), and finally validating them through an empirical application. This empirical test was carried out for a new engine development project at one of the leading European carmaker: Ferrari Spa. This Italian sports car manufacturer is one of the most known brands in the world. From the year of foundation, 1929, the firm has always aimed at the research of excellence for its product, which reached and maintains through the invaluable experience on the highest-level racing competitions.

This work is a part of a 9-month collaboration project between the Department of Management of the University of Bologna and the Technical Division of Ferrari. Firstly, in conducting the research, we reviewed the instruments and techniques developed by the literature, studying the potentialities of Design Structure Matrix (DSM) [5] and the previous applications for product architecture and organizational analyses (see for example [6]). Thereafter, thanks to some previous experiences on SNA studies [7], we argued the potential of applying instruments and techniques already developed in this different field of research (SNA) in relation to the analysis of DSM. Relying on these instruments, we worked up a structured method for the optimization of the development team organization (i.e. subunits composition, responsibility boundaries and integration arrangements) based on the product architectural analysis. The parallel development and testing approach we adopted and the in-depth collaboration with the Ferrari Technical Division and in particular with the team that worked on the firm's last platform engine, allowed us to formalize several steps that could be followed up in similar development projects. The next two sections briefly describe the method and the results of its application in this project.

2 THE METHOD

The first stage of our methodology aims at defining the appropriate level of analysis, from the product-system hierarchy. This mainly consists of identifying the key subsystems-components included in the product to be studied as the initial step to build the Design Structure Matrix. In this regard, in conducting the field research, we performed a series of initial interviews with the Director of the Engine Development Unit and the Project Manager responsible for the development of the new engine. We first analyzed the bills of materials of the engine in order to identify the main sub-groups at the first level of decomposition. We further decomposed each sub-system into lower-level components, thus identifying 46 components that represent the unit of analysis of the Ferrari case. We decided to stop our study at this level of aggregation, considering the need of identifying different components, performing specific functions and generally designing by a single engineer (or by an external organization) and following the managers' indications.

The second phase consists of the identification of a set of relevant design dependencies among components, for the specific product, following the typology proposed by Sosa et al. [4]. We discussed with the unit and team leaders about each type of dependency proposed by the authors and decided to study three of them, considered absolutely relevant for the project: 1) *Spatial*, indicating the need for components' physical proximity for system assembly and functionality; 2) *Structural*, indicating the existence of a functional requirement for transferring design loads, forces, or vibration energy; 3) *Material*, indicating the need of fluid exchange (air, oil, fuel, water) between components, for system functionality. After that we constructed three different DSM by asking five team members, selected on the basis of their level of experience and knowledge of the product, to map out all the different types of dependencies between each pair of components. We asked to measure the criticality of each dependency using a 11 point scale, from -5 to +5, in order to analyze positive (required) and negative (undesired) dependencies. We then collapsed the data into an overall matrix .

The third and most important phase regards the analysis of the dependencies data collected into the final DSM. In this regard, we extended some concepts developed by social network studies to the product architecture analysis: i.e. the notion of *cohesive subgroups* (group of actors, within a network, who interact with each other to such an extent that could be considered distinctive entities [7]). We concluded that using the dependency data collected, we were able to identify *cohesive subgroups* from the network of the engine. We could define them as groups of components with high levels of internal dependency (within group dependency) and relatively low level of intra-groups dependency (between groups dependency). We performed this analysis using the software UCINET 6.1 and following three stages (for a complete description of each stage see [8]): Firstly, we defined an appropriate number of subgroups using the NETWORK > SUBGROUPS > CLIQUES procedure and analyzing the "clique overlap" patterns; Secondly, we clustered the network according to this number of subgroups, using the procedure TOOLS > CLUSTERING > OPTIMISATION; Thirdly, we developed a centrality scale for the components through the procedure NETWORK > CENTRALITY > EIGENVECTOR.

The fourth and last step of the method concerns the organizational design of the development team: assigning each designer to an organizational subunit, defining the responsibilities boundary of these units (either in term of design responsibility and in term of coordination and control of external organization's development work), identifying the needs of integration roles between the groups and finally establishing a criticality scale for the components development activity. We performed these tasks by studying the results of the previous analyses and defining the organizational design according to them: i.e. grouping designers according to the architectural clusters, assigning priorities to the first components of the centrality scale, etc.

3 RESULTS

From the data collected into the DSM and following the clustering procedure, previously described, we were able to provide the two representations of the engine clustered network. Figure 1 shows one of them. The nodes in the figure represent the engine components and the arrows represent the existence of a positive dependency between them. We evidenced four different *cohesive subgroups*, with four different colours. The node's sizes are defined according to the degree of centrality of each component in the system architecture.

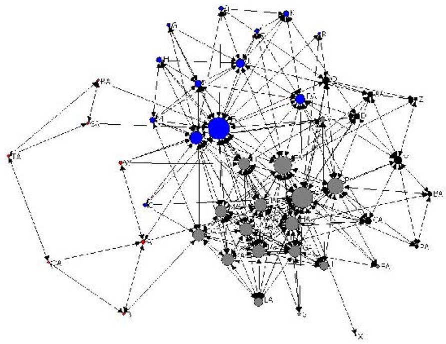


Figure 1. Network representation of the components' dependencies within the engine architecture.

The second representation is the matrix form of the same network, in which the rows and the lines of the original DSM were permuted, in order to distinguish the dependencies within each group of components (close to the diagonal), from the dependencies between the groups (upper-right and bottom-left part of the matrix). Together with the managers of the Technical Division, we analyzed these two representations and assessed their organizational implications. The definition of integration roles between two groups (the blue and the grey ones in the figure) , for example, was crucial for the development activity. The main managerial implications of the work therefore regards the analysis of product architecture (definition of key modules, assessment of "cohesiveness" of each module and dependencies between them, evaluation of the cascading effects of innovating central components, etc.) and the organization of development team (analysis of the degree of overlap between product modules and existing development units and definition of alternative organizational structures). The empirical test thus evidenced the importance of identifying *cohesive subgroups* within the system architecture and defining a *component centrality* scale. These system's architectural properties could facilitate the organizational design and the development activity planning and coordination. Social network techniques appear to be powerful instruments for studying product architectures and assessing the related organizational implications.

REFERENCES

[1] Ulrich K. The role of product architecture in manufacturing firm. *Research Policy*, 1995, 24, 419-440.

[2] Ulrich K. and Eppinger S. *Product design and development. Third Edition*, 2004 (McGraw-Hill, New York).

[3] Simon H. *The Sciences of the Artificial. Second Edition*, 1981 (MIT Press, Cambridge Mass).

[4] Sosa M., Eppinger S., Rowles C. Identifying modular and integrative systems and their impact on design team interactions. *Journal of Mechanical Design*, 2003, 125, 240-252.

[5] Browning T. Applying design structure matrix to system decomposition and integration problems: a review and new directions. *IEEE Transaction on Engineering Management*, 2001, 48, 292-306.

[6] Eppinger S. Innovation at the speed of information. *Harvard Business Review*, 2001, January, 149-158.

[7] Wasserman S., Faust K. *Social Network Analysis. Methods and Applications*, 1994 (Cambridge University Press, Cambridge).

[8] Borgatti, S.P., Everett M.G. and Freeman L.C. *Ucinet 6 for Windows*, 2002 (Harvard: Analytic Technologies).

Contact: Michele Liberati
University of Bologna
Department of Management
via Saragozza 8
40123 Bologna
Italy
Phone: 0039-0512093960
Fax: 0039-0512093949
e-mail: michele.liberati@unibo.it

9TH INTERNATIONAL DSM CONFERENCE

Social Network Techniques Applied to Design Structure Matrix Analysis. The Case of a New Engine Development at Ferrari Spa

Michele Liberati
Department of Management, University of Bologna, Italy.
Federico Munari
Department of Management, University of Bologna, Italy.
Paolo Racchetti
Ferrari Spa, Maranello (Mo), Italy.
Tazio Splendiani
Ferrari Spa, Maranello (Mo), Italy.

Agenda

- Introduction
- Background
- The Study
- Results
- Conclusions

Introduction

Complex Products: *"made up of a large number of parts that interact in a non-simple way, in which the whole is more than the sum of the parts"* (Simon, 1981)

> Aligning product architecture and organizational design is a necessary condition for the successful development of *complex product* (Sosa, Eppinger and Rowles, 2003)

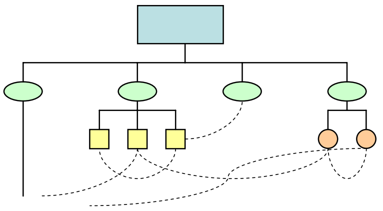
AIMS OF THE RESEARCH

1. **Developing a structured approach** to link architectural analysis and team organization for complex products development, using new instruments (Social Network Analysis);
2. **Validating the method** through its application to a new engine development project at Ferrari Spa.

Background

PRODUCT ARCHITECTURE

1) **Ulrich, 1995**:
   - *"The arrangement of functional elements;*
   - *The mapping from functional elements to physical components;*
   - *The specification of the interfaces among interacting physical components".*
2) **Ulrich and Eppinger, 2004**:
   The scheme by "*which the chunks [of a product] interacts*".



ORGANIZATIONAL DESIGN

1) **Eppinger, 2002**:
   A development organization is composed by *teams*, further decomposed into *working groups* and *individual assignments*.
   - Definition of teams and sub-unit composition;
   - Assignment of coordination responsibilities;
   - Definition of intra-units integration arrangements;
   - Creation of the conditions to facilitate the communication flow between members.

Background

"The development of complex product is a highly interactive social process involving hundreds of people designing thousand of interrelated components and making millions of coupled decisions [.....]"

" [.....] we expect to find that firms in which the interaction patterns across product components and organizational units are well aligned will outperform the others."
(Eppinger, 2002)

PRODUCT ARCHITECTURE ⟺ ***ORGANIZATIONAL DESIGN***

" [.....] importance of identifying design interfaces during the project planning stage so that corresponding design team are managed efficiently during project execution [.....] " (Sosa, Eppinger and Rowles, 2003, p. 240)

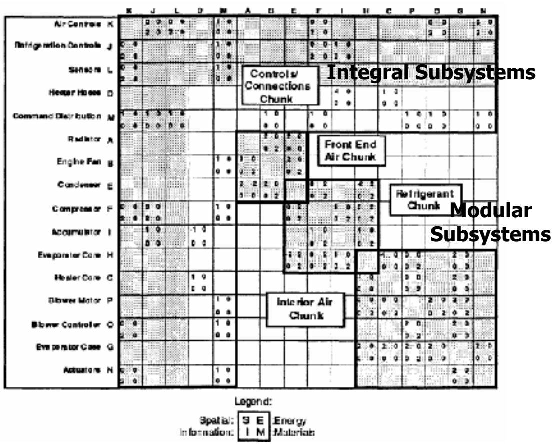
Background

DESIGN STRUCTURE MATRIX AND PRODUCT ARCHITECTURE

Component-Based (Architecture) DSM

Modeling system architectures through the analysis of the components (subsystems) and their interactions (Browning, 2001):

1. Decompose the system into elements;
2. Document the interactions between the elements;
3. <u>Analyze the matrix through clustering procedures</u>.

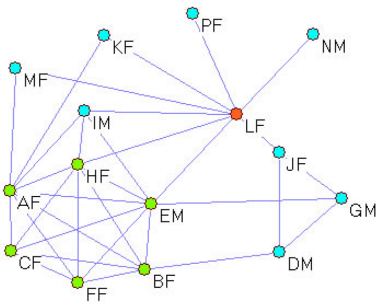


Component-Based DSM for automotive climate control system (Suorce: *Pimmler and Eppinger, 1994).*

Background

SOCIAL NETWORK ANALYSIS (SNA)

"[.....] The phrase "social network" refers to the set of actors and the ties among them. The network analyst would seek to model these relationships to depict the structure of a group. One could then study the impact of this structure on the functioning of a group [.....]" (Wasserman and Faust, 1994)



An example of network representation.

<u>Traditional Fields of Application</u>

Social, Economic, Political research.

<u>The Social Network Perspective</u>

- Actors and their actions are seen as interdependent;
- The network structure provides opportunities for or constraints on individual actions;
- Network models conceptualize structure as patterns of relations among actors.

<u>Object of Investigation</u>

SNA focuses the relationships among social entities and on the patterns of these relationships (Network Unit of Analysis).

The Study

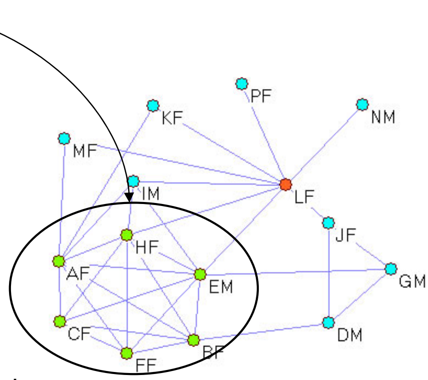
THE APPLICATION OF SNA TO PRODUCT ARCHITECTURE AND TEAM ORGANIZATIONAL DESIGN

Cohesive Subgroups

SNA Domain

Groups of actors, who interact with each other to such an extent that could be considered distinctive entities (Wasserman and Faust, 1994)

<u>Closeness of relationships among a set of actors within a network</u>.



Product Architecture Domain

Groups of components, who interact with each other to such an extent that could be considered distinctive entities (*<u>Modules</u>*)

<u>Closeness of dependencies among a set of components within a product</u>.

The Study

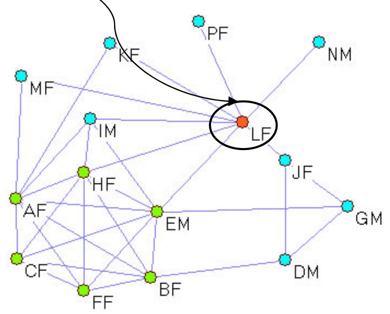
THE APPLICATION OF SNA TO PRODUCT ARCHITECTURE AND TEAM ORGANIZATIONAL DESIGN

Centrality

SNA Domain

Central actors within the network are the ones extensively involved in relationships with other actors (Freeman, 1979)

Number (and strength) of relationships the actor is involved in.



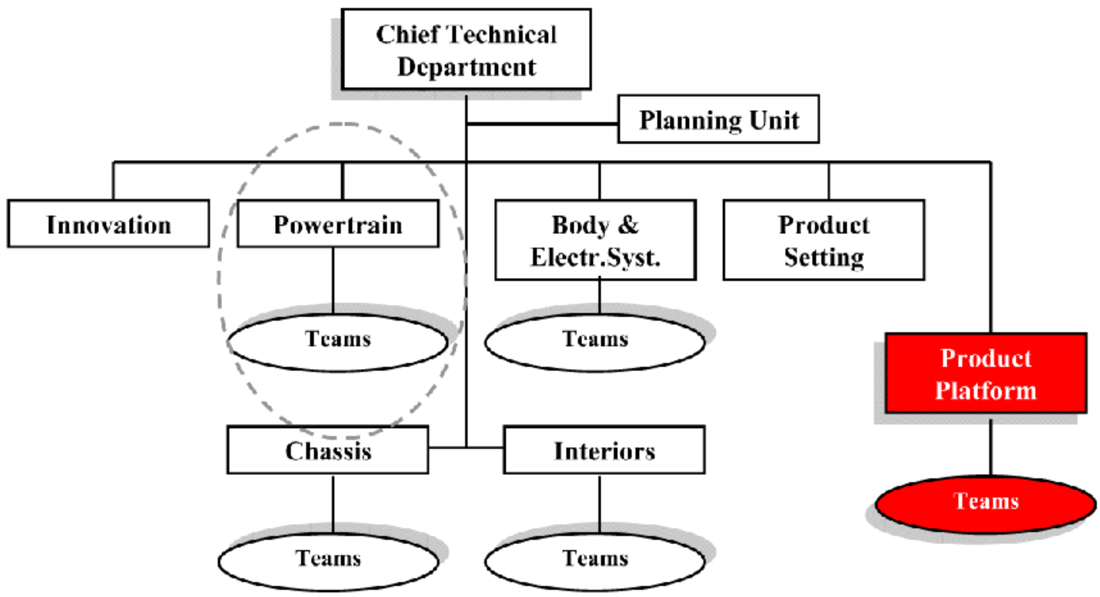
Product Architecture Domain

Central components within the product are the ones extensively interdependent with other components

Number (and strength) of dependencies the component shows.

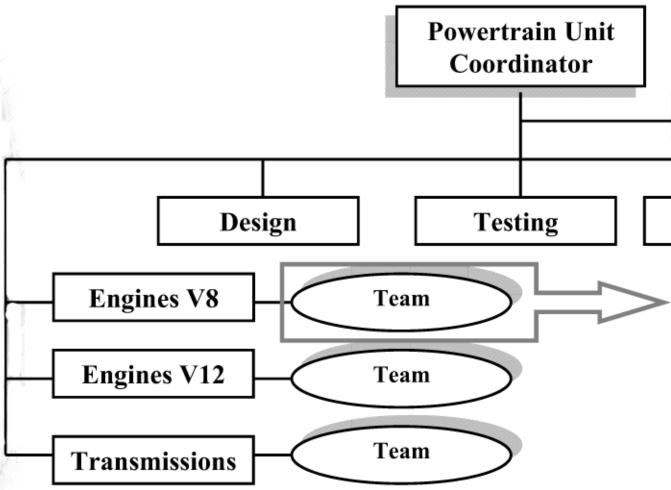
The Study

FERRARI TECHNICAL DEPARTMENT



The Study

FERRARI POWERTRAIN DIVISION AND THE PROJECT



- Platform Engine;
- Increasing engine power (Hp);
- More than 30 engineers;
- Deep collaboration with selected suppliers.

The Study

METHODOLOGY, STEP 1-2

1. Defining the level of analysis, from the product-system hierarchy: identifying the key subsystems-components included in the product to be studied as the initial step to build the Design Structure Matrix;

   Ferrari Case (46 components)

   - Function performed by the component;
   - Responsibility of component development activity;
   - External organization involvement.

2. Identifying a set of relevant design dependencies among components, for the specific product and constructing a DSM for each one (Sosa, Eppinger and Rowles, 2003-2004).

   Ferrari Case (3 design dependencies)

   a) *Spatial*
   b) *Structural*
   c) *Material*

The Study

METHODOLOGY, STEP 3-4

3. Analyzing the dependencies data collected into the DSM, using Social Network Analysis techniques (software UCINET 6.1 © 2006, Analytic Technologies);

   Ferrari Case

   a) Identification of *cohesive subgroups:*
      - Identification of an appropriate number of subgroups, through "*cliques overlap*" analysis: procedure NETWORK > SUBGROUPS > CLIQUES;
      - Clustering the network: procedure TOOLS > CLUSTERING > OPTIMISATION.

   b) Definition of a *components' centrality scale:*
      - Computation of a centrality index for each component: procedure NETWORK > CENTRALITY > EIGENVECTOR

4. Designing the development team organization:

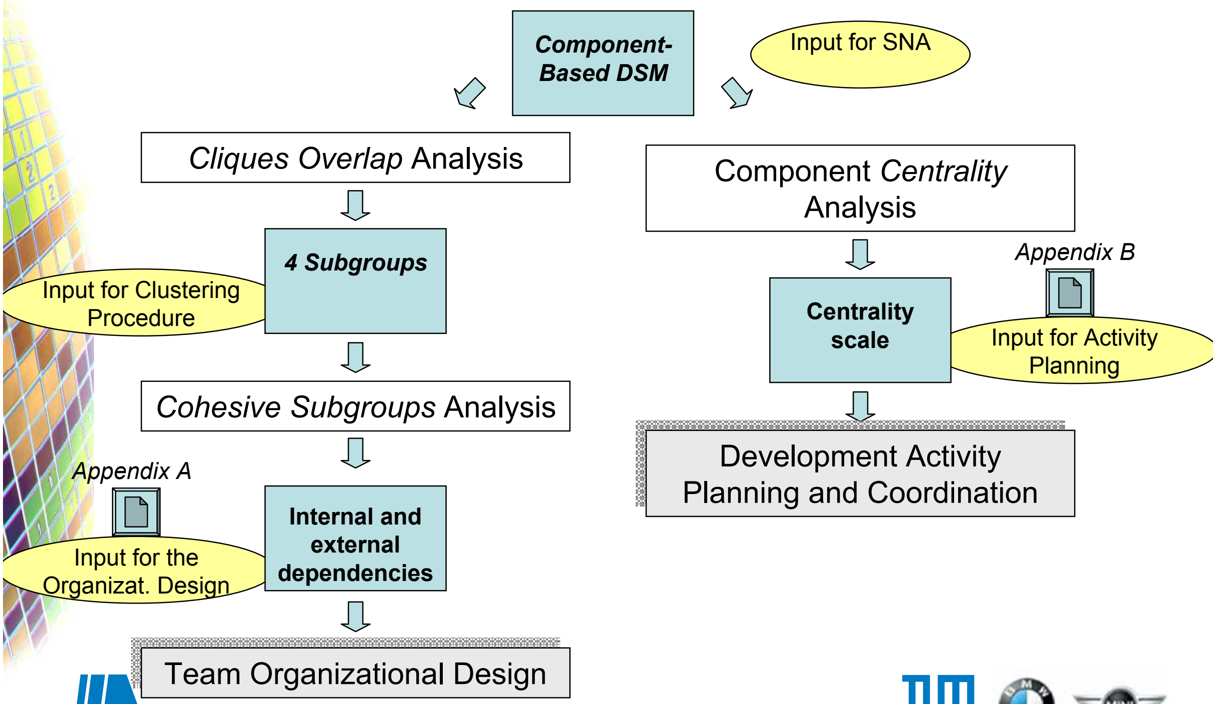
   Ferrari Case

   a) Definition of subunits composition and responsibilities;

   b) Creation of integration mechanism between the subunits;

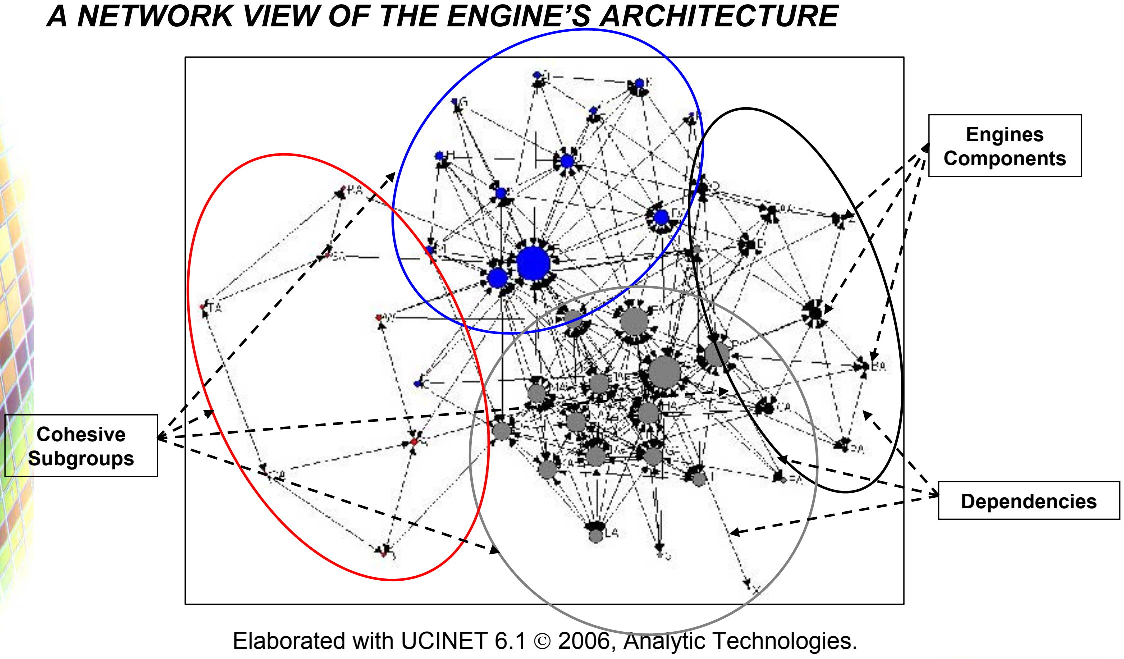
   c) Assessment of a criticality level for each component development project.

Results



Results

A NETWORK VIEW OF THE ENGINE'S ARCHITECTURE



Elaborated with UCINET 6.1 © 2006, Analytic Technologies.

Conclusions

THE DESIGN OF PRODUCT ARCHITECTURE

- 4 different subgroups were identified and the components were clustered maximizing the internal dependencies and minimizing the external ones;
- 2 powerful representations of components' dependencies were provided;
- Central components in the system architecture were identified.

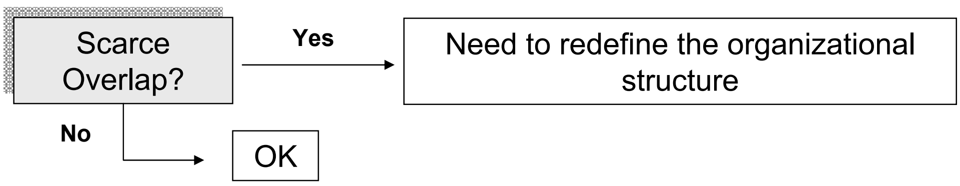
IMPLICATION FOR DEVELOPMENT WORK

- Identification and optimization of key modules composition;
- Inter and Intra-modules dependencies assessment;
- Focusing on central components and evaluating the possible cascading effects of innovating these components.

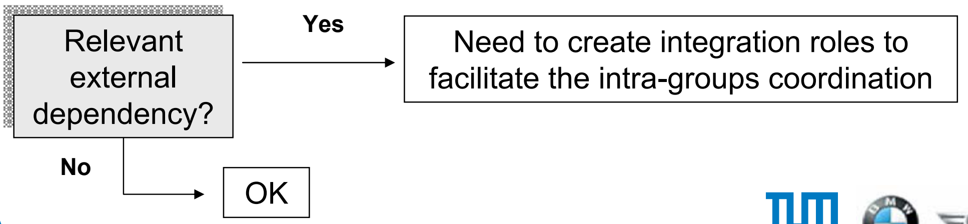
Conclusions

ORGANIZATIONAL DESIGN

1. Analysis of the degree of overlap between product subgroups and existing organizational units:

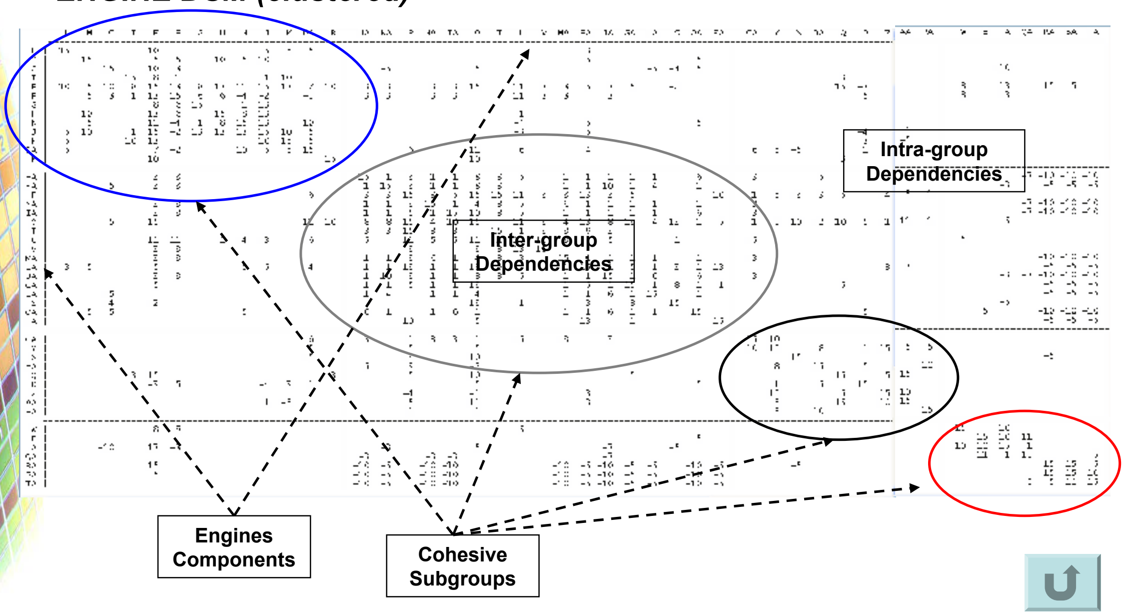


2. Analysis of internal vs external dependencies between the components of each organizational unit:



Appendix A

ENGINE DSM (clustered)



Appendix B

Bonacich Eigenvector Centralities

		1 Eigenvec	2 nEigenvec
1	A	0.050	7.135
2	B	0.010	1.435
3	C	0.053	7.454
4	D	0.050	4.759
5	E	0.299	42.222
6	F	0.135	19.114
7	G	0.060	8.462
8	H	0.099	13.946
9	I	0.047	6.536
10	J	0.139	19.671
11	K	0.076	10.743
12	L	0.067	9.474
13	M	0.084	11.820
14	N	0.102	14.353
15	O	0.397	55.118
16	P	0.262	37.101
17	Q	0.154	21.848
18	R	0.089	12.616
19	S	0.078	11.055
20	T	0.118	16.644
21	U	0.213	30.055
22	V	0.077	10.840
23	W	0.067	9.471
24	X	0.059	7.574
25	Y	0.068	8.593
26	Z	0.026	3.741
27	AA	0.110	15.794
28	BA	0.021	2.972
29	CA	0.036	5.054
30	DA	0.154	21.837
31	EA	0.253	35.849
32	FA	0.133	18.841
33	GA	0.181	25.573
34	HA	0.183	25.915
35	IA	0.198	27.953
36	JA	0.201	28.378
37	KA	0.142	20.097
38	LA	0.089	12.527
39	MA	0.155	21.943
40	NA	0.164	23.157
41	OA	0.147	20.831
42	PA	0.015	2.055
43	QA	-0.038	-5.336
44	RA	-0.174	-24.559
45	SA	-0.213	-30.107
46	TA	0.212	30.024