

Connectivity maps: modeling and analysing relationships in product development processes

ALI YASSINE^{†*}, DANIEL WHITNEY[†],
STEVE DALEIDEN[‡] and JERRY LAVINE[‡]

This paper describes a new systems analysis technique called the ‘connectivity map’ for representing dependency relationships within a product development process. The technique is suitable for capturing and analysing relations between development tasks, design parameters, architecture concepts, information flows, and organizational relationships. The connectivity map is matrix based, using the inner cells of the matrix to capture the nature of the connection between the two axial parameters. The utility of the method is demonstrated using two real examples from the automobile industry. The first application looks at the sources of iteration for a safety belt development process, and the second application deals with flexibility analysis of a product’s architecture.

1. Motivation

Engineering complex products pose both technical and managerial challenges. The technical complexities are usually addressed through decomposing the development process into smaller and simpler elements (i.e. tasks, subsystems) handled by various development teams. Decomposition helps in containing the technical complexity of the design; however, it increases its managerial complexity. The synthesis of the different elements (or subsystems) into a final product (or systems) requires the identification and understanding of the inter-relationships among the different elements. These elements can be many things: physical components that make up the product, performance attributes expected by the consumer, engineering requirements the product must meet, or tasks that create the product. Each of these different element types contributes to the development of a product.

Relationships can exist within element types and between element types. This makes the product and its development complex, and makes capturing relationships between affiliated elements necessary if the product engineering is to be handled (i.e. analysed, engineered, documented, developed, and managed) efficiently and effectively, and if the product is to be robust and successful. This complexity also makes the elemental relationships difficult to visualize and hard to communicate. The main source of difficulty is the lack of models capable of capturing all of these elements and their interdependencies in a single, but simple, way.

For simpler products, the elemental relationships are usually anecdotes or heuristics. On more complex products, written specifications, computer-aided engineering models, drawings, and work plans are used effectively to capture relationships within element

Revision received January 2003.

[†] MIT Center for Technology, Policy, and Industrial Development, Cambridge, MA 02139, USA.

[‡] Ford Motor Company, Dearborn, Michigan, USA.

* To whom correspondence should be addressed. e-mail: yassine@mit.edu

types, while basic attentive engineering handles relations between element types. On very complex products, these tools are still able to support lower-level (within element) efforts. However, they break down and cannot manage the number of relations between element types. To fill this void and improve the development of the product, a simple, compact, comprehensible method is needed to capture these relationships. In the present paper, we propose a matrix-based method called the ‘connectivity map’ that is suitable for capturing and analysing relations between development tasks, architecture concepts, information types, organizations and dependency relationships.

The following sections will outline simple relationship maps and introduce the more complex connectivity map as a method for compactly capturing relationships. This will be followed with specific examples of how the concept of the connectivity map has been applied to automobile subsystems.

2. Relationship maps

One simple, compact approach that has proved successful in representing a system is the creation of a matrix that maps and captures the relationship between different system elements. Generally, these relationship maps are simple A versus B matrices. Two different sets of elements define the matrix, with one set on each axis of the matrix. Marks in the cells of the matrix represent relationships or dependencies between the intersecting subelements. Figure 1 shows a generic example. In this example, the relationships between element type A (with subelements A1, A2, A3, A4, A5, and A6) and element type B (with subelements B1, B2, B3, B4, B5, B6, B7, and B8) are mapped. The marks in the cell indicate which particular subelements are related.

In this example, the relationship map does not specify the direction or nature of the relationship. Information about precedence, dependence or influence can be added via the use of different marks in the cells to represent different types of relationships. For instance, numerical rankings, symbols, or colours can be used in the cells to indicate the relative strengths, direction, or significance of the relation.

		ELEMENT TYPE A					
		element A ₁	element A ₂	element A ₃	element A ₄	element A ₅	element A ₆
ELEMENT TYPE B	element B ₁	X	X				
	element B ₂				X		
	element B ₃		X	X			
	element B ₄			X			
	element B ₅			X			
	element B ₆				X		
	element B ₇				X	X	
	element B ₈						X

Figure 1. Relationship map example.

The idea of using a matrix representation of a system is not new. Several researchers proposed matrix-based methods for system modelling and analysis more than 30 years ago (Steward 1962, 1981, Kehat and Shacham 1973, Warfield 1973). However, it is not until recently that these methods started to attract attention for managing the complexity of large engineering systems and complex product development processes (Hauser and Clausing 1988, Kusiak and Park 1990, Eppinger 1991, Yassine et al. 1999. Steward (1962, 1981) introduced a matrix model for a system called the 'design structure matrix' (DSM) and developed some algorithms for manipulating the matrix as tools for systems design and analysis [1]. Warfield (1973) presented a tutorial paper on the use of binary matrices in system modeling. Kusiak and Park (1990) suggested the use of a module-activity incidence matrix as a tool for clustering the design activities into concurrent groups. Finally, the quality function deployment (QFD) method described by Hauser and Clausing (1988) is another matrix-based model for systems design and analysis in an engineering context. QFD is an example of a relationship map used to translate customer needs into engineering requirements through a matrix known as the 'house of quality'. Using symbols and numerical rankings, the matrix compactly but comprehensively represents the relationships between customer wants, competitive products, engineering metrics, and specifications. Suh (1990) utilizes a 'design matrix' to identify relationships between requirements and product design parameters. More recent papers on matrix-based models utility in engineering management are mainly based on the DSM method (Eppinger 1991, Eppinger et al. 1994, Lockledge and Salustri 1999, Yassine and Falkenburg 1999). Of particular relevance to our work is the paper on DSM by Lockledge and Salustri (1999). They have developed a modified DSM model to capture the relationships between functional requirements and physical components.

All the aforementioned matrix-based tools provide some insight into the dependency structure of a complex system or process (i.e. a dependency map); however, they fail to expose and explore the logic behind these dependencies. That is, if an element B is shown to relate to another element C, it is not clear why and how this relation exists and where it stems from. An added value to these relationship maps is to be able to show/capture that element B is related to C through some characteristic A. In doing so, the analyst is capable of understanding the complexity of a system better and in turn has more information to manage these dependencies and leverage them to the improvement of system performance. The connectivity maps are one way of developing such expanded relationship maps, as discussed in the following.

3. Connectivity maps

A more complex matrix-based tool is the connectivity map. A connectivity map is a B versus C via A matrix. Essentially, it is a combination of two relationship maps into a single $N \times M$ matrix. Subelements of B and C are the axial parameters. Information on A, whose dependency relationships with B and with C are known (and could be documented in another relationship map), is used to fill the cells of the connectivity map. The cells of the connectivity map identify the subelements of A that influence subelements of B and subelements of C. In this way, using the subelements of A, the connectivity map captures the nature of the relationships between subelements of B and C.

Figure 2 provides a simple example of a connectivity map and presents a schematic of how it might be constructed. As shown in the figure, the roots of the connectivity map can be found in relationship maps. The element types B and C are put on the axes, while their respective relationships with element type A are used to map their connection. In this example, the numbers in the cells refer to the particular subelements of A that connect particular subelements of B and C. A couple of specific examples highlighted in Figure 2 are:

- subelement A1 connects subelement B1 and C1; and
- subelement A3 connects subelements B3 and C7.

Note that some subelements of B and C might be connected by multiple subelements of A. For instance, subelements B7 and C5 are connected by both subelements A4 and A5. This means that both A4 and A5 have relationships with B7 and C5.

If the 'X' marks in matrix A and B are replaced by a numerical scheme (such as using '1' to indicate the existence of a relationship and '0' to indicate the absence), then a numerical connectivity map can be generated. A numerical connectivity map can serve as an accounting ledger for identifying what connections are complex. For instance, if the '1' or '0' scheme is used, then those cells that have numbers greater than 1 are clearly indicating the existence of multiple relationships [2].

Because it is a matrix, the connectivity map is compact and easily constructed and modified, which are important attributes in any project management tool. However, the

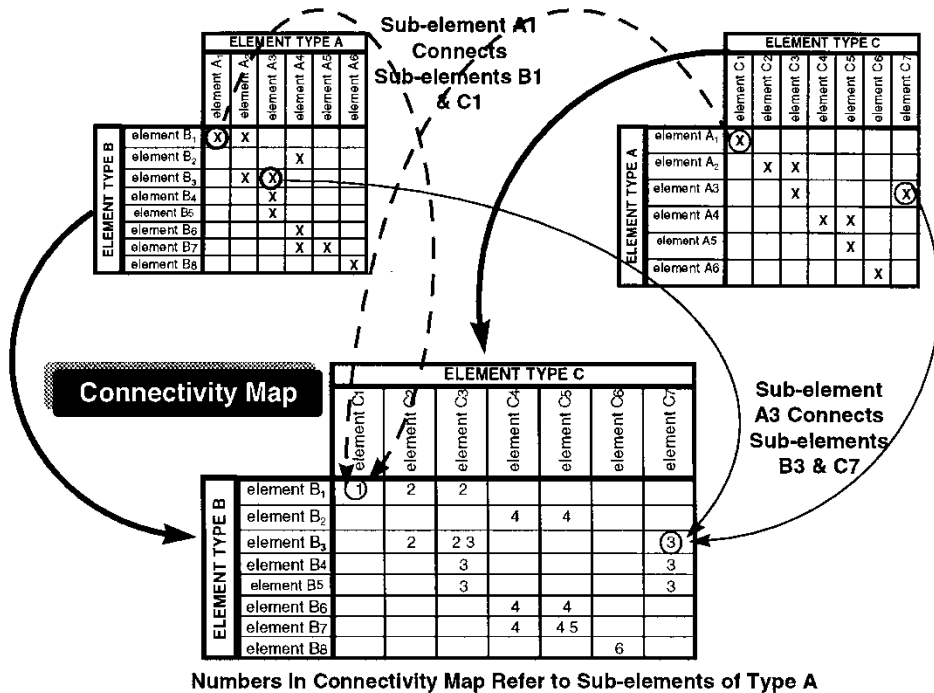


Figure 2. Connectivity map example.

complexity in the connectivity map is introduced by the method or code used to capture the three-way relationships.

The key step in the development of the connectivity map is representing the connections between the three types of subelements in such a way that the map remains compact yet is still able to communicate information and support design or management analysis. The marks in the cells, which represent the relationships (e.g. influence, dependence and association) between subelements must be carefully designed. The challenge is for the coded marks in the cells to not just indicate the existence of a relationship, but also to inform on an aspect of that relation by illuminating the nature of the connection between the three types of elements. The specific coding or legend chosen is highly dependent on the particular elements being studied and the complexity of their inter-relationship. Numerical rankings, symbols, or colours can be used to indicate the relative strengths, direction, or significance of the relation. However, unless the A-C (or A-B) relation is very simple, an alphanumeric code (or symbol or colour) will probably have to be developed for the subelements of A to keep the connectivity map relatively compact and intuitive. If the relationships are extremely complex, the connectivity map can become overwhelmed or equivocal.

The simple example in figure 2, although instructive, cannot communicate the valuable insight that can be developed using the connectivity map, or the various challenges and shortcomings of the method. To better illustrate the method, the next sections present examples of how a connectivity map can be generated and analysed. The examples, both based on the development of automobile subsystems, look at the safety belt development process and the chassis architecture, demonstrating the application of the methodology and discuss the insights gained.

4. Applications from the automotive industry

One application of a connectivity map is to model the relationships between the development process tasks and the physical product via functional requirements. This type of model identifies the product architecture type, requirements requiring coordination during development and identifies potential process improvements. The format for this model has the vertical axis being the tasks that make up the process with the other axis being the components that aggregate to form the physical system. The functional requirements are represented at the corresponding intersection of the row and column where the process and product interact to fulfill requirements (figure 3a) [3]. The focus of this map is to determine whether each task of the process is value-added and satisfies a functional requirement, either partially or completely. The physical system can be evaluated to determine its value based on its relations with functional requirements. That is, if tasks or components are identified that do not relate to a functional requirement, they should be investigated with respect to mating systems to determine whether they are non-value added holistically and, therefore, can be eliminated or can be managed independently. Section 4.1 will discuss this map type in detail.

Another application of a connectivity map shows how it can be used to model and evaluate product architecture for characteristics such as modularity or flexibility. This map examines the relationships between three element types that drive and compose the product architecture. The element types of interest are consumer-perceived attributes, objective engineering metrics, and components design parameters. The format for this model has the vertical axis being the consumer attributes, with the other axis

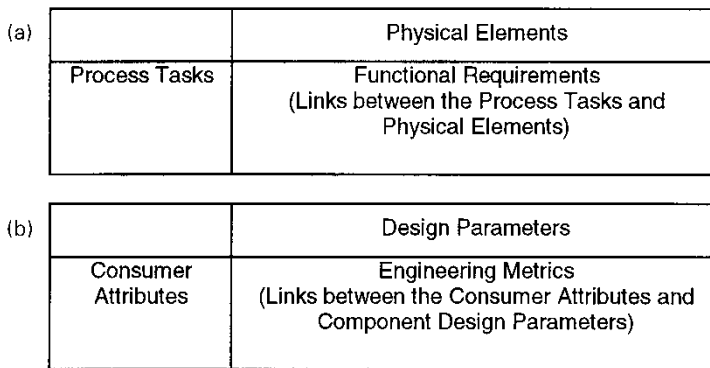


Figure 3. Two applications of a connectivity map. (a) System, process and functional requirements connectivity map. (b) Parameters, attributes and engineering metrics connectivity map.

being the design parameters that are specified to deliver the intended function. The engineering metrics are represented at the corresponding intersection of the row and column where the design parameters and consumer attributes link via engineering metrics that are managed as a part of the development process (figure 3b). Section 4.2 will describe this map in detail.

4.1. Safety belt development process (Lavine 2000)

4.1.1. The development process. The safety belt development process was decomposed into 15 design tasks (as shown later in the row titles of the matrix in figure 6). These tasks were identified by interviewing participants in the process, reviewing work procedures, and participating in some of these tasks by one of the co-authors.

4.1.2. The physical system. Figure 4 depicts the physical decomposition of the safety belt system. The number classification for an element identifies the grouping and level of abstraction. The elements at the lowest level are those defined as a part of the detailed design phase. An example is for the underbody (1.1.1) attachment (1.1.1.1) that requires an attachment type, along with the required physical attributes (e.g. hole shape and size) that are determined as a part of the detailed design. Understanding an element's requirements is a pre-requisite to detailed design and has an impact on the efficiency and effectiveness of the process and organizational structure.

4.1.3. The functional requirements. Figure 5 illustrates the functional requirements decomposition for the same system. Here, the elements represent evaluation criteria that can be used to evaluate the physical system's and subsystems' performance. Based on these requirements, detailed designs and concepts can be analysed and optimized.

4.1.4. The connectivity map: analysis and discussion. Figure 6 is the connectivity map for this safety belt application. The callouts in the cells of the connectivity map represent the elements of the functional requirements that relate the process

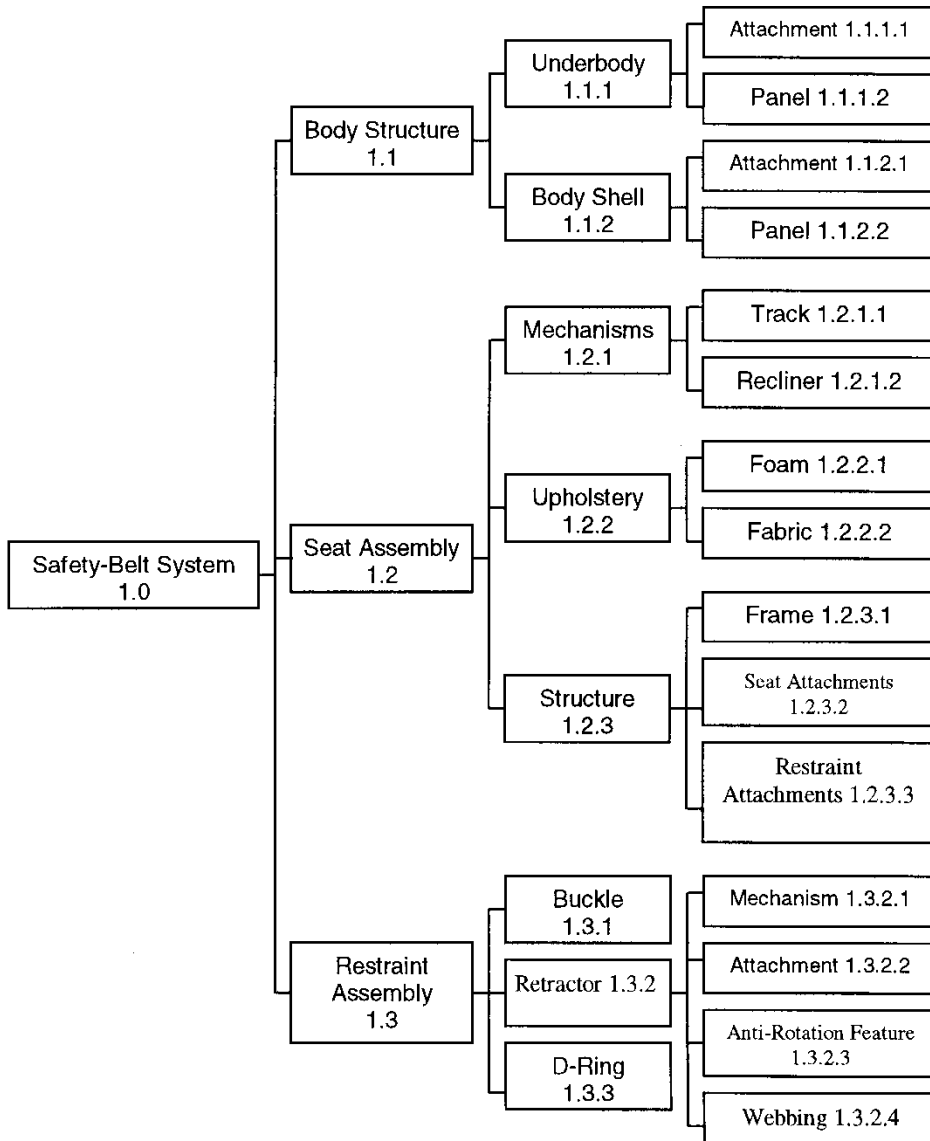


Figure 4. Physical system decomposition.

task to the physical system. A blank cell means that the column's physical element is independent of that row's task. Where multiple functional requirements are called out, the task is providing and/or manipulating information for that component that is intended to support multiple requirements.

Consider the 'determine seat travel' task in figure 6 as an example of how to analyse the connectivity map. The connectivity map shows that this task is partially responsible for the detailed design of the 'track' and 'recliner', and the designs must satisfy the 'accommodate seating' and 'comfort' functional requirements. As a result of

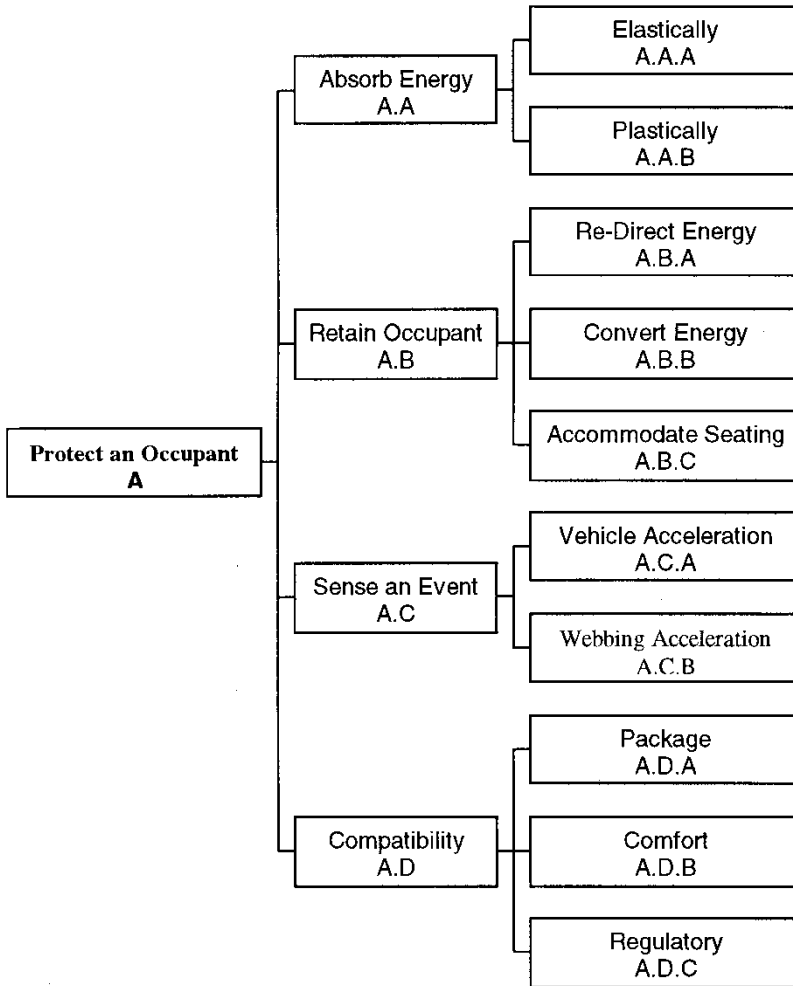


Figure 5. Functional decomposition.

the physical components of the safety belt subsystem fulfilling multiple requirements, the connectivity map reveals a coupled product architecture. The connectivity map can also be analysed to identify why tasks are iterative within a process. For the previous example, the task 'determine seat travel' is iterative because of information requirements from the define seating hardpoints task. This implies that common physical elements, element boundaries, and functional requirements are parts of these tasks.

For these tasks, two of the physical elements affected are common (i.e. track and recliner) and need to satisfy the same functional requirements (i.e. accommodate seating and comfort). In this case, the connectivity map qualified the elements and requirements being shared between the tasks. Analysing the third task, define allowable anchorage locations, shows that this task has three elements in common with the two preceding tasks, and one common functional requirement. The common functional requirement is the information being transferred between the tasks. In general, the

Physical components	Functional requirements	Cause and type of iteration
Common between tasks	Common requirements between tasks	Design refinement during process, and/or inconsistent evaluation for performance attribute
Common between tasks	Unique requirements for each task	Trade-off between functions based on related parameters of the physical system
Unique components for each task	Common requirements between tasks	Integral architecture requiring coordination for system level performance
Unique components each for task	Unique requirements for each task	Component interfaces interact to affect system level attribute

Table 1. Connectivity map metrics for type and cause of iteration.

evolution during the process. The connectivity map identifies the type and cause of iteration by relating the process to the physical system through the functional requirements. Multiple tasks that affect the same component or functional requirement need to be managed throughout the process. This management needs to ensure that downstream tasks understand the motivation of the preceding tasks and the impact/requirements of succeeding tasks.

The seat assembly fabric is the only physical element that is affected by one functional requirement (comfort) and one task (style and develop exterior of seat) that implies a modular engineering single attribute optimization approach.

Finally, this methodology also identified that the following tasks do not satisfy a functional requirement of the physical system (i.e. they have empty rows):

- specify market availability;
- determine applicable safety belt regulations; and
- interpret regulations.

Based on this analysis these tasks are independent process requirements that can be instituted outside of the development process and be performed by a different organization.

4.2. Automobile chassis architecture mapping (Daleiden 2000)

4.2.1. The consumer-perceived attributes. Consumer-perceived attributes represent the critical sensible perceptions that customers use to evaluate their vehicles. When considering chassis architecture and component design, the customer attributes of interest relate to noise, ride, handling, and vibration, also known as vehicle dynamics and 'noise, vibration, harshness'. These attributes are strongly influenced by the design of chassis systems, although body mass and stiffness also play a significant role in these attributes. Discerning and capturing consumer differentiation among these attributes is imprecise, because consumers have difficulties verbalizing what they are sensing and from where it is coming. This leads to coupling between the attributes, which will affect the utility of the connectivity map.

The key consumer-perceived attributes used in the analysis are ride, steering, and handling. However, for clarity and ease of exposition, we will only consider the ride attribute in this paper. Furthermore, we have decomposed this attribute into three sub-attributes as follows: smooth ride, rough ride, and abruptness, as shown in the row headings of figure 7.

4.2.2. The design parameters. To develop the connectivity map, a specific architecture is required. For the case study, a very common architecture was chosen:

- front-wheel drive layout with half-shafts and front subframe; and
- front MacPherson suspension

Furthermore, this example focuses on the front suspension and ignores the rear suspension, steering system, brake system, body and powertrain elements. The key systems and components for the front suspension are listed as column headings in figure 7. Notice how these parameters are divided into two primary categories: stiffness/compliance and geometry/dimension [4]. This division is not necessary for the development and analysis of the connectivity map; however, it was convenient for this particular application.

4.2.3. The objective metrics. To properly translate customer-perceived attributes, vehicle and system objective metrics for chassis performance are needed. Objective metrics are parameters that can be measured and predicted, and to which engineers

SUSPENSION COMPONENT DESIGN PARAMETERS	Stiffness/ Compliance														Geometry/ Dimension					
	front tyre / wheel compliance	rear tire compliance	front stabilizer/torsion bars	front springs/ strut/ shock	front control arm bushings	body mounts	bump stop	engine mounts	rear control arm bushings	rear stabilizer/torsion bars	rear springs	front control arm ball joints	body stiffness	front spindle	front hub & bearing	front knuckles	front control arms assy.	front stabilizer bar linkage	front struts/ shock absorbers	bump stop location
RIDE ATTRIBUTES / SUB-ATTRIBUTES	c1	c2	k1 c3 d2	k1 k2 k4 k5 c1 c3 d1 d2 g8	k5 c1 c7 d2	k1 d1 d2	d1 g8	d1	k6 c2 c4 c8 d2	k1 d2	k1 k6 c2 c4 d1 d2	k4 k5 c1 d1 d2 d2 g4	k1 k5 k6 d1 d2							
Smooth Road - primary ride/ body motion; bounce, displacement, float, pitch, balance																				
Rough Road - ride/ body motion under perturbations	c1	c2	k1 d2	k1 k2 k4 k5 c1 d1 d2 g8	k5 c1 c7 d2	k1 d1 d2	d1 g8	d1	k6 c2 c8 d2	k1 d2	k1 k5 c2 d1 d2	k4 k5 c1 c7 d1 d2 g4	k1 k5 k6 d1 d2			k1 k2 k5 d2 g8	k1 k2 k4 k5 g4 g8	k1 d2	k1 k5 g4 g8	g8
Abruptness - impacts, bump through, topping			k1 d2	k1 k4 k5 c1 d1 d2 g8	k5 c7 d2	k1 d1 d2		d1	k6 c8 d2	k1 d2	k1 k6 d1 d2	k4 k5 d1 d1 d2 g4	k1 k5 k6 d1 d2			k1 k5 d2 g8	k1 k4 k5 g4 g8	k1 d2	k1 k5 g4 g8	g8

Figure 7. Connectivity map for the chassis front suspension example.

can design. The objective metrics, when assigned specific values or target envelopes, become functional requirements that the system, subsystem, or component must achieve to be acceptable to the consumer.

Many objective metrics for chassis are vehicle dynamics metrics. They are standard within the automotive industry and documented in the vehicle dynamics literature (Gillespie 1992). Essentially, these metrics fall into three major categories.

- Kinematic objective metrics describe the relative motion of chassis components along with the forces, moments, and axes that enable the motion.
- Compliance objective metrics describe the compliance, stiffness, spring rates, damping and natural frequencies of chassis systems and components.
- Static design objective metrics describe the geometry, shape, material, and functional qualities.

The objective metrics used to develop the connectivity map are listed in figure 8.

4.2.4. The connectivity map: analysis and discussion. The connectivity map for the chassis front suspension example is shown in figure 7. For this example, the objective was to capture, visualize, and draw conclusions about a product's architecture, its complexity, and its ability to support efficient, manageable changes that can alter the customer-perceived attributes. This meant studying and interpreting the patterns displayed in the connectivity map. The patterns communicate what attributes are emerging, what combinations of objective metrics are interacting to cause or create particular attributes, and which components' interactions are influencing particular objective metrics. The patterns also indicate which functional boundaries are being crossed by the interactions.

The number of functional boundaries that are crossed by component or system interactions is often used to assess complexity. Understanding the ability to support efficient, manageable attribute-impacting change means capturing modularity and flexibility. Modularity exists when each function or attribute of a product is implemented by a single physical chunk and the interactions between chunks are few and well defined (Ulrich and Eppinger 2000). Flexibility describes the ability of a product to respond to changes in design direction and support derivative product or to act as a platform for a portfolio of products (Daleiden 2000).

A particular difficulty in assessing flexibility is encompassing the physics of the relationship between components, objective metrics, and the attributes. This requires the existence of mathematical models, which do not necessarily exist or are not able to completely and reliably model the real world. An automobile chassis system, including the front suspension, is an example.

Table 2 summarizes the characteristics (or 'genes') that different architectures can have regarding modularity and flexibility. This table provides descriptions of the genes, how they can be identified in the connectivity map, and what their existence implies.

With this evaluation framework in place, the ride-front suspension connectivity map can be evaluated, and the flexibility and modularity of the architecture can be assessed. The first observation is that all the attributes have a cell filled for each of the component design parameters listed, except for the front spindle and front hub/bearing geometry/dimension parameters. Also note that if a component design parameter impacts smooth ride, it also impacts rough road ride and usually (but not always) impacts abruptness. This indicates that this is not a modular architecture.

<p>Suspension Kinematic (k)</p> <ul style="list-style-type: none"> ⇒ roll centers, roll center migration (k1) ⇒ kinematic steer front axle (k2) ⇒ kinematic steer rear axle (k3) ⇒ camber change (k4) ⇒ front axle anti-dive/ anti-lift (k5) ⇒ rear axle anti-dive/ anti-lift (k6) ⇒ kinematic steer asymmetry – front (k7) ⇒ kinematic steer asymmetry – rear (k8) ⇒ asymmetric suspension set up (k9)
<p>Suspension Compliance (c)</p> <ul style="list-style-type: none"> ⇒ lateral steer front axle (c1) ⇒ lateral steer rear axle (c2) ⇒ lateral stiffness - front axle (c3) ⇒ lateral stiffness - rear axle (c4) ⇒ longitudinal compliance steer front axle (c5) ⇒ longitudinal compliance steer rear axle (c6) ⇒ longitudinal compliance front axle (c7) ⇒ longitudinal compliance rear axle (c8) ⇒ camber compliance front/ rear(c9) ⇒ castor stiffness (c0) ⇒ ride frequencies (d1) ⇒ roll stiffness (d2)
<p>Static Geometry (g)</p> <ul style="list-style-type: none"> ⇒ static toe front (g1) ⇒ static toe rear (g2) ⇒ hub trail (g3) ⇒ static camber front (g4) ⇒ static camber rear (g5) ⇒ camber asymmetry front (g6) ⇒ camber asymmetry rear (g6) ⇒ wheel stroke/ suspension travel (g7) ⇒ castor (g8) ⇒ ackerman angle (g9)

Figure 8. Objective metrics used in the connectivity map.

However, the connectivity map also shows that the conditions for pseudo-modularity do not exist—there are no unique combinations of components uniquely connected to any attribute.

The conditions for multi-attribute flexibility and multi-path flexibility do, however, exist. Whether they are truly useful is dependent on the existence and reliability of predictive models that would allow the responses of the many interactions to be predicted. For instance, front spring/strut/shock stiffness/compliance influences nine smooth ride objective metrics, more than any other component design parameter. The geometry/dimension counterpart influences one additional objective metric that impacts smooth ride, front wheel stroke/suspension travel. Based on the number of connections between the design parameter and the attribute, the implied complexity would suggest that this design parameter cannot be used to achieve flexibility. However, the component is really a series of springs, dampers, and masses with simple dimensions, which

Gene	Description	Indicator on map	Implication
Multi-path flexibility	More than one component that can be used to achieve the same response in an attribute	Several cells filled for component column	<ul style="list-style-type: none"> • Different ways to achieve desired response • Flexibility dependent upon complexity of physics and models
Multi-attribute flexibility	One component can affect or control the response of several attributes simultaneously	Several cells filled for attribute row	<ul style="list-style-type: none"> • Possibly very efficient way of achieving flexibility • Dependent upon complexity of physics and models
Pseudo-modular	Unique combinations of physical chunks implement unique combinations of attributes or functions	Unique set of filled cells for attribute row and unique set of filled cells for component column	<ul style="list-style-type: none"> • Possibly very efficient way of achieving flexibility • Flexibility dependent upon complexity of physics and models
Modular flexibility	Each attribute of a product is implemented by a single physical chunk and the interactions between chunks are few and well defined	One filled cell for attribute row	<ul style="list-style-type: none"> • Flexibility achievable but limited by singular, unique relationships

Table 2. Summary of product architecture genes.

allows the physical relations to be approximated and accurately modeled with software tools such as ADAMS. This in turn allows the stiffness/compliance of the front strut/shock absorber to be used to tune vehicle suspensions and achieve flexibility via differentiation of ride attributes.

As the connectivity map shows, many objective metrics are shared or influenced by multiple components. This implies that they are areas of interactions between these components and their design parameters. For instance, roll centre/roll centre migration (K1), which influences all three ride subattributes, is connected to nearly every component. Again, without knowledge of the physical models, this seems like too many interactions to be useful for supporting flexibility. However, the components that influence fewer objective metrics (bump stops for example) might support some flexibility. The connectivity map shows that objective metrics that impact the rough road and abruptness attributes also impact road attributes, and that, in general, fewer objective metrics impact abruptness.

The connectivity map does uncover some relationships that appear more manageable. The control arm ball joints compliance and the control arm bushing compliance are the only design parameters that connect the ride attributes via longitudinal compliance (C7). The stiffness/compliance of the front springs/struts/shocks, control arm ball joints, and control arm geometry are the only design parameters that connect to the ride attributes by influencing camber change. These simpler relations and the more manageable number of connections suggest that these components might be useful for achieving flexibility. They have the benefits of the multi-path and multi-attribute genes without an overwhelming number of interactions.

5. Summary and conclusion

This paper has documented the construction and use of a connectivity map, a simple, compact, matrix-based approach for capturing, identifying, and assessing relationships between different types of elements. It can be used with processes, requirements, physical elements, and product attributes, and can be used to evaluate processes or products. A process for developing the connectivity map was presented, along with simple examples to demonstrate how the matrix is constructed and analysed. More complex, real-world examples, drawn from the automobile industry, were also presented. These examples showed two different applications of the method, and also showed some of the benefits, difficulties and pitfalls of the method.

The examples have suggested some of the qualities that allow differentiation between the elements that sit on the rows, columns and in the cells, and whether there are special qualities that the element that will fill the cell (let us call it the connector element) should have. From the generic discussion of section 3, it can be stated that there are three elements in a connectivity map. These different element types can be labeled as: row elements, column elements, and connectivity elements.

The row elements should be considered the starting point. They should be some known, subjective state or desired condition. Since this paper focuses on the development of products, the row elements can be defined further as a known, desired state that supports the development of a desirable product. The column elements are physical objects that develop or emerge from objective needs that necessarily flow out of the row elements.

	What	What	How
General	Row element: what is the desired state to produce a desirable product?	Connectivity element: what the row element demands from the column element	Column element: how the connectivity element will be satisfied or supported
Example 1, restraint system	Process tasks: what steps are needed to develop the product?	Functional requirements: what the process elements demand from the physical elements	Physical element: how the functional requirements will be satisfied
Example 2, suspension system	Consumer attributes: what attributes are needed to satisfy the consumer?	Objective metrics: what the attributes demand from the physical elements	Design parameters: how the functional requirements will be satisfied

Table 3. Comparing the two connectivity maps.

The connectivity elements capture these objective needs and link or align the column elements with the row element. The connectivity element should be objective, subject to quantification, and should support the development or emergence of the column element from the row element. The connectivity elements should say what needs the column elements should satisfy in order to create or support the desired state represented by the row elements. From the perspective of the column elements, the column elements represent how the connectivity elements will be satisfied or supported. Following this line of thought, in general, the relationship structure of the elements in the connectivity map can be thought of as:

What \iff What \iff How

That is, the row elements describe 'What' is the desired state. The connectivity elements describe objectively 'What' the row elements demand from the column elements. The column elements describe 'How' the objective demands of the connectivity elements will be met.

Table 3 summarizes this distinction between row elements, connectivity elements, and column elements by aligning and mapping them to the What/What/How concept. To further illustrate the concept, information from the two examples has been incorporated. Note that, although the examples were different in nature, there are similarities between the row elements, connectivity elements and column elements that can be exploited to provide guidance on identifying good connectivity elements.

One obvious quality is that the connectivity element must link or align two other elements. A quality that is less obvious is that the connectivity element should be an objective measurable, such as a specification. This in turn implies that the column element, the 'How' piece, should be a physical element that would be developed to meet the specification.

Other qualities are less obvious from this discussion but emerge from the examples. Neither of the individual relationship diagrams (see table 3) between the connectivity element and the other two elements should be completely filled. A completely filled relationship diagram would mean that every subelement of the connectivity element was linked to or influenced every subelement in the row element or column element. As the second example shows, the connectivity map becomes unmanageable as the number of linkages increases. However, the individual relationship diagrams must show that all subelements have at least one linkage. Extremely complex situations can overwhelm the ability of the method to support insightful analysis and generation of useful results. This occurs when all the subelements of the connecting element type appear in all the cells (or a preponderance of cells) of the connectivity map. One early indication that this might become an issue is when there is significantly fewer of one type of element. This situation might be circumvented by partitioning down to finer levels of detail on the components (perhaps drilling down to the level of specifying design parameters such as dimensions or material) and the subattributes.

Notes

1. The DSM is a special kind of a relationship map, where only a single set of elements is used to define the matrix (i.e. the matrix becomes a square one) (Warfield 1973, Steward 1981). The off-diagonal entries in the matrix represent the relationship structure between the elements, and the diagonal elements normally carry no significance.

2. Mathematically, the connectivity map can be constructed conceptually by matrix multiplication methods. For instance, with matrix A having N rows and L columns, and matrix B having L rows and M columns, the connectivity maps cells on row '1' are filled by the equation:

$$CM(i, j) = \sum_{j=1}^M \{A(i, j) \cdot B(j, i)\} \quad i = 1, \dots, N$$

3. In a completely de-coupled process with an axiomatic design solution (Suh 1990), there will only be one item in each row and in each column. Ideally, if the process can be conducted in manner so that the tasks can be performed independently either in parallel or sequentially, all of the relationships will be contained along the diagonal of the map.
4. The suspension components, identified in figure 7, have two primary types of design parameters: (1) stiffness/compliance are design parameters for springs, shocks, rubber parts (bushings, tires, mounts), linkages (ball joints) and torsional elements; and (2) geometry/dimension are the physical length, shapes, and diameters of the components.

References

- DALEIDEN, S., 2000, Chassis flexibility: assessing architectural differences and their effect on managing the development of product derivatives. SM Thesis (Cambridge, MA: MIT).
- EPPINGER, S., 1991, Model-based approaches to managing concurrent engineering, *Journal of Engineering Design*, 2, 283–290.
- EPPINGER, S., WHITNEY, D., SMITH, R. and GEBALA, D., 1994, A model-based method for organizing tasks in product development, *Research in Engineering Design*, 6, 1–13.
- GILLESPIE, T., 1992, *Fundamentals of Vehicle Dynamics* (Warrendale, PA: SAE Inc.).
- HAUSER, J. and CLAUSING, D., 1988, The house of quality, *Harvard Business Review*, 66, 63–73.
- KEHAT, E. and SHACHAM, M., 1973, Chemical process simulation programs—2: partitioning and tearing of system flowsheets, *Process Technology International*, 18, 115–118.
- KUSIAK, A. and PARK, K., 1990, Concurrent engineering: decomposition and scheduling of design activities, *International Journal of Production Research*, 28, 1883–1900.
- LAVINE, J., 2000, Parametric design constraints management using the design structure matrix: creation of an electronic catalog for safety belt system. SM Thesis (Cambridge, MA: MIT).
- LOCKLEDGE, J. and SALUSTRI, F., 1999, Defining the engine design process, *Journal of Engineering Design*, 10, 109–124.
- STEWART, D., 1962, On an approach to the analysis of the structure of large systems of equations, *SIAM Review*, 5, 321–342.
- STEWART, D., 1981, The design structure system: a method for modeling the design of complex systems, *IEEE Transactions on Engineering Management*, 3, 71–74.
- SUH, N., 1990, *The Principles of Design* (New York: Oxford University Press).
- ULRICH, K. and EPPINGER, S., 2000, *Product Design and Development* (New York: McGraw-Hill).
- YASSINE, A. and FALKENBURG, D., 1999, Design process specifications management, *Journal of Engineering Design*, 10, 223–234.
- YASSINE, A., FALKENBURG, D. and CHELST, K., 1999, Engineering design management: an information structure approach, *International Journal of Production Research*, 37, 2957–2975.
- WARFIELD, J., 1973, Binary matrices in system modeling, *IEEE Transactions on Systems, Man and Cybernetics*, 5, 441–449.