Proceedings of DETC '02
ASME 2002 International Design Engineering Technical Conferences
28th Design Automation Conference
Montreal, Canada, September 29-October 2, 2002

DETC2002/DAC-34119

ARCHITECTURAL OPTIMISATION USING REAL OPTIONS THEORY AND DEPENDENCY STRUCTURE MATRICES

David M. Sharman, Ali A. Yassine+, Paul Carlile
Massachusetts Institute of Technology
Cambridge, MA 02139

ABSTRACT

This paper outlines a methodology for optimising the multi-domain architecture of a relatively integrated system through an appropriate level of modularisation to maximise societal value created. This method is developed through the application of real options theory and the dependency structure matrix (DSM), and illustrated using a reference example of an industrial gas turbine.
(Keywords: Dependency Structure Matrix (DSM), Real Options Theory, Modular, Integral, Architecture, Domain, Optimisation)

1 INTRODUCTION

Eppinger & Salminen (2001) identified the existence of three interdependent domains of interaction in product development processes. The three domains investigated are: product, process, and organisation. In this paper, we propose a method for optimising the multi-domain architecture of a system to maximise societal value created using a reference example of an industrial gas turbine of ~10MWe. The optimisation of the reference example takes place in the physical domain, but the paper closes by explaining how the process may be implemented across multiple domains.

Various architectural terms used in this paper are explained in the companion paper "Characterising Modular Architectures" (Sharman et al., 2002). For an explanation of Dependency Structure Matrices, see http://mit.edu/dsm/. This paper extends and applies the theoretical work on modular design rules (Baldwin & Clark, 2000) and the resultant simple modular case study (Sullivan et al, 2001) to more complex and ambiguously non-hierarchical architectures that span multiple domains. Most of this paper takes the societal perspective rather than that of an actor such as an individual firm.

This topic is important to designers of complex systems that may evolve substantially but not entirely predictably, e.g. an automobile or gas turbine. In such systems, the way in which the design is split up into more or less 'modular' subsystems may either facilitate or impede future technological innovation to improve aspects of the system's performance (it's 'value'). This paper does not deal with the very relevant topic of who is able to capture the system value (e.g. actors such as the customers, the system integrator, or sub-system providers) and instead uses the intermediate notion of a holistic 'societal value', but does lay out basic concepts that allow discussion of this important issue.

The rest of the paper proceeds as follows. The next section introduces briefly Baldwin and Clark's (2000) work on applied real options theory and modularity. Section 3 applies the results of Section 2 to the valuation of a gas turbine and then we recalculate this valuation based on improved visibility data based on information contained in an uncluttered DSM model. An extension to the options value method that allows the valuation of a clustered DSM is presented in Section 4. Sections 5 and 6 introduce the concepts of multi-domain DSMs, hypothesised inter-domain relationships, and multi-domain optimisation. Finally, we conclude the paper in Section 8 and present possible extensions and our future work.

2 APPLIED REAL OPTIONS THEORY

In this section, the relevant parts of the theory of design rules (Baldwin & Clark, 2000) - referred to by B&C from this point on - will be outlined insofar as they pertain to the calculation of option values for elements.

+ Corresponding author: MIT, CTPID, Rm. E40-253, Cambridge, MA. 02139.
Tel. (617)258-7734, Fax (617)452-2265, E-mail: yassine@mit.edu

This theory proposes that design and industry evolution is an example of a complex adaptive system. The essence of the theory is that modular designs evolve via a decentralised search, by many designers, for valuable options that are embedded in the following six modular operators:

- Splitting a system into two or more modules;
- Substituting one module design for another;
- Augmenting – adding a new module to a system;
- Inverting to create new design rules;
- Porting a module to another system.

B&C propose that when a fully modular design emerges the option space increases and the societal value increases. They propose that a modular design process has three basic stages: (1) Stage 1: The formulation of design rules, (2) Stage 2: Parallel work on hidden modules, and (3) Stage 3: Testing and integration. This permits economic analysis of nested modular architectures by calculating the net option value of a probabilistic (redesign) payoff function that is complexity dependent, minus a series of costs. In the simplest form with no costs incurred, the overall value (performance[1]) of the system, X_{system}, is split into a system level value measure, S_0, and j modules' value measures ($X_1, \ldots, X_j$) :

$$X_{system} = S_0 + \sum_{i=1}^{j} X_i \qquad \text{(B\&C p.253)}$$

Where:
X_{system} represents the value of a system consisting of j modules,
S_0 is considered to be a constant, and
X_i is the contribution of the i^{th} module to overall system value.

The module value for a given design is assumed to be normally distributed with zero expectation and a variance proportional to complexity. Therefore, the expected value (V) of a one-module design is:

$$V_1 = S_o + E(X_N^+) \qquad \text{(B\&C Eq. 10.1)}$$

Where:

$$E(X^+) = \int_0^\infty X f(X) dX ,$$

X is normally distributed with mean 0 and variance $\sigma^2 N$,
X^+ means that expectation only applies to outcomes above 0,[2]
$f(X)$ is the probability density function of a normal distribution,
N denotes the total number of tasks.

Then the design is split into j independent modules whilst keeping the total number of tasks at N. The expected value V_j of this modular design is:

$$V_j = S_o + E(X_1^+) + .. + E(X_j^+) \text{ (B\&C Eq. 10.2)}$$

The effect of splitting the design into modules is to increase the design's expectation total value as any new module that is better than the existing module will be incorporated into the total design, whilst new modules that are worse will be culled. This implies that for a given distribution of outcomes for X, increased modularity increases value.

A system of N tasks that is equally (symmetrically) split into j modules with normally distributed outcomes, such that there are N/j tasks per module, has a value V_j that can be expressed in terms of the value of the one-module design, V_1: $V_j = j^{1/2} V_1$, where S_0 is normalised to zero to ease comparison. This square root relationship of declining (but ever-increasing) returns from increased modularisation can be generalised to the case of asymmetric modules. If X_α is the performance of a module size αN in a system of N tasks split into j independent modules with normally distributed outcomes, then the asymmetric modular design can again be expressed in terms of the value of the one-module design, V_1.

$$V_j = (\alpha_1^{1/2} + \ldots \alpha_j^{1/2})\ V_1\ ;\ \sum_{i=1}^{j} \alpha_i = 1 \text{(B\&C p263)}$$

Now B&C introduce the concept of attempting more than one new design per module so that there are j modules and k independent (parallel) design efforts (trials or experiments) per module. The value V of the design can then be written:

$$V(\tilde{X}_1..\tilde{X}j) = S_o + Q(\tilde{X}_1; k_1) + .. + Q(\tilde{X}_j; k_j) \text{(B\&C Eq. 10.4)}$$

Where $Q(\tilde{X}_j; k_j)$ is the value of the best of k designs for payoff function $\tilde{X}$, provided it is better than zero.

Once again the assumption is made that the payoff function is a standard normal distribution, and $\tilde{X}$ is suppressed to reduce clutter so the value of the best of k designs is denoted as $Q(k)$ where $Q(1)=E(X_1^+)=0.3989$ as in B&C, Equation 10.1.

For the case where modules are symmetric the value $V(j,k)$ of the design process is:

$$V(j,k) = S_0 + \sigma(N_j)^{1/2}.Q(k) \qquad \text{(B\&C Eq. 10.5)}$$

At this point B&C introduce the concept of costs in order to account for the investment required for the three basic stages of: formulate the architectural design rules, create alternative module designs, and test and integrate the resultant systems. They make three assumptions:

- Stage 1 costs of formulating design rules are proportional to the number of modules ($c_j j$)
- Stage 2 costs of experimentation are proportional to the number of experiments ($c_k k$)
- Stage 3 costs of testing & integration are proportional to the numbers of modules and experiments ($T(j,k)$)

So the total cost of a modular multi-experiment process is:

[1] Value and performance are used interchangeably at this stage in the development of B&C. Later on value becomes detached from performance: when market considerations cause cost to diverge from price.
[2] This is because new designs with values of less than 0 are 'worse' than the current design and so are discarded (culled).

$$C(j,k) = c_j j + c_k k + T(j,k) \quad \text{(B\&C Eq. 10.6)}$$

This yields a formula for the ***net option value*** (NOV) for the combined operators of splitting and substitution:

$$NOV(j,k) = S_0 + \sigma(N_j)^{½}.Q(k) - c_j j - c_k k - T(j,k) \quad \text{(B\&C Eq. 10.7)}$$

The S_0 term can generally be omitted as it represents a base value that need never be forgone as unworthy module designs can be culled. The stage 1 costs differ from the stage 2 costs in that the stage 1 costs need only be incurred once (provided the architecture survives for more than one generation) whilst the stage 2 costs recur every generation.

Whereas the costless function (B&C Eq. 10.5) exhibited ever-increasing returns from ever-finer modularisation and increased numbers of experiments (albeit at a decreasing rate), the convex cost-laden trade-off function (B&C Eq. 10.7) describes a trade space with a single 'best' solution from a societal perspective. Other cost relationships can yield multi-peaked solutions e.g. if there is a fixed cost involved in creating a modular design.

B&C propose a two level hierarchy for analysing testing costs: system level tests where modules are embedded in prototypes, and module level tests that rely on principles to allow independent testing of module performance using appropriate metrics. The extreme form of the system level test suffers from combinatorial explosion as, if the cost per test[3] is c_{ts}, then the overall cost $T_s(j,k)$ of testing j modules and k experiments per module is:

$$T_s(j,k) = [(k + 1)^j - 1]\ c_{ts} \quad \text{(B\&C Eq. 10.8)[4]}$$

This overall cost increases dramatically with either j or k. Even reducing the system level testing costs by a factor of one million does not alter the message: combinatorial explosion of system testing costs dramatically reduces the attractiveness of either further modularisation, or further experimentation, or both. Whilst further experimentation can be easily prevented, it may be this explosion result that causes system designs with ~10+ modules to be notable by their rarity (i.e. the 7 ± 2 rule).

As design rules become understood sufficiently that modularisation becomes conceivable, it becomes possible to design tests for modules that do not rely on embedding them in prototypes. B&C point out the practice shifts from "*build and test the system*" to "*model the system and test the modules*". At this level of maturity, for a testing cost per module of c_{ts}, the overall cost $T_m(j,k)$ of testing j modules with k experiments per module is:

$$T_m(j,k) = j\ .\ k\ .\ c_{tm} \quad \text{(B\&C Eq. 10.9)}$$

This outflanks the combinatorial explosion problem of system level testing costs, allowing for higher levels of modularity and experimentation to become viable and desirable. This is a dynamic process as successive generations of designers gain knowledge about how to architect, design, and test, thereby reducing costs and increasing the optimal numbers of modules and trials. This evolution requires investment in:

1. design rules to establish the architecture and interfaces;
2. independent experiments to explore the possibilities inherent in the design;
3. module-level tests to identify superior combinations efficiently.

Not all modules are equal. But if module values are additive the net option value is likewise additive, so for a system of j modules the net option value of the total system, NOV_{tot} is:

$$NOV_{tot} = S_0 + NOV_1 + \ldots + NOV_j \quad \text{(B\&C Eq. 11.1)}$$

And the individual terms expand:

$$NOV_i = \max\left\{\sigma_i(n_i)^{1/2}Q(k_i) - C_i(n_i)k_i - Z_i\right\} \quad \text{(B\&C Eq. 11.2)}$$

Where: n is the number of tasks to redesign the module,
σ is the standard deviation of potential value, and
Z is the visibility[5] of the module to the system.

Given that the main variables in equation (B&C Eq. 11.1 and 11.2) are module size and visibility it is instructive to view the results of calculating the incentives for four different modules – small or large, hidden or visible, shown in Figure 1. This suggests that smaller modules will attract more investment (which is not surprising), as well as hidden modules (which is less obvious). Moreover, the degree of differentiation in attractiveness goes some way towards explaining the degree of competition in various parts of the design space in open and efficient markets, without having had to create a specialised expression. Note how some combinations destroy value as benefits fail to meet costs.

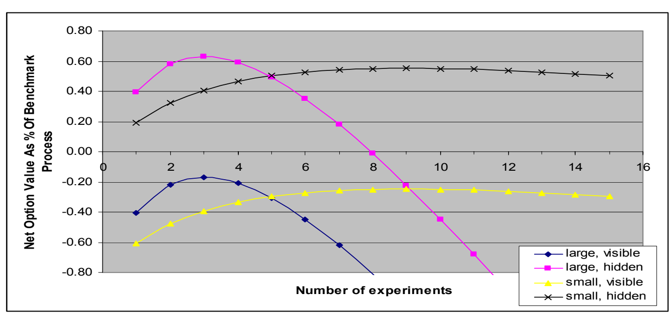

Figure 1: Module Net Option Value For Different Combinations Of Size And Visibility (B&C)

[3] We depart slightly from the B&C notation of using c_t to denote both system level or module level testing costs. Instead we discriminate by using T_s, c_{ts} and T_m, c_{tm} to represent system and module level test costs respectively.

[4] At first glance one might expect this to be $T_s(j,k) = k^j.\ c_{ts}$ but not only are there k experiments per module there are the combinations which include the prior generation's 'seed' module to be tested as well, i.e. $(k + 1)^j$ combinations – of which the value of one combinatorial package is already known.

[5] B&C define visibility as being "the number of other modules that 'see' the visible information contained in it".

An extension of this explain why multi-generation evolution can start off with results that are worse than the zero'eth generation, then add value for a series of generations, and finally lose value by failing to return their costs. This is analogous to the curves shown in Figure 1, except that the x-axis becomes successive generations and the driver is now the learning process that skews the distribution of outcomes. The initial losses are the result of having to cope with restrictive legacy design rules so as to ensure backwards compatibility.

3 BASE VALUATION OF A SYSTEM OF ELEMENTS

In this section, we apply the net value option results, presented in Section 2, to a complex product with varying technical potentials. In doing so, the 'σ' or potential value term may be adjusted to reflect the varying technical potentials of the elements (or modules). The results of this exercise illustrate how heterogeneous the value landscape can become with even simplistic models.

A 10MWe industrial gas turbine (genset, for short) was crudely decomposed into 31 elements as shown in Figure 2 (Sharman, 2002). For each of these elements the size, technical potentials, visibility, etc. were estimated (Sharman, 2002). These point estimates were then plugged into the valuation formula for a heterogeneous system (B&C Eq. 11.1 updated to reflect 10.7 and 10.8) to obtain the curves shown in Figure 3.

3.1 Improved Valuation Using DSM Visibility

A parameter that strongly influences an element's valuation is its visibility, as this determines the amount of costly testing. The valuation shown in Figure 3 was based on a point estimate of an element's visibility. In order to reduce the error in estimating visibility, a dependency structure matrix (DSM) for the observed 31 elements was constructed (se Figure 4) by setting a weighted relationship denoting a material influence from one element to another on a four-point linear scale to denote the relationship strengths (Sharman et al., 2002). This is termed the physical domain DSM.

An element's visibility can be assessed by summing element outputs vertically (Sharman et al., 2002). After normalising the results, this simple visibility was used to recalculate the element valuations. The results are very similar to those in Figure 3 and exhibited the same four characteristic types of valuation curve as discussed above.

Whilst minor changes have occurred, in this instance the genset's architecture was sufficiently well understood that the point estimates of visibility correlated fairly well with the more sophisticated DSM assessment of visibility.

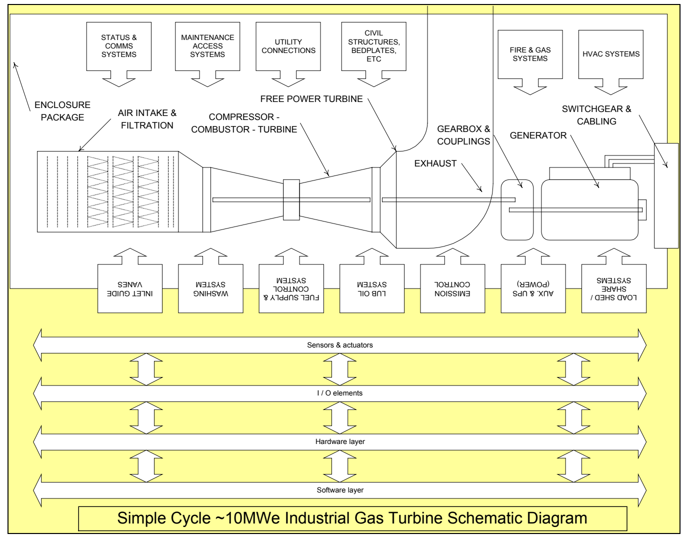

Figure 2: Schematic of Gas Turbine (i.e. genset)

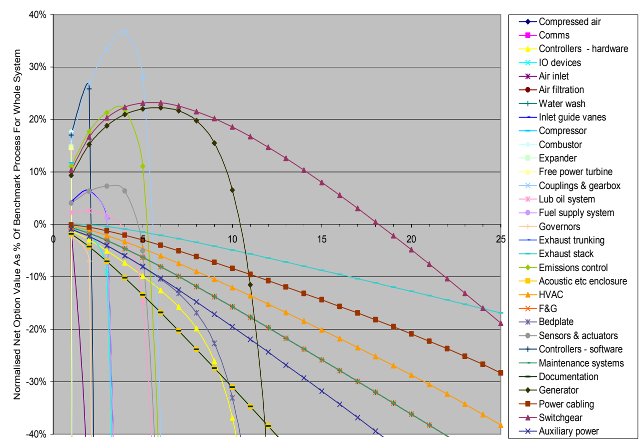

Figure 3: Initial Valuation Of Genset's 31-Elements[6] (see a larger version of this figure in the Appendix)

[6] Inspection of these curves provides insightful strategic implications (Sharman, 2002). The nature of these implications depends on the perspective of the viewer (e.g. different firms will respond differently to different characteristic curve types). In this paper, the perspective we use is that of a society as a whole and therefore the sum of the maximal net value is the relevant metric. For further exploration of this note, see Sharman (2002).

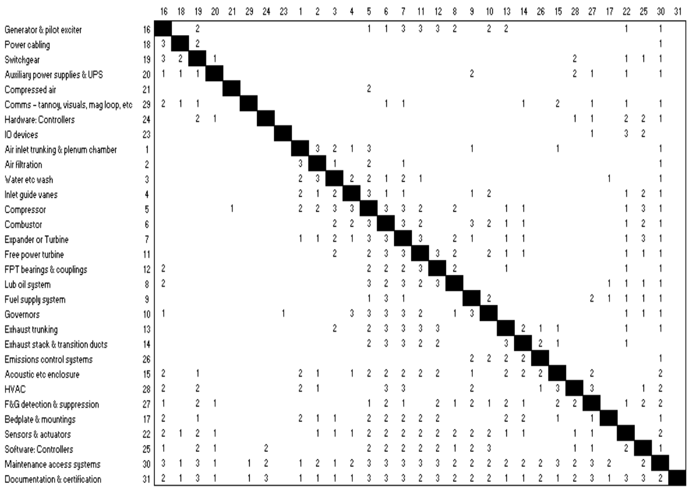

Figure 4: 31-Element Physical Domain DSM (see a larger version of this figure in the Appendix)

3.2 Architectural Valuation Using DSM Visibility

By taking the optimum number of trials for each element, and summing the maximum value yielded for each of the 31 elements, the value of this architecture can be condensed to a single metric, the net system value of 1.87.[7] At this stage, the architecture is defined by the way the total system (comprising tens of thousands of parts) has been grouped into only 31 elements and a different value would have been achieved if those thousands of parts had been grouped into 49 or 81 elements. In this paper we are concerned with effects that occur at this level of decomposition, but in a more realistic valuation it may be necessary to drill to lower levels of decomposition.

Item	**Optimum number of trials**	**Result**
Inlet guide vanes	3	0.07
Compressor	1	0.11
Combustor	1	0.13
Etc.	Etc.	Etc.
System Value		**1.87**

Table 1: Net Option Value Of 31-Element System

3.3 Optimal Number of Modules and Experiments

Until now the system has been assumed to comprise the 31 heterogeneous elements with a basic scaling assumption that the integrated design broke even; so, all what we really did was to allocate the value amongst the heterogeneous sub-systems. However by assuming homogenous sub-systems and holding all other assumptions constant it is possible to investigate what is the optimal mix of modularisation and experimentation with the anticipation that it would be considerably less than 31. This is a somewhat artificial exercise since there are in fact technical constraints to decomposition into an arbitrary number of modules but it should give us a better directional understanding of the system. The results in Figure 5 indicate that 3 experiments on a system of ~25 modules is optimal. The number of experiments appears directionally sound – numbers from 2-6 are indicative of healthy competition - however a result of 25 modules is somewhat odd as in the heterogeneous system above only 8 modules exhibited positive value if 3 experiments are run.

The reason this somewhat optimistic outcome occurs is that the costs of splitting are so low. These costs are both the front-end ones of adjusting the design rules and the back-end ones of testing and integration. Whilst the back-end ones have been covered quite exhaustively, the front-end costs bear a little more attention.

The current cost formula for creating the design rules is:

$$\textit{Cost of formulation} = (c_j \,.\, j) + c_{j\text{-fixed}}$$

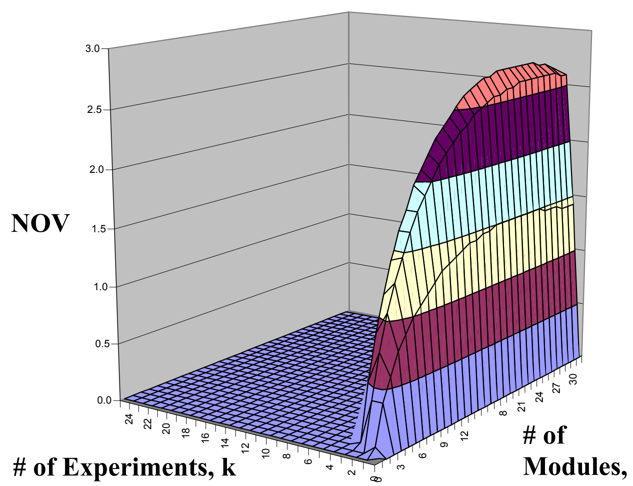

Figure 5: Optimal Number of Modules and Experiments for the Gas Turbine

This underestimates the cost associated with splitting a module. For a module to exist it must have codified design rules that describe its relationships with all other modules. When it is split each of these relationships must be reassessed, **especially if there are significant feedback relationships as in the case of an integrated or semi-integrated product**. If there are j modules in the new design there are $j\text{-}1$ relationships to be considered. Each of these new or amended relationships must also be considered for its effect on the other relationships, i.e. $(j\text{-}1)^2$ combinations of which one can be discounted as it is the sundered module's relationship to itself.

$$\textit{Cost of formulation} = ((j\text{-}1)^2 - 1) \,.\, c_j) + c_{j\text{-fixed}}$$

Updating the net option value calculation yields the very different picture of Figure 6.

This suggests that 2 experiments on 4 modules would maximise value with a maximum of 4 experiments on 6 modules being profitable. This precise outcome is debatable.

[7] So, the value of a design of 31 modular elements is 1.87 times the value of a design of an unmodularised design consisting of a single element.

However, the basic point that design costs are proportional to j^2 rather than to j makes a big difference irrespective of the details. A more subtle equation would weight the power term to correspond to the population density of the DSM.

This suggests that 2 experiments on 4 modules would maximise value with a maximum of 4 experiments on 6 modules being profitable. This precise outcome is debatable. However, the basic point that design costs are proportional to j^2 rather than to j makes a big difference irrespective of the details. A more subtle equation would weight the power term to correspond to the population density of the DSM.

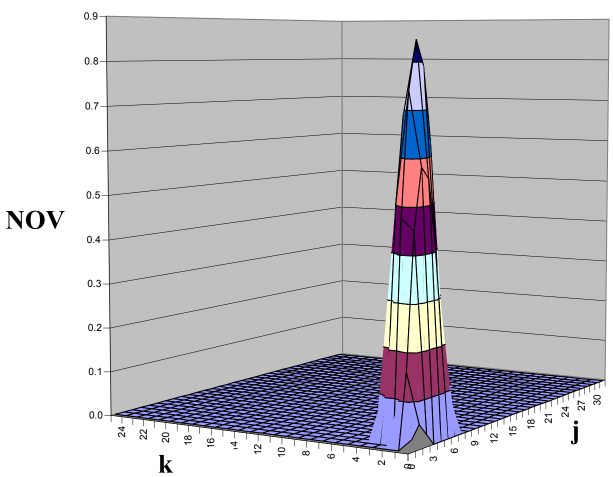

Figure 6: Revised Optimal Number of Modules and Experiments for the Gas Turbine

4 VALUATION OF A SYSTEM OF MODULES

4.1 Identification Of Modular Clusters

Modular clusters were identified by a semi-manual process and shown in Figure 7 (Sharman et al., 2002) that resulted in the following DSM sequence and cluster boundary. Although this particular interpretation is not necessarily the optimum, it appears qualitatively encouraging.

4.2 Visibility; Rules; Boundaries

In order to improve the system value it is necessary to adjust the determining factors such as element size, technical potential, and visibility. This can be done by combining highly related elements into modules following a series of rules (Sharman et al., 2002). The basis of these rules is that information regarding relationships between elements within a module is encoded in the module's design rules, which are graphically depicted by the module's boundary.

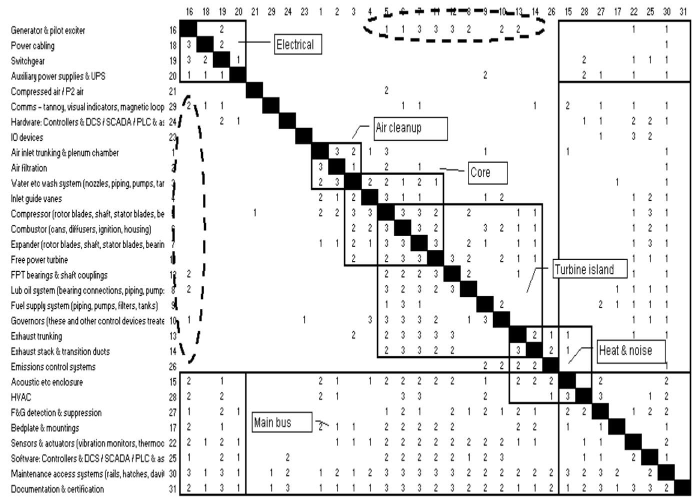

Figure 7: Clustered DSM of Gas Turbine (see a larger version of this figure in the Appendix)

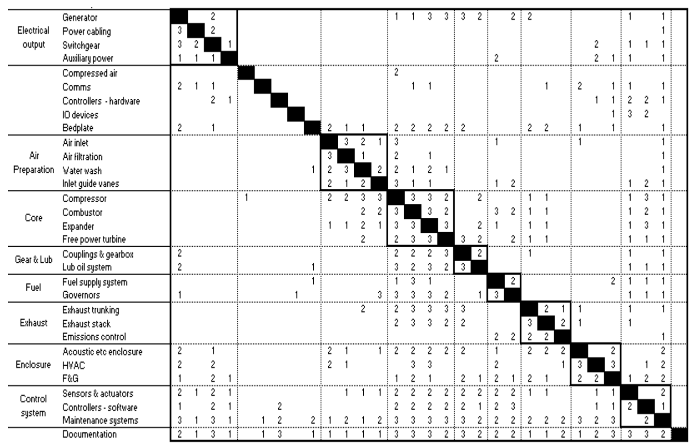

Figure 8: Unpinned Modules in Genset DSM (see a larger version of this figure in the Appendix)

In order to ease the computation process for this example the modules were first refined to the unpinned (i.e. non-overlapping clusters) form of Figure 8 even though this qualitatively sacrifices some of the value. The following simplified rules were then applied.

4.2.1 External Visibility

The simple sum in a vertical direction of all stage two relationships **outwith the module boundary**. This approach maybe problematic; however, other more subtle approaches are discussed further in Sharman et al. (2002).

4.2.2 Module Size

The product of the individual element sizes is used, multiplied by a fill-dependent scale factor that is defined as:

*Scale factor = Total internal visibility / (number of non-diagonal elements * max relationship weight)*

Where the total internal visibility is the sum of all relationships within the module boundary. This is not a perfect module size equation as it can end up lower than the sum of the individual element sizes, and makes little account of any additional rule costs incurred at this intermediate level of decomposition. It can be used as it stands with a minimum value set as the sum of the elements, or a more fundamental formulation introduced.

The result of applying these first two rules is depicted by collapsing the 31-element DSM to a 14-element DSM comprising six singleton elements and eight modules as shown in Figure 9.

4.2.3 Standard Deviation (Technical Potential)

Where there are N elements in the module the following formula is used:

$$\sigma_{module} = \sqrt{}\,(\sigma^2_{element\ one} + \sigma^2_{element\ two} + \ldots\ \sigma^2_{element\ N})$$

This notes that the element standard deviation already in use is weighted to account for technical potential.

4.2.4 Coefficients

To ease comparison the same co-efficients are used for the 31-element calculation that resulted in an NOV of 1.87. The value of the basic process [V(1,1)] is not allowed to float freely and is held constant at 1.0 through a normalisation process that adjusts for the fluctuating number of modules. This is because V (1,1) should always represent the value of one experiment on the unmodularised system and since the underlying system remains constant at 31-elements this should not change.

	Core	Compressed air	Comms	Controllers - hardware	IO devices	Bedplate	Electrical output	Air preparation	Gear & Lub	Fuel	Exhaust	Enclosure	Control system	Documentation
Core	33	1						21	9	8	8		17	
Compressed air	2													
Comms	2						4				1	3	2	
Controllers - hardware							3					2	5	
IO devices												1	5	
Bedplate	8						3	4	2		4	2	1	
Electrical output	8						16		5	4	2	5	8	
Air preparation	17					1		22		4		1	7	
Gear & Lub	19					1	4		5		1		5	
Fuel	16				1	1	1	3	1	5		2	6	
Exhaust	21							2	5	4	12	2	4	
Enclosure	18						11	7	5	6	10	12	10	
Control system	28		1	4		2	18	9	12	13	8	12	7	
Documentation	11		1	3		1	7	4	5	4	4	6	8	

Figure 9: Collapsed Form 14-Element DSM

4.3 Valuation

Until now, the valuation assumed the system is composed of 31 heterogeneous elements, which resulted in a system net option value of 1.87 per Table 1. To illustrate how the same valuation might be applied at a higher level of decomposition, the DSM (shown in Figure 9) is valued in Figure 10 and Table 2.

The valuation of the unclustered 31 individual elements was 1.87 and the valuation of the same 31 elements after clustering is 4.31. This increase in value indicates that the clustering identified qualitatively has hidden sufficient detail that the increased costs of the larger modules are outweighed by the increased benefits. This system valuation makes a good objective function for an optimising clustering algorithm. It is quite likely that a good search strategy would reveal an even better solution (a higher valuation) than the one identified in the example above.

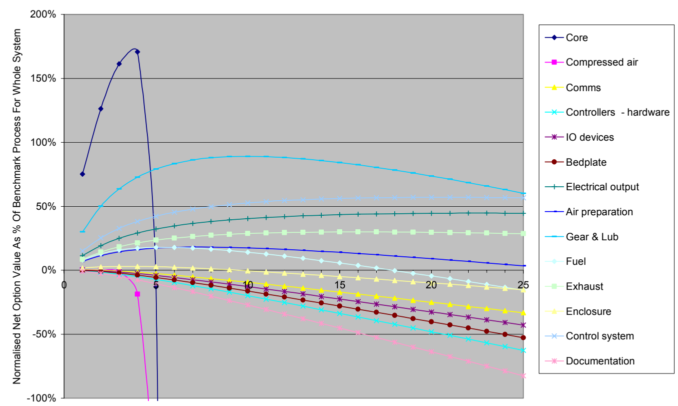

Figure 10: Gas Turbine Option Value Curves for Second Level Decomposition (see a larger version of this figure in the Appendix)

Item	Number of elements	Optimum number of trials	Maximum NOV_{module}
Core	4	4	1.71
Compressed air	1	0	0
Comms	1	0	0
Control hardware	1	0	0
IO devices	1	0	0
Bedplate	1	0	0
Electrical output	4	25	0.45
Air preparation	4	7	0.18
Gear & Lub	2	10	0.89
Fuel	2	6	0.18
Exhaust	3	16	0.30
Enclosure	3	3	0.03
Control system	3	21	0.57
Documentation	1	0	0
System Net Option Value			**4.31**

Table 2: Revised Gas Turbine Option Value

This shows how it has been possible to relate the organisation of a relatively integrated product or system's architecture to its expectation value based on the fundamental principles of option value and design rules. Whilst some simplifications and approximations have been made the principles of valuation as suggested by B&C have been respected and extended to an integrated architecture.

5 HYPOTHESISED INTER-DOMAIN RELATIONS

Eppinger and Salminen (2001) discuss the possible relationship between DSMs in the physical, task, and organisation domains as illustrated in Figure 11. To the best of our knowledge, no one system has been traced exhaustively through all three domains; so, any conclusions are preliminary. The data set on which these observations are made is bounded by the observed physical domain elements. For example, a study compared the organisation domain (e.g. product teams) with the decomposition of the ***tangible*** system elements in the physical domain. By definition, this omitted any ***intangible*** physical domain elements (Rowles, 1999).

This situation is illustrated in the revised Figure 11, which indicates a possible location of ***intangible*** elements as being in stage one and stage three of the design process.

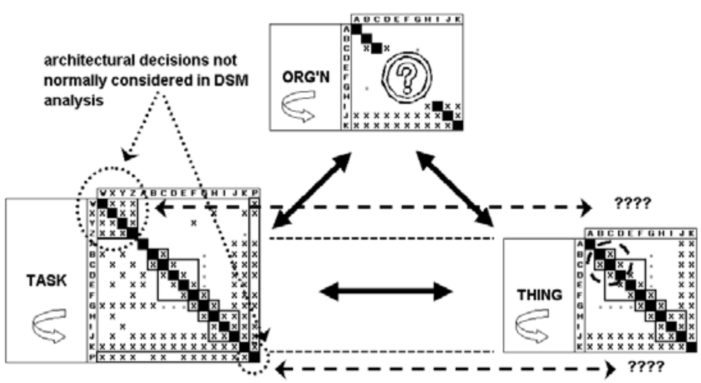

Figure 11: Alternative Hypothesis for Inter-Domain Mapping (Adapted from Eppinger & Salminen, 2001)

This appears to be occurring in the task – physical domain mapping of the generic gas turbine example as depicted in the revised 45-element DSM in Figure 12.

Although elements in one domain need to map to the 'same' element in another domain in a one-to-one manner, this need not necessarily be by a common vector. This is relevant because unpinned and unclustered individual elements in one domain may be pinned and clustered in another domain, and different sequences may result from this. This has the potential to create tension between domains and this trade-space can be at once the source of dynamicism in architectures, and an indication of the potential for disconnects between domains. Since domain boundaries are the natural location for contracts, this qualitatively explains why so many supply chains carry the seeds of their own destruction.

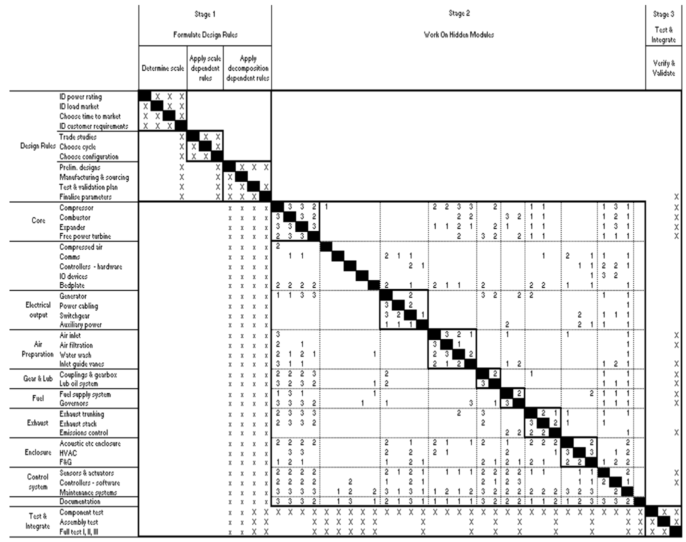

Figure 12: Complete DSM with Intangible Elements (see a larger version of this figure in the Appendix)

Even though 14 new elements have been included, the value of tangible physical elements remains constant. This is because the intangible elements have already been accounted for as value-less costs, and the costs apportioned across the tangible elements. In this industry there is seldom a saleable intangible product so to recoup the cost of intangible elements participants need to retain a tangible presence in some form. This may change if the industry ever restructures completely to the point where dematerialised firms sell intellectual property (designs) to manufacturers, however whilst the design process remains highly integrated with the manufacturing process, and whilst this product remains highly integrated this is unlikely.

6 HYPOTHESISED MULTI-DOMAIN OPTIMISATION

As hypothesized in the previous section, the different domains are related and can interact. This may resolve the problem that there is no physical domain justification for favouring sequences that place relationships near to the diagonal as used by many clustering algorithms, rather than simply inside or outside of a module boundary. The way this may be resolved is that in the task domain the position of a relationship with respect to the diagonal is of significance. So provided there is a link between the sequence in one domain and the sequence in another, the interactions will tend to both cluster elements into modules and force out of module relationships below, or adjacent to, the diagonal.

6.1 Assumptions

In the task domain the system is best considered as a series of feedback loops. The order (i.e. sequence) in which the tasks are arranged determines the network's response. Arranging tasks below the diagonal optimises the task domain. Above diagonal marks indicate the possibility of feedback occurring (either from one generation of design to another or within the same generation). The distance of these feedback marks from the diagonal determines the relative length/duration of each iteration cycle.

In the physical domain the network is transparent. This means that the objective function acts equally in all directions, yielding no net (resultant) force on off-axis relationships. The objective function is binary in the way it acts across the module boundary. This is consistent with tightly linked elements benefiting from being in modules whilst inter-module relationships are equally ‘costly’ irrespective of the off-axis distance: a relationship is either internal or external. Correctly identifying a cluster boundary optimises the physical domain. In this simplistic example, the cluster boundary cut-off is set around any diagonally-centred group with ≥ 50/50 fill ratio (excluding on-diagonal relationships and slightly weighting in favour of enclosing relationships for tie-breaking purposes).

The sequence selected in the two domains need not be exactly the same but should be largely similar. Tasks executed in parallel may be re-sequenced in their physical manifestation provided that module boundaries are respected (i.e. the ‘owner’ of the module is permitted to optimise the module sequence). Tasks executed in series should be kept in the same sequence in the physical domain (i.e. if a sequence is inside a module boundary it is frozen, and if it crosses a module boundary then it makes one module dependent on another) except where there is a weak serial link between modular clusters they may evolve over different cycle times provided they periodically temporally intersect – this allows the modular design to benefit from different evolutionary rates whilst still keeping the overall system in harmony (to allow this to occur the out of module relationship needs to be fixed in the period between temporal intersections by e.g. an accepted standard).

6.2 Example Of Jointly Optimising Two Domains

The effect of this is best illustrated with a small example showing how two domains can interact. In this eight-element example two empty elements are included as spacers, and the domains chosen are the task and physical.

If the information in these two domains is combined then the sequence and module boundary shown in Figure 13 represents a good compromise. This is relatively harmonious as the B:C module can now evolve independent of the D:G module provided the B→G relationship is respected.[8] The reason the task domain drives forwards from B to G is as a result of a deliberate selection. There may indeed have been an alternative solution that allowed G→B to drive the design, and which now represents unshown inter-generational feedback. In the physical domain the small module B:C would be expected to evolve at a different rate than the larger module D:G as they attract different numbers of designers and manufacturers and require different resource levels, etc. This compromise element sequence would require the two objective functions to be balanced, as over-emphasising one domain is the same as ignoring the other.

This might represent the relatively clean decomposition of a telephone into a handset (B:C) and a control unit (D:G). In the task domain B→G represents the choice of the impedances, voltages, currents, etc. In the physical domain B↔G represents the actual electro-mechanical mates, etc.

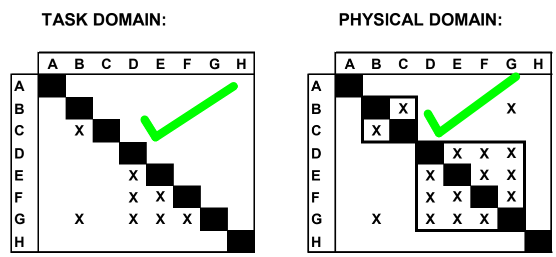

Figure 13: Good Optimisation of Two Domains

However if the information contained in the physical domain had not been known, then efficient sequencing of the task domain by an “as early as possible” algorithm (Cho and Eppinger, 2001) could easily have yielded Figure 14.

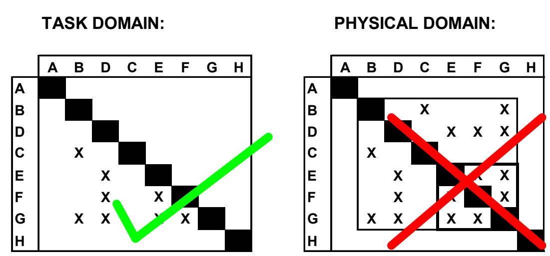

Figure 14: Poor Optimisation from Concentrating on Task Domain

If this had happened it would be nigh on impossible to separate out two modules in the physical domain as the sequence of choosing inter-element interfaces makes it unlikely that they would decompose cleanly. It is possible that one module might be extractable as E:G, but much of the evolutionary potential would have been lost. This is akin to the situation faced by poorly understood products as the physical manifestation is forced to be more integrated than it need be. This means that the evolutionary trajectory is suppressed, as it requires the entire product to be redesigned at each generation, and the inter-generation period is therefore likely to be slower. The upside is that when redesigns do take place they might be slightly faster.

Likewise, if the information contained in the task domain had been concealed then any algorithm that sought tight modularisation might produce Figure 15 instead. Whilst this

[8] This relationship may remain ‘owned’ by the B:C module or may be broken out into a separate ‘design rule’ element. At present only societal valuation is being discussed, however when considering firm-level valuations these two cases correspond to a closed or open system provided that there are multiple contenders for module D:G. Similarly, the dependency of the ‘design rule’ element would indicate the extent to which it was proprietary or non-proprietary thereby yielding a classic 2 x 2 (open versus proprietary) standards grid.

might be good for the physical product it causes an iterative design loop around G→B. In the example of the telephone this might be acceptable however in other products less so, and reflects a situation where a poorly understood design process is locked in place by an over-rapid consolidation on an inefficient dominant design.

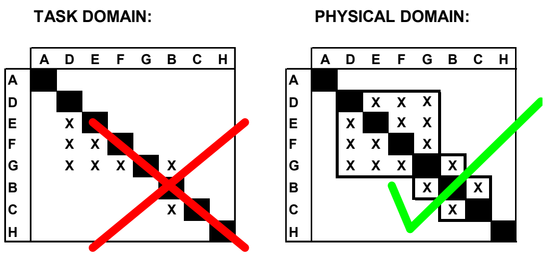

Figure 15: Poor Optimisation from Concentrating on Physical Domain

These issues do not pose a problem if in practice there are rapid design iterations at low cost. However, if design cycles are lengthy and / or expensive then single domain optimisation represents a considerable inefficiency. Such situations might occur if either the design or manufacturing process of a large product is poorly understood, or if there are power imbalances between e.g. design and manufacturing organisations (or, say, between 'engineering' and 'styling'). It is in these more serious situations that blind evolution is most undesirable.

Ideally one should have all relationships mapped in all domains and individual domain optimisations identified, then a global optimum selected, however application of all the different objective functions to single-domain data may qualitatively suggest areas of tension. As an intermediate solution simply posing the question 'how different are relationships in different domains' and 'how different are the objective functions' may yield some insight.

As other domains are added in – especially the organisational and individual, each with their own objective function and rules on sequence relationships, the overall effect is to create a highly complex value surface, as in real life.

6.3 Multi-Domain Societal Objective Function

For multi-domain optimisation the objective function will be of the form:

$$Max.\ \alpha_1\ .\ (Value\ task\ domain) + \alpha_2\ .\ (Value\ physical\ domain) + \ldots + \alpha_j .\ (Value\ j\text{'th}\ domain)$$

Where α_j etc. represent the co-efficients of the j'th domain's contribution to value, and $\sum_j \alpha_j = 1$.

At present the societal value of the physical domain is maximised by objective functions of the form of Equations B&C Eq. 11.1) and (B&C Eq. 11.2) which can be crudely applied using the same series of assumptions as is used to develop Table 2. We believe the societal value of the task domain is maximised by activity networks with uncertain levels of rework (Browning, 1998). Objective functions for the organisational domain based around the various proximity measures seem most appropriate although the foundation of this rule (Allen, 1977) predates widespread use of high-bandwidth electronic communications. The combination of the spatial proximity in organisations with the task domain will need testing against predictions of technical communication in organisations. Other proximity measures are discussed in Fine et al. (1995).

7 CONCLUSIONS AND FURTHER WORK

This paper shows that relative societal value can be qualitatively calculated for alternative multi-domain decompositions of highly integrated architectures. It also introduces a new objective function (for clustering in the physical domain) and explicitly links multiple domains some of which have known objective functions. Further work is required to automate the process, validate it against reference cases, and to extend it to actor value via the application of game theory. Combining this with other modular operators will allow simulation of dynamically evolving architectures.

REFERENCES

Allen, T.J., "Managing The Flow Of Technology", MIT Press; 1977 (1984 edition ISBN-0262510278).

Baldwin, C.Y. & Clark, K.B. 2000. "Design Rules", MIT Press, Cambridge, ISBN 0-262-02466-7.

Browning, T.R., "Modelling And Analysing Cost, Schedule, And Performance In Complex System Product Development", Ph.D. Thesis (TMP) MIT, 1998.

Cho, S., Eppinger, S.D., Product Development Process Modeling using Advanced Simulation, DTM 2001 Conference Proceedings.

Eppinger, S.D., Salminen, V.; "Patterns Of Product Development Interactions"; Proceedings of International Conference On Engineering Design, ICED'01, August 2001.

Fine, C.H., Gilboy, G., Oye, K., Parker, G.; 1995; "The Role Of Proximity In Automotive Technology Supply Chain Development: An Introductory Essay"; MIT Working Paper; May 1995.

http://mit.edu/dsm/, the MIT DSM Home Page.

Rowles, C.M., "System Integration Analysis Of A Large Commercial Aircraft Engine", MIT MSc (SDM) thesis, 1999.

Sharman, D., "Valuing Architecture For Strategic Purposes" MIT MSc (SDM) thesis, January 2002.

Sharman, D., Yassine, A., Carlile, P., "Characterizing Modular Architectures," ASME 2002 International Design Engineering Technical Conferences, DTM-34024, Montreal, Canada, Sep. 29- Oct. 2, 2002.

Sullivan, K., Cai, Y., Hallen, B., Griswold, W.G., "The Structure And Value Of Modularity", ESEC/FSE, 2001.

APPENDIX

In this Appendix the more detailed figures are reproduced at a larger scale.

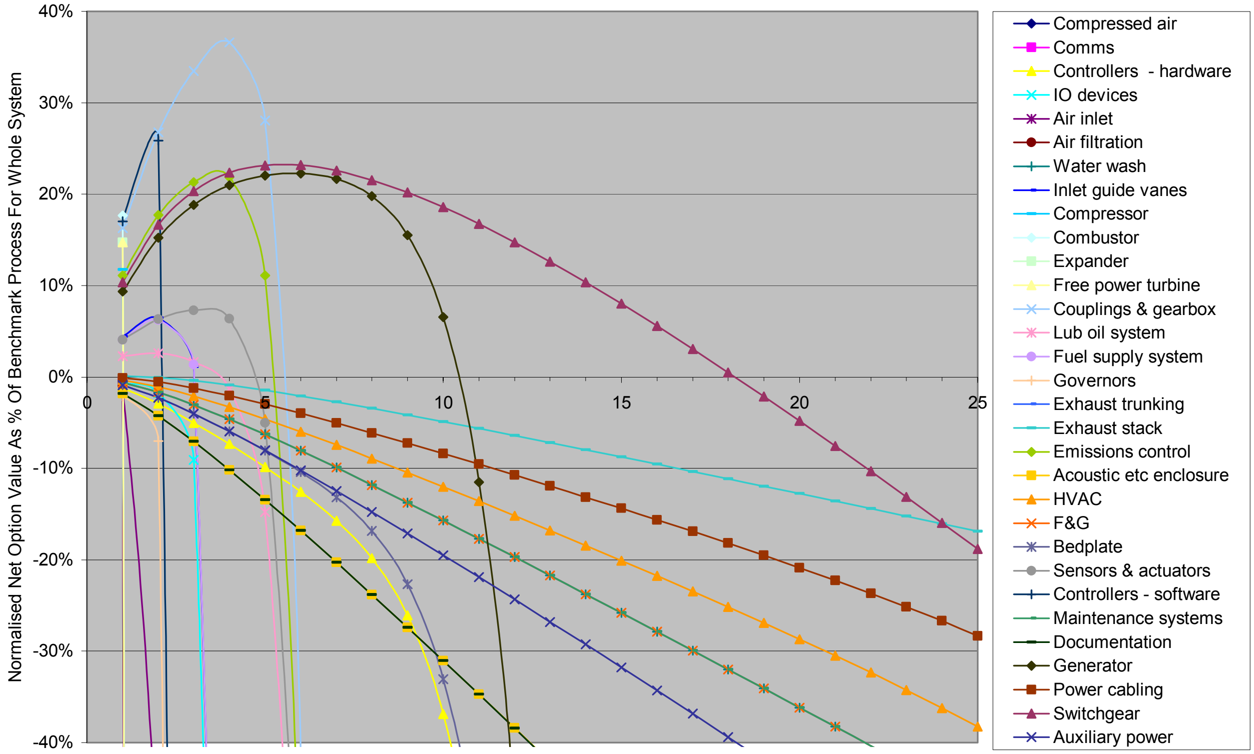

Figure 3: Initial Valuation of the Genset's 31 Elements

		16	18	19	20	21	29	24	23	1	2	3	4	5	6	7	11	12	8	9	10	13	14	26	15	28	27	17	22	25	30	31
Generator & pilot exciter	16			2										1	1	3	3	3	2		2	2							1		1	
Power cabling	18	3		2																											1	
Switchgear	19	3	2		1																					2			1	1	1	
Auxiliary power supplies & UPS	20	1	1	1																2						2	1		1		1	
Compressed air	21													2																		
Comms – tannoy, visuals, mag loop, etc	29	2	1	1											1	1							1		2		1		1		1	
Hardware: Controllers	24			2	1																					1	1		2	2	1	
IO devices	23																										1		3	2		
Air inlet trunking & plenum chamber	1										3	2	1	3						1					1						1	
Air filtration	2									3		1		2		1															1	
Water etc wash	3									2	3		2	2	1	2	1											1			1	
Inlet guide vanes	4									2	1	2		3	1	1				1	2								1	2	1	
Compressor	5					1				2	2	3	3		3	3	2		2			1	1						1	3	1	
Combustor	6											2	2	3		3	2			3	2	1	1						1	2	1	
Expander or Turbine	7									1	1	2	1	3	3		3		2	1		1	1						1	3	1	
Free power turbine	11											2		2	3	3		3	2		2	1	1						1	1	1	
FPT bearings & couplings	12	2												2	2	2	3		2			1							1		1	
Lub oil system	8	2												3	2	3	2	3										1	1	1	1	
Fuel supply system	9													1	3	1					2						2	1	1	1	1	
Governors	10	1							1				3	3	3	3	2		1	3									1	1	1	
Exhaust trunking	13											2		2	3	3	3	3					2	1	1				1		1	
Exhaust stack & transition ducts	14													2	3	3	2	2				3		2	1				1			
Emissions control systems	26																			2	2	2	2								1	
Acoustic etc enclosure	15	2		1						2	1		1	2	2	2	2	2		1		2	2	2			2				2	
HVAC	28	2		2						2	1				3	3				2				1	3		3			1	2	
F&G detection & suppression	27	1		2	1									1	2	1		2	1	2	1	2	1		2	2			1	2	2	
Bedplate & mountings	17	2		1						2	1	1		2	2	2	2	2				2	2		1		1				1	
Sensors & actuators	22	2	1	2	1						1	1	1	2	2	2	2	2	2	2	2	1	1			1	1				2	
Software: Controllers	25	1		2	1			2						2	2	2	2	2	1	2	3					1	1		2		1	
Maintenance access systems	30	3	1	3	1		1	2		1	2	1	2	3	3	3	3	3	2	2	2	2	2	2	3	2	3	2		2		
Documentation & certification	31	2	1	3	1		1	3		1	1	1	1	3	3	3	2	3	2	2	2	1	1	2	1	2	3	1	3	3	2	

Figure 4: 31-Element Physical Domain DSM

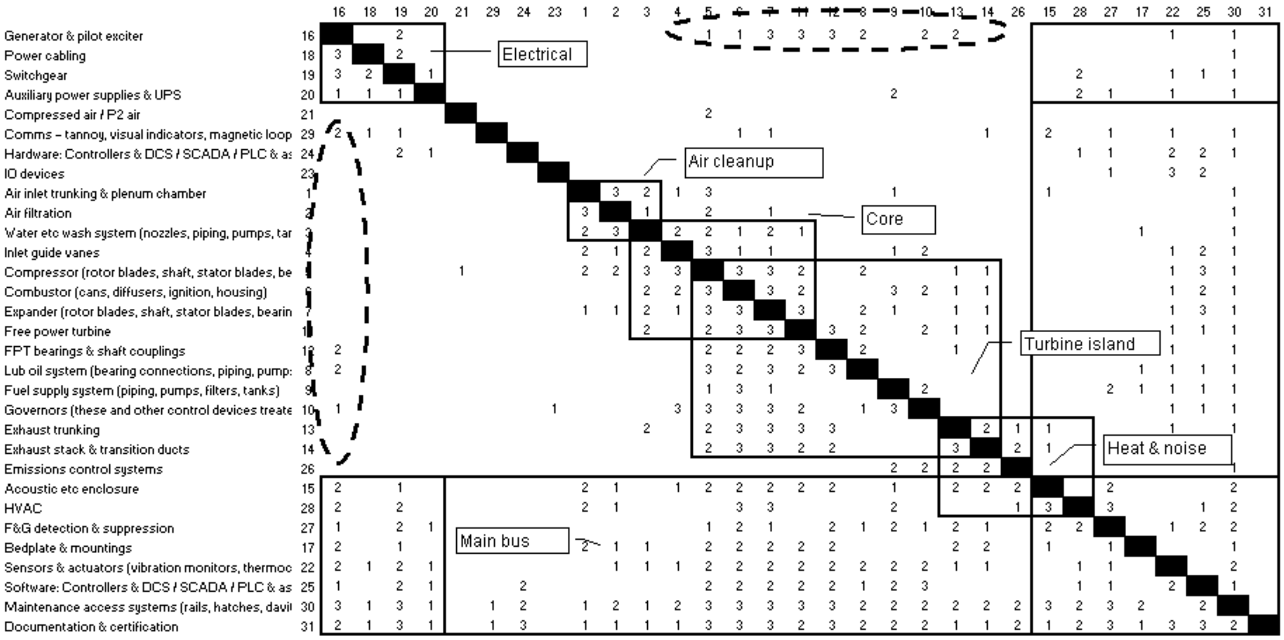

Figure 7: Clustered DSM of Gas Turbine

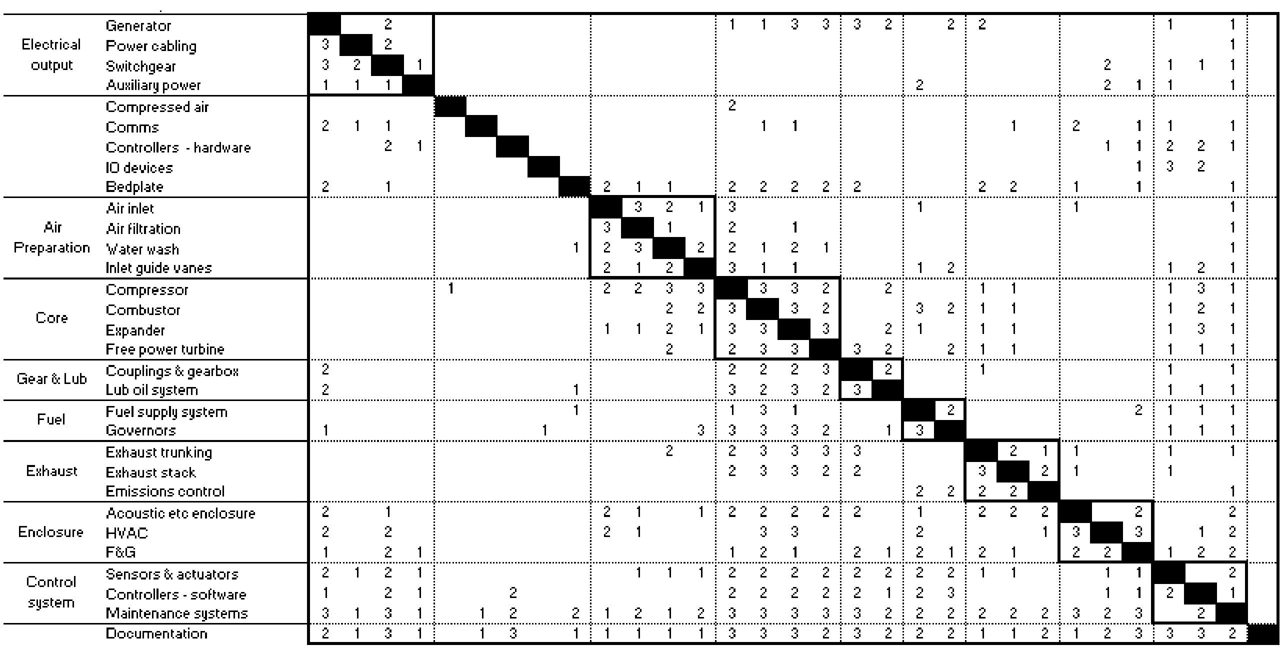

Figure 8: Unpinned Modules in Genset DSM

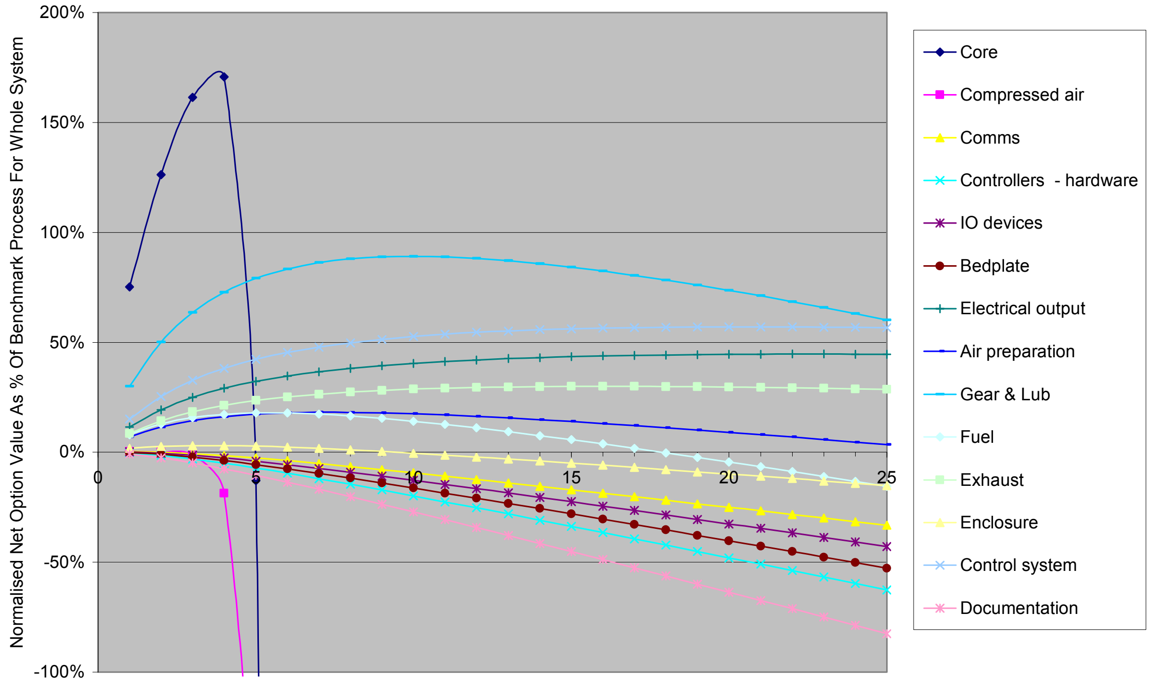

Figure 10: Gas Turbine Option Value Curves for Second Level Decomposition

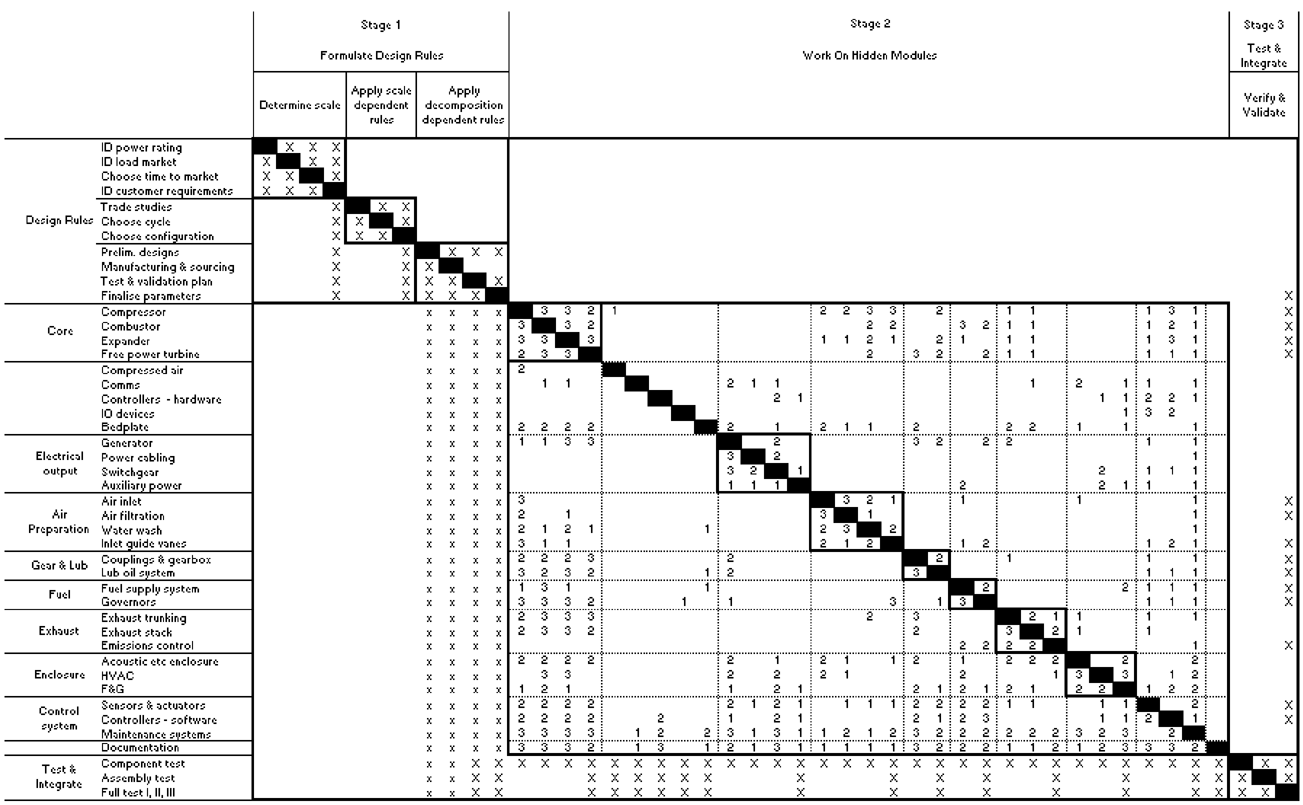

Figure 12: Complete DSM with Intangible Elements