2018 Conference on Systems Engineering Research

A Value-Centric Tradespace Approach to Target System Modularization

Adam M. Ross[a]*, Hank H. Roark, III[a]

[a] *Massachusetts Institute of Technology, 77 Mass Ave, Cambridge, MA 02139, USA*

Abstract

Deciding where to modularize a system can have long-term impact on that system's value over its entire lifecycle. The modularity of a system can impact the system's flexibility, evolvability, scalability, mass, costs, and development schedule. Making these modularization decisions is a key job of the system architect. There is a need to provide the system architect tools that will help focus modularization efforts on the areas of the system that are most likely to provide value to stakeholders of the system. Using a terrestrial vehicle as a case study, an approach is developed that links component modularity to system design variables which are likely to change levels. The approach utilizes dynamic value-driven tradespaces and network measures of component modularity to identify components which are most likely to need to change as well as the components' ability to make a modular change. The approach is shown to provide early design insights about value-centric system modularizations; the approach does require a network representation of the system earlier in the design cycle than may be typically available. Using explicit knowledge, the approach developed can focus designers' modularization efforts on the elements of the system that may need to change to accommodate changes in stakeholders' preferences and use contexts.

Keywords: modularity; value-centric; design; multi-attribute; tradespace exploration; network measures

1. Introduction

During conceptual design of a system much of the lifecycle cost is committed. One of the architecture choices that can be committed in this phase is how a system's components are interconnected, and hence the level of modularity of the components. This modularity level can have long term impacts on the changeability of a system over its lifespan. One of the difficulties with the previous statement is that modularity has many definitions, introducing ambiguity into how to measure and value modularity.

With so many definitions, potential lifecycle benefits, and potential trade-offs, design engineers are faced with many objectives and challenges when deciding how to modularize components within a system. Some work has been done on this front, including the development of clustering algorithms that take into account the endogenous system structure to determine component clusters[1] and the engineering system matrix that uses qualitative knowledge of endogenous and certain exogenous system factors as a screening for potential areas of high system change[2]. One of the author's industry experience, however, is that the decision on organizing system components into modules during conceptual design seems to be based on expert experience, rules of thumb, heuristics, and iteration. Stakeholders and system designers can benefit from practical tools to aid in making design decisions on where to focus modularity efforts during conceptual and early design.

* Corresponding author. *E-mail address:* adamross@mit.edu

A dynamic tradespace-based approach can help make design decisions regarding incorporating changeability into a system, based on changes in customer needs over time and physics-based models of system performance[3]. This paper describes an approach that uses dynamic tradespaces coupled with a network representation of a system's components to aid in value-based decisions about if, and where, to modularize a system to make it more robust to stakeholder requirements changes. This approach will provide design engineers with tools to make modularization decisions of a system in support of product variety and evolvability.

1.1. On Modularity

One of the more difficult aspects of this research was to settle on an operational definition of modules, modularity, and modularization. MIT's Engineering System Division defines a module as "a part of a system that is constructed to have minimal, standardized interactions with the rest of the system" and modularity as "the degree to which the components of a system can be designed, made, operated, and changed independently of each other"[4]. Several authors have defined modularity as something along the lines of the one-to-one mapping of a system function to the system form and including certain component interface characteristics[5], while others have defined modularity as a series of design rules that enable various operations, called modularity operators, on the system[6]. While these definitions suffice as conceptual definitions, this research requires a definition with an operational measure of modularity.

This research is primarily concerned with increasing the changeability of a system based on how well certain components are connected as modules, and their connectivity to the other elements in the system. With this desire, and the need for an operational measure, the definition provided by Ref 7 will be adopted by this research: modularity is "a measure of the lack of technical interface connectivity between components of a system." Based on network analysis and graph theory, there are three metrics for measuring component modularity and their impacts on component redesign: degree modularity M(D), distance modularity M(T), and bridge modularity M(B), which are a normalization of existing network centrality metrics (Degree, Freeman Closeness, and Freeman Betweenness) to a range of [0, 1], where 1 is the highest level of modularity for each metric[7]. From these metrics there is a correlation between the set of outdegree modularity and outdistance modularity and the likelihood of planned redesign of a component to change the system's performance level. These metrics and this correlation finding will be used for the modularity analysis of components needed to change a system's performance.

1.1.1. Benefits and Challenges to Modularity

Modular systems have been proposed to provide many lifecycle benefits to systems and products, including, but not limited to, increasing overall system economic value[6], aiding in system flexibility and evolvability[8], increasing product variety[9], and as an aid to complexity management[10]. Ref 11 proposes modularity as a mitigation strategy for certain types of uncertainties. In this regard, modularity can be viewed as a means to achieve desired goals, and may be critical for achieving other lifecycle properties[8,12].

While modular architectures have many potential benefits, these benefits do not come without costs: for systems with relatively high energy density, modularity increases weight, increases cost, and/or results in lower performance[13]. For systems with business and performance constraints where stakeholders value lower weight, smaller size, or higher performance, one tends to find more integral architectures; conversely when stakeholders value commonality and reuse across products in order to achieve cost savings, one tends to find more modular architectures[14].

1.1.2. Existing Modularization Approaches

The primary methods to aid a design engineer seem to be focused on clustering or heuristics. A clustering approach, using Design Structure Matrix (DSM) representations of a system, can be used to cluster a proposed system into blocks where the interactions within a module cluster are maximized and between clusters are minimized[1]. A shortcoming of this approach is that it only looks endogenously at the system and lacks tools necessary to identify areas where modularization may aid in achieving desired lifecycle properties and system variability across decision makers or through time. Function-cluster was proposed as a way to consider potential changes in mass, energy, and information flow between components and to cluster a system into modules, and suggested system cleavage points to introduce interfaces between modules that will minimize the likelihood of propagation of changes should a module need to change[15]. This approach takes into consideration exogenous factors that could require changes in the system but leaves it up to the system architect to recognize potential changes that may emerge due to changing or different stakeholder needs.

On the other end of the modularization approach spectrum are heuristics[16,17]. The shortcoming with these approaches is that they suggest extensive experience is required by the system architect to make modularity choices, leaving the developing organization to rely on tacit knowledge and the designer with little experience to blindly trust in the heuristic's applicability.

2. Approach

The scope of this work includes the development of an approach to focus system modularization efforts and its application to a case. The Responsive Systems Comparison (RSC) Method, based on Multi-Attribute Tradespace Exploration (MATE), will be used, as it develops value focused designs and indicators of potential design changes[18]. DSMs will be used to model the system architecture and the connectivity between the components, and network centrality measures based on the DSM will be used to measure modularity of the components. These two different techniques, RSC and DSM, will be combined into one approach (Fig. 1). To demonstrate the approach, the case study will be a mobile terrestrial machine (LifeTrac), which was chosen because it has multiple use scenarios and is available as open source. The goals of the approach are as follows:

1. Provide a mechanism to understand potentially desired changes to design variables, based on decision maker preferences, and link those changes to components that may need to be altered in response. The purpose of this goal is to provide designers a value-centric approach to target modularization efforts.

2. Build upon the Responsive Systems Comparison (RSC) method by making connections to RSC's existing process steps. The purpose of this goal is to use the decision analysis tools from RSC to make trade-off decisions in terms of decision maker utility, system cost, and modularization.

3. Utilize the component modularity metrics to evaluate the component modularity of a proposed system architecture[7]. The purpose of this is to be able to measure component modularity during early stage architecture synthesis or to be able to reverse engineer existing systems and quantify the modularity of components in that existing system.

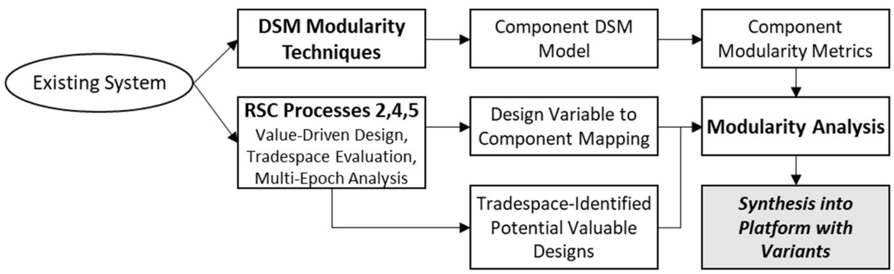

Fig. 1. Approach for considering modularity with RSC

Component modularity analysis consists of two activities: quantification of the modularity of the components of a system and linkage of the components to the design variables used in RSC. Quantification of the components is done by representing the system in a way that is susceptible to network centrality measures, such as a DSM representation, and then calculating the component modularity metrics[7]. It should be noted that constructing the form-form interaction network requires either development of a DSM or network representation of the form-form interactions of the system. Because new systems are often an evolution of an existing system, for many development efforts this assumption is achievable. There are other scenarios, such as green-field design, where representation of the form-form interactions of the underlying system cannot be built due to lack of necessary inputs.

Another needed activity is to provide a linkage between components of the system and design variables. This linkage is an indication of the system components that are likely to need to change because of a change in level of a design variable. This linkage is represented as a multiple domain mapping table between the components and the design variable.

If a designer chooses to have design(s) able to be changed readily over its life, then the designer can place appropriate modularity requirements on the components to reduce the time and/or costs to execute the change. The component modularity analysis is used to derive a table that links the design variables to the components that are likely to change because of a design variable changing. This linkage would only be in place if there is a system architecture being proposed at this stage, as may be the case in changes to existing products or systems. Modularity requirements should also be carried into downstream architecture synthesis activities.

3. Case Example: LifeTrac Tool

The case study system is the OpenEcology Project's LifeTrac tool, which was designed for two functions: acting as a simple wheeled skid steer, and as a simple agricultural tractor[20]. The purpose of the skid steer function is to push, lift, and move material around a work site. The purpose of the agricultural tractor function is to provide tractive energy for pulling agricultural implements, such as tillage or seeding equipment, through a field. It is interesting to note that the LifeTrac tool is described as a "modular" design, but little is given on the website to backup that claim. The LifeTrac tool was chosen for this case study for the following reasons:

- The LifeTrac tool is targeted for use by two types of users. The first type is the farmer who desires a tool to aid in field operations; the second type is a construction work site operator who desires a tool for moving material around a job location. With these two types of users and use cases one might expect two different measures of utility that might result in different desirable system.

- The LifeTrac tool is an existing system, demonstrating the use of the proposed modularization design approach in the evolution of a system. This is considered an acceptable starting scenario as many design efforts are incremental in nature.
- While a simple design, LifeTrac is sufficiently complex to demonstrate the proposed modularization design approach. Complexity was determined by the number of components (n = 47), and interactions (i = 218) between the components, based on the analysis of the component DSM model of the system. By keeping with a simple design, it is hoped that the research is more approachable.
- The design and costing information is covered under an open source license, allowing ease of research access[21].

3.1. DSM Modularity Techniques

The first activity in the proposed approach is to construct a component DSM model of the underlying system. This was done for the LifeTrac tool by reverse engineering open source computer aided design (CAD) models of LifeTrac[22]. The component DSM model in Fig. 2 considers four types of dependencies: spatial (P), mass flow (M), information flow (I), and energy flow (E). Overall, there are 47 DSM elements in the DSM, with 218 element-to-element dependencies, leading to an interaction density of 0.10.

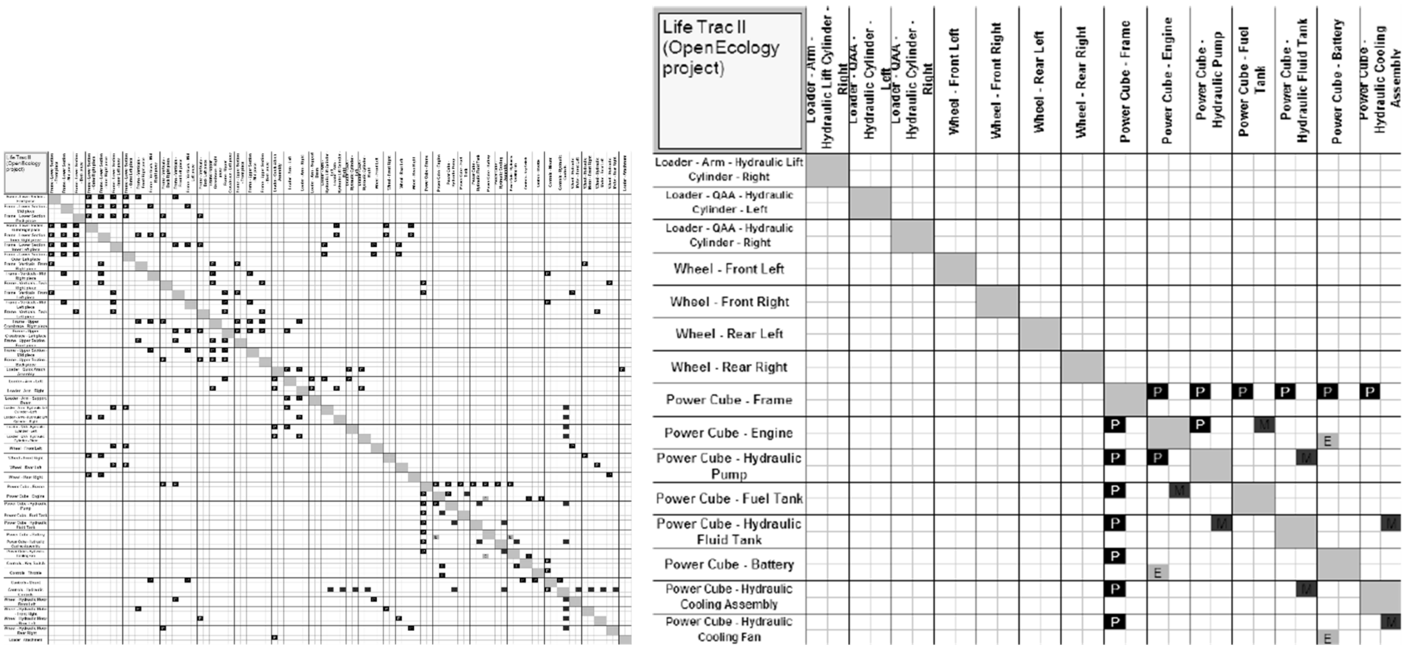

Fig. 2. Component DSM for the LifeTrac system; full (left), lower-right quadrant zoom (right)

These interaction types were simplified to binary, single type (there is either an interaction between two elements or there is none) interactions for the remainder of the case study. The decision to use binary, single type interactions is a trade off in model fidelity versus execution time for later analysis. Further, physical and information dependency may be more easily determined than energy or mass flow dependencies[7]; by collapsing to one type it is suggested that at least the connectivity between components will be more completely recorded for use in the subsequent network centrality analysis. Many alternatives are available for increasing the fidelity of the model of interactions, including types of interactions as well as strength of interactions[7].

At this point in analysis it is not uncommon to apply clustering algorithms to the DSM to group components together into modules. However, that step is not necessary here as we are not interested in the grouping of components into modules (clustering does not change the connectivity between the components). Instead, we want to explore how connected each component is to all other components in the system as an indication of the "cost" of propagation of change.

The next activity in the approach is to calculate the metrics for degree, distance, and bridge centrality[7]. Each of these metrics is in the range of [0, 1], with higher values corresponding to a higher level of modularity for that metric. These metrics, which have been calculated for all LifeTrac components, provide the following insights into the connectivity of a component to other components:

Degree modularity: the number of other components that have direct dependencies with a given component. The less direct dependencies component i has with other components, the higher the value of $M(D)_i$. This is the simplest of the three metrics.

Distance modularity: how far away (or how close) a given component is to other components. This is built on the concept of farness (or its inverse, closeness) in network theory. This measure captures the concept that design changes may propagate not just to/from immediate neighbors (as measured in degree modularity) but also through the network of the design dependencies from components. If component i has high distance modularity $M(T)_i$ then changes to that component would have a longer distance to traverse to reach other components in the system.

Bridge modularity tells one about how many design dependency paths a component lives on between other components. This is built on the network theory concept of centrality. The idea with this measure is to capture the degree to which a component is on design dependency paths between other components. If component i has high bridge modularity $M(B)_i$ then it lies on fewer design dependency paths between all other components.

3.2. RSC Processes 2, 4, 5: Value-Driven Design, Tradespace Evaluation, and Multi-Epoch Analysis

For this case, two decision makers are considered for the LifeTrac: one is a consumer that will use the LifeTrac as a skid steer to move materials around a work site (i.e., construction usage), another is a farmer that will use the LifeTrac for tractor field work (i.e., agricultural usage). In practice, this can be represented as four use scenarios ("epochs"): two for construction ((1) demanding needs and (2) balanced needs) and two for agricultural (row spacing (3) 24 inches and (4) 30 inches). For the construction use, there are three attributes of interest: material capacity, maneuverability, and lifting capacity, whereas in the agricultural use there is only one attribute of interest: efficiency, along with a row spacing constraint. Table 1 lists the measurements, units, acceptance range, and utility weights for each of the attributes across these two uses, including the minimum (U = 0) and maximally desired (U = 1) levels. These attributes are now described:

Material capacity – maximum load is a measure of max weight the vehicle has capability to vertically lift and carry around.

Maneuverability – vehicle width is a measure of ability of the vehicle to fit through openings and passages.

Lifting capacity – breakout force is a measure of force to break material apart (e.g., pulling an embedded stone out of ground).

Efficiency – vehicle work rate is a measure of efficiency to perform key agricultural tasks quickly over an area.

Table 1. Attributes for both users: construction (material capacity, maneuverability, lifting capacity) and agriculture (efficiency, subject to row spacing constraint)

Attribute	Measurement	Units	U = 0	U = 1	Epoch 1	Epoch 2	Epoch 3	Epoch 4
Material capacity	Maximum load	pounds	1200	2000	0.0	0.5	n/a	n/a
Maneuverability	Vehicle width	inches	96	72	0.0	0.2	n/a	n/a
Lifting capacity	Breakout force	pounds	1500	2500	0.0	0.3	n/a	n/a
Efficiency	Work rate	acres/hr	2	6	n/a	n/a	1	1
Row spacing	<constraint>	inches	n/a	n/a	n/a	n/a	24	30

For both agricultural and construction usage, lifecycle cost is measured as the acquisition cost of the underlying system plus the fuel usage over the period of use. The period of use is 5 years with 300 hours of engine time per year, typical for usage patterns and the lifespan of products of this type and size. The acquisition cost model for LifeTrac was built from online information[23,24]. For operating expenses, costs were assumed to be USD1 per horsepower-hour.

Next, we conduct design-value mapping (DVM), with the goal to ensure design variables (factors in our control) relate to achieving decision makers' goals and are considered in subsequent tradespace development. The DVM is in Table 2.

Table 2. Tractor and Skid steer attributes to design vector (design-value) mapping

Design Vector		Attributes				
Variable	Range	Material capacity	Maneuverability	Lifting capacity	Efficiency	Row spacing
Bucket width	[56-84] inches	X	X			
Available hydraulic power	[4-40] HP	X		X		
Engine power	[4-40] HP				X	
Vehicle width	72,90 inches					X

For each design variable, ranges were determined by the availability of off the shelf components; in a clean sheet design these dependencies would be relaxed. A description and justification for each design vector element follows:

Bucket width represents the width of the attachment on the front of the machine. Width of the attachment bucket minimum was set in a range typical available from commercial suppliers (based on a review of attachment bucket sizes from Deere, Bobcat, and Caterpillar websites). For the sake of simplicity, only bucket width is determined to affect maneuverability because width of the vehicle is fixed in this epoch and is considered not tradeable.

Available hydraulic power represents the amount of hydraulic power, as measured by the brake power of the engine. The base design of the LifeTrac has an engine power of 28 horsepower (HP). The minimum and maximum power of this design variable is set at 4 HP and 40 HP, as it is the limit of available air-cooled internal combustion engines available off the shelf[24] (same source the LifeTrac team sourced the current engine).

Engine power represents the amount of engine power available, as measured by the brake power of the engine, that will be translated to tractive force. The base design of the LifeTrac has an engine power of 28 horsepower (HP). The ranges were selected for the same reasons as stated in the description of hydraulic power.

Vehicle width represents the tire-center to tire-center spacing. Because this is a constraint, depending on the epoch, only one of the two levels will provide a feasible design in each epoch (i.e., if considering the 24 inch row spacing epoch, only the 72 inch vehicle width will result in feasible designs because 72 mod 24 = 0 and 90 mod 24 ≠ 0). Since the variable will need to have different values in different epochs to produce feasible designs, it could affect the design of components within the system, making those components potential candidates for modularity analysis.

Now that we have a list of value-driving design variables, the next step is to map these to the DSM components to identify which components would be affected by alternative choices for the design variables. A description of the components identified with each design variable now follows, along with a mapping to components in Table 3.

Bucket width: The main component affected by this design variable is the 'Loader - Attachment' component, the DSM entity that is for the attachment element of the LifeTrac.

Hydraulic power: All components associated with generating and transmitting hydraulic power to the loader arm are determined by the setting of this design variable. Also affected is the length of the loader arm as it is a lever that transmits the hydraulic force to the attachment.

Engine power: This list contains all the components that are responsible for generating and transmitting power to the wheels. These largely determine the ability to create tractive force required for pulling implements through the ground.

Vehicle width: These are the main frame structural elements that determine the overall width of the vehicle.

Table 3. LifeTrac design variable to DSM component mapping

Design Variable		DSM Component i	Component Names
Skid Steer (construction)	Bucket Width	47	Loader - Attachment
	Hydraulic Power	20, 21, 23, 24, 25, 26, 32, 33, 42	Loader - Arm - Left, Right; Loader - Arm - Hydr. Lift Cyl. - Left, Right; Loader - QAA - Hydr. Cyl. - Left, Right; Power Cube – Engine; Power Cube - Hydr. Pump; Controls - Hydr. Controls
Tractor (agriculture)	Engine Power	32, 33, 42, 43, 44, 45, 46	Power Cube – Engine; Power Cube - Hydr. Pump; Controls - Hydr. Controls; Wheel - Hydr. Motor - Front Left, Right; Wheel - Hydr. Motor - Rear Left, Right
	Vehicle Width	1, 2, 3	Frame - Lower Section - Front piece, Mid piece, Back piece

Next, we generate our tradespaces. The alternative LifeTrac designs are generated by varying the design variables across their allowed ranges. Combinations of design variables at particular levels result in a unique design, which was then evaluated in terms of the attributes via a physics-based performance model and a cost model. Due to the row spacing constraints in the agricultural epochs (3 and 4), half of the designs were infeasible (i.e. not usable at all). Table 4 lists the number of designs in each of the four epochs.

Table 4. Basic metrics from LifeTrac tradespaces (designs considered $n = 592$)

Tradespace	Feasible Designs	Designs U(X) ≥ 0
Skid steer, full soln. (epoch 1)	592	256
Skid steer, partial soln. (epoch 2)	592	256
Tractor, 30 in (epoch 3)	296	256
Tractor, 24 in (epoch 4)	296	272

A large number of tradespace analyses were performed to identify design variables that drive value. Screening metrics for looking across epochs are useful for identifying valuable designs, that is, designs that are most efficient in utility for cost (i.e. Pareto efficient). Fuzzy Normalized Pareto Trace (fNPT) can be used to identify designs that, while not strictly non-dominated, could be valuable across epochs. The factor K will be used to denote a tolerance to uncertainty and was varied from 0.00 to 0.10 in steps of 0.02 (the situation of K = 0 is the same as the strict Pareto set)[25].

The results for K = 0 and 0.04 are shown in Fig. 3. We would expect to find no designs achieving a $fNPT_i = 1$ because of the constraint on width across epochs 3 and 4 (i.e., one half of designs in those epochs are not feasible). As K is increased, we see more and more designs within being "close" to Pareto efficient. Upon inspection, it is when K = (0.02, 0.04] that we see designs that, across the Epoch 3 and Epoch 4 constraints, have an fNPT of 0.75; this means that a design is good in both construction use epochs and one agricultural use epoch. These designs are now selected as "passively value robust" and provide the reduced set of potential candidates for modularity.

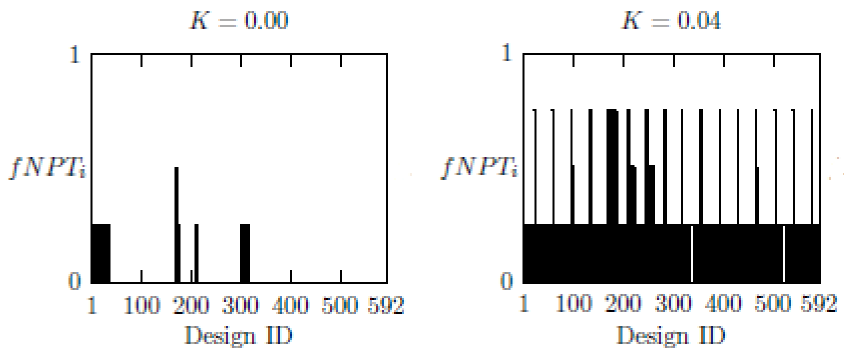

Fig. 3. Fuzzy Normalized Pareto Trace by design for K=0 and K=0.04

Another way to view the previous analysis is as a means to layer changeability onto the most passively robust designs identified through the fuzzy Pareto Tracing. The goal of layering changeability onto the most passively robust designs is to achieve a high effective value robustness, one that leverages both passive and active robustness. Modularity is an enabler for changeability, with the implication being that one can to link together the mostly passively robust designs into a completely robust (passive and active) design through component modularity. This is the same as attempting to find or synthesize, through component modularity, designs with an effective fuzzy Normalized Pareto Trace, efNPT equal to one[26].

3.3. Modularity Analysis

At this point in the approach we have a set of potentially valuable designs, as well as insights into which design variables exhibit variance across these designs. If this were a product for commercial markets, the providing organization might want to offer all the designs to the market as options for different consumers (for example: "good, better, best" product options[27]). Or, the scenario in question could be for a single decision maker that desires a level of skid steer functionality at some time t = 0 but could foresee needing a higher level of functionality at t > 0. An alternative scenario could be that a single tractor decision maker is in epoch 1 (30 inch rows) and may want to switch to epoch 2 (24 inch rows) in the future. With each of these change scenarios we want to understand the relative effort of supporting different designs and design changes inside of the given product architecture. Since the design variables may need to be changed, the next step focuses on answering the question, "What might component modularity tell the designer about the design's ability to support these changes?"

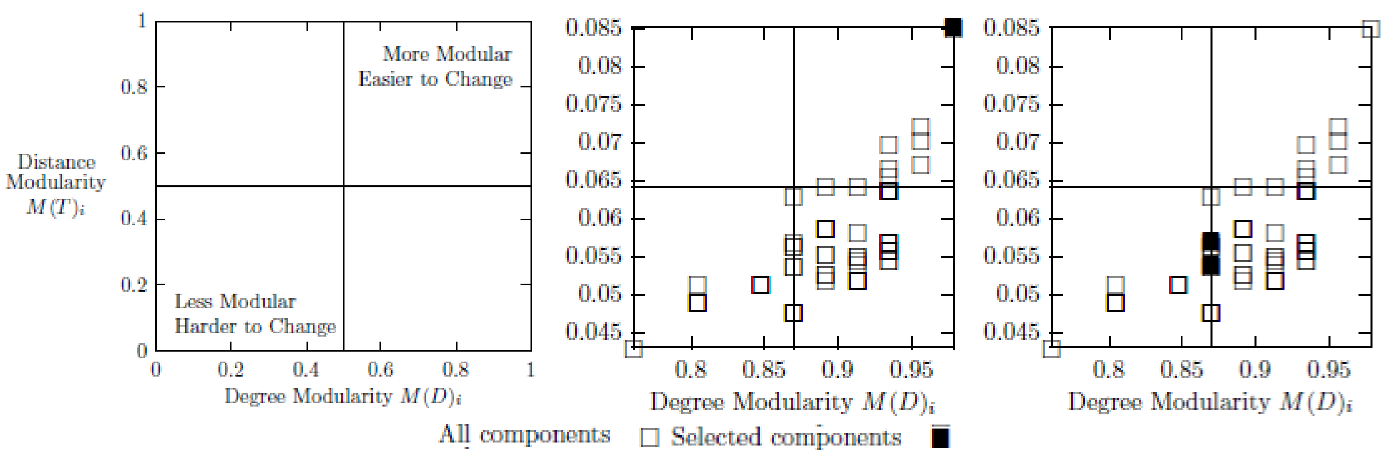

Fig. 4. Description of M(T) vs. M(D) (left), modularity of attachment modules (e.g. bucket) (center), and frame modules (e.g. tractor width) (right)

Returning to our modularity metrics, we can inspect the modularity of the selected components associated with particular design variables (Fig. 4). Those components that are more modular can support easier change. Inspecting all four design variables resulted in identification of two types of modules: *attachment modularity* - readily changed and executable during use or production, and *frame modularity* - harder to change and executable during production only. These modularities enable change paths in the tradespace, allowing a design to "change" into another. As an example, allow us to start with design 22. With modularity path enablers in place for changing the attachment width and changing the width of the product, one could transition from design 22 to the remainder of the designs in Table 5 through executing the modularity path enablers.

Table 5. Example designs reachable through modularity path enablers

Design ID	Hydraulic/Engine Power (HP)	Vehicle Width (in)	Bucket Width (in)	Reachable Via	Reachable Phase[b]
22	25	72	56	*Baseline*	
170	25	72	72	*Attachment Mod*	*Production or Use*
281	25	72	84	*Attachment Mod*	*Production or Use*
318	25	90	56	*Frame Mod*	*Production only*
466	25	90	72	*Attachment and Frame mod*	*Production or Use*
577	25	90	84	*Attachment and Frame Mod*	*Production or Use*

From this shorter list of designs, we will inspect the utilities to determine what designs could be offered into the target markets. Beginning with the construction epochs, the utilities are given in Table 6 for each of the designs from Table 5. This list of designs is sorted in descending order of utility in epoch 1, to make the "good, better, best" offerings easier to discern. The "good, better, best" determination was done via inspection as an example, but other determinations may be made.

Table 6. Product offerings for construction epochs (epochs 1 and 2)

Design ID	Hydraulic/Engine Power (HP)	Vehicle Width (in)	Bucket Width (in)	Utility: Construction - Epoch 1	Epoch 2	Product Offering
577	25	90	84	0.079	0.645	
170	25	72	72	0.060	0.662	Best
281	25	72	84	0.031	0.579	Better
22	25	72	56	0.016	0.341	Good
466	25	90	72	0.015	0.512	
318	25	90	56	0.004	0.191	

The same information can be used for determining the offerings in the agricultural sector. The list of designs is shown in Table 7 with the utilities for the agricultural epochs. Of note is that in each of epoch 3 and epoch 4 the acceptable designs have the same utilities. Upon inspecting the tradespaces for these epochs, we find that utility is a function of the overall engine power. Because the engine power is not varied in the considered designs, and a modularity path enabler for engine power was not in place, there is only one product available to users for each of epochs 3 and 4. The producing entity might now decide to make the cost/time trade-off decision to put an engine power modularity path enabler in place so that it could offer a wider selection of products to target markets.

Table 7. Product offerings for agriculture epochs (epochs 3 and 4)

Design ID	Hydraulic/Engine Power (HP)	Vehicle Width (in)	Utility: Agriculture Epoch 3	Epoch 4	Product Offering
22, 170, 281	25	72	*unacceptable*	0.809091	Yes, all three
318, 466, 577	25	90	1	*unacceptable*	Just 318

3.4. Synthesis into Platform with Variants

Now that four designs 22, 170, 281, 318, have been selected as offerings to two different markets, they can be tied together into an overall product line. Design 22 could be considered the base "platform" by the supplier; design 22 could then be transitioned to the other designs via modularity in production or use of either the frame or the attachment. The family offering of products, based on design 22 as the base platform, is shown graphically in Fig. 5. Also shown is an indication of which variants would be offered to which use types: construction, agriculture, or both. In the case of designs 22, 170, and 281, if a construction customer acquired any of these three designs then they could transition to any of the other two designs via modularity in attachment during the use phase. In the case of designs 22 and 318, if an agricultural customer acquired either of these they could not transition to the other design because modularity in the frame is only available at the time of production; design 22 would be acquired for epoch 4 needs and design 318 would be acquired for epoch 3 needs.

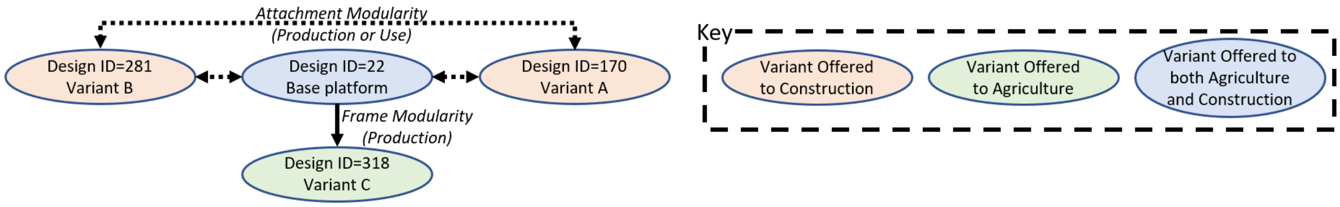

Fig. 5. Design variants for different target users connected via modularity-enabled paths

4. Discussion

While only qualitatively considered, a connection was made between component modularity and system path enablers and transition paths. If one is starting with an existing design, the modularity metrics could be used to indicate which modularity path enablers are present and to what degree, providing higher fidelity as compared to binary existence of modularity. Additionally, these opportunities for targeted modularity can be used to identify families of systems that are, together, value robust across multiple epochs. As an example of the benefits of the proposed approach, the results of the tradespace study and modularity analysis were combined to demonstrated how a complete product line could be developed and offered to the market based on the results of the approach. Finally, this approach has utilized RSC to provide a value-centric focus to modularity efforts. Instead of relying on experiential and tacit knowledge, this approach can focus designer's modularization efforts on elements of the system that may need to change to accommodate changes in decision makers' needs or use contexts, whether the system is an evolution or entirely new.

References

1. Yu T, Yassine AA and Goldberg DE. An information theoretic method for developing modular architectures using genetic algorithms. *Res. in Eng. Des.* 18:91–109, 2007.
2. Bartolomei JE. *Qualitative Knowledge Construction for Engineering Systems: Extending the Design Structure Matrix Methodology in Scope and Procedure.* PhD diss. MIT: Cambridge, 2007.
3. Ross, AM. *Managing unarticulated value: changeability in multi-attribute tradespace exploration*. PhD diss. MIT: Cambridge, 2006.
4. Allen T, Mcgowan D, Moses J, Magee C, Hastings DE, Moavenzadeh F, Nightingale D, Little J, and Roos D. *ESD Terms and Definitions*. MIT: Cambridge, 2002. URL http://esd.mit.edu/WPS/esd-wp-2002-01.pdf.
5. Ulrich K. The role of product architecture in the manufacturing firm. *Research Policy* 24:419–440, 1995.
6. Baldwin CY and Clark KB. *Design Rules, Vol 1: The Power of Modularity*. MIT Press, Cambridge, MA, 2000.
7. Sosa ME, Eppinger SD, and Rowles CM. A Network Approach to Define Modularity of Components in Complex Products. *J. of Mech. Des.*, 129(11), 2007.
8. de Weck O, Roos D, and Magee C. *Engineering Systems : Meeting Human Needs in a Complex Technological World*. MIT Press: Cambridge, 2011.
9. Johnson HT and Broms A. *Profit Beyond Measure: Extraordinary Results through Attention to Work and People*, 2000.
10. Browning, TR and Eppinger SD. Modeling impacts of process architec- ture on cost and schedule risk in product development. *IEEE Trans. on Eng. Mgt.* 49(4):428–442, 2002.
11. McManus HL and Hastings DE. A framework for understanding uncertainty and its mitigation and exploitation in complex systems. *IEEE Eng. Mgt. Rev.* 34(3):81–94, 2006.
12. Ross AM. Defining and Using the New "ilities". Working Paper 2008-4-1. MIT: Cambridge, 2008.
13. Whitney D. Physical limits to modularity. *MIT ESD Symposium 2003*, Cambridge, MA 2003.
14. Holtta-Otto K and de Weck O. Degree of Modularity in Engineering Systems and Products with Technical and Business Constraints. *Concurrent Eng.* 15(2):113–126, 2007.
15. Holtta-Otto K. *Modular product platform design*. PhD diss. Helsinki Univ. of Tech: Helsinki, 2005.
16. Maier MW and Rechtin E. *The Art of Systems Architecting*. CRC Press, 3rd edition, 2009.
17. Stone, RB, Wood KL and Crawford RH. A heuristic method for identifying modules for product architectures. *Design Studies* 21(1):5–31, 2000.
18. Ross AM, McManus HL, Rhodes DH, Hastings DE and Long AM. Responsive Systems Comparison Method: Dynamic Insights into Designing a Satellite Radar System. AIAA Space 2009 (SPACE09). Pasadena, CA, 2009.
19. Roark III HH. *Value centric approach to target system modularization using multi-attribute tradespace exploration and network measures of component modularity*. SDM thesis. MIT: Cambridge, 2012.
20. Open-Ecology-Project. "LifeTrac". URL http://opensourceecology.org/wiki/ LifeTrac, 2011.
21. Open-Ecology-Project. "Open Source Ecology License". URL http://opensourceecology.org/wiki/License, 2011.
22. Open-Ecology-Project. "LifeTrac III (SolidWorks parts & assembly)". URL http://openpario.mime.oregonstate.edu/attachments/download/4304/ LifeTrac_III SolidWorks_partsassembly_.zip, 2011.
23. Open-Ecology-Project. "LifeTrac Bill of Materials". URL http://opensourceecology.org/wiki/LifeTrac/Bill_of_Materials, 2011.
24. SmallEngineSuppliers.com. "Briggs and Stratton Vertical Engines Retail Prices". URL http://smallenginesuppliers.com/shop/html/pages/briggs_ vertical_shaft_engines.html, 2011.
25. Ross AM, Rhodes, DH and Hastings DE. Using pareto trace to determine system passive value robustness. 3rd IEEE Sys Conf. (SysCon09). Vancouver, 2009.
26. Fitzgerald ME, Ross AM and Rhodes DH. Mitigating contextual uncertainties with valuable changeability analysis in the multi-epoch domain. 6th IEEE Sys Conf. (SysCon12). Vancouver, 2012.
27. Meyer MH and Lehnerd AP. *The Power of Product Platforms*. The Frem Press: New York, 1997.