HIGHER ORDER INTERACTIONS: PRODUCT AND CONFIGURATION STUDY ON DSM SATURATION

Phelan, Keith; Summers, Joshua David; Pearce, Brian; Kurz, Mary E.
Clemson University, United States of America

Abstract

Research has shown that higher order interactions are important in evaluating change propagation within a system. This paper presents a systematic approach to evaluate how component interactions beyond the first order affect the interconnectivity of the product/system. A correlation between the higher order component interactions and the betweenness of the components is identified. This information is then used to identify relationships between different products and make inferences about additional systems that exhibit similar patterns. The methods are also applied to subsets of product configuration rules to understand whether or not the same relationship trends occur within product configuration as in product architecture.

Keywords: Product architecture, design structure matrices, Complexity

Contact:
Prof. Joshua David Summers
Clemson University
United States of America
jsummer@clemson.edu

Please cite this paper as:
Surnames, Initials: *Title of paper*. In: Proceedings of the 20th International Conference on Engineering Design (ICED15), Vol. nn: Title of Volume, Milan, Italy, 27.-30.07.2015

1 INTRODUCTION

An important strategy in systems engineering is system decomposition, which can be supported through component-based design structure matrices (DSM) (Browning 2001). This decomposition strategy is integrated into a change management tool for verification and validation planning for line sustaining engineering (Shankar et al. 2014). The planning method is shown in Figure 1. This research is primarily concerned with the highlighted step (Conduct System Analysis). In the original method, only direct interactions between the change component and other components were considered for change propagation pathways during the system analysis (Shankar et al. 2014). While some of the pathways are non-physical, what was not considered is the impact of higher order interactions (if A→B→C, then A→C is a second order interaction) through the propagation pathways. As with any analysis, there is a trade-off between being thorough and being efficient. For instance, if a VVT Plan was developed for a brake system, it is not clear whether the entire vehicle should be also considered at the component level. This complete interaction graph would be too complex for engineers to manage and therefore potential interactions might be missed. The purpose of this research is to explore at what order of interaction the analysis should be conducted in order to maximize the benefits of the trade-off.

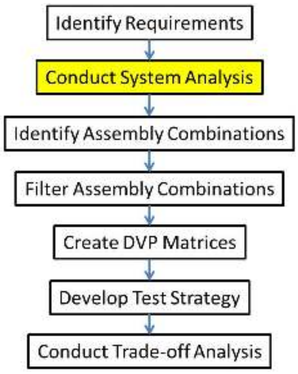

Figure 1. Verification, validation and testing planning method (Shankar et al. 2014)

2 BACKGROUND

2.1 Design Structure Matrices

Design structure matrices (DSM) are commonly used to better understand and analyse product architectures. In one of the earlier examples of using DSMs for system modelling and decomposition, the author provides a classification of existing applications: product architecture, organizational structure, information flow, and design parameter relationships (Browning 2001). Others extend the research by providing a review of the benefits of applying DSMs in understanding product architecture, while acknowledging that a major limitation of DSMs is that they are only applicable in a single domain (Danilovic & Browning 2007). The domain mapping matrix (DMM) is proposed to map the interactions between DSMs from different domains. Similarly, DSMs are used to characterize complex systems by breaking them down into clusters or "building blocks" (Sharman & Yassine 2004). DSMs have also been used to explore how software architectures can be managed (Sosa et al. 2007), introducing architectural metrics, derived from the software architecture DSMs, that can be used in development.

As DSMs become more prominent in engineering design, the number of proposed applications in which DSMs can be implemented has increased dramatically (Xiao & Chen 2010). An early example of using the interdependencies between components to predict some aspect of product development resulted in a proposed method to predict the time required for product development (Carrascosa et al. 1998). This time prediction model uses the amount of dependencies between product development tasks to determine how required changes to specific tasks will affect the other tasks. Other applications that have been discussed include product configuration (Helo 2006), modelling engineering design activities (Kumar & Mocko 2007), and product modularity (Sullivan et al. 2001), to name a few.

2.2 Change Propagation

As DSMs provide an effective method for both viewing and analysing the interactions between components, they are commonly used to better understand change propagation within a product or system. Based on a study on engineering changes and how the interactions between product components could be used to better understand change propagation through the product, a DSM-based engineering change management tool was created to assist in product development (He et al. 2008). DSMs have been used to understand change propagation in a complex system by decomposing the system into a interacting subsystems (Giffin et al. 2009). The change requests over the system's life-cycle was related to one another by considering the decomposed subsystems from which they originated.

Research has also used DSMs not only to understand change propagation, but also to predict it. Based on the use of DSMs in understanding change propagation, the Change Prediction Method (CPM) software tool was developed to assist in visualizing the possible change propagation pathways from a single component prior to the execution of an engineering change (Clarkson et al. 2004). Since its development, the CPM tool has been implemented in case studies, demonstrating the effectiveness of using DSMs to predict change propagation in a system (Jarratt et al. 2004; Keller et al. 2005). While the CPM tool focuses on change propagation in the product development process, the DSMs primarily consist of product components with the change indices being subjectively assigned.

Additional research on predicting requirements change has been conducted using DSMs linking product requirements (Morkos et al. 2012). A historical based approach was used with many different relationship sets to determine which combinations yielded the best predictors. A signification conclusion from this research is the fact that using higher order DSMs, specifically at the second order of interaction, is necessary to best predict future changes to requirements. Thus, the question at hand is whether second or higher order interactions are critical in other DSM applications for understanding and, eventually, predicting change. To better answer this question, one may consider how product complexity could play a role in increasing change propagation within a system or product.

2.3 Product Complexity

When considering product complexity, 29 different metrics are commonly used for evaluating a product or system to predict assembly time from component assemblies (Namouz & Summers 2014). However, when focusing on how complexity can influence change propagation within a product, the researchers focused on the complexity metrics that primarily looked at the interactions between the components. This mirrors the concept of complexity as coupling that is proposed in (Summers & Shah 2010), with an initial algorithm based on decomposability proposed to determine connectivity complexity. Using this algorithm, a simple experiment was conducted using the proposed metric to evaluate multiple existing products represented in different model types (Ameri et al. 2008). The experiment showed the importance of coupling when considering complexity in a product's architecture. Similarly, the connective complexity in a system was used to model and understand design tasks (Mathieson et al. 2011). Using connectivity metrics, the researchers were able to identify patterns and infer additional relationships within the system.

3 APPROACH

The approach presented in this paper consists of two phases. The first phase is the development of the DSMs from models of existing products. The second phase consists of the analysis of the products based on how the individual components or elements interact beyond the first order of interaction. The analysis is done based on both assembly models and also on product configuration (option) graphs.

3.1 Product Architecture

In the development of the product DSMs for this study, previous models are used as reported in the literature to address issues of research objectivity and bias. The DSMs allow the authors to draw comparisons between the physical attributes of the product to the data on how the product's components interacted. This was especially beneficial in looking for patterns between different products that exhibited similar interaction behaviours. Previous work had been done to create connectivity graphs that would provide a visualization of all of the interactions between components within a product based on the analysis of a 3-D CAD model (Namouz & Summers 2014). Figure 2 shows an example of a 3-D CAD model and the corresponding connectivity graph. In this study, the types of relationships between components (elements) is not studied, rather only the topology of the system architecture is investigated.

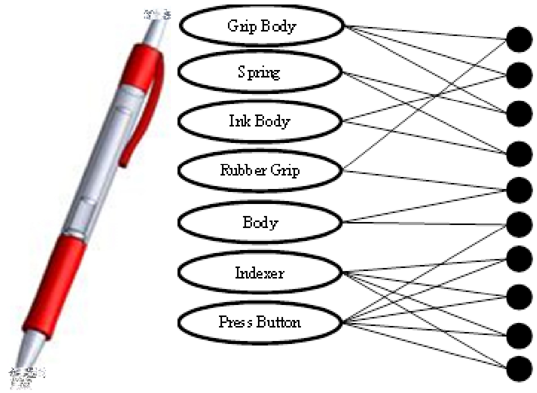

Figure 2. 3-D CAD model for a pen (a) and the resulting connectivity graph (b)

Using the connectivity graphs, it was possible to construct a DSM for each of the products being analysed. It is important to note that the DSMs created from the connectivity graphs only include physical interactions identified in the associated 3-D CAD models. An example of the resulting DSM is shown in Figure 3(a). The second phase began with the identification of component interactions beyond the first order. The DSMs were then populated with the higher order interactions. This is done by finding components with common interactions and then adding the lowest value for common interactions. For example, Grip Body and Body both interact with Rubber Grip at the 1st order, so they have a 2nd order interaction. An example of a completed DSM with higher order interactions is shown in Figure 3(b). The cells in the DSM indicate the shortest path length between the components, or the minimum order of interaction between components.

Component Name		A	B	C	D	E	F	G
Grip Body	A		1		1	1		
Rubber Grip	B	1					1	
Press Button	C						1	1
Spring	D	1				1		
Ink Body	E	1			1			
Body	F		1	1				
Indexer	G			1				

Component Name		A	B	C	D	E	F	G
Grip Body	A		1	3	1	1	2	4
Rubber Grip	B	1		2	2	2	1	3
Press Button	C	3	2		4	4	1	1
Spring	D	1	2	4		1	3	5
Ink Body	E	1	2	4	1		3	5
Body	F	2	1	1	3	3		2
Indexer	G	4	3	1	5	5	2	

Figure 3. Initial (a) and fully populated (b) product design structure matrices for a pen

The final step is to calculate the population density of DSM at each order of interaction. The population density refers to the percentage of existing interactions compared to the total number of possible interactions. The population density includes any interactions that take place up to and including a given order. In the example of the pen (Figure 3(b)), the population density for the 2nd order would be 57.14% (12 interactions out of 21 possible interactions).
Another statistic used to analyse the data is the average shortest path length applied against the entire graph. The shortest path length for a given component is the minimum number of iterations required to reach every other component in the system. In the example shown in Figure 3(b), the shortest path length for "press button" is 4. By averaging the path lengths in the example, the average shortest path length is 4.143.

3.2 Product Configuration

In addition to considering propagation in products, the researchers also evaluated product configuration data. The purpose of this aspect of the research is to determine how DSMs for product configuration are similar to DSMs for product architecture. This is of interest in that previous research has shown the importance of understanding change propagation in configuration management, especially when using a rule-based configuration management system (Phelan et al. 2014). An example DSM for product configuration is shown in Figure 4(a).

Option Name		A	B	C	D	E	F	G	H	I	J	K	L	M	N
161	A														
167	B													1	
169	C								1	1					
1CB	D								1						
2VB	E													1	
6UF	F													1	
823	G									1					
842	H			1	1										
843	I			1				1			1				
858	J									1					
8EB	K													1	
8S9	L													1	
991	M		1			1	1					1	1		1
9AE	N													1	

Option Name		A	B	C	D	E	F	G	H	I	J	K	L	M	N
161	A														
167	B					2	2					2	2	1	2
169	C				2			2	1	1	2				
1CB	D			2				4	1	3	4				
2VB	E		2				2					2	2	1	2
6UF	F		2			2						2	2	1	2
823	G			2	4				3	1	2				
842	H			1	1			3		2	3				
843	I			1	3			1	2		1				
858	J			2	4			2	3	1					
8EB	K		2			2	2						2	1	2
8S9	L		2			2	2					2		1	2
991	M		1			1	1					1	1		1
9AE	N		2			2	2					2	2	1	

Figure 4. Initial (a) and fully populated (b) product configuration DSMs for a product change

In Figure 4(a), the components of the DSM are product options that might be selected independently or within packages by a customer. They relate to other options by a series of rules that might be either engineering based or marketing focused. In the DSM, the connections, through the rules, are shown by 1st order interactions in the DSM. For example, in Figure 4(a), Option 167 is directly related to Option 991 through a configuration rule. Using the same method discussed in the previous section, a higher order DSM is created. Figure 4(b) shows the resulting higher order DSM based on the DSM in Figure 4(a). When analysing product configurations, two levels of DSMs are used: a full configuration ruleset, consisting of approximately 600 components connected by 1400 rules; and a series of configuration DSMs created by considering the affected components from historical changes implemented at an automotive OEM. The DSMs are created by starting with the options affected by the change. The rest of the DSM is populated using the 1st and 2nd order interactions stemming from the initial change components. Once the higher order DSM is created, the same analysis is conducted as with the product architecture DSMs. The purpose behind this is to use the conclusions drawn from analysing product architecture and apply it to product configuration.

4 RESULTS

4.1 Component Saturation

When applied to thirteen products (Namouz and Summers, 2014), the given approach yielded the data shown in Table 1. In the table, the number of components, the betweenness density, and the population density for each product based on the order of interaction are illustrated.

Table 1. Component interaction saturation for product architecture

Product Architecture										
Product (Row Density)	# of Comp	Saturation	Betweenness Density	Population Density by Order of Interaction						
				1	2	3	4	5	6	7
Boothroyd Piston (67%)	7	2	0.314	47.6	100.0	100.0	100.0	100.0	100.0	100.0
Mouse (82%)	12	3	0.186	25.8	86.4	100.0	100.0	100.0	100.0	100.0
Stapler (87%)	16	4	0.245	27.5	83.3	99.2	100.0	100.0	100.0	100.0
Pencil Compass (45%)	12	4	0.431	27.3	66.7	92.4	100.0	100.0	100.0	100.0
Solar Yard Light (64%)	15	5	0.454	19.1	57.1	88.6	99.1	100.0	100.0	100.0
Drill (70%)	28	5	0.259	14.0	61.9	92.1	99.2	100.0	100.0	100.0
Hole Punch (39%)	27	4	0.658	9.1	34.8	74.4	100.0	100.0	100.0	100.0
Vise (44%)	19	5	0.546	15.2	42.1	80.1	97.7	100.0	100.0	100.0
Chopper (38%)	41	6	0.577	8.4	35.5	73.9	93.9	99.3	100.0	100.0
Blender (33%)	43	6	0.620	7.8	32.6	70.2	92.3	99.0	100.0	100.0
Pen (50%)	7	5	0.857	33.3	57.1	76.2	90.5	100.0	100.0	100.0
Maglight (38%)	14	7	1.471	17.6	39.6	58.2	74.7	87.9	95.6	100.0
Brake subsystem (40%)	11	7	Unknown	20.0	43.6	63.6	81.8	92.7	98.2	100.0

Betweenness density is a measure of the average betweenness value for all of the components with a given product. Betweenness is a useful metric for evaluating complexity in that it provides a measure of the importance of a specified component (Freeman 1977). Also shown in Table 1 are the orders of interaction for complete saturation of the DSM and the row density for each product. The row density shows the highest percent of interactions present for a single component of the product. For example, in the pen, there are seven components, resulting in six possible interactions between a single component and the other components. The most interconnected component, the grip body, interacts with three components out of the possible six. Therefore, the pen has a 50% row density. The row density was considered as it may help to explain why some of the saturation rates act as they do.

Using visual inspection, it is possible to separate the resulting graphs into groupings of products that exhibit similar curves. Figure 4 shows the first grouping, consisting of the stapler and the computer mouse. These curves exhibit quick saturations (3rd order) with a steep initial slope. The second grouping is depicted in Figure 5 and consists of the pencil compass, solar yard light, and the electric drill. The curves of this group exhibit slightly slower saturations (4th order) and resemble a parabola. Figure 6 contains the third grouping, consisting of the 3-hole punch, an electric food chopper/processor, a pony vise, and an electric blender. These curves exhibit medium saturation order (4th/5th orders) and have a slight "s-curve" as the order of interaction is increased. The final grouping is shown in Figure 7 and consists of the pen, a Maglite flashlight, and the brake subsystem modelled in the motivating project. The curves of this group exhibit slow saturations (5th-7th orders) and are parabolic in shape.

Figure 4. Product group 1 saturation graph

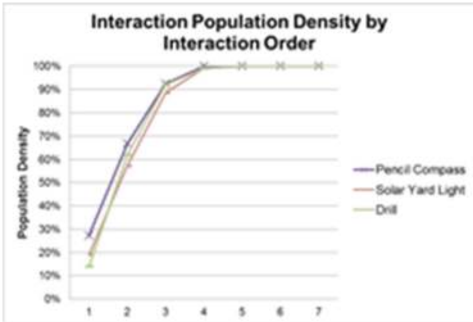

Figure 5. Product group 2 saturation graph

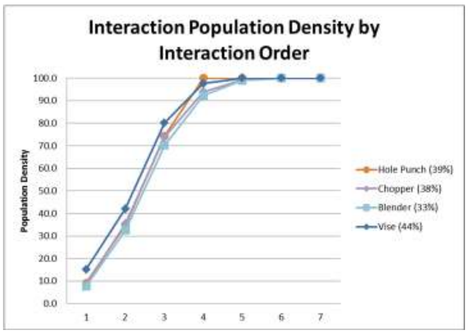

Figure 6. Product group 3 saturation graph

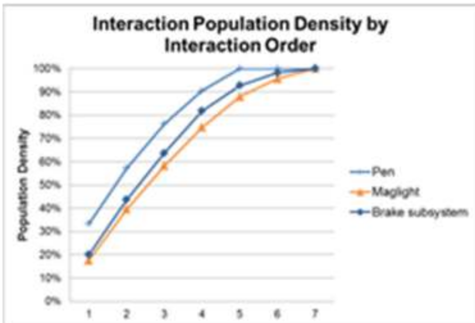

Figure 7. Product group 4 saturation graph

Similar data was created for four product configuration change DSMs from the automotive OEM. The results are shown in Table 2. The only difference is that the betweenness values were not available for the product configuration subsets.

Table 2. Component interaction saturation for product configuration

Product Configuration											
Product (Row Density)	# of Comp	Saturation	Population Density by Order of Interaction								
			1	2	3	4	5	6	7	8	9
Change 1 (35%)	38	7	8.3	32.6	50.9	61.5	73.7	92.9	100.0	100.0	100.0
Change 2 (80%)	26	N/A	21.5	71.1	78.8	78.8	78.8	78.8	78.8	78.8	78.8
Change 3 (46%)	14	N/A	12.1	34.1	37.4	39.6	39.6	39.6	39.6	39.6	39.6
Change 4 (50%)	17	5	19.9	52.9	86.8	98.5	100.0	100.0	100.0	100	100
Ruleset (6%)	395	N/A	0.63	2.24	4.63	7.26	10.35	13.29	15.12	15.92	16.14

When the product configuration subsets are considered alongside the products, three of the changes (1, 2, 4) fit within three of the product groups. The resulting graphs are shown in Figures 8-10. It should be noted that Change 3 and the ruleset were not able to be matched to a specific group due to the low maximum population density. When considering product configurations, it is common that there will be clusters of product options that do not interact in any way with other clusters. It the case of the complete ruleset, this was much more severe, with only 16% of the possible interactions existing at maximum saturation.

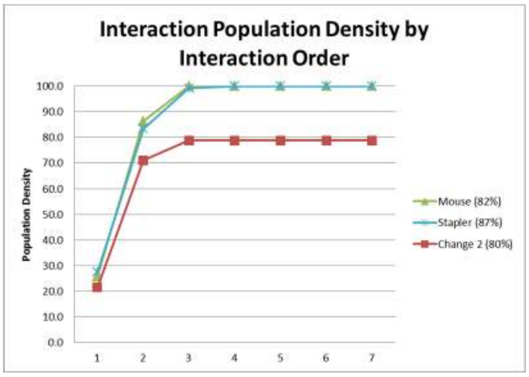

Figure 8. Group 1 saturation graph

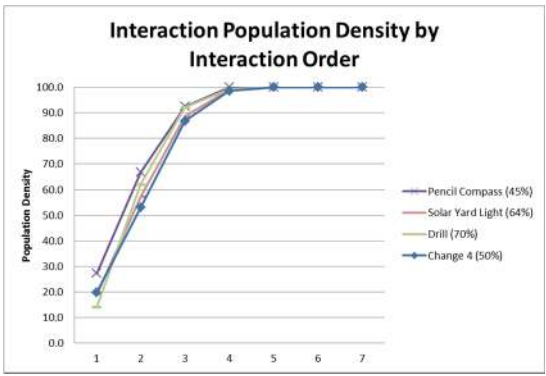

Figure 9. Group 2 saturation graph

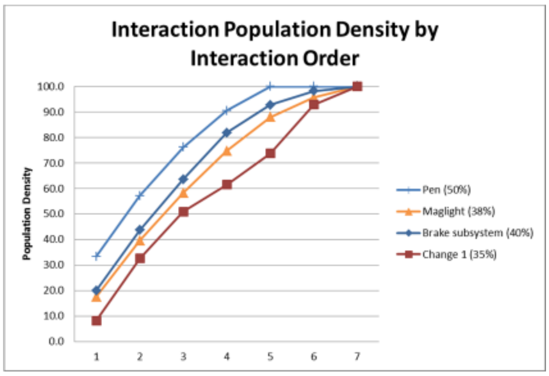

Figure 10. Group 4 saturation graph

4.2 Higher Order Component Interaction

The average shortest path length is also considered as a method for better understanding component interaction within a system. Average shortest path length is useful in that is directly shows the iterations needed to propagate through an entire system. By applying the outlined approach to the product DSMs, the following data was created and is shown in Table 3.

Table 3. Product architecture component interaction

Product Architecture				
Product (Row Density)	Saturation	Avg Path	Comp	Row Dens.
Boothroyd Piston (67%)	2	2.00	7	67%
Mouse (82%)	3	2.58	12	82%
Pencil Compass (45%)	4	3.50	12	45%
Stapler (87%)	4	2.94	16	87%
Hole Punch (39%)	4	3.64	27	39%
Pen (50%)	5	4.14	7	50%
Solar Yard Light (64%)	5	4.00	15	64%
Drill (70%)	5	3.89	28	70%
Vise (44%)	5	4.05	19	44%
Chopper (38%)	6	4.71	41	38%
Blender (33%)	6	4.98	43	33%
Maglight (38%)	7	5.86	14	38%
Brake subsystem (40%)	7	5.45	11	40%

In addition, the same approach was used to analyse the product configuration DSMs. The resulting data is shown in Table 4.

Table 4. Product configuration component interaction

Product Configuration				
Product (Row Density)	Saturation	Avg Path	Comp	Row Dens.
Change 1 (35%)	7	6.16	38	35%
Change 2 (80%)	N/A	2.42	26	80%
Change 3 (46%)	N/A	2.36	14	46%
Change 4 (50%)	5	4.00	17	50%
VRM (6%)	N/A	3.26	395	6%

Visual inspection of both the product architecture and product configuration datasets show that the average shortest path length and the order for complete saturation are closely related. A larger sample population would be required for a more robust correlation analysis. A graph depicting the relationship for both the product architecture and product configuration DSMs is shown in Figure 11.

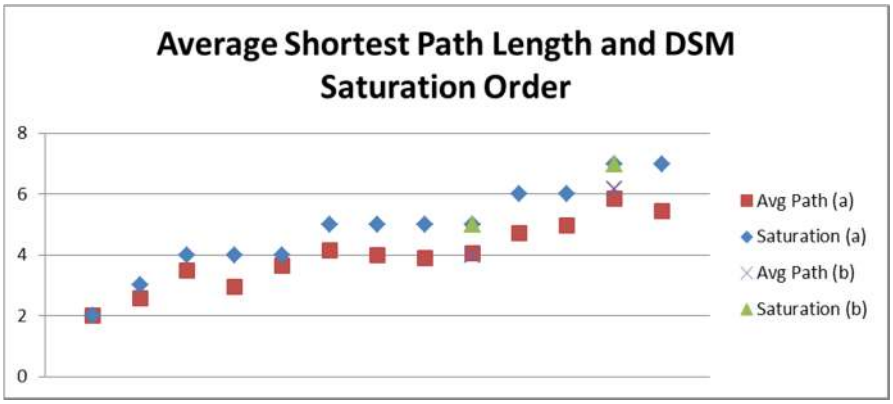

Figure 11. Average path length vs. saturation order for product configuration and architecture

In Figure 11, the product DSMs are shown in group (a), while the configuration subsets are shown in group (b). Only two pairs of data are used for group (b) as the other two configuration subsets did not reach saturation. From the graph, it is clear that the relationships between average shortest path length and complete saturation of the DSM are similar for both product architecture and product configuration.

5 ANALYSIS

When the betweenness values are compared to the saturation curves for the four product groupings, a correlation is suggested; as the betweenness density decreases, the slope of the saturation curve increases, or the individual components more rapidly interact with all of the other components through higher order interactions. It can also be noted that the betweenness density values for those products within a group are similar when compared to the values for the other products (for example, all of the products in group 3 have values between 0.54 and 0.66). This shows that as the slope of the saturation curve increases, the interconnectivity (betweenness) of the product increases.

Because of the familiarity with the products involved, it was possible to identify additional relationships regarding the product groupings with respect to the product architecture. It was noted that the products in group 4 exhibited a stacked-linear assembly structure, which likely corresponds to a decreased saturation rate as components only directly interact with neighbouring components along the body of the product. On the opposite end of the spectrum, those products in group 1 all exhibit chassis product architectures similar to the spokes of a wheel, where a single body or frame component interacts with a large percentage of the other components in the product. This would correspond with a rapid saturation rate as any given component would quickly interact with the other components through higher order interactions as soon as it interacts with the central frame/body

component. Additionally, the high amount of interactions for a single component is also shown in the row density statistic provided.
Average shortest path length is another metric often used when considering the complexity of a system as it correlates to the level of interconnectedness of the components in the system. In the analysis of component interaction a clear relationship is identified between the average shortest path length and the order of interaction at which the DSM is fully saturated. This shows that as the level of interconnectivity of the system increases, the DSM saturation rate also increases.
When comparing the statistics for product architecture and product configuration, it is clear that the metrics for analysing product complexity can also be applied to evaluating the complexity of a product configuration subset or even the entire product configuration ruleset. This analysis is based on the similarities that were identified between the different types of DSMs and the fact that the saturation graphs for the product configuration subsets were able to be matched closely with the graphs depicting product architecture.

6 CONCLUSIONS AND FUTURE WORK

The presented approach provides a foundation for analysing how higher order interactions relate to product complexity and interconnectivity. It is possible to categorize products according to their saturation curves and thereby make inferences about the products' complexity and architectures. The brake subsystem, used in the motivating research as an example for evaluating change propagation pathways, best fits in the 4th grouping (with the pen and the Maglite), likely resulting in a similar level of betweenness density and interconnectedness. Further study is necessary to validate this.
Perhaps a more interesting observation is that a shape function, rather than a single point value, for defining a system's complexity could be used. This has not been seen in previous research. Using a shape function allows for a much larger amount of information to be used when classifying a system and analysing its complexity. One method for implementing this would be to use the set of seven population densities (one at each order of interaction) to define the shape function. Further research is necessary to validate the usefulness of a shape function for describing complexity.
A primary goal of the research is to identify which order of interaction is necessary in considering possible change propagation pathways. While it is likely impossible to provide a specific order for every situation, the results show that the products studied here generally reach saturation at the fourth or fifth order of interaction. Additionally, when considering the average shortest path length, most of the products had a value between four and five. Therefore, to avoid having to consider every component when looking at possible impact of change propagation, it is recommended to consider interactions occurring up to the second or third order of interaction. However, the optimal order is highly dependent on the product architecture as was evident in the distinction between the four product groupings.
When considering the complete product configuration ruleset, a high level of clustering is evident in the low value of the maximum saturation level. Additionally, 10 orders of interaction were required for the DSM to reach near maximum interaction. This would seem to indicate that using a significantly higher order of interaction for analysis may be appropriate when choosing between cost and being thorough. However, the total number of interactions (> 150,000) means that even a low population density of 10% would still include approximately 15,000 interactions. As a result, it is recommended that, prior to conducting a change propagation analysis, a large system be divided into clusters (as was done with the changes that were analysed).
A limitation of this research is that the product architecture DSMs only reflect the physical interactions between components in a given product, as identified from the assembly models. As a result, the analysis does not include other interactions that may occur, such as functional relationships or interactions due to logistical issues (such as components that are transported together). Another limitation of the analysis is that the saturation curves are evaluated/grouped solely through visual inspection. In order to increase the reliability of the research, the products should be analysed using statistical methods, requiring more objects of study. Future research should also be conducted to further evaluate how assembly strategy is related to a product's saturation curve. It may be possible to decompose the product DSMs into component modules to categorize the product according to architecture. Additional relationships could then be identified between the saturation curves and the product architectures.

REFERENCES

Ameri, F. et al., 2008. Engineering design complexity: an investigation of methods and measures. Research in Engineering Design, 19(2-3), pp.161–179.

Browning, T.R., 2001. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. IEEE Transactions on Engineering Management, 48(3), pp.292–306.

Carrascosa, M., Eppinger, S.D. & Whitney, D.E., 1998. Using the design structure matrix to estimate product development time. In 1998 ASME Design Engineering Technical Conference. Atlanta, GA, pp. 1–10.

Clarkson, P.J., Simons, C. & Eckert, C., 2004. Predicting Change Propagation in Complex Design. Journal of Mechanical Design, 126(5), p.788.

Danilovic, M. & Browning, T.R., 2007. Managing complex product development projects with design structure matrices and domain mapping matrices. International Journal of Project Management, 25(3), pp.300–314.

Freeman, L., 1977. A set of measures of centrality based on betweenness. Sociometry, 40(1), pp.35–41.

Giffin, M.L. et al., 2009. Change Propagation Analysis in Complex Technical Systems. Journal of Mechanical Design, 131(8), p.081001.

He, R., Tang, D. & Xue, J., 2008. Engineering change propagation based on design structure matrix. Computer Integrated Manufacturing Systems, 4(4), pp.656–660.

Helo, P.T., 2006. Product configuration analysis with design structure matrix. Industrial Management & Data Systems, 106(7), pp.997–1011.

Jarratt, T.A., Eckert, C.M. & Clarkson, P.J., 2004. The benefits of predicting change in complex products: application areas of a DSM-based prediction tool. In International Design Conference 2004. Dubrovnik, Croatia, pp. 1–7.

Keller, R. et al., 2005. Visualising change propagation. In International Conference on Engineering Design 2005. Melbourne, Austrailia, pp. 1–12.

Kumar, P. & Mocko, G., 2007. Modeling and Analysis of an Ontology of Engineering Design Activities Using the Design Structure Matrix. In 2007 ASME International Design Engineering Technical Conference. Las Vegas, NV, pp. 1–13.

Mathieson, J.L., Miller, M. & Summers, J.D., 2011. A Protocol for Connective Complexity Tracking in the Engineering Design Process. In International Conference on Engineering Design 2011. Copenhagen, Denmark.

Morkos, B., Shankar, P. & Summers, J.D., 2012. Predicting requirement change propagation, using higher order design structure matrices: an industry case study. Journal of Engineering Design, (November 2012), pp.37–41.

Namouz, E.Z. & Summers, J.D., 2014. Comparison of Graph Generation Methods for Structural Complexity Based Assembly Time Estimation. Journal of Computing and Information Science in Engineering, 14(2).

Phelan, K.T. et al., 2014. A case study of configuration management methods in a major automotive OEM. In 2014 ASME International Design Engineering Technical Conferences. Buffalo, NY.

Shankar, P., Summers, J.D. & Phelan, K.T., 2014. A verification and validation planning method to address propagation effects in engineering design. In International Symposium on Tools and Methods of Competitive Engineering. Istanbul, Turkey.

Sharman, D.M. & Yassine, A. a., 2004. Characterizing complex product architectures. Systems Engineering, 7(1), pp.35–60.

Sosa, M.E., Browning, T.R. & Mihm, J., 2007. Dynamic, DSM-based analysis of software product architectures. In 9th International Design Structure Matrix Conference. Munich, Germany, pp. 349–361.

Sullivan, K.J. et al., 2001. The structure and value of modularity in software design. ACM SIGSOFT Software Engineering Notes, 26(5), p.99.

Summers, J.D. & Shah, J.J., 2010. Mechanical Engineering Design Complexity Metrics: Size, Coupling, and Solvability. Journal of Mechanical Design, 132(2), p.021004.

Xiao, R. & Chen, T., 2010. Research on design structure matrix and its applications in product development and innovation: an overview. International Journal of Computer Applications in Technology, 37(3/4), p.218.