INSEAD

The Business School
for the World®

Working Paper Series

2017/47/TOM

Bicentric Diagrams: A Novel Approach to Manage Design Changes of Interdependent Components in Complex Systems

Manuel E. Sosa
INSEAD, manuel.sosa@insead.edu
Corresponding author

Hyunwoo Park
Georgia Institute of Technology, hwpark@gatech.edu

Rahul C. Basole
Georgia Institute of Technology, rahul.basole@ti.gatech.edu

Complex systems can be conceived as a network of interconnected components. Such a network perspective has been used to predict how design changes in one or more components propagate to other components in the system. Despite such advances it remains an important challenge for engineering managers to identify the set of components that could potentially be affected if design changes affect two interdependent components. To address such a design challenge this paper introduces a novel approach, based on bicentric diagrams, to manage design changes of pairs of interdependent components of complex systems. Bicentric diagrams, a novel graph-based visualization technique, enable simultaneous identification of the components that would be directly or indirectly affected if design changes on a pair of interdependent components take place. We illustrate our approach by using bicentric diagrams to examine the impact of changing some critical interfaces of a climate control system and a large commercial aircraft engine.

Keywords: Change Propagation; Design Changes; Design Structure Matrix (DSM); Data Visualization

This research was in part supported by the Tennenbaum Institute at Georgia Tech and INSEAD's R&D funds

INTRODUCTION

Designing complex systems such as automobiles, airplanes, or engines poses large coordination challenges among the various cross-functional teams that design the subsystems and components that formed the system (Alexander 1964, Clark and Fujimoto 1991, Simon 1996, Bar-Yam 2001). When designing complex systems they are typically broken down into subsystems and these into components so that the design of such components and subsystems are carried by cross-functional teams specialized in a given “chunk” of the system (Henderson and Clark 1990, Sosa et al 2003). The challenge of the design teams is not only to design their component but also to ensure that their component designs are well integrated with other interdependent components so that the system works as a whole (Sosa et al 2004, Olson et al 2009, Gokpinar et al 2010).

Considering a complex system as such an interconnected web of components, a critical challenge for engineering managers is to predict and manage how design changes in one or few components propagate to other components in the system (Terwiesch and Loch 1999, Eckert et al 2004, Clarkson et al. 2004, Giffin et al 2011). Moreover, it is important to recognize that many design decisions such as interface standardization, component integration, and defect findings often involve design changes at the interface level (Terwiesch et al 2002, Sosa et al 2004). For instance, when Ford and Firestone discovered that that the tire and suspension subsystems needed to be changed to resolve the instability of the Ford Explorer trucks in the late 90s, how design changes in the tire-suspension interface would affect other subsystem in the vehicle needed to be managed (Pinedo et al 2000). Similarly, when Airbus discovered that the wiring subsystems of two different sections of the fuselage of their A380 airplane needed to be reworked, they needed to identify the other subsystems that could potentially be affected if changes in such interconnected areas of the fuselage were to change (Gumble 2006).

Hence, an important questions for engineering managers to address is *how would design changes of an interface between any two given components in a system might potentially affect other components in the system?* We are addressing this engineering challenge with bicentric diagrams.

Previous work has developed models, representations, and tools to model complex systems as network of interconnected components (Sharma and Yassine 2004, Braha and Bar-Yam 2007). They have paved the way to advanced methods not only to assess various product decomposition options but also to evaluate design change propagation (Chen et al 2005, Chen et al. 2007, Eckert et al. 2004, Clarkson et al. 2004, Jarrat et al. 2005). Despite such advances to model complex systems and predict and manage design change propagation, there is still room to improve the capability of engineering managers to manage design changes at the interface of interdependent components for which there seems to be a lack of tools or techniques that could assist engineering decision makers.

This paper introduces a structured approach, based on bicentric diagrams (a novel graph-based interface representation), to identify and prioritize the set of other system components that could potentially be affected if changes in a pair of interdependent components propagate to other components in the system.

RELATED WORK

Our work builds primarily upon three streams of literature. First, we look into the design structure matrix (DSM) literature because it is the most salient stream of work used to model complex systems as network of interconnected components (Pimmler and Eppinger 1994, Sosa et al. 2003, Sosa et al. 2007, MacCormarck et al. 2006). Second, within the complex systems literature we build upon the stream of work that models change propagation (Eckert et al 2004, Clarkson et al. 2004, Giffin et al. 2011). Finally, we build on the visual decision support system (DSS) literature from which bicentric diagrams emerged as a novel graphical representation technique that is uniquely suited to help engineering managers to manage design changes at the interface level.

The DSM literature has centered on modeling processes, products, and organizations as sets of interdependent tasks, components, and teams. See Browning (2001, 2016) for an extensive review. This literature stream started with modeling new product development (NPD) process as a set of interdependent activities (Steward 1981, Kusiak and Wang 1993, Eppinger et al. 1994). Next, DSM researchers also used DSMs to model both hardware and software products as network of interconnected components (Pimmler and Eppinger 1994, Sosa et al. 2003, MacCormack et al. 2006, Sosa et 2007, Sosa et al. 2011). Finally, this stream of work has also looked into how process and product related decisions impact the organizational side of NPD. For this, NPD organizations have been modeled as interdependent teams that need to coordinate their effort to ensure that the process is well executed and products are well put together and work as an integrated whole (McCord and Eppinger 1993, Morelli et al. 1995, Sosa et al. 2004, Olson et al. 2009, Sosa 2008, Sosa et al. 2015). We complement this stream of work by introducing a new visual representation that allows managers to zoom in on a given interface to visualize the other DSM elements that are directly or indirectly affected by decisions concerning the focal interface.

The challenge of predicting design changes when designing complex systems has attracted great attention of academics in engineering design (Wright, 1997, Eckert et al. 2004, Clarkson et al. 2004, Jarrat et al. 2005, Keller et al. 2005, Giffin et al. 2011). This stream of work has focused on developing predictive models and support tools to anticipate design changes and to manage the impact of such changes during the development effort (Eckert et al. 2005). At the core of this stream of research is the premise that changes in one component may propagate to other components in the product due to the interdependent structure of complex systems. We contribute to this stream of work by focusing on design changes or decisions that involve not one but two interdependent components and how such changes/decisions couple potentially affect other components in the system.

Data visualization is a frequently used method in a range of data-driven decision making contexts (Tegarden 1999, Zhu and Chen 2008). When developed appropriately, visualization can significantly enhance human cognition, enabling users to understand complex structural connections, discover patterns, clusters, and outliers, and communicate results effectively (Card, Mackinlay, and Shneiderman 1999). Interactive visual decision support systems (DSS) have been used in many domains, including finance, healthcare, energy, and engineering (Basole 2014, Basole et al. 2015). Many existing visual DSS typically employ traditional representation techniques, including bar-, line-, and pie-charts, matrix representations, treemaps, or network diagrams (Heer and Shneiderman 2010), mainly due to familiarity and convenience. However, prior visualization research has advocated that the type of visual representation should "fit" the underlying data, tasks, and users (Speier 2006, Basole et al. 2016). Consequently, visual DSS research primarily focuses on developing novel graphical representations and interaction techniques that match the corresponding use context. Prior work has thus examined the effectiveness, usability, and value of various representations and interactions for decision support (North 2006, Stasko 2008, Fekete et al. 2008). We build upon this line of research to introduce a novel visualization technique, bicentric diagrams, to address the challenge of managing design changes at the interface of system components.

THE INTERFACE MANAGEMENT CHALLENGE

Given a system of *n* components and *k* interfaces, the interface management challenge can be framed as the challenge of identifying the other components that could potentially be affected if interconnected components *i* and *j* are involved in a simultaneous design change; we call this a *dyadic design change* of interface (*i*,*j*). There are many reasons for dyadic design changes to occur. For instance, the standardization of an interface, the removal of an interface, or the simultaneous design change of the components which would affect the interface itself. Regardless of the specific reasons that trigger the dyadic design changes, a critical question for managers is to inform to other stakeholders in the development organization of the possible impact of such changes: How could engineering managers identify the other components in the system that might potentially get affected by such an interface design change? That is, how to identify the other product components that are directly and indirectly connected to components *i* and *j*. Note that a direct connection is that which connects two components directly (without any intermediary components). An indirect connection is such that connects two components through an intermediary component. In network terms, we refer to indirectly connected components to those that have a two-degree separation from the focal component.

To illustrate this challenge, consider a climate control system that has been studied in previous studies (Pimmler and Eppinger 1994, Sosa et al. 2011). An schematic representation of such a system is depicted in Figure 1 and its product DSM is shown in Figure 2. The system is formed by 16 components structured in four sub-systems and 34 symmetric interfaces as indicated by the non-zero cells in the DSM shown in Figure 2.

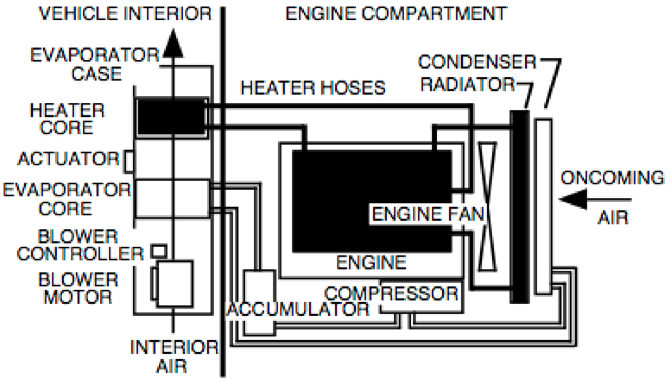
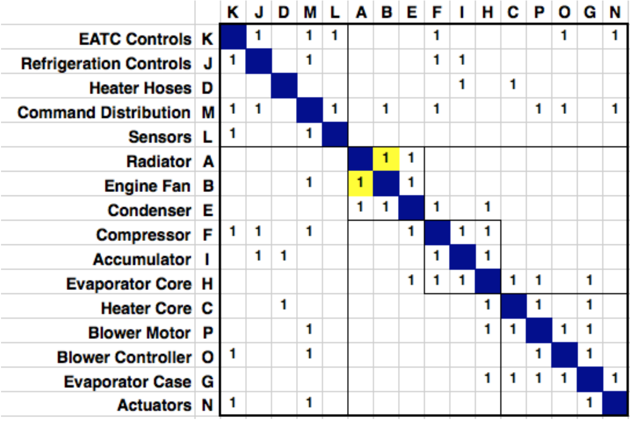


Figure 1. Schematic of a climate control system (Pimmler and Eppinger 1994)



		K	J	D	M	L	A	B	E	F	I	H	C	P	O	G	N
EATC Controls	K	■	1		1	1				1					1		1
Refrigeration Controls	J	1	■		1					1	1						
Heater Hoses	D			■							1		1				
Command Distribution	M	1	1		■	1		1		1				1	1		1
Sensors	L	1			1	■											
Radiator	A						■	1	1								
Engine Fan	B				1		1	■	1								
Condenser	E						1	1	■	1		1					
Compressor	F	1	1		1				1	■	1	1					
Accumulator	I		1	1						1	■	1					
Evaporator Core	H								1	1	1	■	1	1		1	
Heater Core	C			1								1	■	1		1	
Blower Motor	P				1							1	1	■	1	1	
Blower Controller	O	1			1									1	■	1	
Evaporator Case	G											1	1	1	1	■	1
Actuators	N	1			1											1	■

Figure 2. Climate control system DSM

Let us now consider a dyadic design change involving both the radiator (component A) and the engine fan (component B). Note that components A and B are two interdependent components with the fewest number of interfaces with other components: two and three interfaces, respectively. Hence, presumably it should be fairly straightforward to identify the other components affected by a simultaneous design change of components A and B. Specifically, how would examining the system's DSM help reveal the potentially affected components by a simultaneous change in interface (A,B)? To address this challenge, we would need to answer the following questions using the DSM representation above:

- Which are the components that are directly connected to both A and B?
- Which are the components that are directly connected to A and indirectly connected to B?
- Which are the components that are directly connected to B and indirectly connected to A?
- Which are the components that are indirectly connected to both A and B?

Clearly, a simple visual inspecting the DSM does not immediately reveal the affected components asked for in the questions above. To address the first question, we need to find the intersection of rows A and B (or columns A and B). Because the rows (and columns) of these components are located next to each other in the DSM is easy to see that both components A and B are also connected to the condenser component (component E). Certainly, addressing this question is more challenging when the two components are distant in the DSM and they have a larger number of interfaces with other components.

To address the other questions, we first need to identify the components that are indirectly (through at least one intermediary component) connected to either A and B. To do so, we need to square the DSM, whose non-zero cells would reveal the components that would be connected through an intermediary component (Gebala and Eppinger 1991).

In order to answer the second question, we need to find the intersection of row A (or column A) of the original DSM and row B (or column B) in the squared DSM. Doing so, results in an empty set (i.e. there are no components directly connected to A that are indirectly connected to B). Similarly, to answer the third question we need to find the intersection of row B (or column B) in the original DSM and row A (or column A) in the squared DSM. This results in finding that the command distribution component (component M) is directly connected to B and indirectly connected to A.

Finally to find the components that are indirectly connected to both A and B, we need to find the intersection of rows A and B (or columns A and B) in the squared DSM. This results in learning that the compressor and the evaporator core components (components F and H, respectively) are indirectly connected to both A and B.

Clearly, even when considering the least connected components in the system whose interfaces are well documented in a DSM, it is not at all trivial to find out the other directly and indirectly connected components that could potentially be affected by simultaneously changing a pair of interdependent components. Next, we show how bicentric diagrams can provide a leap step forward in addressing this challenge.

BICENTRIC DIAGRAMS

A bicentric diagram is a novel type of visual representation that combines ideas from from graph and set visualization to simultaneously depict sets, relationship between sets, and the reach of set members in the integrated egonetworks of two focal entities (Park and Basole 2016). The technique has been applied to data in a variety of contexts including interfirm alliances (Basole et al. 2015), healthcare activities (Basole et al. 2014), university collaboration (Park and Basole 2015), and technology co-mention in media outlets (Park and Basole 2016).

Fundamentally, a bicentric diagram layout builds on the well-established concentric network visualization technique, where a focal node is placed at the center and its directly connected (or neighboring) nodes are positioned in a circular fashion on the first circle (k=1) and the indirectly connected nodes are placed on the second circle (k=2). A bicentric diagram then smartly overlays two of these concentric egonetwork layouts to provide an effective representation of the two focal nodes as well as their shared direct and indirect nodes organized by tier.

Figure 4 provides a conceptual representation of the bicentric diagram layout technique. In a bicentric diagram there are two focal nodes *A* and *B*, which are positioned at a fixed distance apart. Two concentric circles are drawn around each node. Each circle represents a tier of the focal node. The arcs and intersection points represent areas where corresponding node sets are placed.

- Directly connected nodes shared by both Node A and B are positioned in the center [1].
- Those nodes only connected to Node *A* (or *B*) directly are placed on the arc [A1] (or [B1]).
- Those nodes only indirectly connected to Node A (or B) are placed on the outer arcs [A2] and [B2].
- Nodes one step apart from Node *A* and two steps away from Node *B* are placed around [2a] and [2b]. We differentiate nodes placed at the top ([2a]) and the bottom ([2b]) by whether they belong to the main component or not. We use this approach to minimize long edges (a trait desirable in graph visualizations) and clearly identify clusters.
- Similarly, nodes one step apart from Node B and two steps away from Node A are placed around [3a] and [3b]. The same top ([3a]) and bottom ([3b]) differentiation applies. Nodes in the top cluster belong to the main component; all others are placed in the bottom cluster.
- Nodes two steps apart from both Node *A* and *B* are placed at [4a] and [4b]. The same top ([4a]) and bottom ([4b]) differentiation applies.

We use a consistent visual encoding approach throughout all bicentric visualizations. Nodes represent components. The size of the nodes is proportional to the number of direct connections. Nodes are colored by subsystem category.

In the static version, edges between nodes are not displayed to reduce the clutter and improve readability of the visualization. In the interactive version, users can turn on/off edges as well as node labels. Hovering over individual nodes provides additional information in the tooltip.

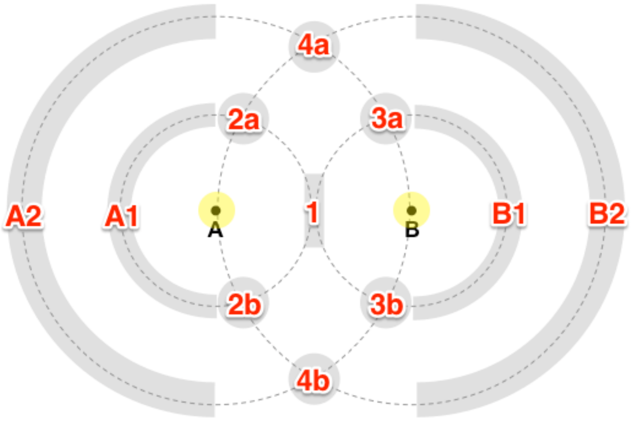


Figure 4. Bicentric diagram of nodes A and B

EXAMPLE 1: A CLIMATE CONTROL SYSTEM

To illustrate the value of this novel visual representation, let us examine the bicentric diagram corresponding to the radiator (component A) and the engine fan (component B) of the climate control system (Figure 5).

The bicentric diagram shown in Figure 5 provides immediate visual answers to the questions posed in the previous section concerning the dyadic design change of components A and B of the climate control system. First, the center of the diagram where the two tier 1 circles intersect (zone 1 of the diagram) corresponds to the components that are directly connected to both components A and B. In this case, there is only one component: the condenser (component E), which is the answer to the first question we posed above. The lack of components in the zone 2a of the diagram indicates that there are no components connected to component A that are indirectly connected to component B. However, the zone 2b in the diagram indicates that the “command distribution” (component M) is directly connected to component B and indirectly connected to component A. The nodes shown in zone 4a of the diagram show the two components (F and H) that are indirectly connected to both A and B.

In addition to providing answers to the questions of interest, the bicentric diagram also indicates the components that are exclusively connected to either A (areas A1 and A2) or B (areas B1 and B2). Because A and B are directly connected the components that directly connect to them must be in either zone 1 or zone 2 or zone 3. Hence, it is impossible to see components in zones A1 and A2, which would correspond to components that are exclusively connected to A (or B) but not indirectly connected to B (or A). However, it is possible to see nodes in zones A2 and B2. They would correspond to components that are exclusively indirectly connected to A (or B).. In this case, component A does not have any other component with which is exclusively connected indirectly whereas component B is exclusively connected (indirectly) to six other components.

In addition to identifying components that are directly or indirectly connected to A and B, we can also identify the nodes that do not appear in the diagram. They correspond to the four components that are not connected either directly or indirectly to A or B. Those are the four unaffected components that are less likely to be affected by a simultaneous design change of A and B. Those unaffected components are components C, D, G, and I.

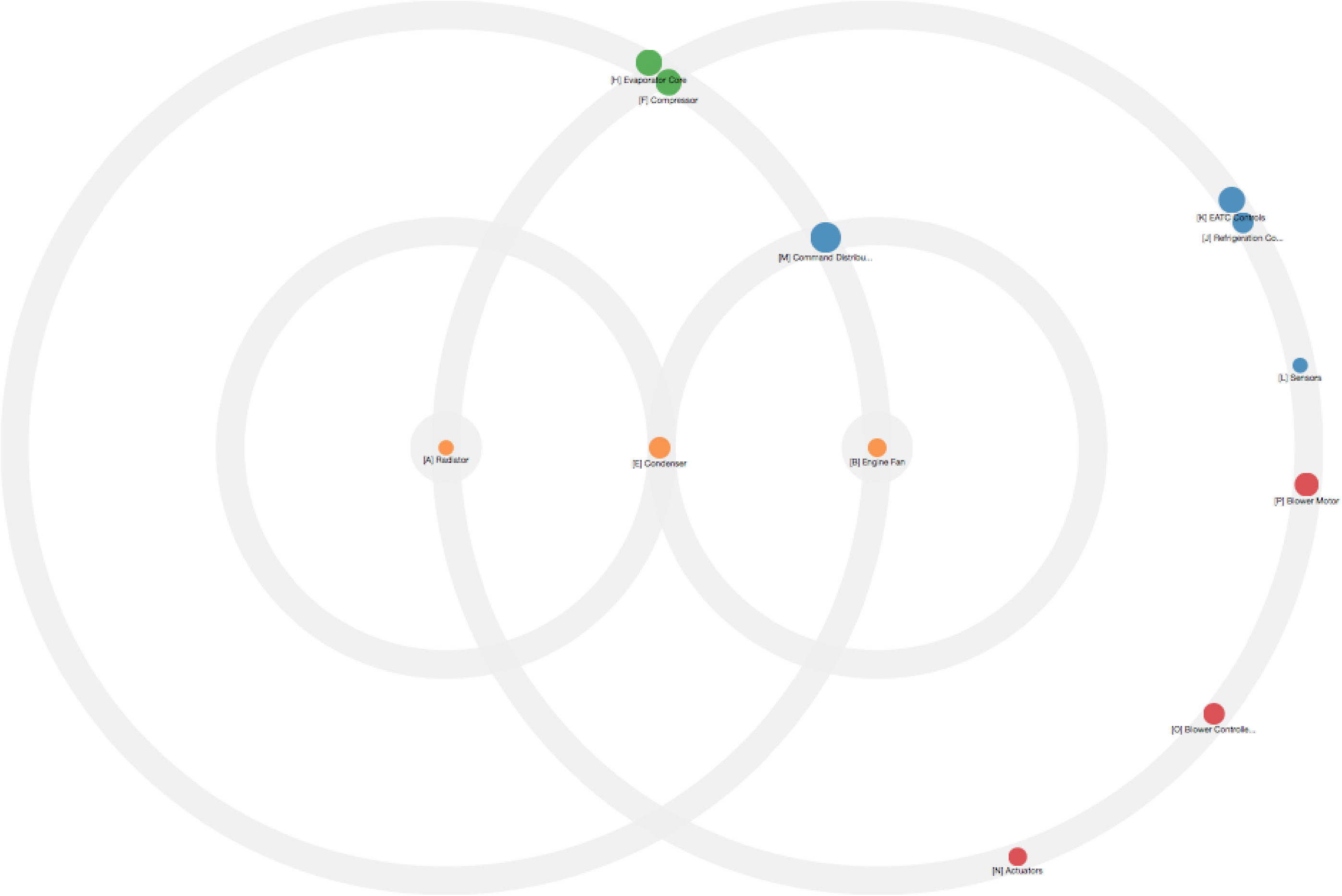
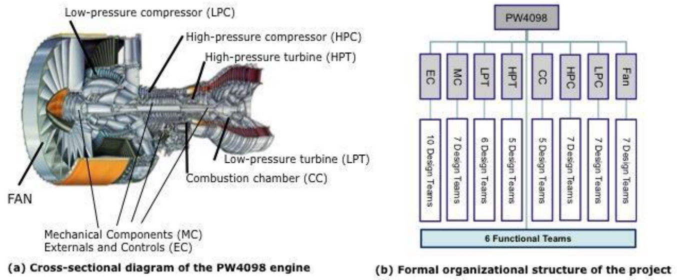


Figure 5. Bicentric diagram of the radiator (component A) and engine fan (component B) of climate control system

EXAMPLE 2: A LARGE COMMERCIAL AIRCRAFT ENGINE

We illustrate further the use of bicentric diagrams for managing dyadic design changes by applying them to various critical design interfaces of a large commercial aircraft engine. The development of this engine has been widely studied in previous engineering design, operations management, and organizational studies (Sosa et al. 2003, 2004, 2007, 2011, 2015). A unique aspect of this dataset is the availability of both product architecture data of the engine and organizational structure data of the teams that designed the engine. In this example we aim to show two cases: i) We will see how examining bicentric diagrams could reduce the chances of critical technical interfaces not being coordinated during the design process; ii) We will also show how bicentric diagrams could help predict the implications of uncovering new technical interfaces between some components.

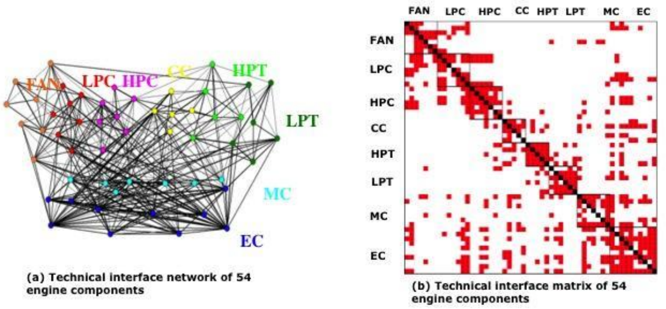
The cross-sectional schematic of the engine along with the formal organizational structure of the teams that design such an engine are shown in Figure 6. The engine is comprised of 54 components structured into the 8 sub-systems shown in Figure 6. On the organizational side there were 54 teams responsible for the design of each of the engine components. In addition there were six functional teams that were responsible for engine level performance evaluation.



(a) Cross-sectional diagram of the PW4098 engine (b) Formal organizational structure of the project

Figure 6. The aircraft engine studied and its formal organizational structure

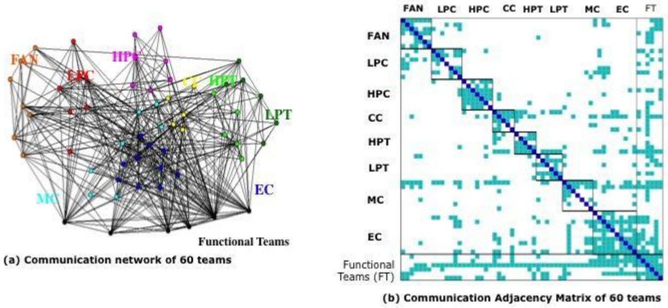
By interviewing the engine's system architects the various types of design dependencies among the 54 engine components were documented. Such a dependency structure can be represented in a network graph or a product DSM as shown in Figure 7. In such a product network (or product DSM), a link between two components indicates that a component depends on another component to achieve its functionality (for details see Sosa et al. 2003, 2007).



(a) Technical interface network of 54 engine components (b) Technical interface matrix of 54 engine components

Figure 7. Network and product DSM representation of the engine studied

On the organizational side, the 54 design teams and six functional teams communicated to coordinate their effort the 10-month engine design phase. Such cross-team communication network, was captured by surveying and interviewing key members of the teams (for details see Sosa et al. 2004, 2015). Figure 8 shows the network graph and team DSM representations of the cross-team communication network.



(a) Communication network of 60 teams (b) Communication Adjacency Matrix of 60 teams

Figure 8. Network and team DSM representation of the organization that designed the engine studied

By overlaying the product and organizational networks shown in Figs 7 and 8 respectively, we can identify not only the *matched interfaces* between engine components that were matched by the corresponding cross-team interactions but also *mismatched interfaces* (interfaces that were not addressed by the corresponding cross-team interactions) and *mismatched communications* (cross-team communications that occurred in the absence of an identified technical interface). (For details see Sosa et al. 2004, 2015.) Next we will show how bicentric diagrams can be used to get additional insights about some critical mismatched interfaces and mismatched communications which did impact the performance of the development of the engine.

Case 1: Managing a critical mismatched technical interface

One of the insights uncovered by analysing the systematic factors that were associated with the occurrence of mismatched interfaces was the effect of sub-system boundaries (interfaces between components of different sub-systems were more likely to be mismatched interfaces) and interface criticality (more critical interfaces were less likely to be mismatched interfaces). Interestingly, critical interfaces across sub-system boundaries were equally likely to be mismatched as non-critical interfaces (see Sosa et al. 2004 for details). The consequence of this is that there were some critical interfaces across sub-system boundaries that were not attended by the corresponding cross-team communications and such an omission had negative performance impact on the development of the engine (Sosa et al. 2004, p. 1687). Examining the bicentric diagram of a couple of such critical mismatched interfaces between one engine components that belongs to the low-pressure compressor (LPC) and two components of the high-pressure compressor (HPC) subsystems give us additional insights about the reasons for the negative impact of such mismatched interface on the development of the engine and offers new ways in which such possible pitfall could be mitigated.

Figures 9 and 10 show the bicentric diagrams of those two critical mismatched interfaces. In both figures, the centered node (colored in dark yellow) on the left hand side of the diagrams represents the component of the LPC while the centered node (colored in green) on the right hand side of the diagrams represents the HPC components.

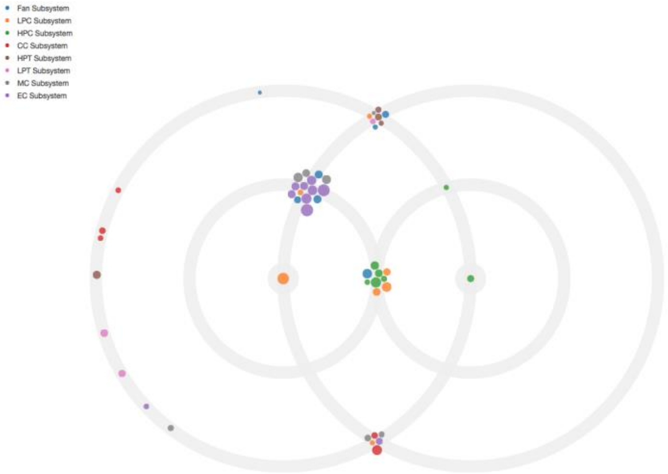


Figure 9. Bicentric diagram between the LPC component and HPC component1

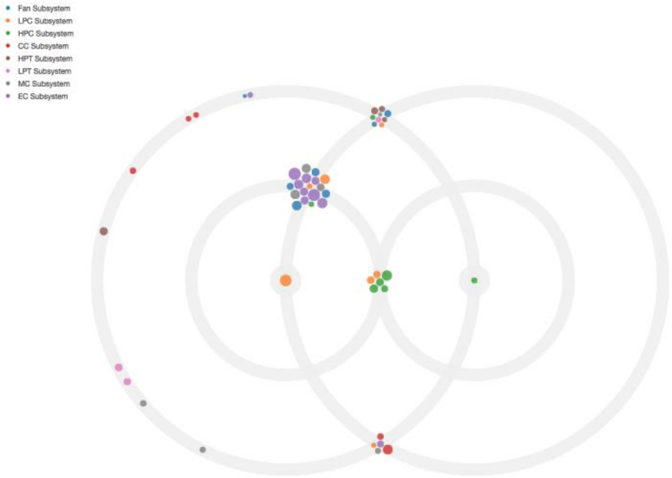


Figure 10. Bicentric diagram between the LPC component and HPC component2

The visual inspection of both diagrams show very similar patterns. They both show that the LPC component (on the left hand side of the diagram) shares many direct interfaces with components that are indirectly connected to the two HPC components. In other words, in addition to strong design dependency of the LPC component on the HPC components, the LPC component can also be affected by design changes on the HPC component that can propagate not only through the common components (in zone 1 of the diagrams) but also through the 15 and 19 other components that were indirectly connected to the HPC components and directly connected to the LPC component (zone 2a in the diagrams).

In order to understand the value that examining these diagrams could have had it is important to understand some of the qualitative observations associated with some of mismatched interfaces. As Sosa et al. (2004, p. 1687) reported, "one reason for these unmatched interfaces is that teams across boundaries did not have opportunities for indirect interactions to communicate or discover changes associated with them. We found this to be particularly relevant for structural and thermal design dependencies." In this project, the design requirements of some cross-boundary interfaces were well documented at the outset of the project and therefore the teams involved did not see the need to communicate and instead adhered to the predefined requirements. "Had they done so, they would have discovered additional load transfer interfaces not explicitly defined" (Sosa et al. 2004, p. 1686), which led to problems later in the project.

This suggests that even if the technical interface between the two focal components is believed to be well understood, it is important to understand that design changes can propagate through intermediary components. As a result, any visual representation, such as the one provided by bicentric diagrams, that can help managers identifying the number of direct and indirect paths in which design changes of two focal components can propagate is poised to be helpful for managers. In this case design changes in the HPC's component could lead to unexpected consequences in the LPC component due to the large number of components that were indirectly connected to the HPC components and directly connected to the LPC component.

Case 2: Managing a critical mismatched cross-team interaction

Mismatched cross-team communications occur in the absence of an identified technical interface. One of the reasons for this occurrence is the existence of a "hidden" technical interface which was uncovered during the design process by the teams involved in such an interface. As Sosa et al. (2004, p. 1687) indicated "many of these [mismatched cross-team communications] were reportedly related to investigations into possible engine-level design conditions which manifested in adverse structural or thermal load transmission or insufficient pressures. Some teams were using their experience with prior generation engines to uncover new direct and indirect design interface characteristics prior to the development of tests where they would be evaluated. This type of team interaction is almost universally positive as it serves to improve product performance and reduce downstream design iterations." This raises the question of how the rest of the organization gets informed about the existence of such a new interface between two given components.

A bicentric diagram of a newly uncovered technical interdependence can inform engineering managers about the other teams that need to be informed of the existence of such a new technical interface. To illustrate this consider the mismatched inter-team communication between a LPC component (component A) and a component of the high-pressure turbine (HPT) subsystem (component B). Figure 11 shows the bicentric diagram corresponding to such a newly uncovered interface.

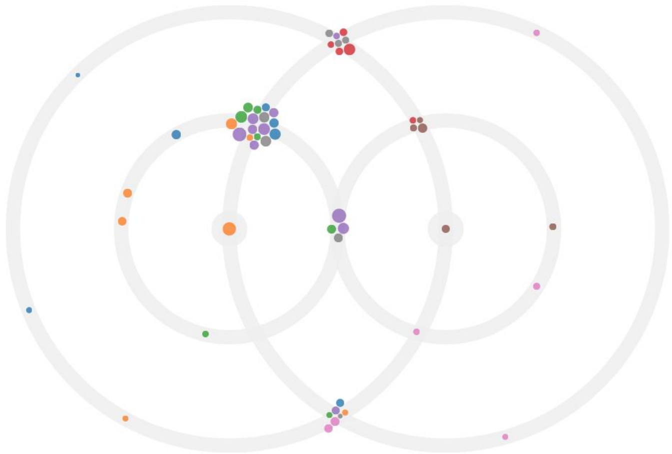


Figure 11. Bicentric diagram of mismatched inter-team communication between a LPC component and a HPT component

By examining the diagram shown in Figure 11 we can immediately identify four components (that belong to three other subsystems) that are directly connected to both components A and B (they are located in zone 1 of the diagram). Interestingly, the diagram also exhibits a high concentration of components in zone 2a indicating that there is a high risk of design changes propagating from component B (the HPT component) to component A (the LPC component) through any of the 17 intermediary components that are two steps away from component B and directly connected to component A. This diagram is consistent with the observation that such a mismatched inter-team communication could serve as an important coordination vehicle to preempt possible design changes propagating through intermediary components between two (a priori) disconnected components.

DISCUSSION

This paper introduces bicentric diagrams as a novel visualization aid that could help engineering managers in managing design decisions concerning any pair of components in a system. The power of this visual representation relies in its way to organize in a simple way the complexity embedded in the architecture of a system defined by its number of components and the dependency structure among them. Moreover, this representation allows engineering decision makers to zoom-in into the dyadic relationship between two system components while simultaneously visualizing how the other components in the system relate directly or indirectly to the focal interface.

This paper illustrated some applications of bicentric diagrams in the design of complex systems. However, the applications of these diagrams can be generalized into distinct family of applications:

i) Manage highly changing interfaces: Once such critical and unstable interfaces are identified, engineering managers can use bicentric diagrams of such interfaces to identify the other components that could get affected by such dyadic changes. This would allow them to prioritize who to coordinate efforts with to manage such design changes.

ii) Removing an existing interface: Contrary to highly changing interfaces, some interfaces can be standardized or removed altogether. Given the proliferation of modularity principles to manage the design of complex systems, it is crucial to understand the impact to the rest of the system of "freezing" a given interface (Baldwin and Clark 2001, Holtta-Otto and de Weck 2007). Bicentric diagrams can assist managers not only to identify prominent interfaces to "freeze" but also to manage how to inform other stakeholders which could be affected by freezing an interface.

iii) Adding a new interface: As illustrated in the second example above, technical interfaces between components are often uncovered during the design process. In such situations, bicentric diagrams can help managers to identify the other stakeholders concerned by the addition of such a new interface to the architecture of the system. For instance, in the development of software systems, the equivalent of adding a new interface is establishing a new function call between two previously independent software objects (e.g. java classes). When this happens is critical to announce to the responsible of other software objects affected of such a new interface addition into the system (MacCormack et al. 2006, Sosa et al. 2014).

This paper has introduced the use bicentric diagrams to manage dyadic design changes based on a product DSM representation of the system. However, often what managers have is the collaboration network of the engineers (i.e., the people DSM of the organization). For instance, Sosa (2008) and Sosa and Marle (2013) documented how engineers in the R&D department of a software organization communicate for work-related reasons. In such, a collaboration network of engineers, bicentric diagrams would highlight the other people that would potentially need to be informed when something important for the organization happens in a dyadic collaborative relationship (e.g., two people get collocated, or separated into distinct teams or locations; or two people make an important discovery or analyze some test results).

Bicentric diagrams can also be used to complement process DSMs, which captures the design process as a network of interdependent tasks (Steward 1981, Eppinger et al. 1994). In this case, bicentric diagrams help managers identify the other tasks that would be affected by changes of a given pair of highly interdependent activities. Again, if something changes between two activities (e.g., two activities become concurrent or overlapped, or a new person gets responsible to handle the interdependence with another activity), then the bicentric diagram would help identifying the other activities that could potentially be affected by such a dyadic change.

We hope this introduction of bicentric diagrams to manage dyadic design changes opens new possibilities for research in engineering design that can lead to both academic and managerial implications for better complex systems development.

ACKNOWLEDGMENTS

This research was in part supported by the Tennenbaum Institute at Georgia Tech and INSEAD's R&D funds.

REFERENCES

- Alexander, C., 1964, *Notes on the Synthesis of Form*. Harvard University Press, Cambridge, MA.
- Baldwin, C. Y., and Clark, K. B., Design Rules: The Power of Modularity (MIT Press, Cambridge, MA, 2000).
- Bar-Yam, Y., 2001, *Introducing Complex Systems.* Boston, MA: NECSI Press.
- Basole, Rahul C., et al. "Bicentric visualization of pediatric asthma care process activities." *Proceedings of IEEE VIS 2014 Workshop of Electronic Health Records, IEEE*. 2014.
- Basole, Rahul C., et al. "Understanding business ecosystem dynamics: a data-driven approach." *ACM Transactions on Management Information Systems (TMIS)* 6.2 (2015): 6.
- Basole, R. C., Qamar, A., Park, H., Paredis, C. J., & McGinnis, L. F. (2015). Visual analytics for early-phase complex engineered system design support.*IEEE computer graphics and applications*, *35*(2), 41-51.
- Basole, R. C. (2014). Visual business ecosystem intelligence: Lessons from the field. *IEEE computer graphics and applications*, *34*(5), 26-34.
- Basole, R. C., Huhtamäki, J., Still, K., & Russell, M. G. (2016). Visual decision support for business ecosystem analysis. *Expert Systems with Applications*, *65*, 271-282.
- Braha, D., and Bar-Yam, Y., 2007, "The Statistical Mechanics of Complex Product Development: Empirical and Analytical Results," Manage. Sci., 53(7), pp. 1127–1145.
- Browning TR (2001) Applying the design structure matrix to system decomposition and integration problems: a review and newdirections. IEEE Trans Eng Manage 48(3):292–306
- Browning TR. 2016. Design structure matrix extensions and innovations: a survey and new opportunities. IEEE Transactions on Engineering Management.
- Card, S. K., Mackinlay, J. D., & Shneiderman, B. (1999). *Readings in information visualization: using vision to think*. Morgan Kaufmann.
- Chen L, Ding Z, Li S (2005) A formal two-phase method for decomposition of complex design problems. ASME J Mech Des 127:184–195
- Chen L, Macwan A, Li S (2007) Model-based rapid redesign using decomposition patterns. ASME J Mech Des 129:283–294
- Clark, K.B., and Fujimoto, T., 1991, "Product Development Performance: Strategy, Organization and Management in the World Auto Industry," Harvard Business School Press, Boston, MA, USA.
- Clarkson, P.J., Simons, C.S. and Eckert, C.M., 2004, "Predicting change propagation in complex design", *ASME Journal of Mechanical Design*, Vol. 126 (5), pp. 765-797.
- Eckert, C.M., Clarkson P.J. and Zanker, W., 2004, "Change and customisation in complex engineering domains", *Research in Engineering Design*, Vol. 15, pp. 1-21.
- Eckert, C., Clarkson, J., Earl, C. 2005. Predictability of change in engineering: A complexity view. Proceedings of ASME IDETC/CIE conference. Long Beach, CA. USA.
- Eppinger, S. D., Whitney, D. E., Smith, R. and Gebala, D., 1994, "A Model-based Method for Organizing Tasks in Product Development", *Research in Engineering Design*, Vol. 6 (1), 1-13.
- Fekete, Jean-Daniel, Jarke J. Van Wijk, John T. Stasko, and Chris North. "The value of information visualization." In *Information visualization*, pp. 1-18. Springer Berlin Heidelberg, 2008.
- Gebala, D. and Eppinger, S. 1991. Methods for analyzing design procedures. Proceedings of the ASME Design Theory and Methodology conference.
- Gokpinar, B., Hopp, W. J., and Iravani, S. M. R., 2010, "The Impact of Mis- alignment of Organizational Structure and Product Architecture on Quality in Complex Product Development," Manage. Sci., 56(3), pp. 468–484.
- Gumbel, P. 2006. Trying to untangle wires. *Time*. October 16 (European edition), p. 36-37.
- Heer, J., Bostock, M., & Ogievetsky, V. (2010). A tour through the visualization zoo. *Commun. Acm*, *53*(6), 59-67.
- Henderson, R. M., and Clark, K. B., 1990, "Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms," Adm. Sci. Q., 35, pp. 9–30.
- Holtta-Otto, K. and de Weck, O. 2007. Metrics for assessing coupling density and modularity in complex products and systems. Proceedings of the ASME IDETC/CIE conference. Las Vegas, NV, USA.
- Jarratt, T., Eckert, C., and Clarkson, J., 2005, "Pitfalls of Engineering Change: Change Practice During Complex Product Design," *Advances in Design*, pp. 413–424.
- Keller, R., Eger, T., Eckert, C., and Clarkson, J., 2005, "Visualising Change Propagation," 15th International Conference on Engineering Design, pp. 62– 63.
- A. Kusiak and J. Wang, "Efficient organizing of design activities," *Int. J. Prod. Res.*, vol. 31, pp. 753–769, 1993.
- MacCormack, A., Rusnak, J., and Baldwin, C. Y., 2006, "Exploring the Struc- ture of Complex Software Designs: An Empirical Study of Open Source and Proprietary Code," Manage. Sci., 52(7), pp. 1015–1030.
- K. R. McCord and S. D. Eppinger, "Managing the integration problem in concurrent engineering," MIT Sloan School of Management, Cam- bridge, MA, Working Paper No. 3594, 1993.
- Morelli, M.D., S.D. Eppinger, R.K. Gulati. 1995. Predicting technical communication in product development organizations. *IEEE Trans. Eng'g. Management* 42(3): 215-222.
- North, C. (2006). Toward measuring visualization insight. *IEEE computer graphics and applications*,

26(3), 6-9.

- Olson J, Cagan J, Kotovsky K (2009) Unlocking organizational potential: A computational platform for investigating structural interdependence in design. ASME J. Mech. Design 131(3): 031001–031001-13.
- Park, Hyunwoo, and Rahul C. Basole. "Bicentric diagrams: Design of a graph-based relational set visualization technique." *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2015.
- Park, Hyunwoo, and Rahul C. Basole. "Bicentric diagrams: Design and applications of a graph-based relational set visualization technique." *Decision Support Systems* 84 (2016): 64-77.
- Pinedo, M., S. Sehasdri, and E. Zemel. 2000. The Ford-Fireston case. *Teaching case*. Department of Information, Operations, and Managment Sciences, Stern School of Business, NYU.
- Pimmler T.U. and Eppinger S.D., "Integration analysis of product decompositions", ASME Design Theory and Methodology Conference, Minneapolis, 1994.
- Sharman, D. M., and Yassine, A. A., 2004, "Characterizing Complex Product Architectures," J. Syst. Eng., 7(1), pp. 35–60.
- Simon, H. A., The Sciences of the Artificial (MIT, Cambridge, MA, 1996).
- Sosa, M., S. Eppinger, C. Rowles. 2003. Identifying modular and integrative systems and their impact on design team interactions. *ASME Journal of Mechanical Design*, 125(2): 240-252.
- Sosa, M., S. Eppinger, C. Rowles. 2004. The misalignment of product architecture and organizational structure in complex product development. *Management Science*, 50(12): 1674-1689.
- Sosa, M., S. Eppinger, C. Rowles. 2007. A network approach to define modularity of components in complex products. *ASME Journal of Mechanical Design*, 129(11): 1118-1129.
- Sosa, M. 2008. A structured approach to predicting and managing technical interactions in software development. *Research in Engineering Design*, 19(1): 47-70.
- Sosa, M., J. Mihm, and T. Browning. 2011. Degree Distribution and Quality in Complex Engineered Systems. *ASME Journal of Mechanical Design*, 133(10): 101008:1-15.
- Sosa, M., J. Mihm, and T. Browning. 2013. Linking Cyclicality and Product Quality. *Manufacturing & Service Operations Management* 23(5).
- Sosa, M., F. Marle. 2013. Assembling creative teams in new product development using creative team familiarity. *ASME Journal of Mechanical Design* 135(8): 081009:1-13.
- Sosa, M., M. Gargiulo, C. Rowles. 2015. Can informal communication networks disrupt coordination in new product development projects? *Organization Science* 26(4): 1059–1078.
- Speier, Cheri. "The influence of information presentation formats on complex task decision-making performance." *International Journal of Human-Computer Studies* 64.11 (2006): 1115-1131.Terwiesch, C. and Loch, C. H, 1999, "Managing the Process of Engineering Change Orders: The Case of the Climate Control System in Automobile Development", *J Product Innovation Management*, Vol. 16, pp. 160-72.
- Stasko, J. (2014, November). Value-driven evaluation of visualizations. In*Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (pp. 46-53). ACM.
- D. V. Steward, "The Design structure system: A method for managing the design of complex systems," *IEEE Trans. Eng. Manage.*, vol. 28, pp. 71–74, 1981.
- Tegarden, David P. "Business information visualization." *Communications of the AIS* 1.1es (1999): 4.
- Terwiesch, C. Loch, C. De Meyer, A. 2002. Exchanging preliminary information in concurrent engineering: Alternative coordination strategies. Organization Science.
- Wright, I.C., 1997, "A Review of Research into Engineering Change Management: Implications for Product Design," Design Studies, Vol. 18, pp. 33-42.
- Zhu, Bin, and Hsinchun Chen. "Information visualization for decision support." *Handbook on Decision Support Systems 2*. Springer Berlin Heidelberg, 2008. 699-722.