Valuing Architecture For Strategic Purposes

Comments On Applying The Dependency Structure Matrix With Real Options Theory

By David M. Sharman
B.Eng. (Hons) Engineering, Royal Naval Engineering College, 1989

Submitted to the Systems Design and Management Programme
in partial fulfilment of the requirements for the degree of

Master of Science in Engineering and Management
at the
Massachusetts Institute of Technology

January 2002

Signature of Author...

Author: David M. Sharman
System Design and Management Program
11th January 2002

Certified by...

Paul Carlile
Thesis Supervisor
Professor, Sloan School of Management

Certified by...

Ali A. Yassine
Thesis Supervisor
Research Scientist, Centre for Technology, Policy, and Industrial Development

Accepted by...

Steven D. Eppinger
LFM/SDM Co-Director
GM LFM Professor of Management Science and Engineering Systems

Accepted by...

Paul A. Lagace
LFM/SDM Co-Director
Professor of Aeronautics & Astronautics and Engineering Systems

THIS PAGE DELIBERATELY BLANK

Valuing Architecture For Strategic Purposes

Comments On Applying The Dependency Structure Matrix With Real Options Theory

by

David M. Sharman

Submitted to the Systems Design and Management Programme
on 11th January 2002
in partial fulfilment of the requirements for the degree of
Master of Science in Engineering and Management

1 ABSTRACT

Analysis of product and organisational architecture using dependency structure matrices to describe a series of domains, in conjunction with real options theory, assists in predicting the strategic capabilities of either existing or potential products and organisations, and likely optimal or maximal rates of change. This assists in predicting the extent to which technologically dependent organisations can realistically create and capture value from a portfolio approach as a number of technology conglomerates currently seek to do. It also goes some way towards explaining why existing organisations find it difficult to create or exploit new knowledge and thereby helps explain why many synergies remain unrealised.

This suggests that strategic leadership of technology conglomerates must be by people who possess either the tacit knowledge of the financial, organisational and technical aspects of the business, or who possess explicit tools to bridge any gaps. Given that explicit financial tools are available, in the absence of unique individuals the strategic planning process needs to incorporate measures designed to a priori check that the proposed strategies will result in technical knowledge creation and organisational value capture.

Thesis Supervisor: Paul Carlile
Title: Professor, Sloan School of Management
Thesis Supervisor: Ali A. Yassine
Title: Research Scientist, Centre for Technology, Policy, & Industrial Development

"What does make design a problem in real world cases is that we are trying to make a diagram for forces whose field we do not understand."

Christopher Alexander, 1964

"Only connect."

E.M.Forster, 1910

2 ACKNOWLEDGEMENTS

Firstly I would like to thank you, the reader, for taking the time to consider what I have to say. It is not perfect and if I were to start again I would try to do better, but I hope it helps you nonetheless.

Secondly I would like to thank my various teachers over the years. At school, college, and university; on ships and oil installations; in offices, at sea, on lakes and in jungles; many people have shown me what works and does not, and why. And then they have all pointed out that in the end it is the people that matter much more than the processes.

Particular thanks are due to my advisers Paul Carlile and Ali Yassine of MIT's faculty and research staff, who have been at once helpful and demanding. Various personnel from United Technologies Corporation's Pratt & Whitney Division gave me a valuable insight into a particularly relevant development effort for which I am grateful. The very busy Carliss Baldwin of Harvard Business School gave me her time and in doing so helped clarify some crucial points regarding the theories she has developed jointly with Kim Clark.

My fellow students Stella Oggianu and Hernan Joglar collaborated with me in developing examples for wind turbines and pebble bed nuclear reactors using various methodologies, and in doing so helped hone my thinking. As with all of the people to whom I am grateful any errors are mine alone.

Lastly but by no means least I would like to thank my partner Marja van Gestel for her support and patience that makes me a very fortunate person indeed.

3 TABLE OF CONTENTS

4 LIST OF TABLES

5 LIST OF FIGURES

6 INTRODUCTION

6.1 Origin Of This Thesis

This thesis began as an attempt to explore how a manufacturer might best take advantage of the likely changes in the energy industry. However few quantitative tools were found for conducting this exploration that were not ad-hoc in nature and limited in their scope of application, and so the thrust of the thesis changed to become one that explores a possible quantitative tool using an illustrative example of a power generation product.

6.2 From The Energy Industry To Valuing Architecture

Worldwide the energy industry is entering a period of great change. This is not the first time this has happened in the energy industry nor is this the first industry to undergo rapid and unsettling change. It is not clear what source(s) of energy will be favoured nor what conversion technologies will be required. Manufacturers of energy conversion equipment (nuclear reactors, coal-fired power stations, gas turbines, etc.) are faced with the problem of deciding how best to allocate limited resources between the different options so as first to create value for society as a whole, and secondly to capture value and minimise risk for their stakeholders in particular.

There are two great and intertwined pieces in this problem. The first great question is how to determine the likely future value landscape – for example when, where, and how one energy source such as coal might be preferred over another such as wind. This is related to the second great question which is how best to create and capture value in this landscape. The two are intertwined since there is a feedback relationship between societal investment in any particular energy conversion technology and the value of that energy source. In this respect societies' past choices determine to an extent their range of future choices. There are tools that can be applied to determine what might be the value landscape for any particular industrial sector, such as the energy sector. Examples of the unifying tools that exist are econometric models (e.g. as described in Greene, 1999; Kennedy, 1998) and systems dynamics models (e.g. as described in Sterman, 2000). There is a significant uncertainty in the output of these models but they do go some way to answering the first great question – the nature of the value landscape – by way of qualitative and quantitative outputs that represent the system's behaviour in an integrated manner.

There is rather less certainty in answering the second great question – how best to create and capture value – which is a fundamental strategic question. Until recently the best tools have come from economists explaining the role of firms (e.g. Coase, 1937) in quantitative terms and the strategists (e.g. Porter, 1980 & 1985) who have developed many qualitative models. Localised success has been achieved in reconciling these two viewpoints for particular industries but even where locally successful these attempts have been bounded by the industrial paradigm on which they are based. There has been remarkably little work that integrates these two viewpoints in a unified manner that is able to extend beyond any one industrial paradigm. Historically this has been less relevant but increased globalisation and accelerating technological advances have provided many examples of where failure to understand the situation has led to rapid extinction of many firms and industries. Failure to unify creates problems as no matter how good a systems dynamics or econometrics model is created, a variety of studies and meta-studies (e.g. UN & WEC, 2000) are used to determine both the models' structures and the governing parameters. This in turn means that value inherent in unforeseen opportunities afforded by development of key elements will never be observed a priori in these models. In the energy sector examples of such elements are the possible restructuring towards a more distributed model, and the possible convergence of the transportation system's energy supply system and the fixed electrical system. Such a restructuring is equivalent to the introduction of the personal computer and the Internet; and it may be that the current non-unified models will be as poor in predicting the development pathway of the energy system as they were in predicting that of the information technology sector, as the most important phenomena were emergent in nature.

Because this turmoil is most evident in the information technology sector much academic study has focussed in this area to understand what is occurring (e.g. Teece, 1998; Moore, 1999; Porter 1980 & 1985; Baldwin & Clark, 1997 & 1999; Fine, 1998; Christensen, 1997). Themes that have emerged from this work are that: one must understand any industrial sector as being an integrated multi-dimensional system; and that rate of change depends on the extent to which local (sub-system) improvements can be made that lead to emergent revolutionary change at the system level. This suggests that system designs that stress modularity over integration are capable of more rapid evolution but prone to high volatility as different dominant designs emerge and then markets are captured and value harvested by entities that are able to control key aspects of these dominant designs.

Taking this line of enquiry to its logical conclusion (Baldwin & Clark, 1999) suggest a unifying theory that is applicable to modular designs. The theory has two parts; the first deals with the subject of value creation from a societal perspective and the second deals with the subject of value capture from an entity perspective. As befits a theory that is applicable to modular designs the theory is itself modular in nature. The subject of value creation is handled using the notion of evolution applied to architecture, in combination with the theory of the contract as exemplified by entities such as Coasian firms, and the value of real options. The (as yet largely unwritten) subject of value capture can be handled using a combination of multi-stage / multi-player game theory with classical micro and macroeconomics, in particular price / demand etc. elasticity curves.

This Baldwin and Clark theory is explicitly written with highly modular architectures of modern information technology products in mind. This thesis attempts to extend it and apply it to the relatively integrated architectures of energy conversion technologies and in doing so investigates the extent to which it may provide a more general unifying theory of the value of architecture.

This is of relevance to firms in the energy sector as to a great degree all possible desirable architectures are highly integrated in one or more ways. So this thesis sets aside the first great question regarding the nature of the value landscape and focuses in on the second great question that is in essence a problem of holistically valuing a multi-dimensional architecture. Because of the breadth and novelty of the topic more problems are described than answers are found and so in exploring these problems the thesis does not arrive at any answers of direct relevance to the energy industry but to a limited extent matures the applicable tool kit.

In passing comment is made on the relevance of the Baldwin and Clark theory in developing a new language to enable better communication. This is relevant to work on the dynamic capabilities of the firm (Teece et al, 1997, 1998) and that on boundary objects (Carlile, various).

6.3 Objectives & Major Findings

Even though there exist a number of instances where the selection of one total architecture (product, organisation, etc.) over another has clearly influenced both the societal value created and the net value captured by any given economic entity, there exist few tools to objectively analyse the best architecture for a given problem. This is essentially a strategic question and there are strategic analysis tools available, however these are unsatisfactory in objectively and quantitatively resolving such questions in anything other than an ad-hoc manner.

It has been proposed (Baldwin & Clark, 2000) that the societal value of highly modular architectures can be analysed objectively by the application of real options theory. This thesis attempts to extend this proposal to more integrated architectures. It then outlines a methodology that may allow determination of the value capture question.

In exploring a series of problems in measuring the architecture new insights are gained into alternative ways of describing architectures using Dependency Structure Matrices (DSMs) and in the limitations of DSMs. This leads to the definition of new architectural terms, illustrations of alternative solutions using the notion of domains, dimensions, and topologies, and the proposal for new DSM manipulation algorithms. Proposals are made and illustrated regarding quantification of the degree of modularity and integration in architectures. The ability to objectively and quantitatively describe any architecture is an important first step in selecting between competing alternatives. Architectures are defined as spanning multiple domains and the principles of decomposition and aggregation are applied in the identified dominant domain. This requires a holistic view of the entire architecture in order to identify the appropriate domain over the relevant life cycle. It also requires a systematic understanding and recording of the relationships between the primitive elements at a lower level of decomposition that are arranged to form an architecture at a higher level.

A holistic integrating explanation of the relationships between DSMs, Quality Functional Deployment matrices (QFDs) and the inter-domain transforms described by axiomatic design's Design Matrices (Suh, 2001) is described.

The introduction of penalty terms into the real options valuation to represent boundaries is new. The suggestion that architectural control may also be exercised through test and integration competency is new, as is the description of why these are so easily subject to inversion from one to another, and the implications of this.

The proposed extension of societal valuation to determination of value capture applies the dynamic process of multi-stage multi-player game theory in a new context.

The hypothesis is that the application of this method will quantitatively demonstrate, and conceptually link, the relationship between dealing with "knowledge boundaries" (Carlile, 2002) and the more strategic discussions of a firm's “dynamic capabilities” (Teece, 1997). This demonstration is particularly helpful since it might be able to explain some of the difficulties historically experienced by technology conglomerates in harvesting synergy.

A discussion is included of the extent to which the question value creation has been (and can be) answered, and of the domain issues in conducting further work. This is followed by a discussion of the proposed methodology for the question of identifying value capture which suggests that, leaving size aside, an element's visibility largely determines it's intrinsic societal value whilst the relationship between an element's visibility and dependence in the context of the concentration of dependence in other elements determine an element's ability to defend or capture actor value.

Overall the complexity inherent in this suggests that strategic leadership of technology conglomerates must be by people who possess either the tacit knowledge of the financial, organisational and technical aspects of the business, or who possess explicit tools to bridge the gap. Given that explicit financial tools are available, in the absence of unique individuals the strategic planning process needs to incorporate measures designed to a priori check that the proposed strategies will result in knowledge creation and value capture. The option value of an architecture represents a potential tool for holistically expressing this value and so can be used to substitute explicit language for tacit knowledge.

6.4 Approach; Methodology; Structure of Thesis

The first five sections are administrative in nature; table of contents, list of figures, etc. The sixth section is the first real ‘chapter’ and explains how a thesis that was originally intended as an investigation into the energy industry's future structure became instead a case study application of a particular technique to gas turbine power generation equipment.

The sixth, seventh, and eighth chapters cover the relevant literature, define various specialist terms such as *architecture, domain, dimension, decomposition, modular,* and *integral*, and introduce a series of tools. A short discussion of Dependency Structure Matrices (DSMs) is contained here and an extended introduction to calculating the Net Option Value (NOV) inherent in a modular architecture together with my reproductions of a series of worked examples. Although lengthy, this introduction to NOV is only a partial summary of the topic and further information on the topic, various subjects I do not summarise, and more worked examples can be found in Baldwin & Clark (2000).

Much of the remainder of this thesis uses the test case of a small gas turbine. This is because the initial aim was to explore the possibilities inherent in current dominant designs of generating plant (primarily gas turbines) and then those afforded by newer designs (e.g. wind turbines, fuel cells, small nuclear reactors), and finally the possibilities afforded by various combinations and to relate these results to the manufacturer of medium sized power generation equipment. In the event the tool set was found to be sufficiently immature that only the test case of the gas turbine was investigated in isolation. For this reason only the architecture of a generic small industrial gas turbine generator set ("gen-set") is described in chapter nine. Whilst this test case is generic it has sufficient detail as to be both representative of a typical energy transformation product and of a typical relatively – but not entirely - integrated product and the nearby multi-dimensional architecture. As such it demonstrates the problems and benefits of applying this method to actual situations. On occasion another product is used as a representative example and this is often a wind turbine.

Chapter ten makes an initial valuation is made of the gas turbine sub-systems using Baldwin & Clark's theory of option value. Various assumptions are made and then challenged. This leads to changes in the assumptions, the valuation process, and the valuation itself. Strategically useful results emerge which are then contrasted with more idealised results obtained if no sub-systems are identified a priori.

Chapter eleven uses Dependency Structure Matrices (DSMs) as a tool for analysing the architecture of the generator set at the second and third level of decomposition. Most of this is performed in the physical domain and therefore the principal concept is that of clustering. The meaning of clustering is investigated using simplified cartoon DSMs and cartoon architectures so as to be able to introduce and define terminology and develop the underlying issues with the minimal amount of complexity. These cartoons are woven in between consideration of the real gas turbine. Towards the end of the chapter some discussion takes place regarding automated clustering algorithms. This leads to a discussion of the meaning of cluster boundaries and boundary objects on one hand, and on network / graph theory on the other. The apparent structure in the gas turbine's generator set is qualitatively related to the tension seen in the alternative option value calculations seen in the previous chapter. Indicative quantitative resolutions are attempted. The notion of the missing architectural decisions is introduced and the pathway to the other domains indicated. This leads to a discussion of multi-domain optimisation. In passing comments are made on characterising the signatures of integral and modular architectures.

In chapter twelve a further series of valuations are made of the gas turbine's architecture using different ways of calculating visibility. Initial valuations focus on applying the improved understanding of the principles of visibility. Later valuations show the impact of imposing a different intermediate level architecture on the underlying reality. The task domain is compared with the physical domain and some missing intangible elements are identified. This allows the full architectural system signature to be characterised. Finally a proposal is made that seeks to justify joint optimisation of multiple domains, together with a suggested objective function for maximising societal value that may resolve the apparent contradictions between what is 'best' in different domains.

Chapter thirteen discusses the difference between creating value and capturing value and contrasts societal and actor value. It is suggested that visibility determines societal value created whereas the balance between visibility and dependency determines the ability to defend and/or capture value. The way in which multiple actors can compress S-curves is discussed, as is the potential for one product to contain modules or elements with different cycle times. This leads to a discussion of the different sorts of discontinuities that may occur and what characterises them. These discontinuities are then compared with the notion that firms need dynamic capabilities in order to make successful transitions, and then it is suggested that these discontinuities map fairly well to proposed types of knowledge boundary objects. This allows for a discussion on bridging methodologies of which the notion of Net Option Value is an example.

Chapter fourteen and onwards conclude and make some suggestions for further work. In general the various research 'dead-ends' are included. Whilst this detracts from the readability it may assist future researchers.

7 BACKGROUND: ARCHITECTURE AND MODULARITY

This chapter outlines some of the relevant literature and definitions, except where it is only locally relevant or where the narrative thread is best served by a later introduction. The first part of the chapter primarily surveys the literature whilst the second part summarises the relevant portions of work by Baldwin & Clark on valuing modular architectures.

7.1 Defining Architecture

The term architecture will be used frequently in this thesis, but often in a more holistic manner than many of the traditional definitions. The way I use the term "architecture" seeks to be independent of product type, discipline, or industry thereby avoiding the specificity of usage found in information systems and product development literatures as described by Mikkola (2000). In information systems the architecture describes data flow through networks of hardware and software, whilst in product development the focus is on the physical components and their relationships. Both of these literatures try to avoid entanglement with social and organisational aspects of architecture. The 'standard' definition of a product's architecture is ***"the scheme by which the functional elements of the product are arranged into physical chunks and by which the chunks interact"*** (Ulrich & Eppinger, 1995). Drafting a generic definition is extremely difficult as described by Rechtin & Maier (1991), however I go further and define it as ***"the arrangement of a product or process in all the relevant domains and at all the necessary levels of decomposition required to fully describe the essential product or process"***. Implicit in this definition is the notion that the manner of arrangement of the components is as important as the list of components, as there is information contained in the topological relationships. This definition extends beyond the 'standard' definition by being open-ended about the number of domains (e.g. physical, functional) required to fully specify the architecture and is in the spirit of the system architect's definition of architecture, ***"The structure (in terms of components, connections, and constraints) of a product, process, or element."*** (Rechtin & Maier, 1991).

7.2 Defining Decomposition

Decompositioning is a systems engineering term used to describe the process of breaking down things into ever-smaller chunks until the primitive elements are arrived at (after Rechtin &

Maier, 1991; Boppe, 2001; Crawley; 2001). This is essentially a slicing (partitioning) process and so it is subject to artistic interpretation (like much non-trivial engineering) with no one 'right' way of decomposing a thing. The thing can of course be in one or many domains. In a sense this entire thesis is about how best to decompose in a connected fashion across multiple domains, and the evolutionary dynamics that may result.

7.3 Defining Domain & Dimension

"Domain" is normally used to describe ***"a recognised field of activity and expertise, or of specialised theory and application"*** (Rechtin & Maier, 1991). I use it in the sense of 'field' so as to be able to discriminate between dimensions and domains. Typical domains would be organisational, functional, physical, individual, i.e. the '***als***'. In my usage a dimension is 1) a subset of a domain (e.g. energy, mass, momentum in the physical domain); 2) a parameter that describes a location in space. I apologise for having the two usages of dimension but when I commenced this thesis I had not appreciated that high-dimension topologies might be important.

7.4 Defining Modularity

Modularity is an ambiguous and elusive notion that has been loosely used in different ways by different people at different times. In its architectural usage is easiest to define as an antonym, i.e. modular is the opposite of integrated. Webster's New World Dictionary defines the noun "module" as "***a compact assembly functioning as a component of a larger unit***" and the adjective "modular" as "of a module". Alternatively the adjective "modular" is defined as "***whole or complete***". So in this sense at the two extremes modular architecture is one made up of assemblies of components, whilst an integrated architecture is one made up purely of the lowest level of component without having intermediate assemblies. In the physical domain this is how I shall use the two pure extremes, and in other domains (e.g. the organisational domain) the same concept will apply except that the notion of component and assembly will require translating, either to domain-neutral terms such as primitive element or domain specific terms such as department. I interestingly the most domain-neutral terms I can identify for assembly are '*module*' and '*cluster*' (i.e. a cluster of relatively closely related elements): the former risks confusion and appears too extreme whilst the latter is rather fuzzy. However I emphasise the concept that there is a continuum between a fully modular and a fully integrated architecture. A fully modular architecture is one with clear clusters of elements at an intermediate level to the top level concept, and where the relationships between the elements within an assembly is hidden to the elements outside the assembly. This incorporates the notion that a module not only contains elements, but also contains a higher density of relationships between those elements than to elements outside the module. In a sense whether one uses the term cluster or module is irrelevant.

Many of these notions of what is modularity span multiple disciplines. By comparing the usage in the various disciplines it is possible to derive a thematic definition of modularity (Schilling, 2001). Because of the process of creating the definition this is inherently a common denominator definition, and where most disciplines are silent on a theme it is possible for a minority of disciplines to introduce a potentially irrelevant theme. An advantage of this approach is that by comparing an example with the reference themes degrees of modularity can be recognised.

This modular – integrated continuum is very much the thrust of Fixson's (2001) three-perspective view on modularity. This excellent literature review thoroughly covers the alternative definitions of modularity and relevant literary usage primarily from a engineering hardware product design viewpoint. Three perspectives are developed: the systems perspective; the hierarchical perspective; and the lifecycle perspective and like most multi-perspective views these are intended to give a complete and consistent description of the concept. It is suggested that only when a product can be described as being modular from all of these perspectives is it indeed fully modular. To a certain extent these perspectives are pretty much akin to Schilling's themes, and to a certain extent different and akin to my notion of domain. The notion that an architecture is only fully modular when all themes are satisfied in all domains is very much my assumed definition of an absolutely pure modular architecture except that I substitute the notion of domain for perspective and do not limit the physical domain's application to hardware products – the physical domain can as easily contain software code. The reason I substitute the notion of domain for Fixson's use of a perspective on a dimension is that a single perspective can view multiple domains. This point is emphasised and made easier to integrate with other work if the notion of domain is separated out from the notion of dimension.

Although the survey of Fixson (2001) is very thorough it does omit a line of work by Mikola (1998, 2000, 2001) that focuses on measuring the opportunity for modularisation in a given architecture. This develops a mathematically expressed modularisation function which yields reverse S-curves for given rigidities of interface constraints. Although the conclusions can be disputed this is interesting as it is the first such formal quantitative measurement and moreover one that explicitly defines the degree of modularisation as a continuum from fully integrated to fully modular.

Baldwin and Clark (2000, p.11) define modularity "***as a particular pattern of relationships between elements in a set of parameters, tasks, or people. Specifically, modularity is a nested hierarchical structure of interrelationships among the primary elements of the set.***" The first part of the definition is in accordance with my multi-domain definition, but the second part of the definition presupposes that all modular architectures possess open topologies and I shall attempt to show that this need not be the case (indeed some of Baldwin & Clark's examples show cross-linked structures). The subject of 'hierarchic nesting' recurs widely (as reported by Schilling, 2001) and it may be that the level of detail at which the non-hierarchic cross-linking occurs is simply not recognised when viewed from a perspective that does not value these cross-links at their true worth: this may be in issue in high dimension topologies.

Modularity as a strategy can be employed for different reasons. In the right circumstances it allows increased product or organisational[1] variety (Schaefer, 1999; Ulrich & Eppinger 1995, 2000) increased rates of technological or social innovation (Baldwin & Clark, 1993); reduces costs through re-use (Shirley, 1990); more sharply defines the opportunities for market dominance through interface capture (Moore, 1999); and the increased specialisation at firm level may allow more flexible response to environmental change (Sanchez & Mahoney, 1996).

The issue of change being enabled by modularity is discussed by Fine (1998). He points out that there is a correlation between horizontally structured industries and modular product architectures on the one hand, and vertically structured industries and integrated product architectures on the other. These extreme structures in combination with the forces of disintegration (competitor entry, breadth of challenge, organisational rigidity) and the forces of integration (concentrations of power, bundling opportunities, proprietary lock-ins) have the

[1] The development of organisational modules predates the development of product modules by many centuries. The evolutionary continuum is illustrated in the transitions from networked pre-historic clan structures; then layered feudal society; and then hierarchical industrial civilizations (e.g. the introduction of functions and departments).

potential to create a double helix. Some view this helix as being dynamic, but unless it is viewed as being an evolving spiral it can become a static repetition. Irrespective this is clearly a more holistic view of architecture as being a joint product-industry construct.

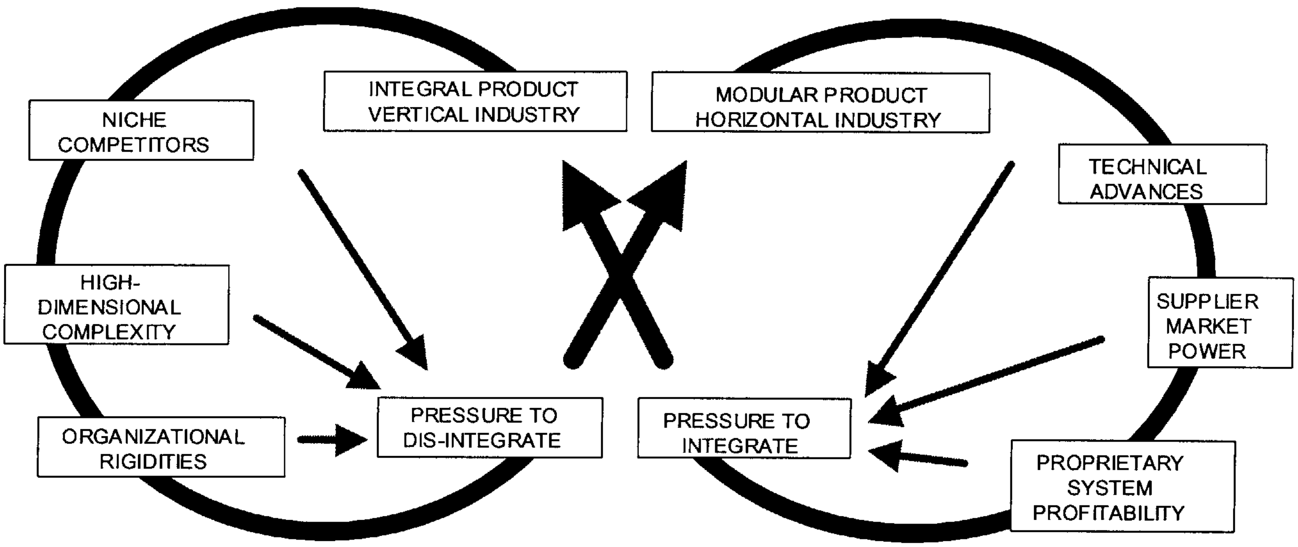


Figure 1: The Double Helix Of Industry & Product Structure (Fine & Whitney, 1996)

But modularity has drawbacks. In its extreme form the inter-module interfaces must be fully defined so as to allow change to occur within modules without adversely affecting inter-module working. This is beneficial only if the choice of element division into modules is optimal, if all elements are divided into unique modules, and if all inter-module relationships are then completely described in interface descriptions that also fully describe the emergent system-level characteristics. Such a harmonious state only occurs when designers well understand the product so that they are able to describe completely and clearly the visible design rules (i.e. the inter-module interface descriptions, Baldwin & Clark, 2000). If premature modularisation is attempted it adversely affects the system-level trade-space by imposing unnecessary or contradictory constraints. Furthermore the use of modularity to increase variety may have costs that exceed the advantages of scale and scope. For these reasons some literature argues that high-performance products tend to be more integrated in nature although this may reflect a poor understanding of the crucial details of the architectures under consideration, e.g. high performance combat aircraft, vessels, and vehicles share more sub-systems than are generally understood (e.g. missiles, radars, communications, electronic warfare suites, etc.).

7.4.1 Modularity Is Not Standardisation

Simply because a product is comprised of modules need not necessarily imply that a product cannot be to some degree unique. If modularisation is carried out within a carefully crafted set of design rules it actually increases the opportunities for customisation. Whether this customisation is superficial or deeper is again a separate issue. Likewise the simple act of standardisation does not automatically imply modularisation – many highly integrated processes and products have clearly defined standards that are only of value in the context of the whole. So although modularisation implies some degree of standardisation at the interface, one should not be lulled into thinking that the two are thoughtlessly interchangeable terms.

7.5 From Modularity To Option Value

In this section the relevant parts of Baldwin & Clark's (B&C) theory of design rules will be outlined. The formulae shown are Baldwin & Clarks' and the illustrative examples are my reproductions of the equivalent examples in their work. The purpose of recalculating their examples is that they well illustrate the development of Baldwin & Clark's argument, and give the reader some confidence in my ability to do the sums.

7.5.1 Introduction

Baldwin and Clark's (2000) theory of design rules 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 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

They propose that when a fully modular ("truly modular") design emerges the option space increases and the decision-making becomes decentralised. This is because once the design rules are codified new designers (of modules) are encouraged to enter the market and need not refer to the initial system architects. The codification of the design rules lowers entry costs for substitute modules, as modules may only need creating & testing in isolation. Furthermore, although any one individual new design (of a module) stands no greater chance of being an improvement, the overall chance of a better module design increases as more designers make the attempt.

After laying out a series of definitions Baldwin and Clark (B&C) propose that a modular design process has three basic stages:

1. The formulation of design rules.
2. Parallel work on hidden modules.
3. Testing and integration.

This then 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.

7.5.2 Benefits Of Splitting And Substitution

In the simplest form with no costs incurred, the overall value (performance[2]) of the system, X_{system}, is split or partitioned into a system level value measure, S_0, and j modules' value measures $(X_1, \ldots, X_j)$:

$$X_{system} = S_0 + \sum_{i=1}^{j} X_i$$

(Ref. B&C p.253)

where X_{system} represents the value of a system consisting of j modules,

[2] Value and performance are used interchangeably at this stage in the development of Baldwin & Clarks' work. Later on value becomes detached from performance: when market considerations cause cost to diverge from price.

and S_0 is considered to be a constant.

where X_i is the contribution to overall system value of the i th module

The module value for a given design is assumed to be normally distributed[3] with expectation of zero[4] and a variance proportional to complexity. So the expected value (V) of a one-module design is:

$$V_1 = S_o + E(X_N^+)$$

(Ref. B&C p.255 (Eq. 10.1))

where $E(X^+) = \int_0^\infty Xf(X)dX$

and X is normally distributed with mean 0 and variance $\sigma^2 N$,

and X^+ means that expectation only applies to outcomes above 0[5],

and $f(X)$ is the probability density function of a normal distribution

and 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:

$$V_j = S_o + E(X_1^+) + E(X_2^+) + ... + E(X_j^+)$$

(Ref. B&C p.259 (Eq. 10.2))

[3] Given that most natural phenomena are either normally or binomially distributed, and that the symmetrical normal distribution is more pessimistic than an asymmetric binomial, this appears a reasonable assumption.

[4] This is a conservative assumption, which implies that on average development efforts make no progress.

[5] This is because new designs with values of less than 0 are 'worse' than the current design and so are discarded (culled).

where X_j is the contribution to overall system value of the j th module

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 and it works because there is no cost (value penalty) associated with modularisation. Such a cost penalty is a special example of the more general point that any changes in the distribution of X must be considered carefully – for example if modularisation makes the distribution of outcomes worse then there may not be a net gain. As Baldwin and Clark point out premature modularisation of a system, before its design rules are understood, will tend to cause such a loss in value because it is unlikely that the system will be efficiently split.

Next Baldwin and Clark show that 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:

$$V_j = j^{1/2} V_1$$

(Ref. B&C p.261 (Eq. 10.3))

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:

$$V_j = (\alpha_1^{1/2} + \alpha_2^{1/2} + \dots \alpha_j^{1/2})\, V_1$$

$$\text{where } \sum_{i=1}^{j} \alpha = 1$$

(Ref. B&C p.263)

Now Baldwin and Clark 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:

$$V\ (\widetilde{X}_1 ... \widetilde{X}j) = S_o + Q(\widetilde{X}_1; k_1) + Q(\widetilde{X}_2; k_2) + ... + Q(\widetilde{X}_j; k_j)$$

(Ref. B&C p.264 (Eq. 10.4))

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

Once again the assumption is made that the payoff function is a standard normal distribution, and $\widetilde{X}$ is suppressed to reduce clutter so the value of the best of k designs is denoted as $Q(k)$[6] 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:

$$V(j,k) = S_0 + \sigma(N_j)^{½}.Q\ (k)$$

(Ref. B&C p.265 (Eq. 10.5))

This formula [B&C Eq. 10.5] represents the value of splitting a design into multiple modules and running multiple design trials for each module where the best outcome is substituted for the initial design. This function is graphed below in three dimensions with S_0 normalised to zero and the value of one experiment in a one-module design scaled to one, $V(1,1) = 1$, termed ***the value of the basic process***.

[6] See Appendix A for the calculation of Q(k).

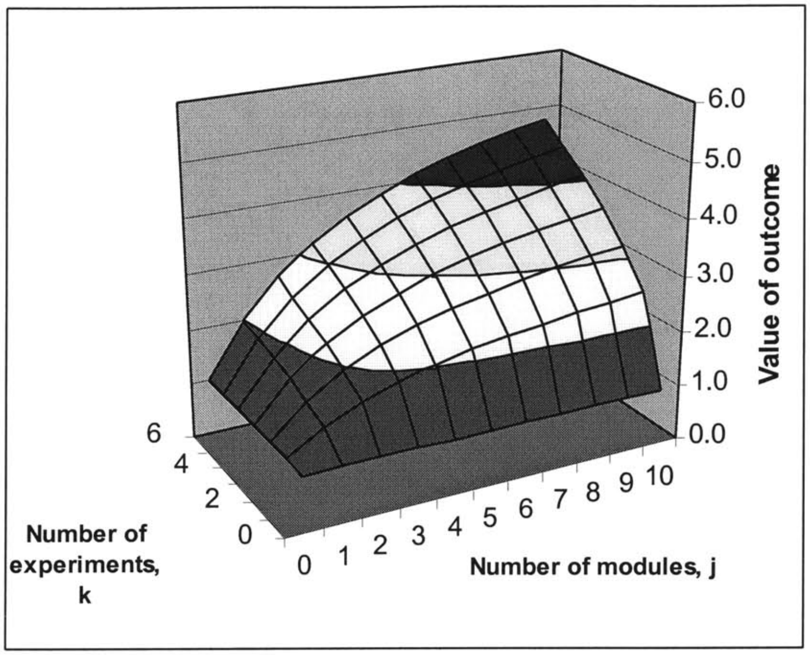


Figure 2: Value Of Splitting & Substitution With No Penalties (ref. B&C plate 10.1)

This graph (Figure 2) is a visual depiction of the potential benefits of modularisation for designs where there are no penalties; the coloured bands trace the iso-value contour lines. It shows that from a societal perspective not only is it desirable to split a design into modules and to run multiple independent design experiments, but that these interact together to create even more potential value for society. The value of the basic process is one (i.e. one design trial on a single module comprising the whole system creates one unit of value; V(1,1) = 1) whereas if, say, six experiments are conducted on a system divided into ten equal modules, the expectation is for almost five units of value to be created (i.e. V(10,6) = ~5). From a firm's perspective the benefit is more ambiguous as the increased substitutability of one module design for another reduces the firm's control, and by making market entry more attractive creates a risk of additional competition in all of the relevant module design spaces. Strategies to minimise this risk include reducing the appropriability of a module (e.g. by securing the intellectual property so that one can either deny entry to an alternative design, or harvest its value through a licence fee) or by retaining complementary assets of which the most powerful are by not fully/openly codifying the design rules such that the test and integration phase remains pivotal and/or a new 'release' of the architecture can be imposed on subsequent design generations.

7.5.3 Cost Of Splitting And Substitution

At this point Baldwin and Clark 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:

$$C(j,k) = c_j j + c_k k + T(j,k)$$

(Ref. B&C p.269 (Eq. 10.6))

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

$$NOV(j,k) = S_0 + \sigma(N_j)^{½}.Q\,(k) - c_j j - c_k k - T(j,k)$$

(Ref. B&C p.269 (Eq. 10.7))

The S_0 term of this 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.

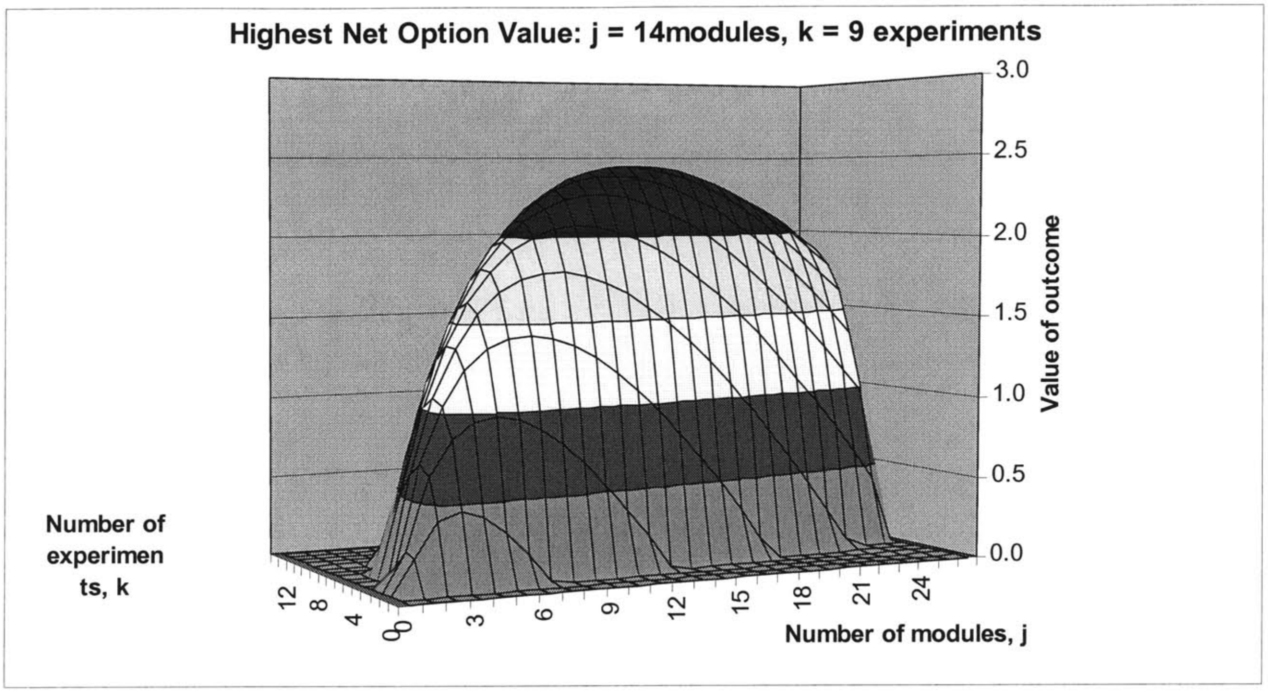


Figure 3: Net Option Value In The Absence Of Testing & Integration (ref. B&C plate 10.2)

Setting aside the stage 3 cost of testing and integration $T(j,k)$ this function can be graphed as in Figure 3 above. Once again this has been normalised with S_0 set to zero, the value of the basic process $V(1,1)$ set to one, and $c_j j = c_k k = ½$ so that NOV(1,1) = 0, i.e. it breaks even. The effect of including the stage 1 and 2 cost terms is to reduce the desirable parts of the solution space and to create a single peak in the objective function. So 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) of 14 modules with 9 experiments per module and very steep slopes towards value-destroying regions. Baldwin and Clark then point out that other cost relationships can yield multi-peaked solutions such as are shown in Figure 4 below where a fixed cost of ½ $V(1,1)$ is subtracted from all designs having more than one module to illustrate what happens if there is a fixed cost involved in creating a modular design.

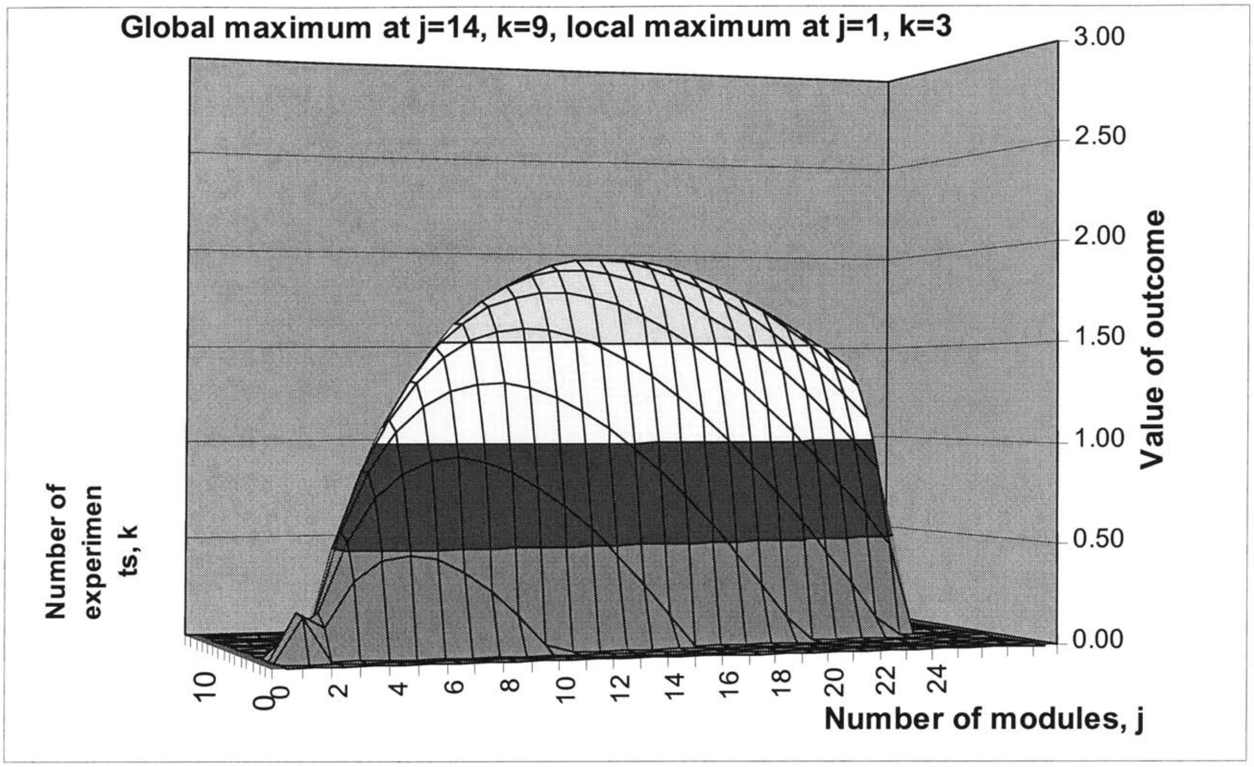


Figure 4: Net Option Value With Multi-Peaked Value Landscape (ref. B&C plate 10.3)

This example yields the same location for a global maximum as before at j=14, k=9, and has a local maximum at j=1, k=3. This is an entirely reasonable possibility for some systems and illustrates both the importance of understanding what the cost drivers are, and the care to be exercised if seeking to optimise value through simple hill-climbing heuristics such as many firms (or societies) tacitly employ. In this instance it is necessary to aggressively invest in both modularisation and experimentation to obtain positive results: firms (or societies) that are over-cautious might easily be trapped in the low-value state, whilst firms that seek over-controlled single-variable experimentation would be even worse off as on average they would squander value. Only the co-ordinated investment in discontinuous change will be successful. Baldwin and Clark suggest that three conditions must be met for an enterprise to succeed in making such a leap:

1. Designers must see that the value is there.
2. Those who perceive the value must have the resources to make the necessary investments.

3. The enterprise must have a means to capture the value it creates (and perhaps to reward its designers).

As they observe there are many and substantial barriers to meeting these three conditions.

7.5.3.1 System Level Tests

The inclusion of the stage 3 costs of testing and integration further reduce both the potential value and the number of desirable solutions. Given that Baldwin and Clark propose a two level hierarchy of system and module it is not surprising that they likewise propose a two level approach to 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. They point out that the extreme form of the system level test suffers from combinatorial explosion as, if the cost per test[7] is c_{ts}, then the overall cost $T_s(j,k)$ of testing j modules and k experiments per module is:

$$T_s(j,k) = [(k + 1)^j - 1]\ c_{ts}$$

(Ref. B&C p.274 (Eq. 10.8))[8]

Obviously this overall cost increases dramatically with either j or k. To illustrate this Figure 5 below is the same as Figure 3 above but with the inclusion of system level testing costs at 10% of V(1,1) and the other costs reduced so that NOV(1,1) is normalised at 0.0. Not only has the maximum NOV reduced from ~243% to a mere ~16%[9] but the envelope of positive outcomes has dwindled to a mere two from the ~300 previously seen.

---

[7] I depart slightly from the Baldwin and Clark notation of using c_t to denote both system level or module level testing costs. Instead I discriminate by using T_s, c_{ts} and T_m, c_{tm} to represent system and module level test costs respectively.

[8] 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.

[9] In these two cases I obtain a slightly different optimal result than Baldwin and Clark. These formulae are sensitive to rounding.

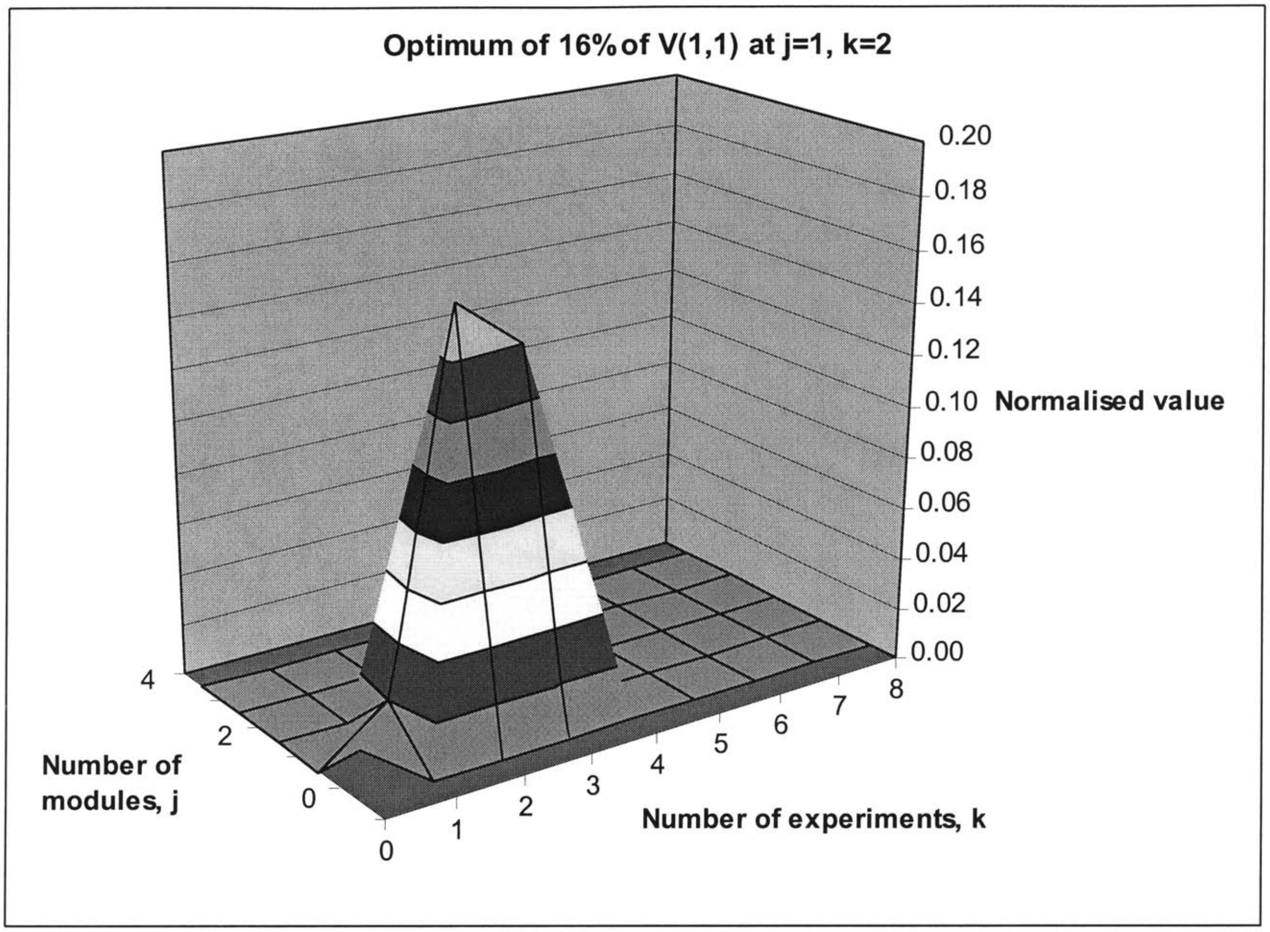


Figure 5: Net Option Value Of Splitting & Substitution With System-Level Tests

(c_{ts} = 0.1 V(1,1); $c_j = c_k$ = 0.45 V(1,1) so NOV(1,1) = 0.0); (ref. B&C plate 10.4a)

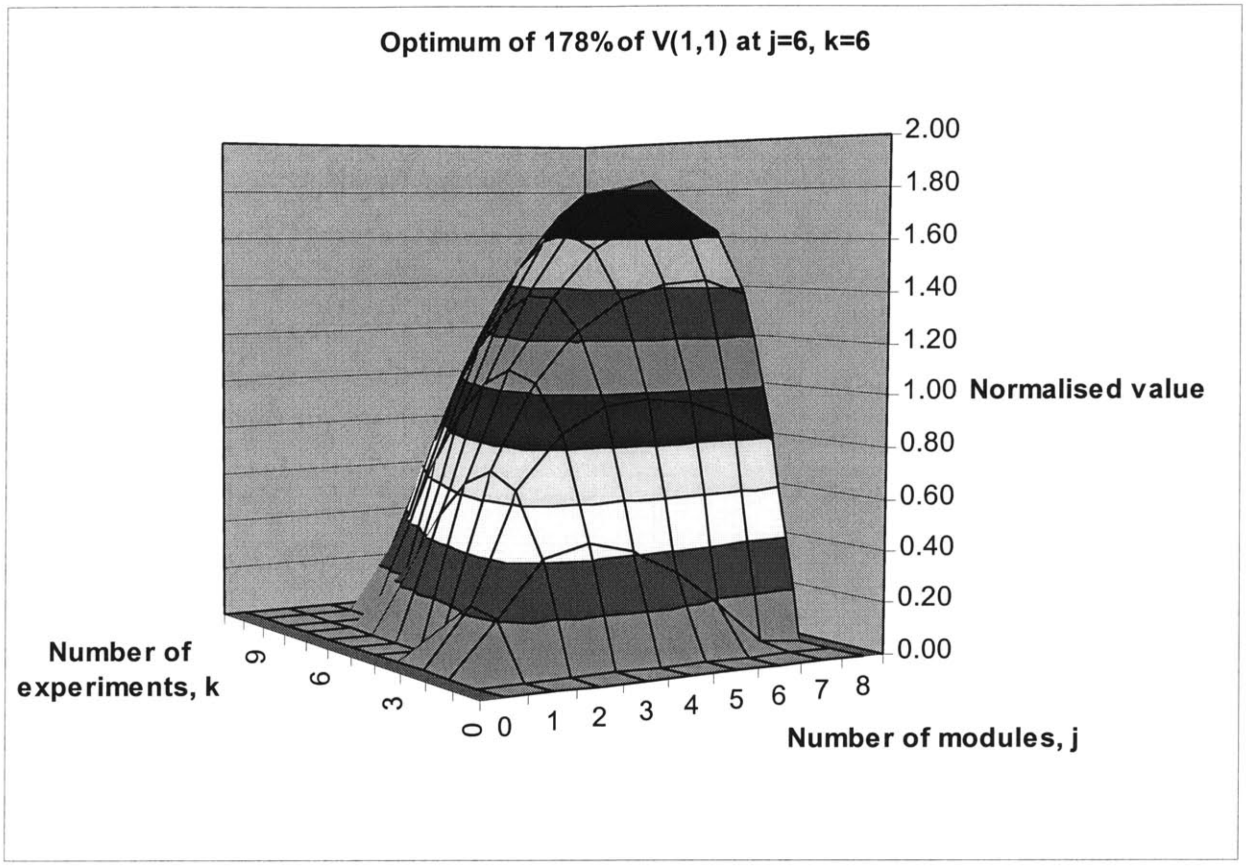


Figure 6: Net Option Value Of Splitting & Substitution With Very Low Cost System-Level Tests

(c_{ts} = 0.0000001 V(1,1); $c_j = c_k$ = 0.49999995 V(1,1) so NOV(1,1) = 0.00000000); (ref. B&C plate 10.4b)

Even reducing the system level testing costs by a factor of one million as shown in Figure 6 does not alter the basic 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 (Miller, 1956) may be founded on the application of this characteristic to natural systems) as costs increase like driving into a cliff.

7.5.3.2 Module Level Tests And Evolution

As design rules become understood sufficiently that modularisation becomes conceivable so it becomes possible to design tests for modules that do not rely on embedding them in prototypes. As Baldwin and Clark 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 and k experiments per module is:

$$T_m(j,k) = j \,.\, k \,.\, c_{tm}$$

(Ref. B&C p.278 (Eq. 10.9))

The effect of this is to outflank the combinatorial explosion problem of system level testing costs, so much higher levels of modularity and experimentation 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 testing costs and increasing the optimal numbers of modules and trials. This evolution requires investment in:

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

7.5.4 Unequal Modules

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:

$$NOV_{tot} = S_0 + NOV_1 + NOV_2 + \ldots + NOV_j$$

(Ref. B&C p.283 (Eq. 11.1))

and the individual terms expand:

$$NOV_i = \max\{\sigma_i (n_i)^{1/2} Q(k_i) - C_i(n_i)k_i - Z_i\}$$

(Ref. B&C p.284 (Eq. 11.2))

where n is the number of tasks needed to redesign the module, where σ is the standard deviation of potential value,

where Z is the visibility of the module to the system,

where max indicates the desire to maximise the value of any decision variables such as the number of experiments in the i th module, k_i.

An important point is the manner in which Baldwin and Clark define visibility as being "the number of other modules that 'see' the visible information contained in it". Since they have defined modularity as being a nested hierarchy, in their schema it is only necessary to consider the modules that "look up" either directly or indirectly as described in a system design hierarchy diagram (c.f. section 11.7.2.2 for more detailed visibility discussions). Also they make it quite clear that not all the information in a module need be visible – this can give rise to legacy design rules.

Given that the main variables in this 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 as shown in Figure 7:

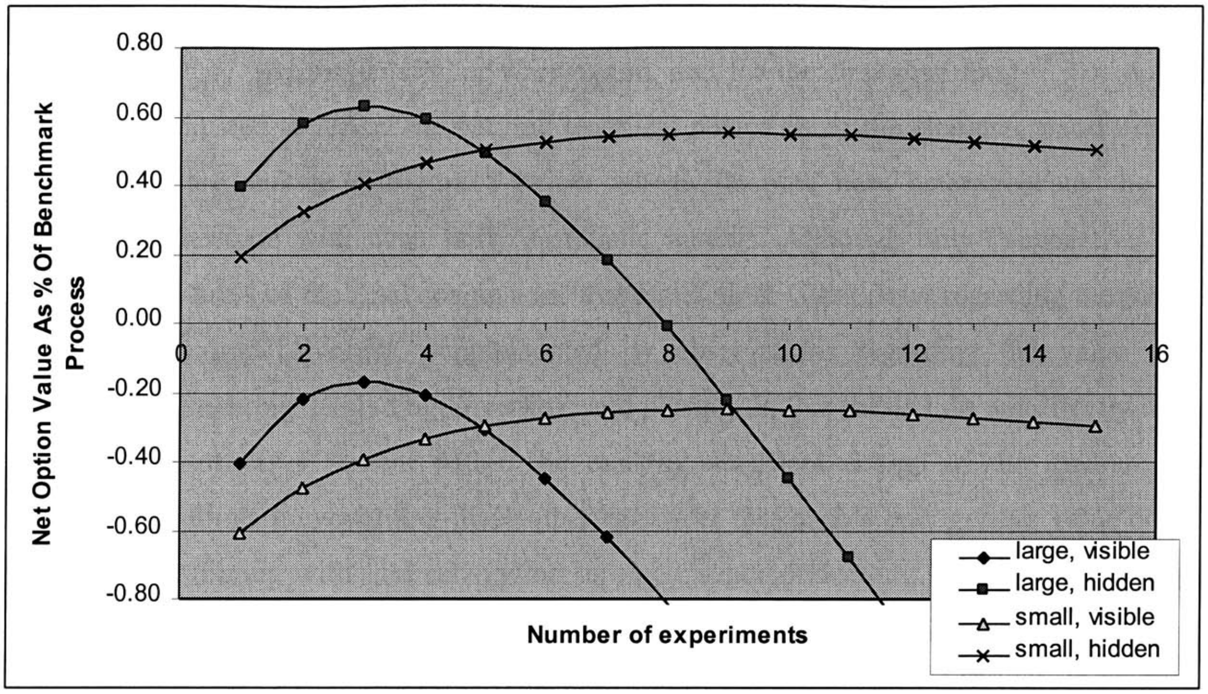


Figure 7: Module Net Option Value For Different Combinations Of Size And Visibility[10]

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[11]. Note how some combinations destroy value as benefits fail to meet costs.

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 7 above, 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.

[10] This example uses slightly different seed values than the equivalent example in Baldwin and Clark, 2000 (Figure 11.1). A negative NOV indicates value destruction as costs of experimentation outweigh benefits.

[11] It seems to me that the more fundamental a theory the more likely it is to explain a wide range of particular results without needing to resort to specialised cases.

The next step is to apply this to a complex product with varying technical potentials. This is done for an early (1990) personal computer workstation, and for the first time the 'σ' or potential value term is adjusted to reflect the varying technical potentials of the elements (modules). The results of this are shown in Figure 8 below which illustrate how heterogeneous the value landscape can become with even fairly simplistic models. Although only "suggestive" these results form the basis of the final conclusions that Baldwin & Clark draw regarding the intensity of sub-market competitiveness, supplemented by observations regarding the value of the operators of augmenting, excluding, inverting, and porting. In this thesis I do not directly discuss the other operators as I remain within the existing dominant design of the medium sized industrial gas turbine, however it is these other operators that enable endogenous value creation through e.g. combining with fuel cells, wind turbines, superconductors, etc. to potentially yield other dominant designs.

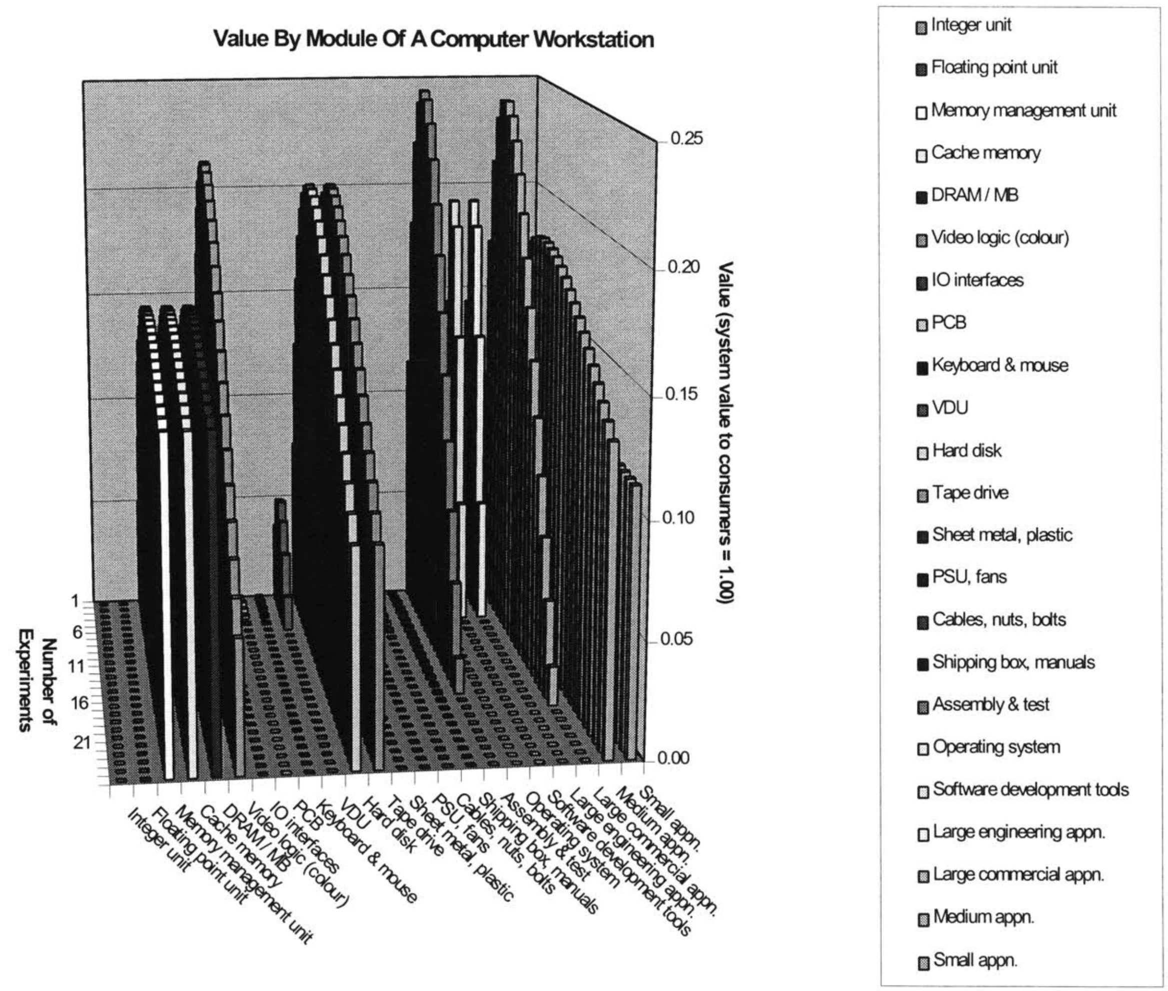
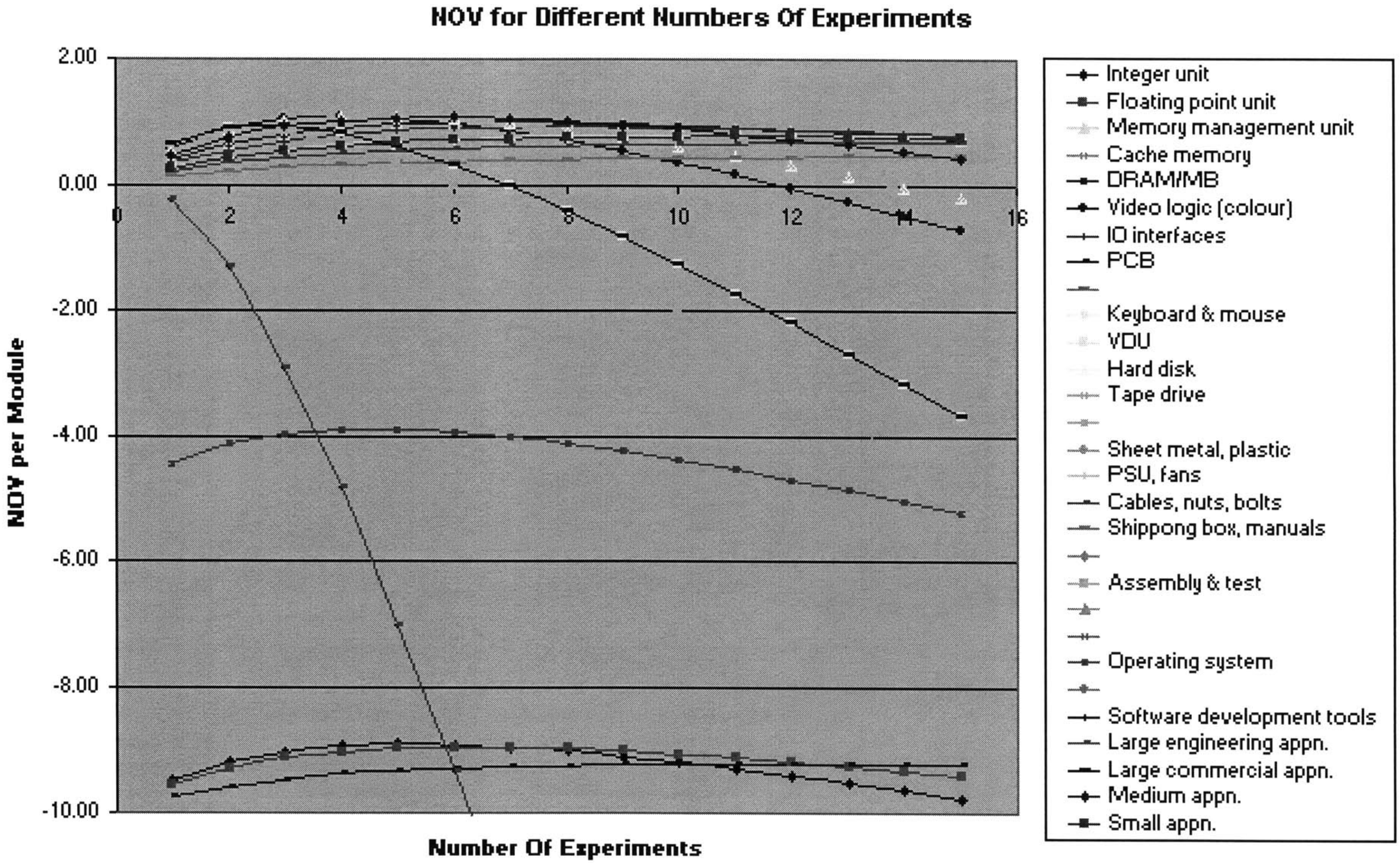


Figure 8: Two Views Of Value By Module Of A Computer Workstation[12]

[12] The two views of this example shown on adjacent pages use slightly different scaling assumptions than the equivalent Baldwin &Clark example (Ref. plate 11.1). The view above is that shown in B&C whilst the one on the next page is the representation more normally used in this thesis.


NOV for Different Numbers Of Experiments
NOV per Module
Number Of Experiments
2.00
0.00
-2.00
-4.00
-6.00
-8.00
-10.00
0
2
4
6
8
10
12
14
16
Integer unit
Floating point unit
Memory management unit
Cache memory
DRAM/MB
Video logic (colour)
IO interfaces
PCB
Keyboard & mouse
VDU
Hard disk
Tape drive
Sheet metal, plastic
PSU, fans
Cables, nuts, bolts
Shippong box, manuals
Assembly & test
Operating system
Software development tools
Large engineering appn.
Large commercial appn.
Medium appn.
Small appn.

8 BACKGROUND: DESIGN DOMAINS

8.1 Introduction

This chapter explains what domains are and outlines three different ways of describing information in a domain. It also touches on dimensions, inter-domain transforms, decomposition, and stages in the product lifecycle.

8.2 Domain Definition

Engineered products can be thought of as being the result of multiple sorts of interactions. For example a product can be defined as the result of **tasks** performed by an **organisation** comprising **people** creating a **physical** embodiment that performs **functions**. In this example there are five domains (people, organisation, tasks, physical, function) and within each domain there can be multiple dimensions – for example the physical domain may have dimensions of **force**, **momentum**, **mass**, **energy**, and **information** (after Pimmler, 1994).

This notion of domains is useful inasmuch as it allows precise definition of what comprises a product in a consistent and complete manner. As yet there is not full agreement in the literature on how domains and dimensions should be defined and indeed it may never be possible to reconcile the alternative views. This disagreement may simply be a natural outcome of there being more than one consistent and complete scheme for describing a product. Alternatively it may be that all proposed schema are to some degree incomplete and inconsistent and therefore the selection of a scheme for any particular purpose should depend upon which has greatest utility for solving the question in hand.

8.2.1 Domains Usage Per Axiomatic Design

The most advanced work on defining domains in a formal manner has been carried out under the name axiomatic design (Suh; 1996, 2001) and which recognises four domains of the design world: the customer, functional, physical, and process domains.

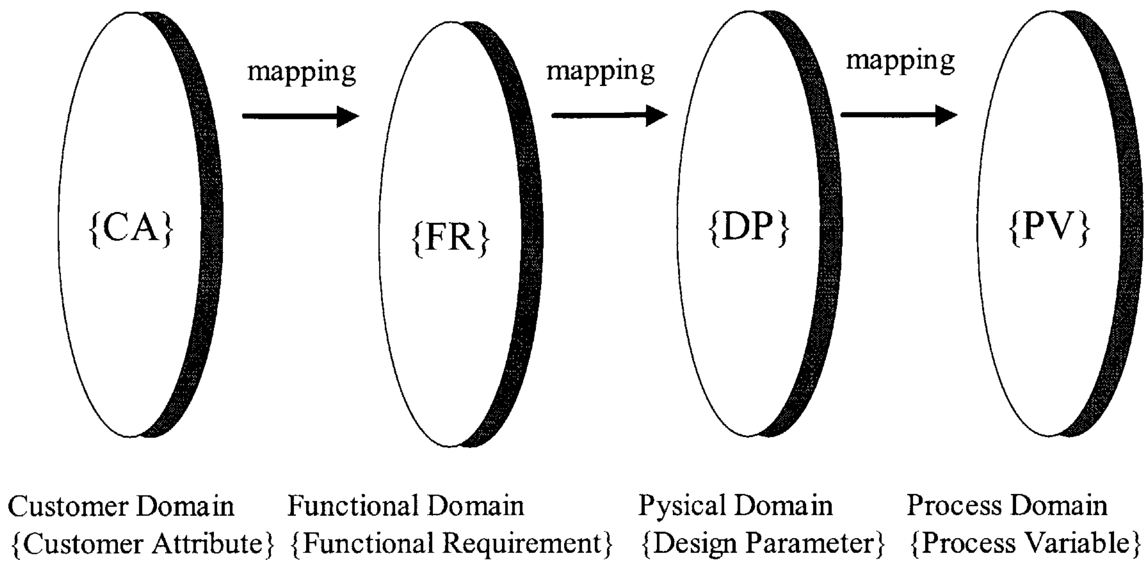


Figure 9: The Four Domains Of The Design World per Axiomatic Design[13]

Axiomatic design is interesting inasmuch as it places great emphasis on the mapping process that takes place from one domain to another. These mappings are described by transform matrices or design matrices that show one domain on each axis and where the interactions are recorded within the matrix as illustrated in Figure 10 below. Suh proposes a series of axioms that, if followed, result in better designs. A key element of this is the need to zig-zag backwards and forwards amongst the domains as one decomposes the problem statement and formulates alternative concepts for satisficing the customer. These transform matrices need not be square nor symmetrical and may be very complicated. They only describe forward flows of information from one domain to another and any backwards flow is described by the zig-zagging process illustrated in Figure 11 combined with rules of consistent decomposition.

A problem with axiomatic design's propositions is that they are highly dependent on the appropriate phrasing of the elements of each domain. However even cursory examination of this work, and of a series of products designed by applying these axioms, suggests that the concept of domains, the process of mapping between domains, and the relationships within domains are important.

[13] After Suh, 2001.

Design Matrix Mapping Second Level Process (Functional Requirement) to Form (Design Parameter)

		8	9	10	11	12	13	14	15	26	27	28	29	30	23	24	25	16	17	18	19	20	21	22	1	2	3	4	5	6	7
		HPU	Plumbing	Electronic controller	Communication	Instrumentation	Wiring loom	UPS	PSU	Tower	Nacelle	Foundation	Slew ring support frame	Access & conduits	Slew ring assy.	Yaw mechanism	Chassis & mounts	Hub	Blades	Aero brakes	HSS	LSS	Shaft brake	Gearbox & coupling	Generator	Switchgear	Power electronics	Cabling	Cooling	Soft start	Transformer
FR 11	Maintaining aerofoil surface at desired angle of attack																	X	X												
FR 12	Maintaining rotor disc perpendicular to wind	X	X	X		X	X		X	X	X	X	X		X	X	X														
FR 21	Transmitting lift forces along rotor blade to hub																	X	X												
FR 22	Constraining hub in all but one (axial) degree of freedom									X	X	X	X		X	X	X	X													
FR 23	Transmitting reactive forces									X	X	X	X		X	X	X	X	X												
FR 31	Constraining LS shaft to only desired axial rotation (plus acceptable float)									X	X	X	X		X	X	X					X		X							
FR 41	Connecting LSS to driven gear																					X		X							
FR 42	Constraining driven gear in situ, meshed with similarly constrained driving gear																							X							
FR 43	Connecting HSS to driving gear																				X			X							
FR 44	Transmitting reactive forces									X	X	X	X		X		X														
FR 51	Constraining HS shaft to only desired axial rotation (plus acceptable float)									X	X	X	X		X	X	X					X		X							
FR 61	Rotating conducting circuits in magnetic field																								X				X	X	
FR 62	Constraining parts									X							X								X						
FR 63	Transmitting reactive forces									X	X	X	X		X		X							X	X					X	
FR 71	Providing continuous conductive / insulated path													X	X											X		X			
FR 81	Transmitting electrical waveform through electrical filter (i.e. remove high frequency components)																								X	X	X	X			X
FR 82	Adjusting power factor of waveform																														X
FR 83	Maintaining constant frequency waveform (dependent on concept this is executed different ways, e.g. via variable pitch or via inverters)			X		X												X													X
FR 91	Providing continuous conductive / insulated path																								X	X	X	X			X
FR 101 •	Various FRs to do with ease of manufacturing, installation, cost, etc.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
FR 111 •	Various FRs re likely peak stresses, fatigue, design lifecycles, etc.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
FR 121 •	Slowing, abstracting energy, heating (via turbulence), transforming to noise, etc.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
FR 131 •	Using low energy readily available materials requiring little special skills & tools, etc.	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X

Figure 10: Example Design Matrix For A Wind Turbine

DELIBERATELY BLANK

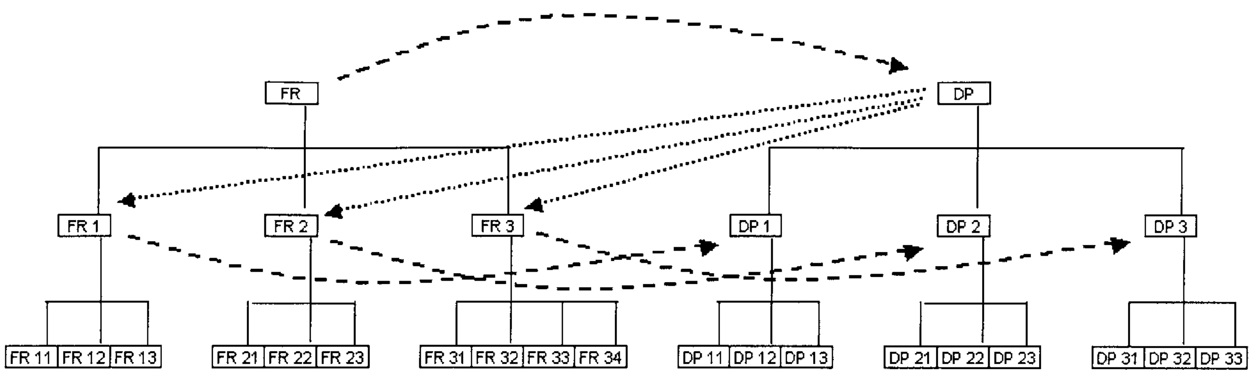


Figure 11: Zigzagging Between Domains per Suh To Decompose A Problem

8.2.2 Domain Usage In Quality Function Deployment Matrices

An alternative view of domains is implicit in the concept of Quality Function Deployment Matrices (QFD matrices) that are often known as a House Of Quality because of the roof-shaped triangle on top. These are a similar attempt to define domains and to flow information forwards through a design process and come from work carried out by Mizuno for Mitsubishi Kobe Shipyards in 1972 (Hauser and Clausing, 1998). As with Suh's four-domain description of the design world there are four formal domains in the QFD process and the matrices represent transforms from one domain to another. Because of their application in systematically and traceably decomposing high level needs they need not be square and often expand greatly in size as needs are flown down to lower level specifications in other domains.

The user has great flexibility to define the domains of a QFD and some users go so far as to define different domains for different stages of a product or system's lifecycle giving rise to a 'stack' of QFDs. As with Suh's design world there is no feedback path between the domains, but unlike Suh it is difficult to separate out the levels of decomposition of the problem[14].

The roof-shaped triangle on top of a QFD represents the relationships between elements of the vertical columns. This is typically used to describe binary (i.e. do exist or do not exist) conflicts between elements, and beneficial positive relationships are not normally included. Implicit in the

[14] QFDs have had a somewhat mixed reception as Whitney, 1996 suggests with "QFDs or other semi-mystical attempts to convert what the customer wants into hard engineering specifications.". In this paper Whitney makes the pertinent point that unless there is a rigorous understanding of the underlying phenomena it is not possible to identify design pathways. He goes on to point out those products that rely on system level emergent properties are hard to model a priori. This applies to most high-energy systems and so reduces opportunities for modularisation.

triangular shape of the roof is the notion that elements possess symmetrical relationships in any given domain.

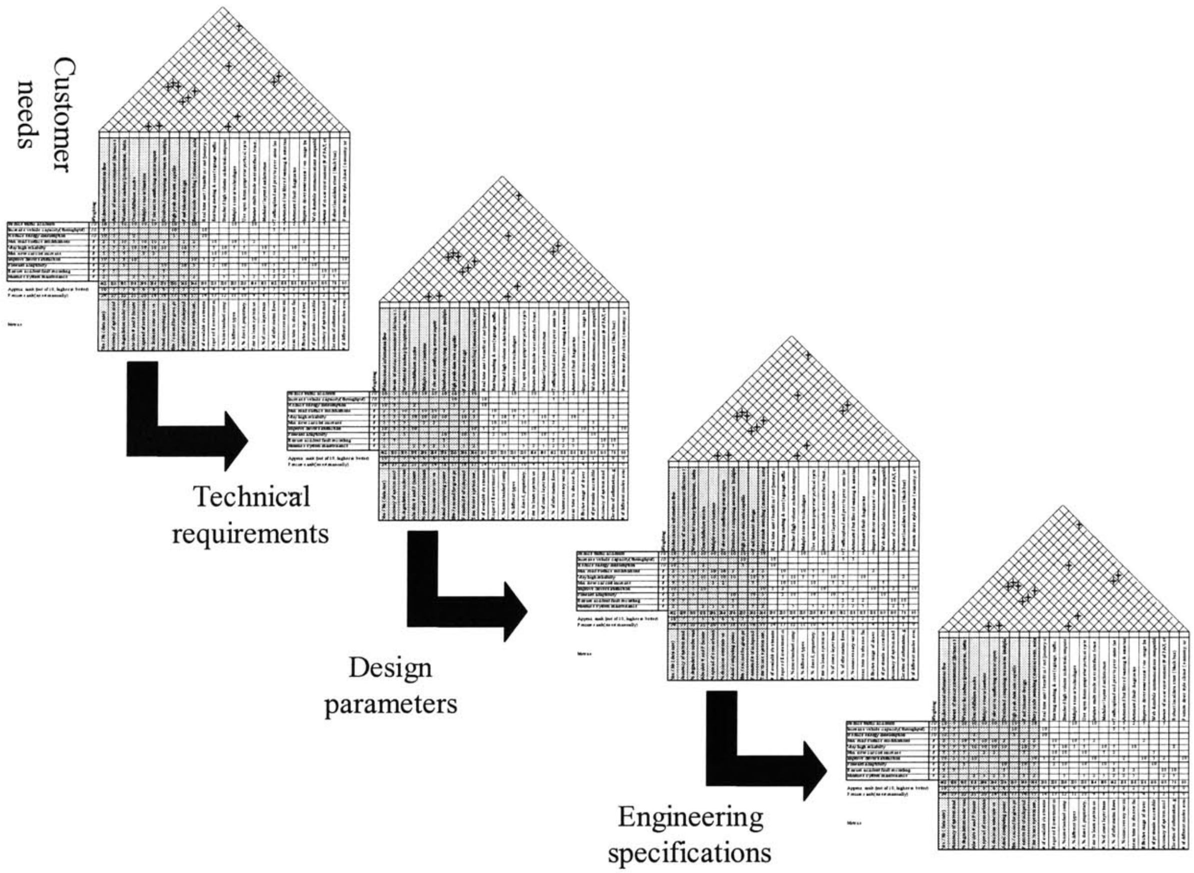


Figure 12: Typical QFD View Of Domains

A cartoon of a QFD flow down is shown in Figure 12; alternative domain definitions might equally well be customer wants > design requirements > part characteristics > key process operations > production requirements.

8.2.3 Domain Usage In Dependency Structure Matrices

A Dependency Structure Matrix (DSM) is a means of representing the interactions between elements[15]. In its classical form the same listing of elements takes place in the same order on two orthogonal axes and interactions between elements are recorded in the matrix. In a classical DSM the element names are in the same domain on both axes, e.g. a task domain DSM that has

[15] For an in-depth description of what DSMs are, and various ways of using them, see a comprehensive survey and collection of papers at http://mit.edu/dsm/ , which is the "MIT Design Structure Matrix (DSM) Home Page".

an element label "handle" is representing the task "design handle" and should not be confused with the physical domain label "handle" which represents the physical artefact itself. Thus a DSM is extremely similar to the 'roof' of the House Of Quality except that where the roof assumes symmetrical and purely binary negative relationships a DSM has wider application.

Mainstream DSM literature recognises three primary domains of interest: the task domain which describes the process of doing something; the physical domain which represents the thing itself irrespective of whether it be a hardware or software artefact; and the organisational domain which represents the groups of people who perform tasks. These three primary domains are well described by Eppinger et al (e.g. Rowles, 1999; Eppinger & Salminen, 2001) and depicted by diagrams similar to Figure 13 below.

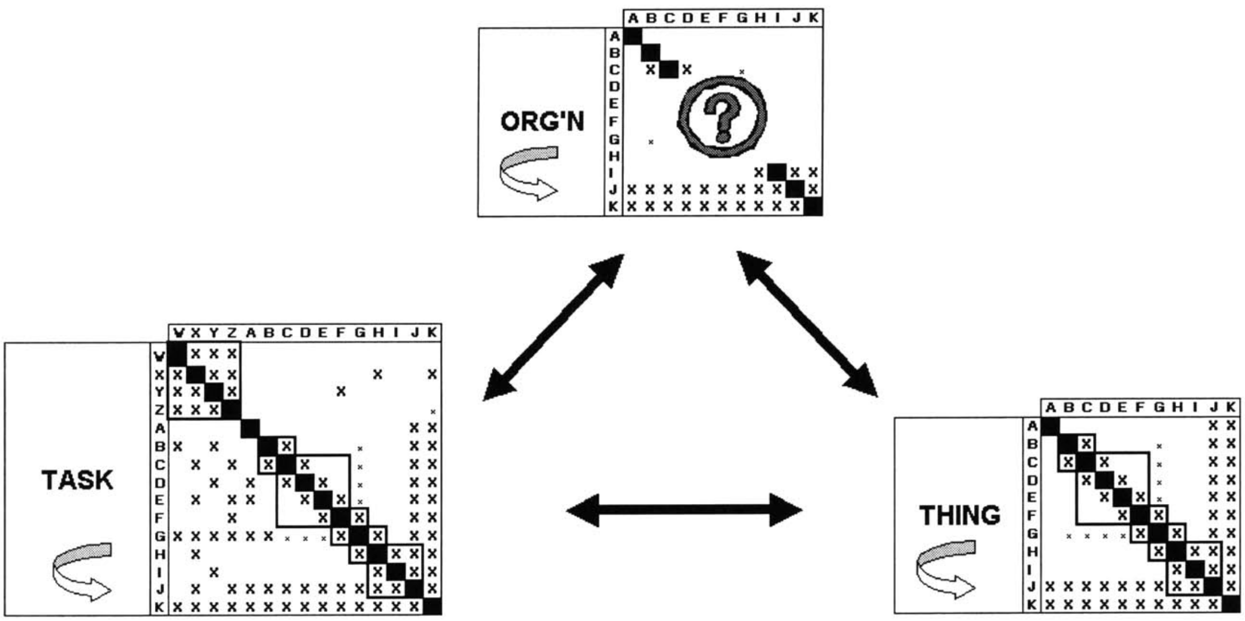


Figure 13: The Three Primary Domains Of DSM Usage

(after Eppinger & Salminen, 2001)

Obtaining satisfactory data regarding the mapping processes between these domains, and then analysing it to test hypotheses is a non-trivial task. For this reason Eppinger et al have so far made limited progress but results to date indicate very strongly that there is frequently a one to one mapping between both elements and relationships in the task (or process) domain and the physical (or thing) domain. In literature to date it appears that 'missing' one to one mappings have occurred as a result of poor designs (of the process and/or thing) or as a result of inability to measure mappings that did in fact occur. As yet there is a sparse data set regarding the other possible transforms.

A fourth domain representing the functions to be implemented by the artefact's elements has been probed but to a lesser extent. Furthermore implicit in the notion of the organisational domain is the notion of a fifth domain where individual people relate to each other in a way that may or may not be different than that described by the organisational domain. Thus overall it may be that there are five domains as depicted in

Figure 14, which may be related by predictable patterns of transforms. If one considers axiomatic design's other domains it is conceivable that there are additional domains and perhaps the easiest way to investigate them is to consider each of the three primary domains (organisation, task, physical) as having supporting or dependent relationships in subsidiary domains.

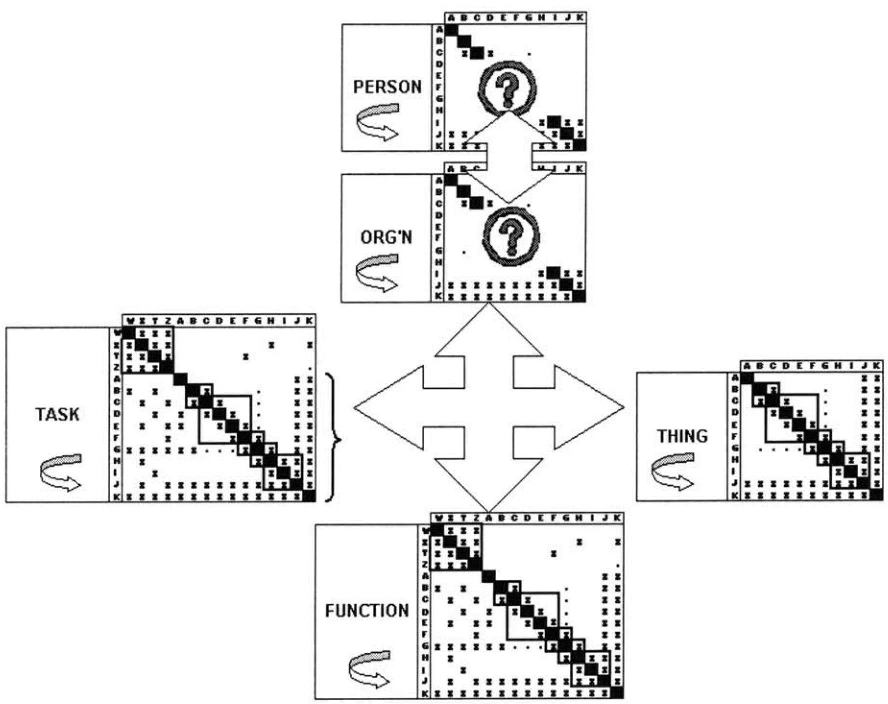


Figure 14: Five Possible Domains As Represented By DSMs

The notion of a DSM is independent of the notion of decomposition and so DSMs are better able to remain consistent in their depiction of reality than QFDs. They also are good at maximising the ability of normal humans to perceive innate structure.

A complicating factor that is as yet inadequately described by domains is the issue of the multiple stages in a product's lifecycle. This is relevant as the balance of power across the lifecycle governs who will be able to extract the surplus value that it is commonly the intent of the system to generate. This is an unresolved question.

9 BACKGROUND: GAS TURBINE GENERATOR SETS

This chapter explains what gas turbines are, why they came into being, what they are used for, and who makes them. This explanation is intended to aid understanding of the worked example of the valuation of a medium sized industrial gas turbine that is threaded through much of the remainder of this thesis.

9.1 A Brief History Of Gas Turbines

Gas turbines were developed in the late 1930s as the limits of the existing internal combustion engines became apparent for aviation use. Immediately before and during the Second World War the British, German, and Italians made intensive development efforts, which resulted in the production of the first jet-propelled combat aircraft in the closing years of the war and their subsequent widespread use in commercial aviation. After the war attempts were made to build gas turbines purely for marine propulsion but these failed and instead so-called aero-derivatives were placed into successful service in a variety of medium sized naval vessels of up to 20,000 tonnes. This brought them to a level of maturity where they were then adapted to commercial use in stationary electrical power generating plant or for driving high speed rotating equipment such as gas compressors or high volume liquid pumps. From the late 1960s onwards there was a resurgence of interest in the larger sizes of gas turbines and successful non-aero derivatives were manufactured. Today typical industrial gas turbine gen sets span the range of 200kW to 250MW and many thousands of units are in use worldwide. Research and development efforts continue to stretch both the upper and lower bounds of power output, and to improve the fuel efficiency and operating characteristics (reliability, cost, start-up, load response, weight, etc.).

9.2 Description Of A Gas Turbine Generator Set

The generic physical design of a nominal 10 MWe[16] simple cycle industrial gas turbine is shown in Figure 15 below[17]. This depicts the thirty-one major assemblies and sub systems that comprise

[16] Mega Watt electricity; MWe is the nominal useful electrical generating capacity of the gen-set.

[17] The data in this section is based upon my experience working with industrial gas turbines and supplemented by discussions with fellow students engaged in the design of gas turbines. My experience is in the nominal electrical power output range from 1.5 MWe to 15 MWe as supplied by a variety of manufacturers such as Rolls Royce,

a generation package, or "gen-set". In defining thirty-one major sub-systems at a second level of physical domain decomposition value judgements have been made about how to cleave the design in order to identify salient architectural features. Whilst it can be argued that this value judgement invalidates any further analysis it is necessary to start somewhere and it is unrealistic to start at part or feature (i.e. sub-part) level such as individual blades, cams, or lines of software code as will be discussed later.

The heart of a gas turbine gen-set is the gas generator where the rotating blades of the multi-stage compressor are mounted inside a casing and force air into a burner where it is mixed with fuel and the mixture ignited. The resulting rise in volume due to ignition of the fuel causes the expanding hot gases to seek an exit and the cross-sections and pressure profile of the gas turbine are such that it is easiest for the hot gases to flow out via the turbine rather than forwards through the compressor. As the hot gases pas through the turbine they drive the turbine shaft, which drives forwards to the compressor and so maintains a constant air flow rate and constant flame front in the burner. In some designs there may be multiple spools of compressor-turbine pairs that rotate on concentric shafts to increase efficiency but this detail is not considered here. Downstream of the turbine is the free power turbine that is spun by the remainder of the energy in the still expanding exhaust gases before they turn the corner via the exhaust transition duct and exit to atmosphere via the chimneystack. In some designs the hot exhaust gases then pass through heat exchangers to create steam that drives a steam turbine and thus increases overall efficiency but this analysis concentrates on the single (or simple cycle) design illustrated. The free power turbine drives a generator, often via a gearbox. There are various shaft couplings and gear drives for auxiliaries and starter motors. The compressor has inlet guide vanes that are manoeuvrable stator blades that swivel to direct the airflow at the optimal angle depending on the load condition.

All of the turbine shafts, the gearbox, and the generator contain thrust and journal bearings that require cooling and lubricating. The lubricating oil is forced around a closed loop from a holding tank, through a pump, filters, a distribution manifold, pressure regulators, flow regulators, the

---

Rustons (then part of European Gas Turbines which has since been acquired by Rolls Royce) and Solar (owned by Caterpillar). Some of these were aero derivatives (e.g. Avon, Olympus, Tyne, and Spey) whilst others were pure industrials (e.g. TB1750 and Mars) and which are sometimes known as 'frame' turbines in the larger sizes.

bearings themselves, coolers, chip collectors, and back into the tank. Dependent on the design some lubricating oil may pass into a hydraulic system to power actuators and controls. There are typically two or three electric pumps plus a back-up gear driven pump that maintains a minimum flow to the turbine in case of power failure so that a controlled shutdown may take place without damage.

The burner assembly typically has several burner cans that are each capable of accepting liquid and gaseous fuels. The liquid fuel supply system typically has an external tank, filters, multiple pumps, a fuel manifold, and manifold control valves with spillback loops. The gas fuel supply system is similar albeit with compressors in place of pumps. A crucial component in both is the fuel supply controller, which governs the energy output of the entire gen-set and which is best thought of as a separate subsystem.

Between the compressor and the combustor high-pressure air is drawn off for use in cooling and actuation. Different manufacturers have different terminology but this is often called "P2 air" and is filtered, dried, regulated, and supplied to pneumatic controllers and as coolant flow to the burners and hotter blades. From an efficiency standpoint this is expensive air and so there may be a supplementary supply that uses standard industrial air compressors for the lower pressure uses so as to alleviate the load although in more modern turbines electronic controllers have supplanted the pneumatic controllers' requirements.

A turbine's efficiency is very dependent upon the surface roughness of the compressor blades and the initial intake pressure. For this reason it is necessary to filter the air to rid it of particulate pollutants and to do so in a manner which requires the minimum pressure drop between atmospheric pressure and the fan face. This requires well-designed air intake ducts, plenum chambers, bag / screen / mesh filters and turbine nosecone. Despite the best efforts of the designer some contaminants will be ingested (or may originate inside the engine, e.g. oil leaks) and build-up on the blades causing fouling. Various types of wash systems are built in to allow the fouling to be removed whilst the gen-set is in service, or at least without the need to strip the gen-set down. Typically these consist of a tank of wash liquid, a pump, and nozzles which are mounted downstream of the filters but upstream of the fan face.

The generator supplies alternating current to a switchboard via incoming connections. The switchboard has a series of bus bars onto which tap racks of breakers, contactors, fuses, and their associated instrumentation. Leaving the switchboard are the outgoing cables to a step-up transformer and the users' system. The instrumentation in the switchboard feeds the governor and the load-share/load-shed system of the user. There will also be a series of auxiliary power units or supplies (APU) to drive turbine auxiliaries, starter motors, etc. via step down transformers and uninterruptible power supplies (UPS) so that the generator can run down in a predictable manner to a safe condition in the even of a system failure, and then be restarted. These APUs and UPSs include battery banks, external feeds, and occasional small internal combustion engines.

The gas generator, the gearbox, and the generator are mounted onto a rigid bedplate that in turn is mounted on the foundations of the user. Often the bedplate contains the lubricating oil tanks and has mounted onto it many of the auxiliaries such as pumps, coolers, filters, etc. so as to minimise the need for transition pieces capable of accepting vibrations however this is somewhat dependent on turbine size and the design philosophy. This bedplate and the associated tightly mounted equipment are often termed the "turbine island" and some are contained within an acoustic enclosure or package enclosure that also serves to enclose the thermal signature of the turbine and any fumes. In the case of a fire the acoustic enclosure can be flooded with extinguishants such as CO2 or water fog, triggered by fire and gas detection equipment. In some designs the entire bedplate and turbine island are mounted within a single acoustic enclosure and in others the generator and gearbox are located outside the main acoustic enclosure.

It is important to maintain the acoustic enclosure, the auxiliaries, and the operations personnel within tolerable conditions and so a heating, ventilation, and air conditioning (HVAC) package is included. Sometimes this is mounted on top of the acoustic enclosure and sometimes it is located next to the primary air intakes.

It is difficult to obtain access to many of these systems, yet it is undesirable to have such a high capital expense item as a gen-set out of service for maintenance. For this reason there are a variety of maintenance access systems built into gen-sets ranging from borescope access ports or simple inspection & maintenance hatches to lifting points, spreader beams, heavy duty travelling cranes, davits, and soft-patches.

Because of the noise, the poor sight lines, the difficulty of rapid egress, and the danger of automatic fire suppression equipment being triggered most gen-sets are equipped with status indicators for the benefit of operations personnel. These may simply be a series of coloured lights or may include alarm bells and communications relay equipment (such as additional loudspeakers or magnetic-loop systems).

The gen-set is monitored and controlled through a variety of pieces of software, input/output devices and converters, sensors and actuators. These are best visualised as 'layers' for the purpose of this thesis although they can be broken down into more detailed elements if useful.

Outside of the gen-set there are many other systems in the typical end-user environment. For the purpose of this thesis the upstream system boundary is defined as the interconnection to the atmosphere, the bulk fuel storage, and any external control input (e.g. from a external SCADA layer). The downstream boundary is defined as being the exhaust stack limit and the switchboard contactor to step up transformer. The foundations and civil structures, spares and consumables stores, the operations and maintenance records and the operations & maintenance staff are defined as being outside the system, but the package's standard delivery documentation (i.e. the operations and maintenance manuals) are within the system. In essence this is what one would expect to find if one walked into a turbine hall to find a turbine that had been pre-commissioned but not yet commissioned. Together these form thirty-one major physical 'chunks' or primitive elements.

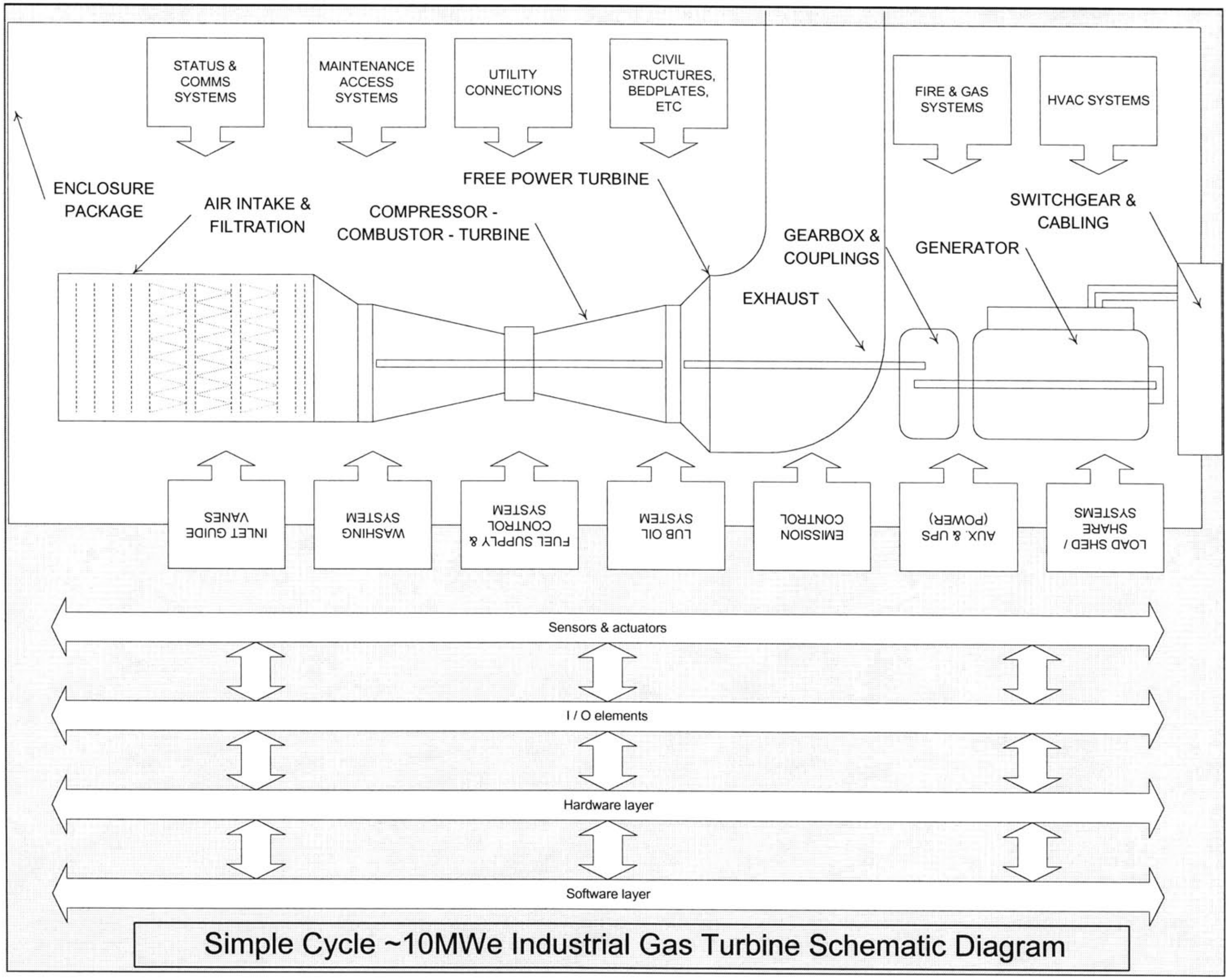


Figure 15: Schematic Of Typical ~10MWe Industrial Gas Turbine

9.3 Gas Turbine Manufacturers

As well as various niche manufacturers there are half a dozen significant players in the manufacture of gas turbine gen-sets including Rolls-Royce, General Electric Corporation, United Technologies Corporation's Pratt & Whitney Division, Mitsubishi Heavy Industries, Caterpillar's Solar Division, Alsthom Power (which includes the ex ABB turbine division), and Westinghouse. Not all of these compete across the whole range of sizes and many have entered into alliances to enter particular markets. This has the effect that at any one gen-set size there are typically four or so competitive manufacturers.

These manufacturers rarely make all of the gen-set themselves but a similar pattern emerges in the various sub-system levels. Although the thrust of this thesis is to demonstrate why there should be an optimum number of players in any portion of the market there will be no attempt made to 'prove' the validity of the hypothesis given the level of detail involved. Instead it will be sufficient to observe gross differences such as if less than two or more than ten players are optimal.

10 INITIAL VALUATION OF A GAS TURBINE GENERATOR SET ARCHITECTURE

In this chapter an initial valuation is made of the gas turbine sub-systems, identified in the last chapter, using Baldwin & Clark's theory of option value. Various assumptions are made and then challenged. This leads to changes in the assumptions, the valuation process, and the valuation itself. Useful results emerge which are then contrasted with more idealised results obtained if no sub-systems are identified a priori.

10.1 Initial Decomposition

As described in the previous chapter the gas turbine genset is decomposable to 31 identifiable sub-systems (or sub-sub-systems, e.g. compressor or free power turbine) per the diagram below, which is a highly stylised version of Figure 15: Schematic Of Typical ~10MWe Industrial Gas Turbine:

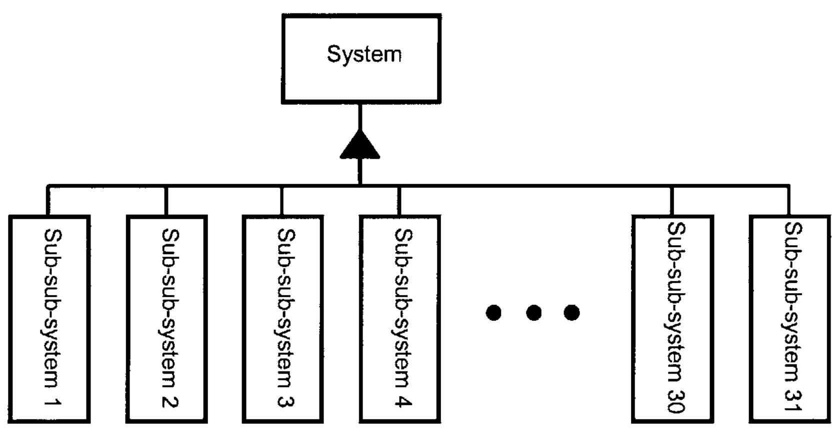


Figure 16: System Decomposition Directly To Sub-Sub-Systems

This decomposition is deliberately made without identifying intermediate level(s)[18] of decomposition (e.g. 'auxiliary sub-systems'). The reasons for bypassing the intermediate level(s) are so as to avoid premature decisions as to the most appropriate manner and depth of decomposition; and at the same to illustrate the problems that occur if no intermediate decomposition is made. Using the 7 ± 2 rule for decomposition (Miller, 1956) there would be

[18] A level could as easily be termed a layer, however one needs to be cautious in using layer terminology given the more specific usage of layers in information communications systems' standards.

expected to be at most two intermediate levels (1, 5, 25, 75) and at minimum one intermediate level (1, 7, 49 or 1, 9, 81) – but even then not everything need decompose via the same number of levels. Assuming hierarchical decomposition the range of options for decomposing exactly the same 31 elements is illustrated in the diagram below:

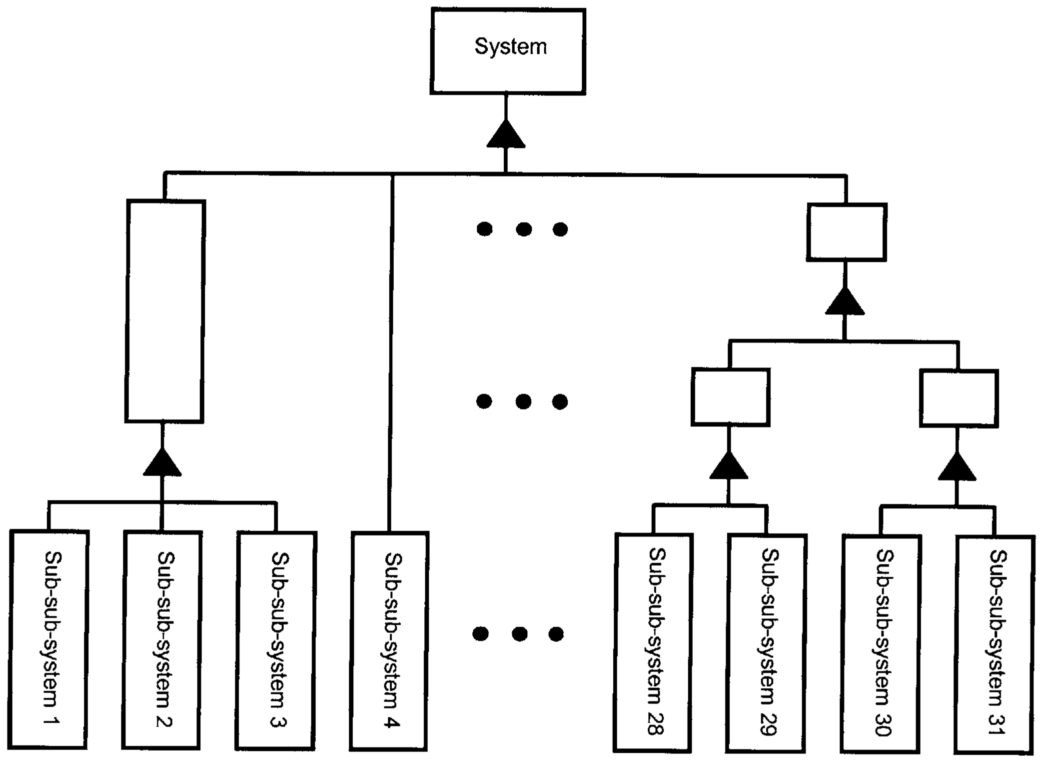


Figure 17: Hierarchical System Decomposition With Intermediate Level(s)

These sub-sub-systems should not necessarily be termed modules except in the loose sense, and for this reason they are often termed 'primitive elements' for now.

10.1.1 Characterisation Of Decomposed Elements

These 31 elements are clearly heterogeneous and the assumptions made about these elements are described in Table 1 below, which is valid for the manufacture and operation phase, i.e. the physical product domain. This table is directly comparable to the Baldwin & Clark equivalent for the computer workstation. The cost is the approximate manufacturing cost of an element after taking into account R&D costs. The calculated complexity uses cost as a proxy for complexity whilst estimated complexity is an independent estimate. Visibility is the estimated visibility (impact) of that element on the overall design and technical potential is again an estimate of the relative benefit of further R&D on this element compared with others.

Table 1: Primitive Elements Of A Generator Set (Physical Product Domain)[19]

Physical Product Domain (Manufacturing & Operation)						
Primitive Elements:		**Cost**	**Calculated Relative Complexity**	**Estimated Relative Complexity**	**Visibility in Operation**	**Technical Potential**
		$ '000s	***%***	***%***	***0 - 1***	***0 - 1***
Compressed air	$	4	1%	80%	0.8	0.1
Comms	$	5	1%	20%	0.1	0.1
Controllers - hardware	$	10	2%	40%	0.3	0.1
IO devices	$	4	1%	30%	0.7	0.1
Air inlet	$	8	2%	30%	0.7	0.1
Air filtration	$	8	2%	60%	0.4	0.1
Water wash	$	5	1%	60%	0.2	0.1
Inlet guide vanes	$	6	1%	50%	0.3	0.4
Compressor	$	30	7%	100%	1.0	1.0
Combustor	$	10	2%	100%	0.6	1.0
Expander	$	15	3%	90%	0.9	1.0
Free power turbine	$	15	3%	90%	0.9	1.0
Couplings & gearbox	$	22	5%	90%	0.6	1.0
Lub oil system	$	40	9%	50%	0.5	0.3
Fuel supply system	$	8	2%	50%	0.6	0.4
Governors	$	8	2%	90%	0.6	0.1
Exhaust trunking	$	3	1%	20%	0.1	0.1
Exhaust stack	$	4	1%	10%	0.1	0.1
Emissions control	$	15	3%	100%	0.3	0.7
Acoustic etc enclosure	$	10	2%	30%	0.1	0.1
HVAC	$	15	3%	30%	0.3	0.1
F&G	$	5	1%	40%	0.2	0.1
Bedplate	$	10	2%	30%	0.2	0.1
Sensors & actuators	$	25	6%	40%	0.5	0.4
Controllers - software	$	5	1%	0%	0.6	1.0
Maintenance systems	$	5	1%	10%	0.3	0.1
Documentation	$	1	0%	0%	0.3	0.1
Generator	$	1,00	23%	80%	0.3	0.7
Power cabling	$	1	0%	40%	0.1	0.1
Switchgear	$	25	6%	70%	0.3	0.7
Auxiliary power	$	20	5%	60%	0.3	0.1
	$	4,41	100%			

[19] The source is discussions carried out with a manufacturer of small gas turbines (both aero and industrial), combined with the author's operational experience.

10.2 Initial Calculation Of Net Option Value; Assumptions Used

The next step is to calculate the net option value (NOV) associated with experiments on this decomposition. Each line on the graph represents the value of the i'th module and each point on a line represents the value associated with the k'th experiment.

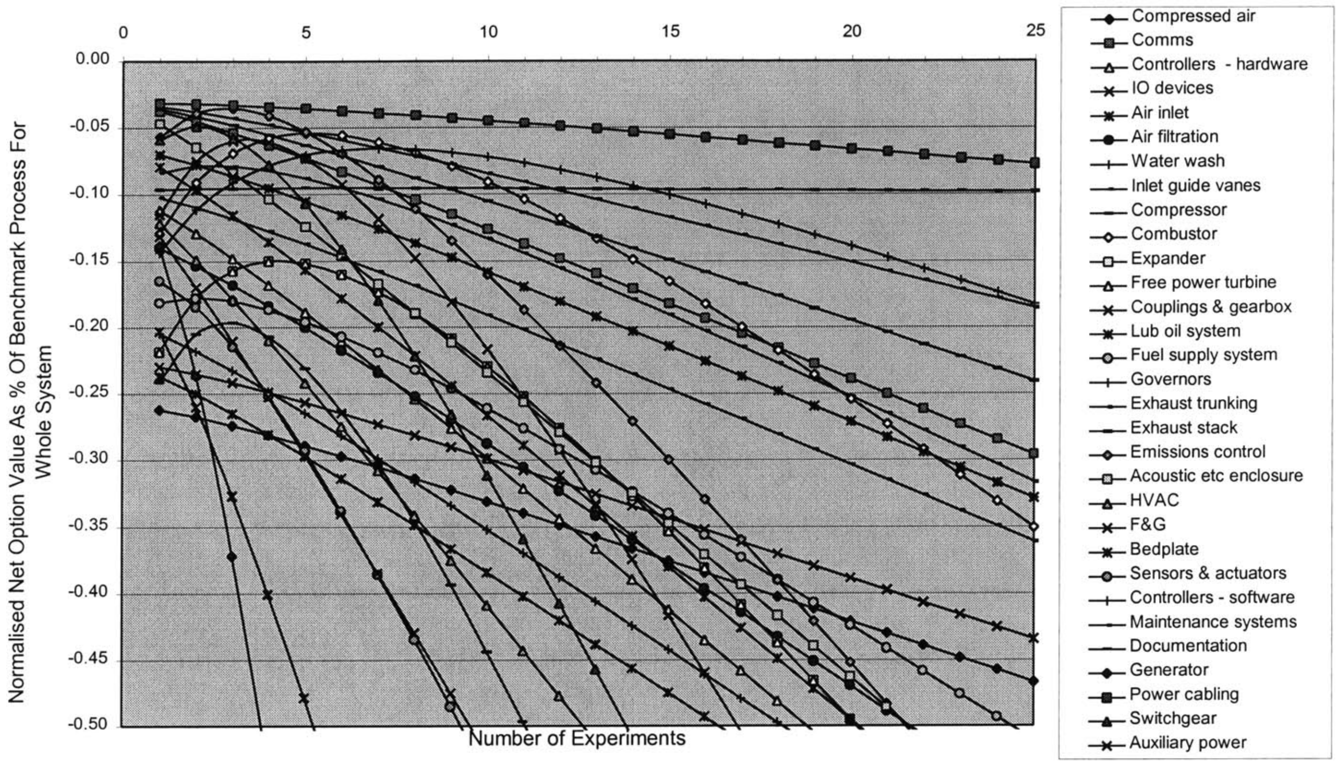


Figure 18: Net Option Value Of Genset – Initial Physical Domain

The value landscape is unattractive as the results for all k-i's indicate a negative net option value. Before examining why this is so, it is necessary to list the assumptions used.

Table 2: Assumptions Used In First NOV Calculation

Item	Assumption
Decompositional:	
Domain	Physical product
Level	Arbitrary on the basis of engineering judgement, but to ~3 levels.
Number of tasks, N	Assumed to be the number of identified sub-sub-systems per the initial decomposition, 31, cf. Table 1
Option Value:	
Distribution function	Normal

Potential, σ	Linearly proportional to technical potential, sets width of distribution function, cf. Table 1. Maximum potential assumed $\sigma = 0.40$.
Complexity	Assumed to be linearly proportional to module size, which in turn is assumed to be linearly proportional to normalised module cost, cf. Table 1.
Experiment Cost:	
1. Costs of Formulation	Omitted.
2. Costs of Experimentation	Combined with testing & integration costs.
3. Costs of Test & Integrate	Assumed to be linearly proportional to module size (i.e. cost) and number of experiments, and a cost per module experiment.
Scaling of cost per module experiment	Adjusted so that the value of the basic process is zero, i.e. NOV(1,1) = 0 excluding visibility costs.
Visibility Cost:	
Visibility	Assumed from raw data, cf. Table 1
Scaling of visibility cost	Assumed equal to cost per module experiment.

These assumptions interact. Whilst some relate to basic data, others relate to how the calculation process is applied, and others relate to the selection of the calculation process itself. What is interesting is to consider the sensitivity of the outcome to the assumptions so that only those that appear material are pursued.

10.3 Domain, Visibility, Level, Potential And Complexity

The effect and validity of these assumptions are now discussed.

10.3.1 Challenging The Domain Assumption

The reason for initially choosing the domain of the physical product is that this is what Baldwin & Clark have done in their equivalent example. Given that they feel the task domain and the product domain to be 'fundamentally isomorphic" this ought to be an irrelevant distinction, i.e. the normalised results should be comparable. Performing the calculation in another domain and seeing if the same result is obtained can investigate this isomorphism. Assuming a one to one mapping of elements from the physical domain to the task/process domain (e.g. the task of "designing the alternator" rather than the alternator itself) the basic data for the task domain is shown in Table 3, and all other assumptions are held constant.

Table 3 : Primitive Elements Of A Generator Set (Task or Design Process Domain)[20]

Task Domain (Design)

Primitive Elements:	Cost	Calculated Relative Complexity	Estimated Relative Complexity	Visibility in Design	Technical Potential
	$ '000s	*%*	*%*	*0 - 1*	*0 - 1*
Compressed air	$ 500	1%	40%	0.8	0.1
Comms	$ 100	0%	40%	0.1	0.1
Controllers - hardware	$ 800	2%	70%	0.3	0.1
IO devices	$ 20	0%	50%	0.5	0.1
Air inlet	$ 500	1%	70%	0.7	0.1
Air filtration	$ 500	1%	60%	0.8	0.1
Water wash	$ 160	0%	50%	0.1	0.1
Inlet guide vanes	$ 200	1%	80%	0.5	0.4
Compressor	$ 1,500	4%	100%	1.0	1.0
Combustor	$ 6,000	16%	100%	0.8	1.0
Expander	$ 1,000	3%	70%	0.9	1.0
Free power turbine	$ 800	2%	70%	0.9	1.0
Couplings & gearbox	$ 3,000	8%	80%	0.4	1.0
Lub oil system	$ 5,000	13%	100%	0.4	0.3
Fuel supply system	$ 750	2%	50%	0.5	0.4
Governors	$ 4,000	11%	100%	0.6	0.1
Exhaust trunking	$ 100	0%	50%	0.1	0.1
Exhaust stack	$ 20	0%	20%	0.1	0.1
Emissions control	$ 3,000	8%	100%	0.4	0.7
Acoustic etc enclosure	$ 400	1%	90%	0.1	0.1
HVAC	$ 400	1%	40%	0.2	0.1
F&G	$ 150	0%	50%	0.1	0.1
Bedplate	$ 100	0%	60%	0.3	0.1
Sensors & actuators	$ 1,000	3%	50%	0.4	0.4
Controllers - software	$ 2,000	5%	90%	0.6	1.0
Maintenance systems	$ 500	1%	50%	0.1	0.1
Documentation	$ 1,000	3%	90%	0.2	0.1
Generator	$ 3,000	8%	60%	0.3	0.7

[20] These numbers draw heavily on the experience of a senior gas turbine programme manager. They were estimated on the basis of an aero-derivative design that does not have to bear the initial costs, i.e. these costs are for industrialisation only. Given that it is a mature design using current design tools there is relatively little design iteration required. It also sets aside many of the system level testing costs as will be discussed later. It does not consider capacity expansion costs (an important consideration in practice). Profit margins are not included in either of these sets of direct costs – so sales price is not a good comparison.

Power cabling	$	50	0%	30%	0.2	0.1
Switchgear	$	1,000	3%	80%	0.1	0.7
Auxiliary power	$	500	1%	60%	0.2	0.1
	$	38,050	100%			

The results are illustrated in the graph below. Whilst the characteristic shapes are similar (given that the same equations are in play this is unsurprising) some notable qualitative differences emerge. Most obviously one 'module' (the switchgear) is sufficiently valuable that it has positive net option values. Secondly structural changes have occurred in the ordering of attractiveness of the modules and, whilst this is a somewhat vague concept given the crossing that occurs, it is the more interesting change from a fundamental perspective.

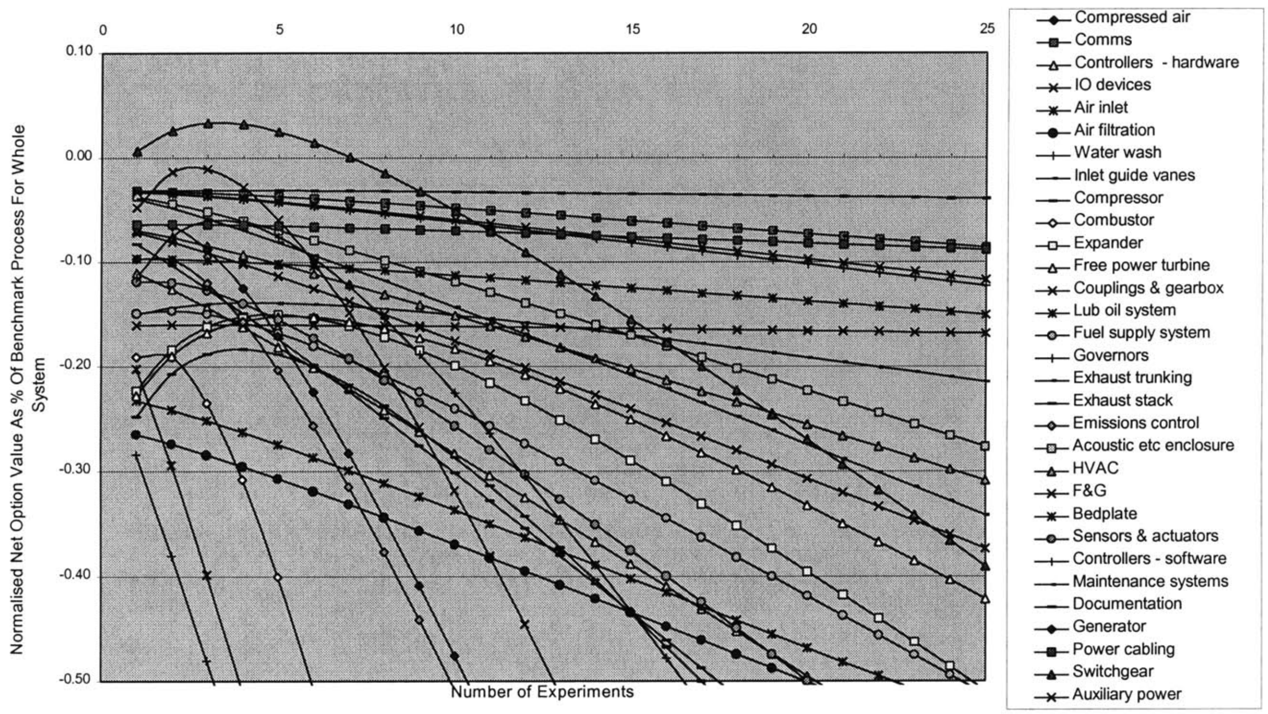


Figure 19: Net Option Value Of Genset – Initial Task Domain

In order to consider why these structural differences emerge it is only necessary to correlate the respective raw inputs of the two domains. There are three such inputs: technical potential, visibility, and complexity and these are shown in Figure 20 below along with a fourth; module cost which is the source of the calculated complexity. Each of the four graphs in Figure 20 shows the value in one domain plotted against the value in the other domain, plus the linear trend line and the R^2 value for the trend.

The technical potential cannot be the source of the structural differences in the NOV results as the same data are evident in both domains. This of course does not necessarily mean that the data are right – merely that they are at least consistent in both domains. Given the difficulty in establishing technical potentials and given the good agreement between the sources for this example I set the problem of obtaining better data aside.

The visibility data is crudely correlated along a 1:1 line with an R^2 value of 0.7955. However this variance is a significant source of the differences as the example of the switchgear shows. If the visibility of the switchgear in design is changed back to 0.3 (i.e. the switchgear sets some of the design rules for 30% of the other modules, as it does when in operation) from 0.1 (it's estimated value) then this is sufficient to reduce its value to being always negative. This sensitivity exists because of the relative size of the cost of visibility and the cost of experimentation. This variation is typical of the range over which the data falls. And once again this does not mean that either value is 'wrong' but does illustrate that it is important to determine visibility so as to minimise erroneous difference accumulation, and that the size of the visibility cost coefficient determines how important it is compared with other difference sources. The actual data difference arises because there is a genuine difference in importance of modules in the design process versus the operational process. These differences are partly a result of the non-aligned commercial interests of the manufacturer versus the operator, and partly a result of technical compromises that need to be made between what is good (or achievable) for design, manufacture and use. More sophisticated quantitative ways exist to calculate visibility using the information contained in Dependency Structure Matrices and this subject will be returned to later.

The calculated complexity correlation is strikingly less good with an R^2 value of only 0.1659 to a linear trend with a slope of ~10:1. The slope is irrelevant because the number is used only in the normalised form, but the extremely poor fit is interesting. An example is the generator where in the physical domain it has a complexity of 23% and in the design domain only 8%, which is sufficient difference to move it from being the most complex module to only the fifth most complex module. Given that this complexity metric is calculated on the basis of cost it is necessary to ask whether such a great cost difference is justifiable. The answer is that generators are an extremely mature product whose design is relatively well understood but whose manufacture (in this size) is still expensive despite being a fairly commoditised product. The reasons for this are that they use relatively expensive materials, require capital-intensive machinery, and skilled workforces – about the worst of all worlds. So the difference appears to be directionally justifiable but given that the effect is certainly material (changing from 28% to 8% has an effect equivalent to changing visibility from 10% to 30%) it is likewise worth considering how best to minimise unnecessary error. Two options are worth considering – finer decomposition (so as to minimise aggregate error) and ceasing to use cost as an indirect proxy for complexity. Ever-finer decomposition will not solve the overall issue: whilst this particular error may reduce other errors will tend to increase. It is partly to avoid this trap that a level of decomposition of ~3 has been chosen in this instance as it is a good trade-off and yields tractable yet representative problems. Instead other proxies for complexity were considered.

The assumption that cost is a good proxy for complexity rests upon the assumption that any particular fundamental task (or part) requires a roughly constant investment of capital and labour to execute in itself, and taking into account its relationships with other tasks and parts. Given that labour and capital (in the form of computational power or manufacturing equipment) are pretty efficiently traded in the internal (or external) markets of modern technology companies it is valid to only consider the range of variation of cost normal for labour. In the design realm engineering salaries vary from ~\$25k/yr to ~\$125k/yr which is a factor of five – however most tend to cluster in the much narrower range \$50k/yr to \$100k/yr which is a factor of two. In the manufacturing realm it is hard to find wages outside of the \$10/hr to \$50/hr with most around \$25-35/hr, i.e. similar levels of variation. If even these levels of variation were randomly used to transform the source data from one domain to another there would clearly be a much better R^2 value in evidence than the 0.1659 seen.

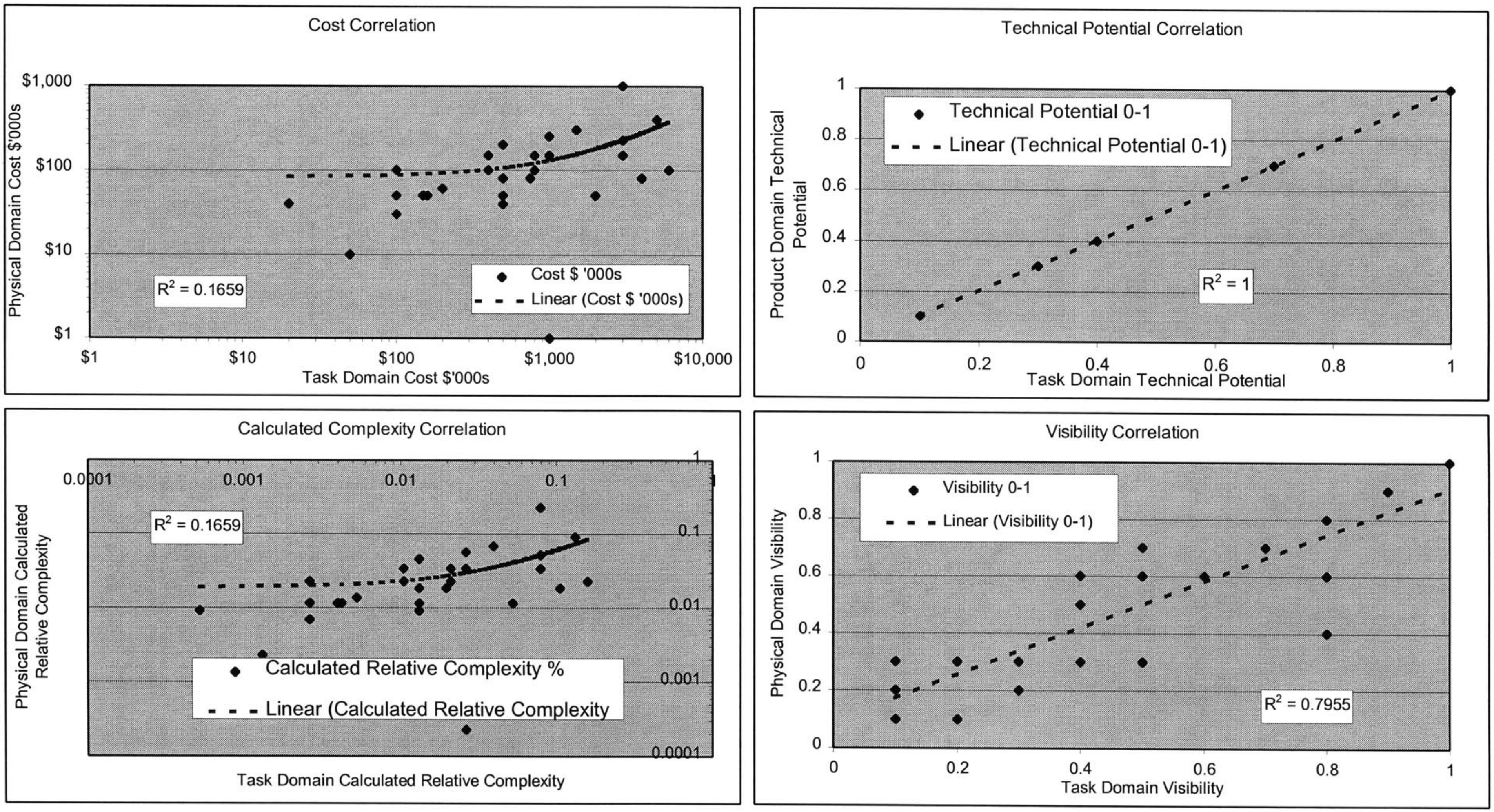


Figure 20: Correlation Of Physical & Task Domain NOV Calculation Inputs

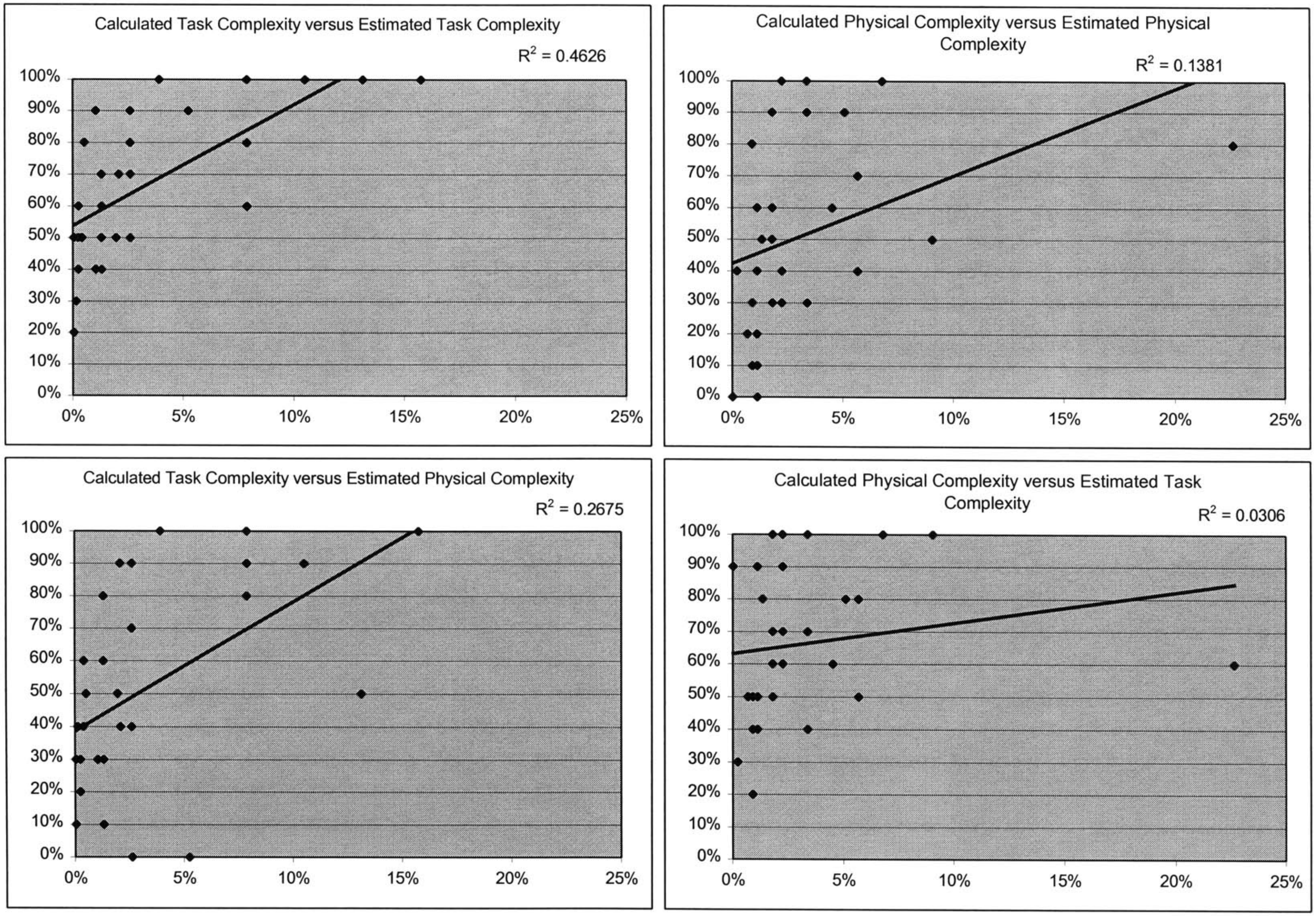


Figure 21: The Four Complexity Correlations

10.3.2 Challenging The Complexity Assumption

It is also worth challenging the assumption that all parts (or tasks) are equal. Unfortunately many ways in which costs are tracked in organisations means that even that side of the correlation would be an uncertain measure. Instead simple engineering judgement was used to answer the independent question "how complex is this module on a scale of 0-100% compared with the other modules". This estimated complexity is shown in Table 1 and Table 3 and is correlated above in Figure 21.

Figure 21 illustrates the four possible comparisons of the physical & task domain, and the calculated and estimated complexity values. The y-axis is always the estimated complexity score on a range of 0-100% and the x-axis is always the calculated complexity score. Since the sum of the latter must always be 100% whilst the former are independent this has the effect of skewing the slope of the linear trend line. Fitting a logarithmic trend seems, at first sight, attractive but trials show it to be not that much better (e.g. R^2 of ~0.35 versus ~0.25) and given the desire to compare the fit with the linear trends in the previous series of correlations this is not after all attractive.

The calculated task complexity fits relatively well with the estimated task complexity – much more so than for the corresponding calculated and estimated physical domain scores (R^2 of 0.46 versus 0.14). This suggests that the data is of a higher fidelity in the task domain than in the physical product domain. This may be because the data sources are more knowledgeable in one domain than the other (which would be odd given that one source was a strong operator whilst the other was a strong designer – and each understands the work of the other), or alternatively that much of the complexity in the physical domain is hidden in nature and so is under-reported.

The complexity mapping between physical and task domains disrupts both the other correlations – most markedly so for the calculated physical complexity versus estimated task complexity. Which one emerges as superior is largely a matter of the way in which the R^2 value is defined and therefore what emerges on the numerator and denominator. However it is worth noting that the degree of variation between the data sets is of the same order of magnitude as the degree of variation seen across the domains. This suggests that further work is desirable in this area so as (once again) to minimise the introduction of avoidable error. However in the interests of time I choose to use the estimated task complexity as being the best data set on hand. This forces calculation into the task domain, but as we shall see this is in any case desirable. It is then necessary to normalise the complexity data so as to eliminate the artificial effect of the way the question was posed. The results of this are shown below in Figure 22 below.

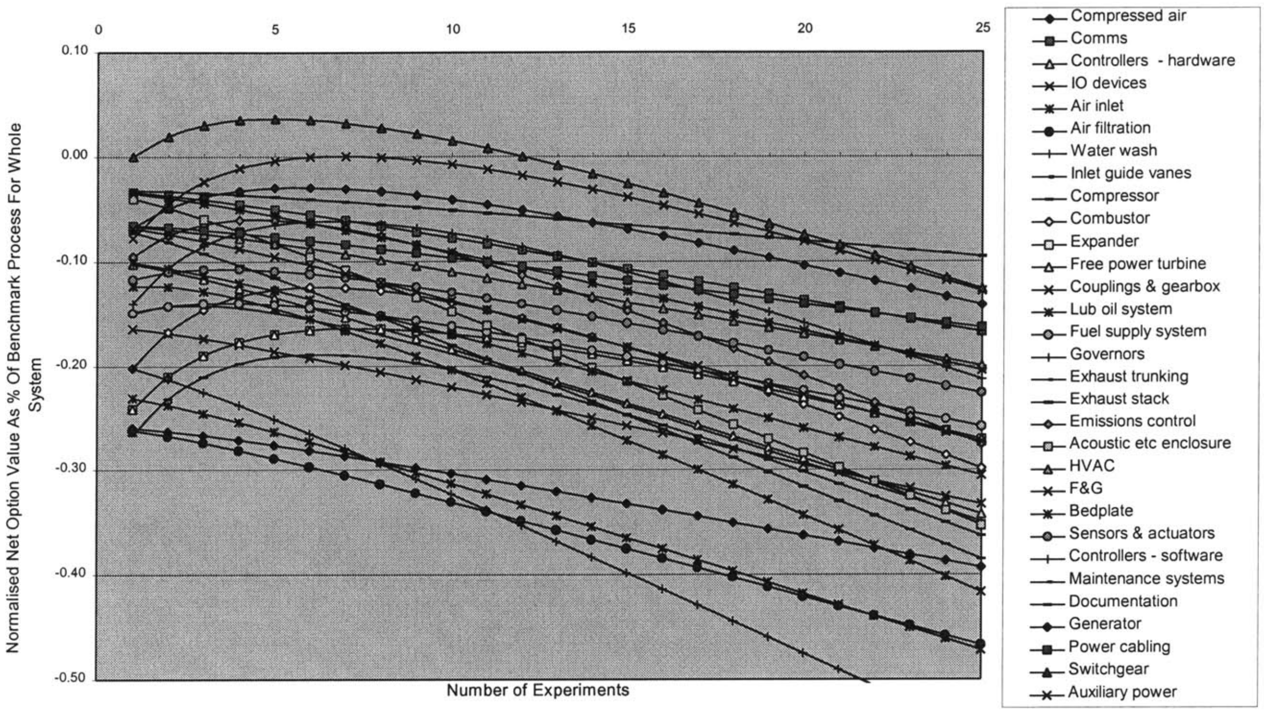


Figure 22: Net Option Value Of Genset – Task Domain With Better Complexity Data

Directionally these results are good as a second module (the gearbox) is emerging as an attractive investment, and in general more of the modules are climbing towards being attractive. This we know to be correct as gas turbine manufacturers and their module makers do make a profit[21], and do exist.

10.4 Scaling Of Visibility Costs

Having chosen to make the move over to the task domain it is now meaningful to consider the actual cost of visibility which has until now been assumed to be the same as the combined cost of experimentation, testing & integration. In the absence of data some assumption was required as

[21] It may only be because of survivor bias and state subsidies that gas turbine manufacturers (and their supplier networks) make a profit. If one were to eliminate subsidies and include the value destruction of failed manufacturers this might not be the case. However it is awkward to do these sums, as one has to back out the effect of modules being sold into other industries – e.g. gearboxes in other rotating machinery markets.

otherwise there are too many degrees of freedom to be able to partition the cost terms when setting NOV(1,1) = 0.

A full assembly, test, and system validation of a new design gas turbine costs between 10% and 17% of the total engineering and design costs[22]. Since the notional 10MWe genset under discussion is a derivative of a mature aero-product a cost at the low end of this range is reasonable. However since the cost base of $38.0 million does not include system level testing it is reasonable to assume 12.5% additional expenditure, i.e. a further $4.8 million that brings the total cost to $42.8 million.

Returning to the general formula for NOV of splitting and substitution:

$$NOV(j,k) = S_0 + \sigma(N_j)^{1/2}.Q\,(k) - c_j j - c_k k - T(j,k)$$

It is now possible to say that if S_0 is discarded and $V(1,1)$ set to unity then for $NOV(1,1)$ to be zero:

$$c_j j = c_k k = 0.44$$

$$T(j,k) = 0.12$$

If these ratios are then plugged into the NOV calculations for alternative j and k then a much cheerier picture emerges in Figure 23 as the overly oppressive assumed T(j,k) is reduced to a more realistic level.

[22] Data from an anonymous current gas turbine development programme.

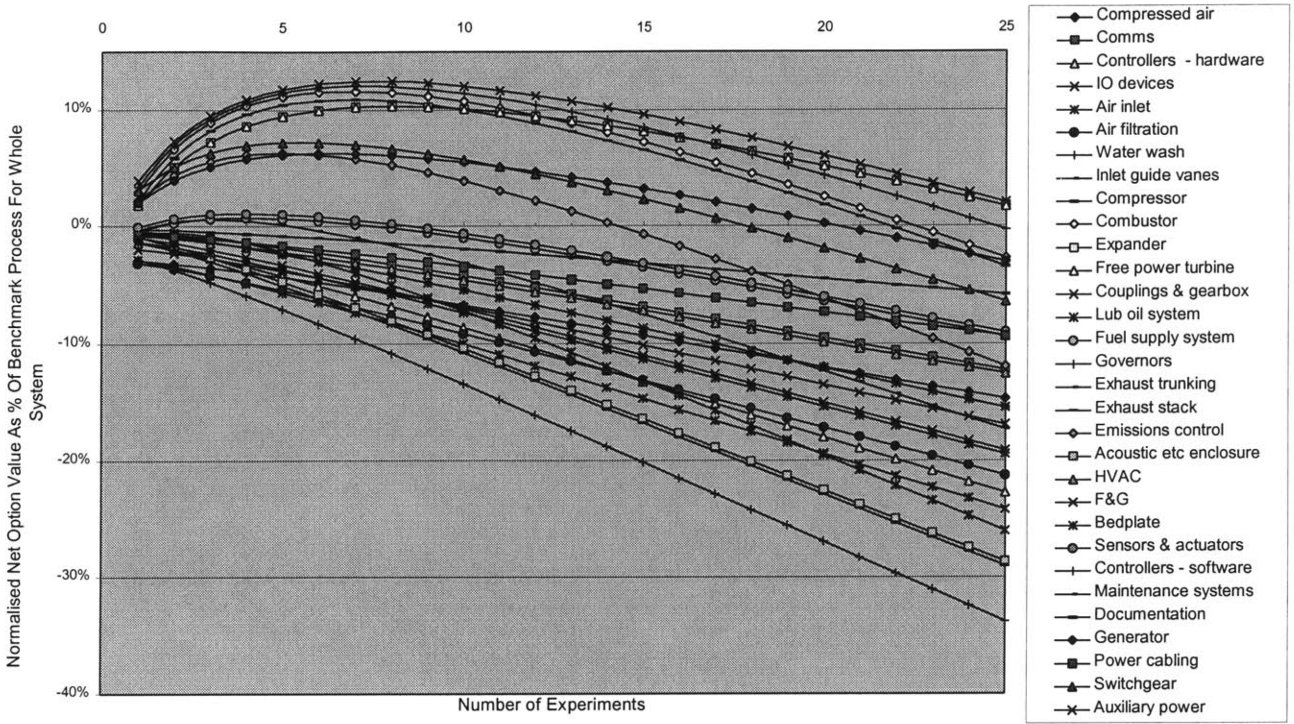


Figure 23: Net Option Value Of Genset –Task Domain, With Better Visibility Data

Even so it is noticeable that there are still a lot of elements that remain an unattractive investment in themselves. However there are still a series of assumptions that have yet to be discussed namely the number of tasks; the distribution function; and all aspects of the costs of experimentation.

10.5 Determining The Number Of Elements

Until now the calculations have assumed that the system has been decomposed to its maximum extent yielding 31 primitive elements or tasks, N. Clearly this is not the case as a gas turbine gen set has many thousands of unique components and related design tasks. Prencipe, 2000 notes that there are thousands of parts in the hydro-mechanical control unit (governor) alone of older gas turbines. Whilst much of this is now enabled in software rather than hardware the number of indivisible elements (parts) has tended to increase rather than decrease and this is only one of the thirty-one primitive elements described, albeit one of the more complicated ones. More modern engines are designed with some degree of parts reduction in mind and so the entire engine comprises 5-10,000 line replaceable units (LRUs) or stock-keeping units (SKUs). In turn many (not all) of these LRUs and SKUs can be further broken down to yield a total parts count of many tens of thousands with typical numbers in the range 50-75,000[23]. To this should be added all the parts in the balance of plant required to house the gas turbine itself and turn it into a useful generator set. Overall one can see a total part count of the order of 100,000 in the physical domain (including lines of software code).

To indicate the relationship between the number of parts and the related design tasks Sullivan, 1998 lists 34 considerations to be taken into account when designing a fan blade[24], which requires 73 separate analyses to make a complete assessment of the design. Clearly not all parts merit or require the same level of design focus as a fan blade, and equally some parts such as the fan blades are replicated but nevertheless it does give some idea of the focussed engineering that goes into a relatively integrated high performance high energy product.

This suggests that decomposing to the third level is not quite half way to the bottom, on average. Even if all branches decompose evenly (something we know not to be true as e.g. the fan blades are fully decomposed yet are higher up the tree than, say, a washer in the fuel system) it would take seven steps to obtain the required number of parts, ie. $7^6 = 117{,}649$, and probably eight or nine steps to account for all the design tasks.

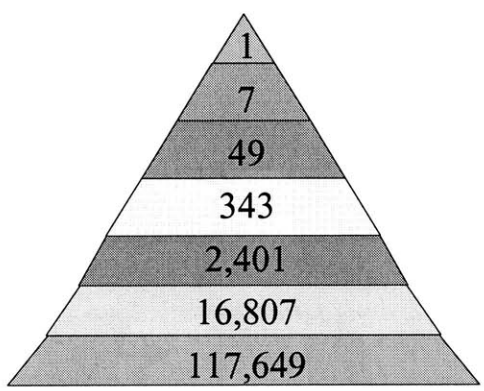


Figure 24: Part Count Sustained By Even Decomposition With A Rule Of Seven

[23] Interviews with gas turbine design engineers.

[24] The particular fan blade that Sullivan refers to is only found in gas turbines in aviation service. It is mounted upstream of the compressor to turn a turbo-jet into a turbo-fan. However the design analysis complexity actually increases when moving further into the engine as thermal effects increase, so if anything Sullivan's data understates the point.

This is all very well but what does it mean for net option value ? If one were to assume that there were in fact 117,649 elements and perform an NOV calculation for all of them there would be rather a lot of lines on a graph. However if the visibility for all lower level parts is equal to those of the higher-level assemblies then these multiple lines reduce to the same (say 31) lines previously seen. Essentially the increased number of elements N cancels through.

There are two ways in which the visibility of the lower level can be the same as the higher level. The first is if there is if the decomposition is fully fathomed at the higher level. For example in the revised illustration below of the same seven layer decomposition the tree 'D' is fully fathomed at level three (D-3), and the tree 'A' is fully fathomed at level five (A-5). Once a tree is fully fathomed it no longer has an opportunity to change it's visibility as no further (lower) cross-connections can be made.

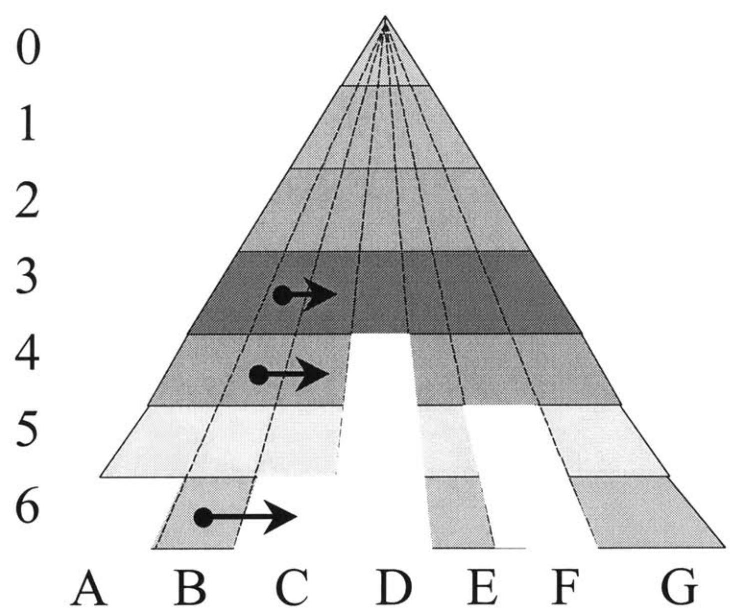


Figure 25: Visibility In Decomposition

A second way in which the visibility can be unchanged is if the topology of the network does not alter as one decomposes. For example if elements in E-2 are cross-linked to elements in F-2 then we know that the more abstract representation of E-2 that we call E-1 must also cross-link to F-1. The same is true in the reverse direction but with a caveat: if the unitary abstract entity F-3 links to the unitary abstract entity G-3 this means that some, but not necessarily all, of F-4 must link to

G-4. Unfortunately if we do not know the nature of the cross-linkage between F-3 and G-3 it is difficult to draw conclusions about topological changes with level of decomposition, yet because visibility is so important it would be nice to get it as right as possible and thereby minimise avoidable errors.

The simplest way to 'get it right' is simply to decompose everything to the bottom at which point all the information is known. The data explosion problem makes this an undesirable approach, so even if one has bottom-up data to hand it is preferable to discover the most appropriate level of decomposition through easier top-down methods. On the basis of these points two heuristics can guide us in this endeavour.

10.5.1 Heuristic One For Bounding Decompositions:

If 'B' were the only branch to extend to level 6 then there is clearly no need to decompose further than level 5 as there is nothing left to cross-connect to – this suggests that when only one tree is left a lower bound can be placed on the decomposition. It is somewhat artificial to say that the reduction to 'B' alone creates the lower bound as this presumes some a priori knowledge that the 'B' branch has insufficient internal detail to be of interest, i.e. one would already have mentally created a module around the 'B' branch. The lower bound might equally well be drawn slightly higher or lower, but nonetheless something significant is occurring at this point. A more neutral explanation might be "***place a lower bound when an arbitrarily significant number of branches terminate***".

10.5.2 Heuristic Two For Bounding Decompositions:

Pure hierarchical structures do not allow cross-linking between branches. Even if the structure has some cross-linking a good decompositional schema will minimise this and ensure that it takes place as high up the structure as possible. By being alert for such clean decompositions one can set an upper bound. If the absence of cross linking (however intuitively observed) is though of as an inter-branch barrier then the rule would be "***place an upper bound when the first barrier is observed***" as above this point it is known that cross-linking does take place. Of course any notional barrier must extend to bottom.

Both these heuristics are illustrated in another version of a seven level decompositional sketch shown below. This is the sort of sketch that a knowledgeable practitioner could rapidly make with some degree of confidence regarding a product or service (e.g. an engineer for a technical product, a bureaucrat for an organisation). It shows that only 'B' requires decomposing seven levels and that D does not require decomposing at all. The most conservative application of the first heuristic is therefore to place a lower bound between the sixth and seventh level of decomposition.

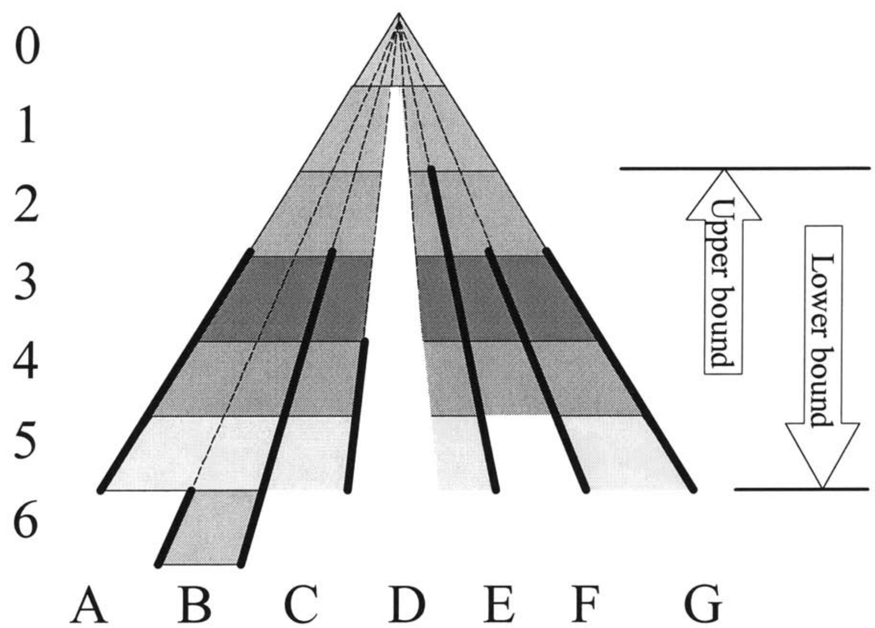


Figure 26: Example Application Of Heuristics To Determine Decompositional Depth

The figure also sketches in barriers where a branch becomes fully independent, i.e. where no further information is gained regarding the degree and nature of cross-linking. In this instance, at one extreme some cross-linking is going on between B-5 and A-5 and at the other extreme the unfathomed E-2 and F-2 are already independent[25]. Thus the most optimistic upper bound that can be placed is between the second and third level.

---

[25] This sketch suffers from the dimensionality issue, which will be discussed later as the E-2 / F-2 barrier is ambiguous regarding direction of permeability, and regarding whether either branch can link to non-adjacent branches. However, provided it is developed carefully and not over-stressed, it provides a sufficiently powerful representation for the intended qualitative use as a crude filter.

The question then becomes where to draw a line with this qualitative analysis and return to quantitative analysis. Inevitably this is a matter of judgement, but in this instance a decision to value at either the third or fourth levels appears reasonable. If in doubt one should always decompose one level further as it is easier to 'climb up' than to recognise the error of having too little detail. The exception to this is if the tools available for quantitative analysis cannot handle over-large problems and then one should err on the side of simplicity.

It might be proposed that one could combine different levels – for example (A,B-4) and (C,D,E,F,G-3). An obvious advantage of this is that it preserves all the necessary detail whilst minimising the data set size and noise. Against this is the objection that one is on the slippery slope to comparing apples with oranges. In truth it is a rather artificial objection as one could as easily have elevated A,B-4 to A,B-3 by decomposing these branches using a rule of nine or greater. Since these are rarely clear-cut decisions, at this stage one should feel able to experiment in this manner. If one were to have a staggered decomposition then, in the event that the visibility was constant (i.e. a homogenous decomposition) the additional part numbers would simply cancel out on aggregate, and in the event that the decomposition was indeed heterogeneous then it would become apparent.

This approach is poor at deciding where to slice the decomposition in order to observe emergent characteristics. This is primarily because decompositional tools are inherently poor at tracing emergent characteristics, and less a fault of premature decompositional termination. So one can at best note what is believed to be emergent characteristics (as these must have some value, either positive or negative) and keep track of them separately.

Fundamentally the decision as to what level to decompose to, i.e. what number of elements to trace, becomes an a priori guess at the likely architecture. Unless an utterly novel product is in play some notion will exist of what is the relevant level to decompose to so that all the essential information is captured without over-decomposing and thereby risking masking of the underlying signal with noise. If in doubt decompose one level further and don't worry too much about doing so in an uneven manner.

It is exactly this process that has taken place with the 10 MWe gas turbine gen-set. My intuition is that most of the interesting interactions take place at one level above the 31 element decomposition I have proposed, yet so as to be sure I have drilled to about the third level (0, 1, 2). To a certain extent this guess is already being supported in Figure 23 as there appear to be two different sorts of elements emerging: those that are of clear positive value and those that barely (if at all) break even. This suggests that there is interaction going on at some aggregate level above the 31 elements or else the system would stall economically. As we have already noted 'visibility' appears key to resolving this and it is likely that this visibility will be seen most clearly as an emergent characteristic of the second level structure. It is also possible that I have over-decomposed some branches but as I note above this need not be harmful.

10.6 Experimental Cost Structure

Even though Baldwin & Clark had developed a three stage cost equation for splitting and substitution this structure was somewhat lost in the expansion to unequal modules where the cost term becomes:

$$C(j,k) = c_i \,.\, (n_i) \,.\, k_i + Z_i$$

If visibility is not that of a nested hierarchy then after looking at the original expressions:

$$C(j,k) = c_j j + c_k k + T(j,k)$$

Which yielded:

$$NOV(j,k) = S_0 + \sigma(N_j)^{1/2}.Q\,(k) - c_j j - c_k k - T(j,k)$$

We can divide the stage three costs up into those attributable to module level and system level testing:

$$T(j,k) = f(A\,, T_s(j,k)) + f(B\,, T_m(j,k))$$

Where visibility and size determines the co-efficients A and B:

$$T(j,k) = f(Z_i\,, T_s(j,k)) + f(n_i\,, T_m(j,k))$$

Which expands to stage three costs of:

$$T(j,k) = [(k+1)^{Z_i} - 1]\, c_{ts} + n_i \,.\, j \,.\, k \,.\, c_{tm}$$

Giving a full expression:

$$NOV_i(j,k) = S_0 + \sigma(N_j)^{1/2}Q(k) - c_j j - n_i\, c_k k - \{[(k+1)^{\Sigma j \text{ sees } i} - 1]\, c_{ts} + n_i\, j\, k\, c_{tm}\}$$

I.e.

NOV = system value + option value – formulation cost – experiment cost – test/integrate cost

The difficulty with this is the need to determine the value of the coefficients. With the genset we can use real data in a similar manner to the previous scaling of visibility costs – a subject it inevitably reopens:

As before:

$$NOV(1,1) = 0$$

$$S_0 = 0$$

$$V(1,1) = 1$$

But now:

Stage one costs (1,1) = 0.05[26]

Stage two costs (1,1) = 0.83

Stage three costs (1,1) = 0.12

It is assumed that the 12% of costs attributable to system testing and module testing are equal for the first experiment on the one-module design. This is somewhat of an artificial distinction as for the first experiment on an integral design it is rather hard to conduct 'modular' experiments. However some sub-system testing would take place in any case and this assumption provides the necessary reduction in freedom.

This then determines all the various coefficients.

[26] Typical front-end engineering costs are 3-5%. An anonymous current gas turbine project spends 9 of 36 months in this phase, which represents 1 of 9 basic tasks for that programme, but this needs to be balanced by the reduced spending rate in this phase. Erring on the conservative side 5% is allowed and this in turn leaves the balance of 83% for stage two costs.

HEALTH WARNING

At this point a health warning is probably due. The results should not be taken literally as some sort of quantitative gospel, and must be handled carefully in even the qualitative sense. The raw data are questionable – I myself can see many more problems with them than I have commented on. Secondly the process is sensitive to the assumptions. Whilst I have chosen these it would have been as easy to have chosen otherwise and thereby forced different results. At the end of the day the GIGO rule applies[27] and the user of the results must interpret them cautiously.

Popping all the right co-efficients in to achieve the desired scaling of NOV(1,1) = 0 etc., the results are seen in Figure 27 below.

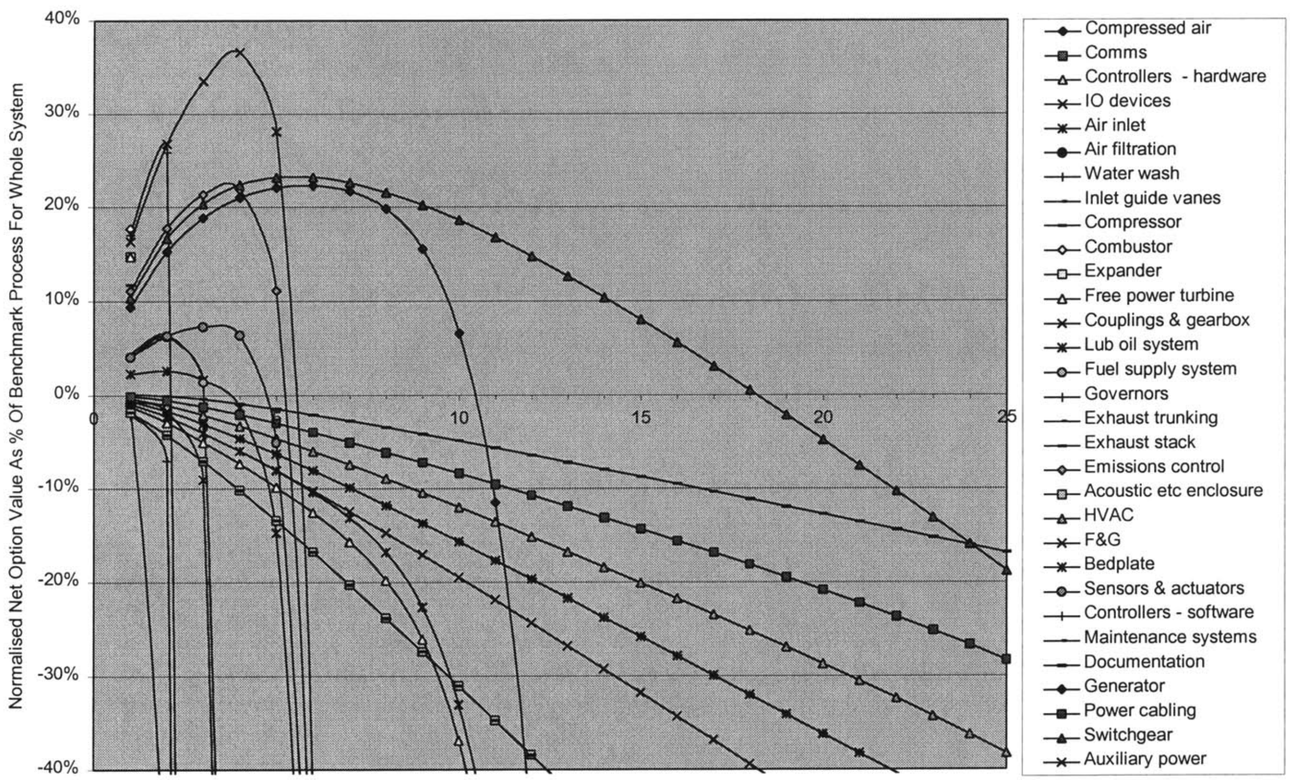


Figure 27: Genset NOV with all assumptions updated

[27] GIGO: Garbage In = Garbage Out.

There are now some extremely interesting characteristics emerging from this rather mathematical treatment of the gen set's 'modules'. However before getting carried away with analysis it is worth summarising the updated assumptions for the record:

Table 4: Assumptions Used In Updated NOV Calculation

Item	**Assumption**
Decompositional:	
Domain	Task (also known as 'process')
Level	To ~3 levels per application of heuristics to system.
Number of tasks, N	Irrelevant as no significant heterogeneity assumed below level of decomposition used in initial decomposition, 31, cf. Table 3.
Option Value:	
Distribution function	Normal
Potential, σ	Linearly proportional to technical potential, sets width of distribution function, cf. Table 3. Maximum potential yields $\sigma = 0.40$.
Complexity	Normalised estimated data used rather than normalised module cost, cf. Table 3.
Experiment Cost:	
1. Costs of Formulation	Included, scaled to ~5%
2. Costs of Experimentation	Included, scaled to ~83%.
3. Costs of Test & Integrate	Revised to include both system & module testing; scaled to ~12%.
Scaling of cost per module experiment	See above.
Visibility Cost:	
Visibility	Assumed from raw data, cf. Table 3.
Scaling of visibility cost	Now forms an element of stage 3 costs

10.7 Use Of Normal Distribution Function

Before commenting on the results it is worth touching on the use of the normal distribution function. The selection of the distribution and the scaling are one of the "formidable problems"[28] faced in implementing practical real options analysis. Normally these are referred to as the volatility of the return for a project and are calculated from stock price data. In the absence of

[28] Bowman & Moskowitz, 2001

volatility data for design projects I simply used a normal distribution on the basis that most natural phenomena are either normal or binomial, and that since the objective was to improve with each design experiment it would be unduly pessimistic to apply a binomial. In this instance the scaling of the normal distribution is a three step process: firstly the baseline standard deviation is selected as 0.4 for no better reason than a) this is what Baldwin & Clark used and b) any higher value yields unduly optimistic results, and secondly interviews were conducted to assess technical potential which was used to further reduce this upside in a linear fashion. Lastly the overall system is scaled so that V(1,1) is 1. In these examples this last scaling actually makes it irrelevant what the absolute numbers are earlier – it is the relative numbers that matter – but in more rigorous cases V(1,1) might be allowed to vary as a result of the second stage of scaling rather than being forced to unity, and then it would matter.

10.8 Interim Comments On The Results

The figure below splits the NOV graph up into four result types. Whilst there are some individual stray results and some types may be over-optimistically skewed the trends align with anecdotal industry experience.

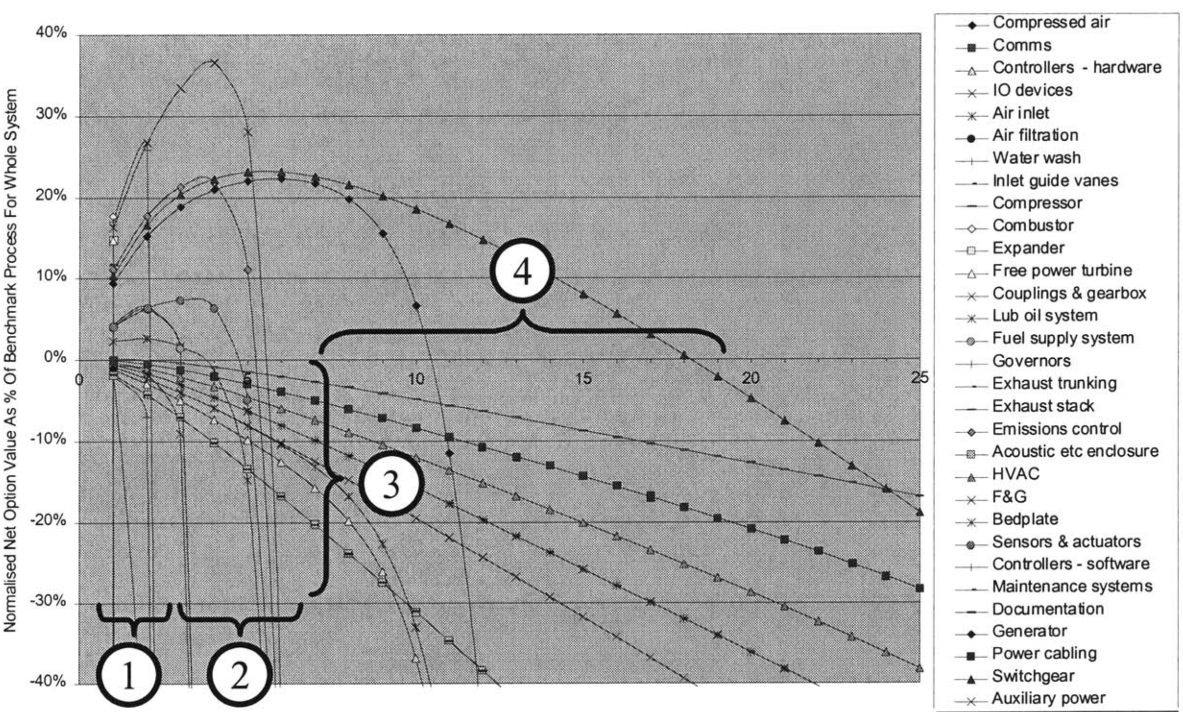


Figure 28: Four Characteristic Element Types

Type One: Crippling Test & Integration Costs

These curves suggest that the market can only sustain one or two products before the testing & integration costs become prohibitive. If there are in fact more players than optimum this will be an extremely unstable market to be in. An extremely important way to manage this problem is to avoid modularisation in the first place by integrating these sub-systems from the start. In this case it is not at all surprising that the core air-flow fall into this sector (compressor, turbine, free power turbine, combustor) and that they are in practice extremely highly integrated products. Certainly one does not normally find a Solar compressor mated with a Rolls-Royce turbine, although extremely carefully choreographed joint ventures do take place of the more easily detached chunks e.g. A Mitsubishi combustor with Pratt & Whitney rotating elements. An exception to this is when one company possesses critical technology (or at least is perceived to possess the a commanding technological advantage, perhaps by customers) as it will simply never be worth investing in developing an in-house capability: the most obvious example of this is Rolls-Royce's lead in vertical lift technology. Likewise it is not uncommon to find players seeking to go beyond single-product integration and the core designs are normally planned around a platform product strategy so as to minimise these crippling test and integration costs.

Type Two: Small Windows Of Opportunity

A small but finite number of design experiments are worthwhile in this area and the systems integrators will make use of the flexibility this permits. However the extremely steep decline in value if over-experimentation occurs is symptomatic of the high technological barriers to entry for such sub-systems as sensors and actuators. These products are typically outsourced but systems integrators will strive to avoid being locked in to any one supplier as they seek to maintain purchasing leverage. Suppliers for their part tend to have quite good margins as they spread their R&D costs over a wide range of industries with limited end user customisation occurring. New entrants tend to occur only at major technological discontinuities in either the product arena - such as the switch to embedded digital devices, or the process arena – for example in metal surface hardening and casting / machining for gearboxes.

Type Three: Ailing Cost Centres

These curves are always negative, i.e. the individual experiments lose money and the more experiments are conducted the more money is lost. If this is an outsourced product then it will be difficult to find a supplier willing to co-operate as, at best, margins will be meagre. If this is an internally sourced product then the supplier division will almost always be regarded as an ailing cost centre dragging down the company's profits[29]. However these products are essential to the success of the system and so ways must be found to make them cost-effective or attractive. Seeking re-use across platforms and industries will achieve economies of scale and scope and several of these sub-systems fall into this category (e.g. use of common off-the-shelf piping systems, re-use of software & hardware controllers across different lines of gas turbines, etc.). It is also attractive to minimise the number of players and so any one prime system integrator will typically only have one supplier of e.g. HVAC systems. In the extreme systems integrators may even agree to co-operate on sharing capabilities in this area – for example Pratt & Whitney has recently strengthened it's supplier of Type 3 sub-systems into a fully fledged division called Hamilton Sundstrand that is now the supplier of engine control systems to Rolls Royce's forthcoming engine for the Airbus A380. As we will explore when we look at DSMs in detail there is considerable scope for effective strategic management to turn these curves into healthier territory by effective use of platform-style integration within meta-modules.

Type Four: Fierce Commodity Suppliers Of Large Systems

The generator and the switchgear are both areas where improvements are possible and many technical avenues are open to achieving them. Given the importance of electrical machinery and relatively low technical (as opposed to financial) barriers to entry there are constant new-entrants to these markets. What is not fully captured by this picture is the way in which the systems integrators of all of these products use this multiplicity of sources to drive down margins. The main barrier to entry is the purchase of the capital-intensive manufacturing machinery and distribution networks, so once a supplier enters these markets they are essentially captive to the consumer (in this case the system integrator). The suppliers will attempt to evade this by upwardly integrating into systems integration itself (this was the ABB model but it failed and was sold off as Alsthom Power; Siemens-Westinghouse and Mitsubishi have been more successful) or will attempt to forge so-called strategic alliances with the systems integrators and horizontally integrate across other related markets. Given the extreme market cyclicity and high

[29] Prencipe, 2000 gives an insight into the differing responses to this common problem used by the three main aero gas turbine manufacturers. It is not difficult to guess who is who and see what has been more effective.

fixed costs pure-play horizontal integration combined with some degree of alliancing is probably the better option, especially since product differentiation is so difficult.

10.8.1 Predictive Strategy versus Reactive Strategy

What is interesting about the construction of these sets of curves is how little data was required about the financial performance of the industries. Instead a good understanding of the technical issues of the underlying product was sufficient to reveal a priori many things that most strategic analyses only perceive in the rear view mirror. Whilst this may be a moot point in an industry with such a slow evolutionary timeframe as gas turbines (let's face it they are hardly fruit flies) it is of critical importance in faster-evolving industries. In this instance an extremely sparse data set was hacked together around just a few interviews and point data and even so, once the underlying technique was worked through, some recognisable and useful results were evident that align well with the outcomes of the normal qualitative (or ad-hoc quantitative) strategic models.

10.9 Problems – Visibility, Structure, And Design Costs

The single most important factor for determining the shape of these curves is the cost of visibility, which is most obviously evident in the stage three costs of testing & integration. In any analysis it is important to get this as right as possible given the data available. A question then is "what tools and techniques exist that can give us useful information on the visibility issue without requiring too much work". Any tools that are used need to be able to cope with data sets of ~50+ elements arranged in non-hierarchical manner, and to yield quantifiable and repeatable results.

The second most important thing that requires further work is the ability to calculate option values for any intermediate level clusters. Since first these need to be observed in some optimal manner once again we need a way of calculating and quantifying optimal structures (topologies). There is a very short list of candidates that fit this bill.

Dependency Structure Matrices would seem to help with both these questions as they offer a convenient way of describing the relationships between things, manipulating this information into a useable format, and then assisting in depicting the answer. It would certainly be helpful if they could serve as more than a piece of expensive wall art for engineering meeting rooms.

10.9.1 Design Costs – Optimal Decomposition Of A Homogenous System

Lastly it is somewhat surprising that the system was economically able to support immediate decomposition to 31 sub-systems (or sub-sub-systems). Of course the basic scaling assumption was that the integrated design broke even and so all that was really being done 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. In order to get sensible results the visibility was set to 15% (approximately what one would expect in a seven module decomposition) and the technical potential was set to 100%. The results 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.

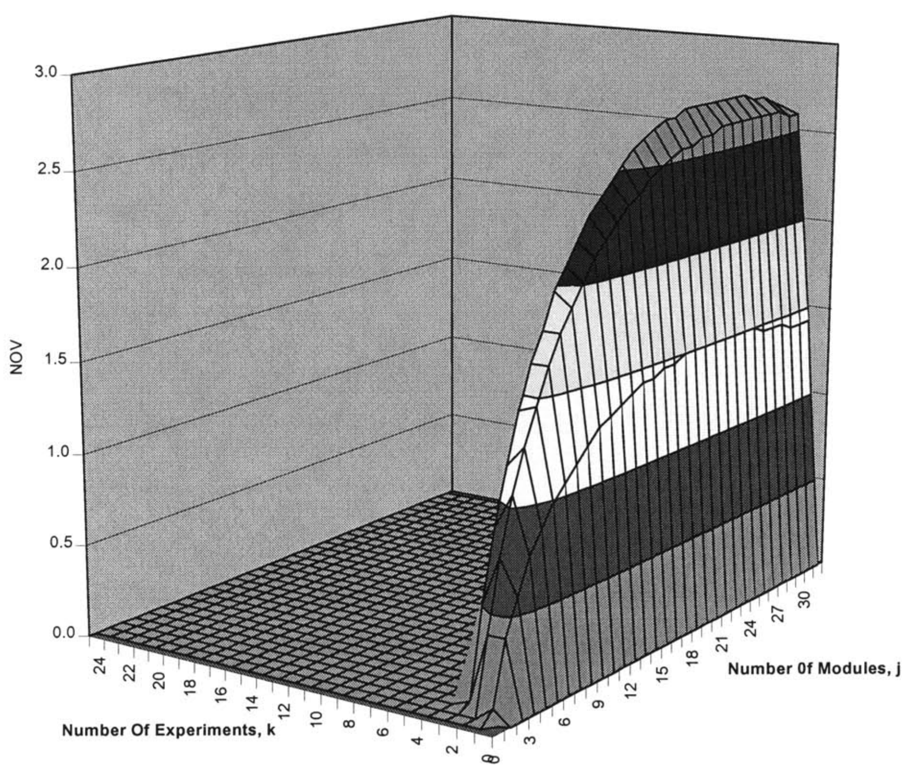


Figure 29: Optimal Number Of Modules And Experiments For The Gas Turbine

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:

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

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

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

Updating the net option value calculation then yields a very different picture:

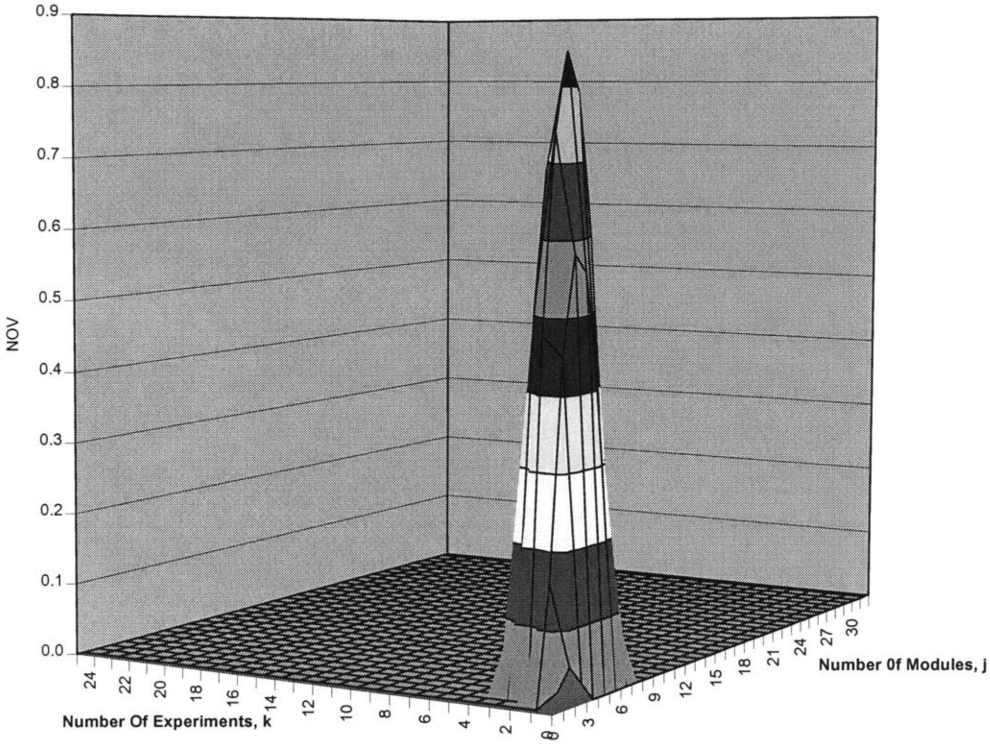


Figure 30: Revised Optimal Number Of Modules And Experiments For The Gas Turbine

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 – for example there may be relationships that do not exist prior to splitting and so there should be no costs associated in creating design rules for these relationships that are equally non-existent post splitting. However the basic point that design costs are proportional $\sim j^2$ rather than to $\sim j$ makes a big difference irrespective of the precise details. A more subtle equation would weight the power term to correspond to the population density of the DSM.

It is not possible to directly translate these results to the 31-module system because the 31-module system is heterogeneous rather than homogenous, and the individual modules of the 31-module system are not directly bearing the stage 1 costs of formulating the design rules.

10.10 Summary Of Chapter

In this chapter an initial valuation crude valuation of the gas turbine's 31 tangible sub-systems has been polished through a process of challenging and refining the assumptions. Qualitatively useful results have been obtained which give an indication of the different tensions that heterogeneous sub-systems are exposed to, and these have been categorised into four types. Such a crude categorisation is in itself of value for developing strategy from the very low level basis of the product's architecture rather than a more high-level basis of industry etc. structures. Then investigation of the optimal decomposition of a homogenous system of a similar nature suggested that four modules would be optimal.

Whilst the precise numbers are disputable, and the exact cost calculation debatable, the underlying issue is the disparity between the 31 or so modules recognised at the third level of decomposition, the four or so main types of module values seen in the NOV calculation for the heterogeneous system, and the optimal decomposition of four modules suggested by the NOV calculation for a homogenous system.

The question that arises from this seeming contradiction must be whether it can be satisfactorily explained. It appears that the notion of visibility may be important in this area and Dependency Structure Matrices may assist in making a better assessment of visibility than the point estimates used hitherto. The next chapter will discuss a possible resolution of this at the intermediate level of decomposition.

11 NOTES ARISING FROM CONSIDERING THE DEPENDENCY STRUCTURE MATRIX REPRESENTATIONS OF THE GAS TURBINE GENERATOR SET

11.1 Introduction

This chapter uses Dependency Structure Matrices (DSMs) as a tool for analysing the architecture of the generator set at the second and third level of decomposition. Most of this is performed in the physical domain and therefore the principal concept is that of clustering. The meaning of clustering is investigated using simplified cartoon DSMs and cartoon architectures so as to be able to introduce and define terminology and develop the underlying issues with the minimal amount of complexity. These cartoons are woven in between consideration of the real gas turbine.

Towards the end of the chapter some discussion takes place regarding automated clustering algorithms. This differentiates between the search strategy itself and the characteristics of the objective function. This leads to a discussion of cluster boundaries and boundary objects on one hand, and on network / graph theory on the other. This returns us to the notion of perspective and metaphor, which takes us towards the next chapter on language. Before concluding we relate the apparent structure in the gas turbine's generator set to the tension seen in the alternative option value calculations seen in the previous chapter, with the intent of qualitatively resolving this. Indicative quantitative resolutions are attempted using various proxies for the full graph theory and boundary object penalty function(s) and a societally optimal gas turbine architecture is proposed.

The notion of the missing architectural decisions is introduced and the pathway to the other domains indicated. This leads to a discussion of multi-domain optimisation. In passing comments are made on characterising the signatures of integral and modular architectures.

11.2 Physical Domain DSM Definition & Initial Analysis For The Gas Turbine

As previously discussed there are many possible domains available e.g. organisational, task, physical, functional. In the previous chapter it was shown that the results of the option value calculations were different in the task domain (aka. process domain) than in the physical domain, and in that chapter the decision was made to primarily use the task domain as better quality data was available. However the differences were the result of differences in data used for option value calculations, not of differences in data that is relevant to DSM generation and manipulation. In this chapter the data is of better quality in the physical domain and since option values are seldom considered whilst DSMs are the focus it is in the physical domain that the bulk of the analysis is done.

11.2.1 Initial Binary DSM & Automated Clustering

A dependency structure matrix DSM for the observed 31 physical sub systems was constructed by setting a tick mark to denote a material influence from one chunk to another[30]. A material relationship is a significant flow of energy, mass, information, or structural (geometrical or force) from one chunk to the next – i.e. the main dimensions of the physical domain. At this stage these are noted on a binary scale of 0 (no dependency exists) or 1 (a dependency exists). This is not a symmetrical DSM, as direction does matter in this instance. This is termed the physical domain DSM, and it expresses the thing's relationship with the thing itself, at about the third level (0, 1, 2) of decomposition.

In its raw unsorted form this physical domain DSM is as shown in Figure 31 below, which is a bit of an untidy mess. A gas turbine is not a fully integrated product as can be seen by the fact that the matrix is not fully populated. Therefore it ought to be possible to partition the DSM to force as many relationships as possible to on the diagonal and thereby expose the irreducible

[30] These relationships were determined by me, drawing on my experience in operating and troubleshooting gas turbines. Some auditing was then conducted with my peers who are more experienced in gas turbine design. No changes were made as a result of these audits. This does not mean that the DSM is 'right' but it does suggest that it is good enough for the purpose of this thesis, i.e. developing a concept via a worked example. This worked example is a synthetic DSM for a generic 10 MWe gas turbine gen-set.

clusters for further analysis. However applying MIT's Excel macro for DSM analysis[31] results in no movement. This is because several of the sub systems have sufficient relationships as to effectively stall the partitioning (clustering) algorithm. This is indicative of a low signal to noise ratio in the information presented to the clustering algorithms. At this point there are the following options to achieve some progress in identifying the clusters that may exist:

1. Intuitively eliminate selected above diagonal feedbacks to enable automated clustering, selecting the feedbacks to be eliminated using implicit relationship strength.
2. Explicitly reveal relationship strengths and analyse using an automated algorithm.
3. Manually and intuitively cluster either on the basis of the binary information or taking into account the strengths of the relationships.

Each of these options is described in turn, however the second option is considered in an interim manner and then returned to after manually clustering.

11.2.2 Intuitive Elimination Of Selected Feedbacks & Automated Clustering

If the feedback relationships[32] from fire & gas systems, software, sensors, and maintenance access are eliminated then the clusters shown in Figure 32 are revealed. These feedback relationships were chosen because they contained the greatest amount of noise, i.e. since they connected to everything they contained least signal content and therefore their elimination in some sense exposed the greatest signal. The two dominant clusters marked out in the diagram relate in some sense to the air intakes, treatment, and compression, and to the rotating machinery and mounting frames. A possible third cluster is observable relating to the switchgear and power cabling. The algorithm also determines cluster bounds by corner values (ticks) and thus the

[31] This is the Excel macro of 2000 by Tyson Browning (banding & simulation), Qi Dong (reachability matrix), Carlos Fernandez (clustering algorithm), Ali Yassine (integration & layout). Note that there is another MIT DSM tool that operates as an Excel add-in and which is better at banding and project simulation whilst the Excel macro referred to here is better at partitioning (clustering).

[32] Feedback is strictly a term that should be kept for the task domain as it indicates a direction of flow. Clearly there is an unambiguous flow of time in a task domain DSM, however the equivalent flows in the physical domain are more ambiguous. Therefore in the physical domain it is probably better to use the terminology 'above diagonal' or 'below diagonal'. In this instance selecting the same elements' below diagonal relationships for elimination yields materially the same (inconclusive) results.

algorithm proposes a series of five large and almost completely overlapping clusters (denoted by the blue shaded regions), even less convincingly than the manually marked two.

Whilst this reveals that the eliminated sub systems play a different role than the two clusters it also suggests that it is worth investing a little more effort in attempting a less intuitive and more explicit methodology as these clusters are not entirely convincing even to an untrained eye.

	Air inlet trunking & plenum chamber	Air filtration	Water etc wash system (nozzles, piping, pumps, tanks)	Inlet guide vanes	Compressor (rotor blades, shaft, stator blades, bearings, housing, blade tip seals, bla	Combustor (cans, diffusers, ignition, housing)	Expander (rotor blades, shaft, stator blades, bearings, housing, blade tip seals, blade	Lub oil system (bearing connections, piping, pumps, filters, tanks)	Fuel supply system (piping, pumps, filters, tanks)	Governors (these and other control devices treated separately)	Free power turbine	FPT bearings & shaft couplings & gearbox	Exhaust trunking	Exhaust stack & transition ducts	Acoustic etc enclosure	Generator & pilot exciter	Baseplate & mountings	Power cabling	Switchgear	Auxiliary power supplies & UPS	Compressed air / P2 air	Sensors & actuators (vibration monitors, thermocouples, control valves, shutdown valv
Air inlet trunking & plenum chamber		1	1	1	1				1						1							
Air filtration	1		1		1		1															
Water etc wash system (nozzles, piping, pumps, tanks)	1	1		1	1	1	1				1						1					
Inlet guide vanes	1	1	1		1	1	1		1	1												1
Compressor (rotor blades, shaft, stator blades, bearings, housing, blade tip seals, blade mounts)	1	1	1	1		1	1	1			1		1	1							1	1
Combustor (cans, diffusers, ignition, housing)			1	1	1		1		1	1	1		1	1								1
Expander (rotor blades, shaft, stator blades, bearings, housing, blade tip seals, blade mounts)	1	1	1	1	1	1		1	1		1		1	1								1
Lub oil system (bearing connections, piping, pumps, filters, tanks)					1	1	1				1	1				1	1					1
Fuel supply system (piping, pumps, filters, tanks)					1	1	1			1							1					1
Governors (these and other control devices treated separately)				1	1	1	1	1	1		1					1						1
Free power turbine			1		1	1	1	1		1		1	1	1								1
FPT bearings & shaft couplings & gearbox					1	1	1	1			1		1			1						1
Exhaust trunking			1		1	1	1				1	1		1	1							1
Exhaust stack & transition ducts					1	1	1				1	1	1		1							1
Acoustic etc enclosure	1	1		1	1	1	1		1		1	1	1	1		1			1			
Generator & pilot exciter					1	1	1	1		1	1	1	1						1			1
Baseplate & mountings	1	1	1		1	1	1				1	1	1	1	1	1			1			
Power cabling																1			1			
Switchgear																1		1		1		1
Auxiliary power supplies & UPS									1							1		1	1			1
Compressed air / P2 air					1																	
Sensors & actuators (vibration monitors, thermocouples, control valves, shutdown valves, RVs, etc)		1	1	1	1	1	1	1	1	1	1	1	1	1		1		1	1	1		
IO devices																						1
Hardware: Controllers & DCS / SCADA / PLC & associated buses, displays, and data loggers																			1	1		1
Software: Controllers & DCS / SCADA / PLC & associated buses, displays, and data loggers					1	1	1	1	1	1	1	1				1			1	1		1
Emissions control systems									1	1			1	1								
F&G detection & suppression					1	1	1	1	1	1		1	1	1	1	1			1	1		1
HVAC	1	1			1	1			1						1	1			1			
Comms – tannoy, visual indicators, magnetic loop, etc.						1	1							1	1	1		1	1			1
Maintenance & installation access systems (rails, hatches, davits, isolation valves, etc.)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		
Documentation & certification	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1		1

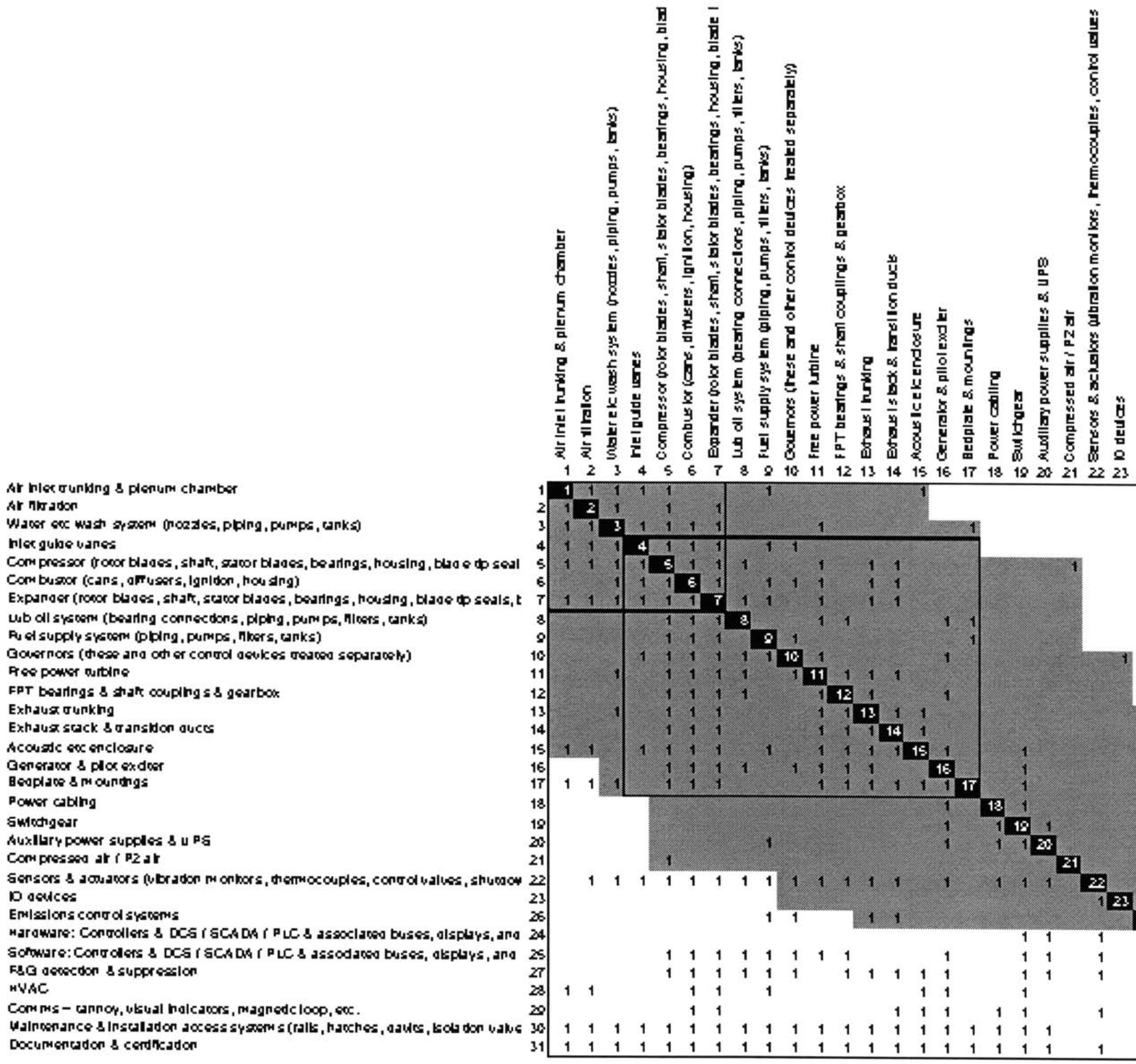
Notes: Complete Binary

This figure shows the asymmetric unsorted physical domain DSM.

Although the source data has four possible relationship strengths (0, 1, 2, 3) this version of it has had all non-zero relationships replaced by 1 to form a binary DSM.

Although the DSM is unsorted the primitive elements were written down in a non-random manner and the clustering algorithms do not randomise the order. This has implications if the clustering algorithm does not branch widely enough (i.e. can be captured by a local optimum) or if the solution is path dependent.

Figure 31: Unsorted Physical DSM Of Typical 10MWe Industrial Gas Turbine



Notes: Artificial De-randomisation

Simply clustering the complete binary DSM above results in no changes as, in the absence of relationship strengths, the data is nearly random. Since all clustering algorithms require that there be information content in the data it is not surprising that they all failed, as the situation is fairly similar to sorting white noise.

In order to increase the information content some data was changed. By inspection the above diagonal main bus elements were set to zero to. This figure shows the DSM after the MIT DSM macro was then run.

Some clustering is evident but the five identified clusters almost completely overlap. This does not represent good cluster definition.

Figure 32: Physical DSM With Clusters Revealed by Intuitive Feedback Elimination

11.2.3 Weighting The DSM, Automated Clustering Using Different Materiality Thresholds

An explicit assignation of weights to relationship strengths is the next step. Returning to the initial physical domain DSM the simple binary (0 or 1) tick marks were replaced with a four-point scale to denote the relationship strengths. In order to guide this intuitive estimation the following qualitative scale was used:

Strength	Title	Explanation
3	High	Significant flow of three of (mass + energy + information + load/geometry)
2	Medium	Significant flow of two of (mass + energy + information + load/geometry)
1	Low	Significant flow of one of (mass + energy + information + load/geometry)
0	Zero	No significant relationship

Table 5: Four Point Scale To Denote Strength Of Relationships Used In Physical DSM

In case of doubt as to the strength of a relationship the question was asked "What would be the relative speed and scale of effects felt by each sub-system as a result of single point damage to another sub-system?". The answers draw upon my experience of troubleshooting faults in these sorts of generator sets.

In conducting this exercise it is difficult to do more than make an estimation of the strength of relationship because of the complexity of the relationships. As an aside in order to fully understand the relationships it would be necessary to decompose each sub-system to its elemental parts, and then to map each of these parts across to the functional domain (which would reveal some one to many and some many to one relationships). Another term for the functional domain is the parametric domain (which is not the same usage as in axiomatic design) and it would then be possible to explicitly determine the strengths of the relationships for a given design (e.g. per Mascoli, 1999). Then it would be necessary to map these strengths back to the physical domain (somehow summing many to one relationships and allocating one to many relationships) and then upwardly aggregating these elemental strengths to arrive at the higher level strengths possessed by the chunks I term sub-systems.

11.2.3.1 How Good Is This DSM ?

At this point it is worth posing the question "To what extent are the relationships sufficiently captured by this synthetic data?". Subsequent to my populating the DSM with synthetic data I was made aware of Whitney's as yet unpublished work on population density in DSMs, which suggests that this example is appropriately populated as the relationships I have identified span the anticipated population density:

Method	Population (Total tick marks)
Whitney's prediction for 31 x 31 DSM	186
Synthetic data (high + medium + low)	349
Synthetic data (high + medium)	204
Synthetic data (high)	66
Mean average of synthetic data	206

Table 6: Predicted Versus Actual Population Density For 10MWe Industrial Gas Turbine

Whitney's data may be worth analysing to see if there are different trends evident in population density for DSMs covering different domains, and certainly worth analysing for differences in population density for modular versus integral products. For example there are ~10% more high & medium strength relationships in this DSM than Whitney forecast which suggests a relatively integrated product. Whitney's data may be more reflective of how humans design products that are within humanity's ability to understand, than reflective of any natural laws about the degree of connectivity. If this is the case then 'only' ~10% differences will be highly material.

This increased population of ~10% is highly sensitive to the materiality threshold in use when populating a DSM. It is well known that no two people ever create the same base data for a DSM. This is partly because of the difficulty in establishing a common materiality threshold for use in establishing when a relationship is sufficiently strong to merit inclusion, and then because of the difficulty in applying this consistently across all the possible relationship pairs. It is of course also a result of failure to agree a common system decomposition and difficulties that exist (in practice) in ignoring the effect of indirect relationships rather than the direct one (e.g. Element 'A' may have no direct relationship with element 'B' but may influence it indirectly via element 'C'. In this case no A-B tick should exist but frequently it is either not possible to understand data regarding the indirect nature of the relationship, or to obtain the data, and in many instances an A-B tick will result). These materiality concerns are eased somewhat when one moves to weighted DSMs as the propensity for error accumulation reduces. A related issue is the nature of the scale in use – for example Thebeau, 2001 uses a non-linear four-point scale of [0, ½, 1, 2] which obviously will affect the outcome.

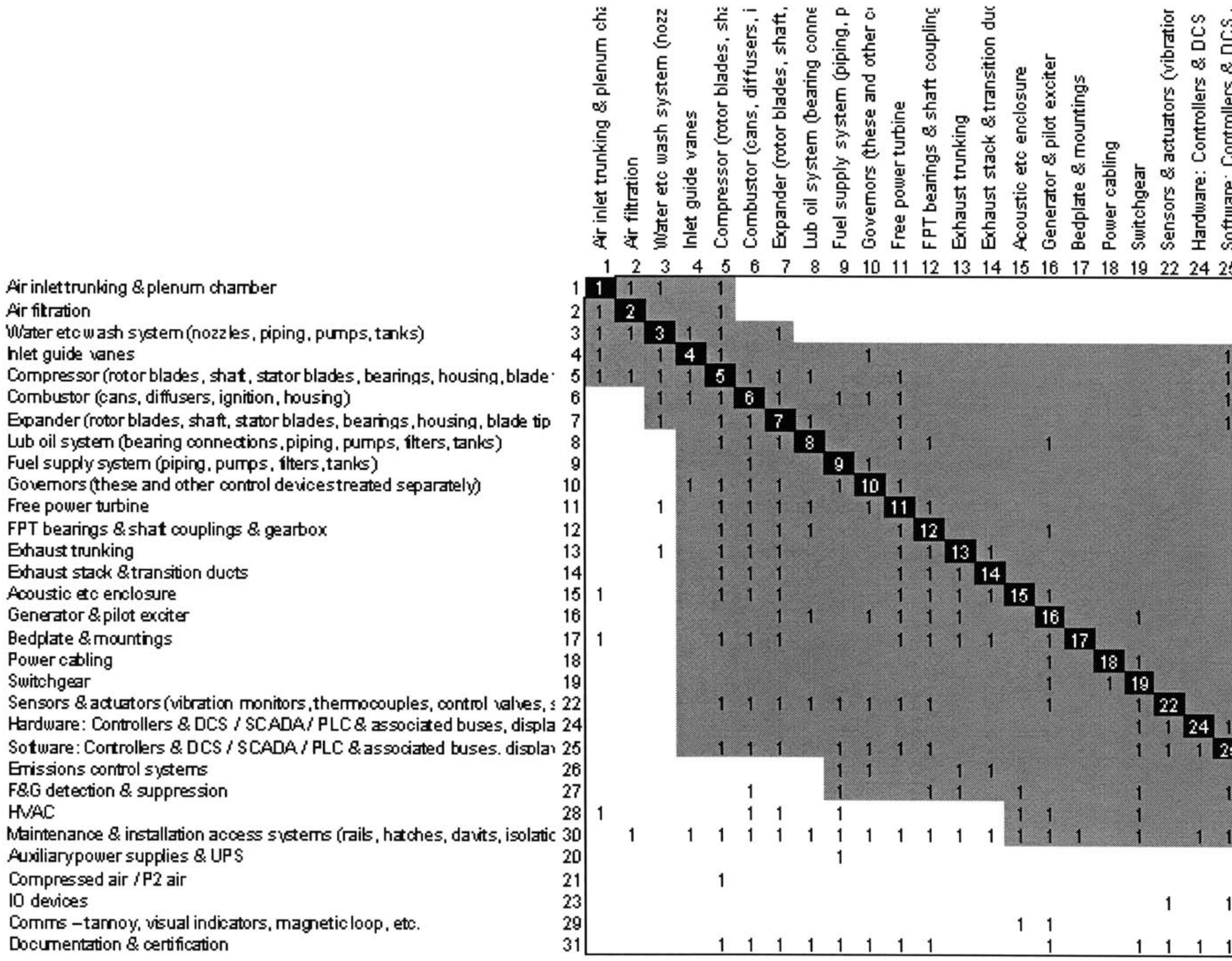
Thus in answer to the question "How good is this DSM ?" no categorical answer can be given. However it appears directionally sound, as it is not dramatically different than the known data set of comparable DSMs as collected by Whitney.

11.2.3.2 Automated Clustering Using Different Materiality Thresholds

It is possible to exploit the materiality threshold to improve the effectiveness of automated binary clustering algorithms[33]. As a first pass all the high and medium strength relationships were replaced by a 1 and all the low or zero strength relationships with a 0. This was then automatically partitioned with the results shown in Figure 33. A second pass was to replace only the high strength relationships with a 1 and to replace all others with a 0 and these results are shown in Figure 34.

Both these trials reveal some clustering even though much of the order is not noticeably different. It appears that the air intake systems and the core gas generator systems pretty much dominate the upper left quadrant in all circumstances whilst depending on where the binary 'cut' is made different subsystems are forced to the bottom. However this does not fully cater for the weights of the relationships and as before the results are rather inconclusive and unsatisfactory. As a next step manual clustering was attempted before seeking either more automation or better understanding.

[33] Dr Dan Whitney, MIT, suggested this approach.



Notes: High & Medium Strength

All high and medium strength relationships (2, 3) were replaced with a one and all low and zero strength relationships (0, 1) were replaced with a zero.

Upon clustering using the MIT DSM macro the following emerged. Once again five clusters have been identified and once again they all overlap significantly. However on this occasion the low above diagonal population density means that the optimal cluster boundary location has to be doubted.

Figure 33: Clusters In Physical DSM Revealed By Partitioning On High & Medium Strength Relationships

PARTITIONING ONLY ON3		Air inlet trunking & plenum 1	Air filtration 2	Water etc wash system (r 3	Inlet guide vanes 4	Compressor (rotor blades, 5	Combustor (cans, diffuser 6	Expander (rotor blades, sh 7	Fuel supply system (pipin 9	Governors (these and othe 10	Free power turbine 11	FPT bearings & shaft coup 12	Software: Controllers & D 25	Acoustic etc enclosure 15	Bedplate & mountings 17	Auxiliary power supplies & 20	Compressed air / P2 air 21	Sensors & actuators (vibra 22	Hardware: Controllers & D 24	Emissions control system: 26	F&G detection & suppress 27
Air inlet trunking & plenum chamber	1	1	1			1															
Air filtration	2	1	2																		
Water etc wash system (nozzles, piping, pumps, tanks)	3		1	3																	
Inlet guide vanes	4				4	1															
Compressor (rotor blades, shaft, stator blades, bearings, housing, blade tip seals, t	5			1	1	5	1	1					1								
Combustor (cans, diffusers, ignition, housing)	6					1	6	1	1												
Expander (rotor blades, shaft, stator blades, bearings, housing, blade tip seals, bla	7					1	1	7			1		1								
Fuel supply system (piping, pumps, filters, tanks)	9						1		9												
Governors (these and other control devices treated separately)	10				1	1	1	1	1	10											
Free power turbine	11						1	1			11	1									
FPT bearings & shaft couplings & gearbox	12										1	12									
Software: Controllers & DCS / SCADA / PLC & associated buses, displays, and dat	25									1			25								
Acoustic etc enclosure	15													15							
Bedplate & mountings	17														17						
Auxiliary power supplies & UPS	20															20					
Compressed air / P2 air	21																21				
Sensors & actuators (vibration monitors, thermocouples, control valves, shutdown	22																	22			
Hardware: Controllers & DCS / SCADA / PLC & associated buses, displays, and da	24																		24		
Emissions control systems	26																			26	
F&G detection & suppression	27																				27
Comms – tannoy, visual indicators, magnetic loop, etc.	29																				
Lub oil system (bearing connections, piping, pumps, filters, tanks)	8					1		1				1									
Exhaust trunking	13						1	1			1	1									
Generator & pilot exciter	16							1			1	1									
IO devices	23																	1			
HVAC	28						1	1						1							1
Exhaust stack & transition ducts	14						1	1													
Power cabling	18																				
Switchgear	19																				
Maintenance & installation access systems (rails, hatches, davits, isolation valves,	30					1	1	1			1	1		1							1
Documentation & certification	31					1	1	1				1	1					1	1		1

Notes: High Strength

Here only high strength relationships (3) were replaced with a one and all medium, low and zero strength relationships (2, 1, 0) were replaced with a zero.

Upon clustering using the MIT DSM macro the following emerged.

Now only two modules have been identified but this time they emerge clearly. Equally clear is the bus module although the algorithm does not identify and delineate it.

Figure 34: Clusters In Physical DSM Revealed By Partitioning On Only High Strength Relationships

11.2.4 Manual Clustering Of The Weighted DSM

Intuitive manual clustering taking into account the previous results and the actual weighting can yield different results depending on the extent to which a single group of system-wide relationships ('buses') is emphasised over 'good' clusters. In any case there is a considerable degree of integration in the product, at least in the physical domain. Two alternative versions are shown in Figure 35 and Figure 36 below. Even though they are slightly different they do exhibit an interesting characteristic, that of the intermediate layer (circled in dashes) which has apparently never before been commented on in DSMs and which I term a weak or auxiliary or subsidiary bus. The generation of these two alternatives are discussed next.

11.2.4.1 A First Manual Clustering

This took the trial runs above as a starting point and then a few manual changes were made in the weighted DSM to reveal these clusters. Clusters were arbitrarily marked off by borders to emphasise their location and nominal boundaries. In the functional domain this selection & marking process is easier as upper right-hand tick marks (relationships) denote the outermost feedback loops and thereby set the boundary, however in the physical domain one has to take into account both above and below-diagonal relationships.

After inspection of the clusters were given names to identify them. These names can be chosen arbitrarily or by looking at the clusters and seeking to identify common characteristics – I used the latter approach.

Notice how some clusters are isolated (e.g. the switchgear) whilst others overlap (e.g. air clean-up with the gas generator) or are completely embedded in a larger cluster (e.g. turbine island within the acoustic sources).

The identification of a cluster of system wide elements that spans the bottom, or the right hand side, or both is characteristic of these sorts of DSMs. In the functional domain such a system wide layer would normally equate to the test & integrate tasks that are performed towards the end of a project sequence. In the physical domain there is no timeline associated with being in the lower right hand corner and so it is better to term this a 'bus' after the 'bus and module' style role it appears to play in the architecture. This bus is actually termed the 'main bus' as it was next observed that system-wide integrating functions need not necessarily be confined to this lower right area.

11.2.4.2 A Second Manual Clustering

This took the previous DSM as the starting point and made a few more manual changes to investigate some of the alternative locations suggested by the explorations of alternative binary clusters. After inspection the clusters were marked out and named and for the first time an unmistakeable auxiliary bus was observed. A dashed ellipse identifies this auxiliary bus. In retrospect it is now possible to see this auxiliary bus feature in other DSMs I have drawn, and it is also visible in DSMs that have been produced by others.

Observe how easy it is to give exactly the same cluster a different yet equally meaningful name, and how easy it is to identify a different cluster boundary in the same sequence. This demonstrates the dangers of manual intervention. One must strive very hard not to be seduced by false certainty in ordering the sequence, identifying the cluster boundaries, and in giving names that create self-fulfilling prophecies. It is for these reasons that at the very least some form of automated clustering algorithm is to be preferred and one must be extremely cautious about any names that are given – perhaps even to the point of developing a studiously value-neutral naming convention.

Before delving further into automated clustering algorithms it is worth stepping back and reflecting on the features seen so far. This allows clarification of the terminology so as to have a firm foundation for the next step. In the first instance this clarification is best done using stylised cartoons of lower levels of complexity.

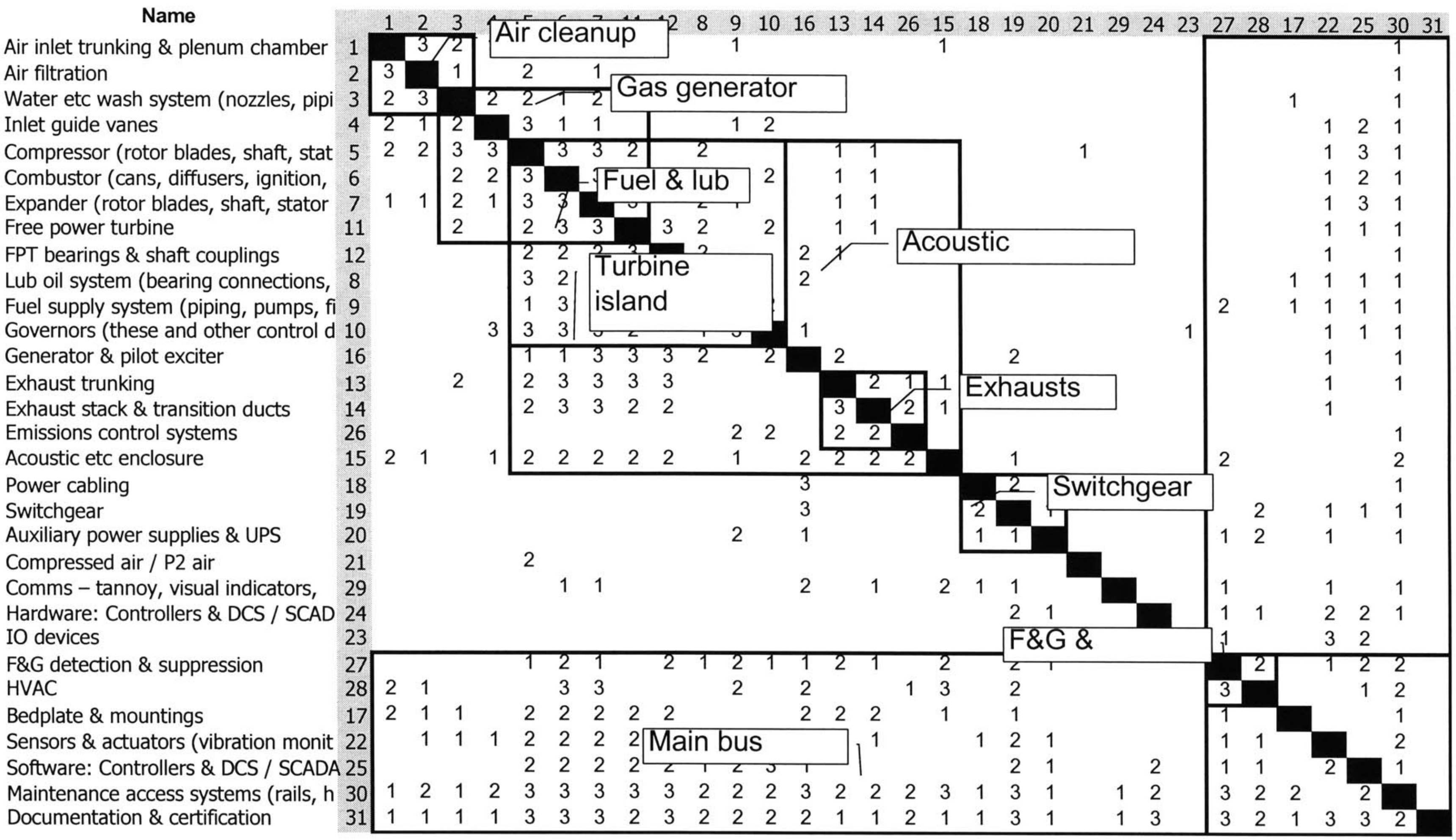


Figure 35: Physical DSM Of 10MWe Gas Turbine With Named Clusters – Initial Manual Version

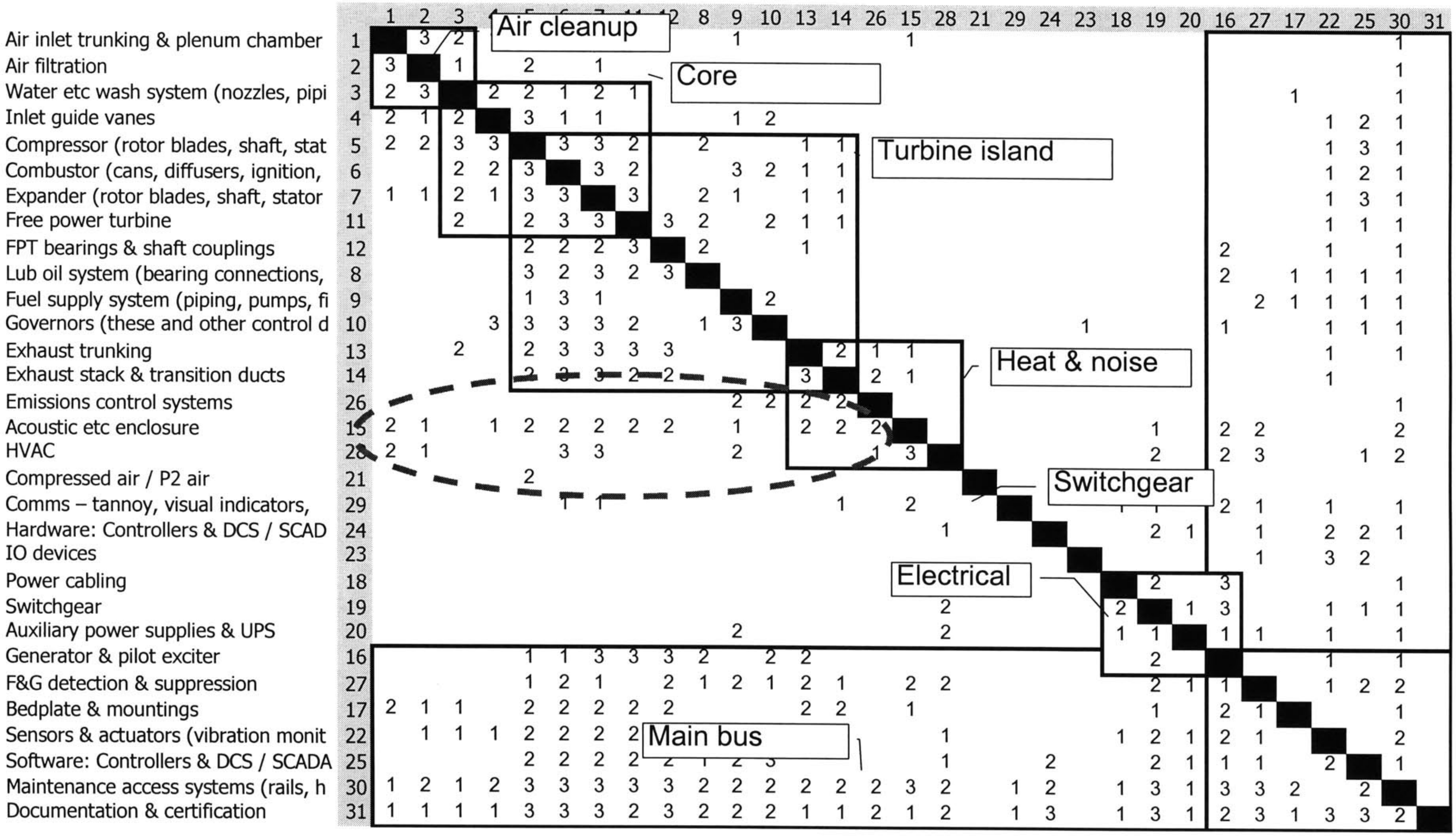


Figure 36: Physical DSM Of 10MWe Gas Turbine With Alternative Clusters – Second Manual Version

11.3 Cartoons, Terminology, Simplicity

It is important to recollect that these physical domain DSMs are simply representations of the degree of interconnectedness of the subsystems in the physical domain. As has been suggested all products are to some degree either integral or modular (Ulrich & Eppinger, 2000). The following diagrams outline how DSMs may be created that equate to those seen in the industrial gas turbine physical domain DSM.

11.3.1 Basic Concepts

Before diving headfirst into a discussion of the industrial gas turbine it is worthwhile considering the basic building blocks of the dependency structure matrix as a way of representing the physical domain.

11.3.1.1 Simple Bus, Four Chunks, No Modules

In Figure 37 below we have a relatively simple example of five related (connected) parts. Four parts are connected to a fifth and three of these four connections are of equal strength. This is shown on the left as a physical schematic that can be thought of as representing a flow of mass, energy, information, or force/geometrical constraint between the parts. This situation may be thought of as the situation where E is some form of system level integrating component, or bus, that is connected to parts A, B, C, and D. In this instance the relationships between the parts are symmetrical, i.e. part A depends on part E in the same way that part E does on A. To the right of the schematic is drawn the DSM for this system. A small 'x' represents a weak relationship or connection or dependence, and a large 'X' represents a strong relationship. To the right again is shown the more conceptual architectural diagram that represents the same situation. This third diagram can be drawn in any orientation as the relationships are symmetrical, thus at this stage no value should be assigned to any convention that chooses to draw part E above rather than below or to the side.

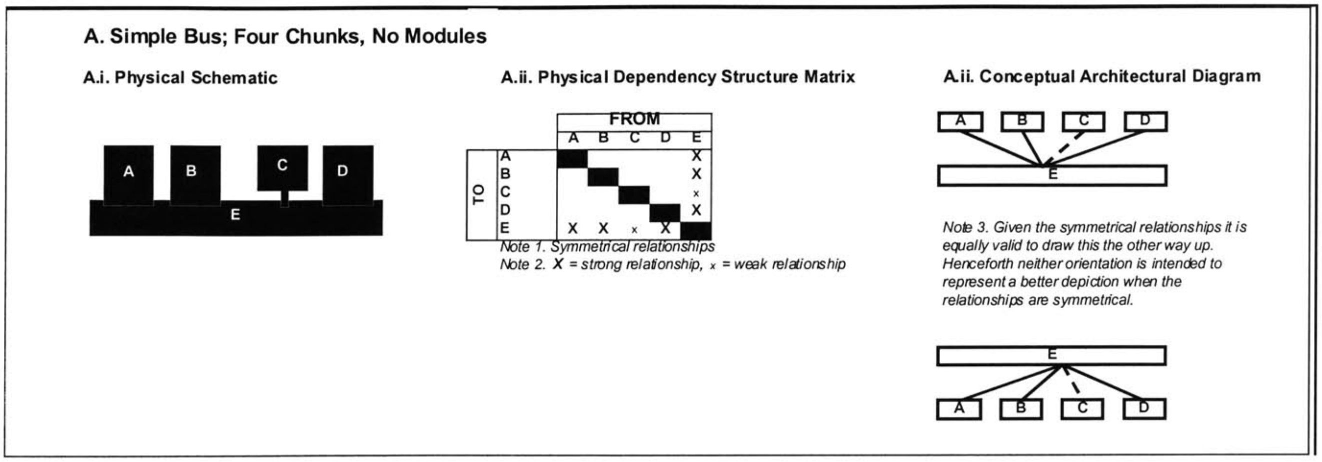


Figure 37: Simple Bus, Four Chunks, No Modules

At this point little discrimination is made between the strengths of the relationship in the architectural diagram although the concept has been introduced by showing a weaker relationship with part C. Parts can also be referred to as elements or chunks. Strictly speaking a part is the smallest possible decomposition of something whilst a chunk or element could be assemblies in their own right. For this reason strict terminology should differentiate between primitive elements (i.e. parts) or higher-level elements although in practice almost all writers are sloppy in this respect (myself included). I term a bus as being 'simple' when it is a primitive element.

11.3.1.2 Simple Bus, Four Chunks, Two Modules

In Figure 38 below we now introduce more relationships between parts. In this instance the pair of parts A and B are related symmetrically to each other as well as each possessing an equal relationship with part E, and a similar structure can be seen in the pair of parts C and D. In the DSM it makes sense to denote this pairing as being a modular cluster and here this is denoted by the (two letter) description Module SS, Module TT, etc.

In this instance the significance of being a modular cluster is that all the parts within the module possess the same relationships with each other and with parts outside of the module – this can be thought of as being the design rules of the module. In this instance the modularity is perfect, but in some cases modules will be similar rather than identical. In this case representing them as modules implies either that the module must be over-specified (in order to achieve all the functions) or that the performance will suffer (where only functionality present in all clusters is included in the standard module design). I term this imperfect modularisation and it is a feature of many designs that use modules as platforms that are then tailored.

In general modules represent clusters of high internal complexity and low external complexity. Ideally the module boundary would be drawn along the contour of least complexity, orthogonal to the flows of information from the inside to the outside and vice versa.

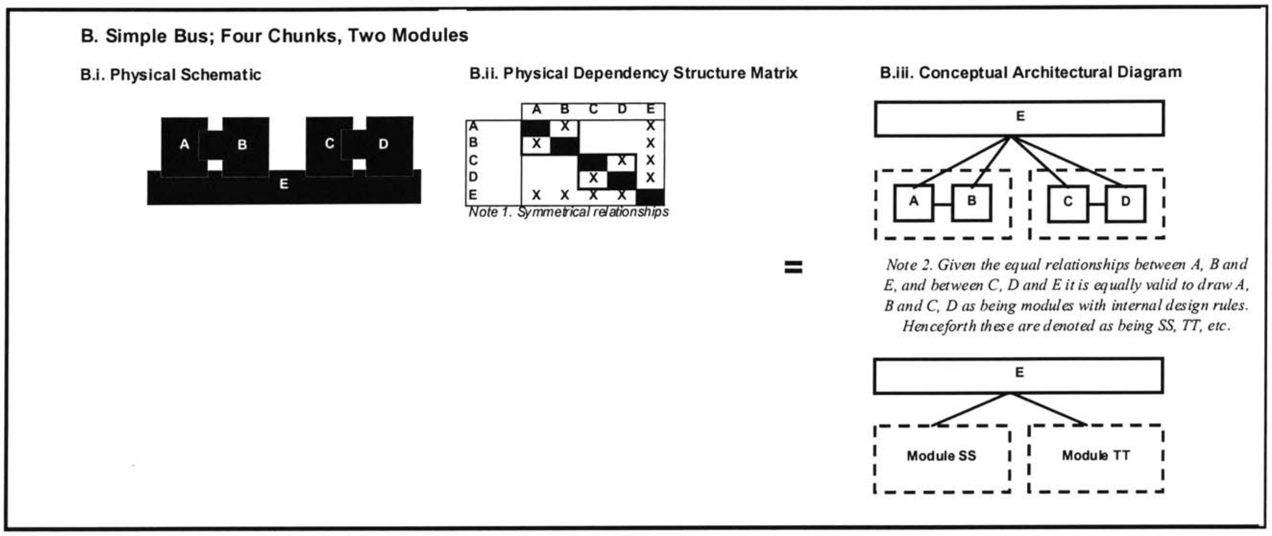


Figure 38: Simple Bus, Four Chunks, Two Modules

11.3.1.3 Bus Module With Four Chunks Forming Two Modules

Now we look at the situation where there are multiple system wide integrating chunks or elements. In this instance the two chunks E and F are connected equally to all other chunks. So whilst chunks A, B and C, D may be thought of as being modules, so too may the chunks E and F which gives a situation where the system is now comprised of two types of module. Here module SS and TT are both of the same type whilst module UU is of a different type which I, for convenience, term the bus module UU. In this instance the multiple buses that comprise the bus module are all perfect. Often I omit the expression 'module' when describing a 'bus module' as it is normally self-evident when the bus is not a primitive.

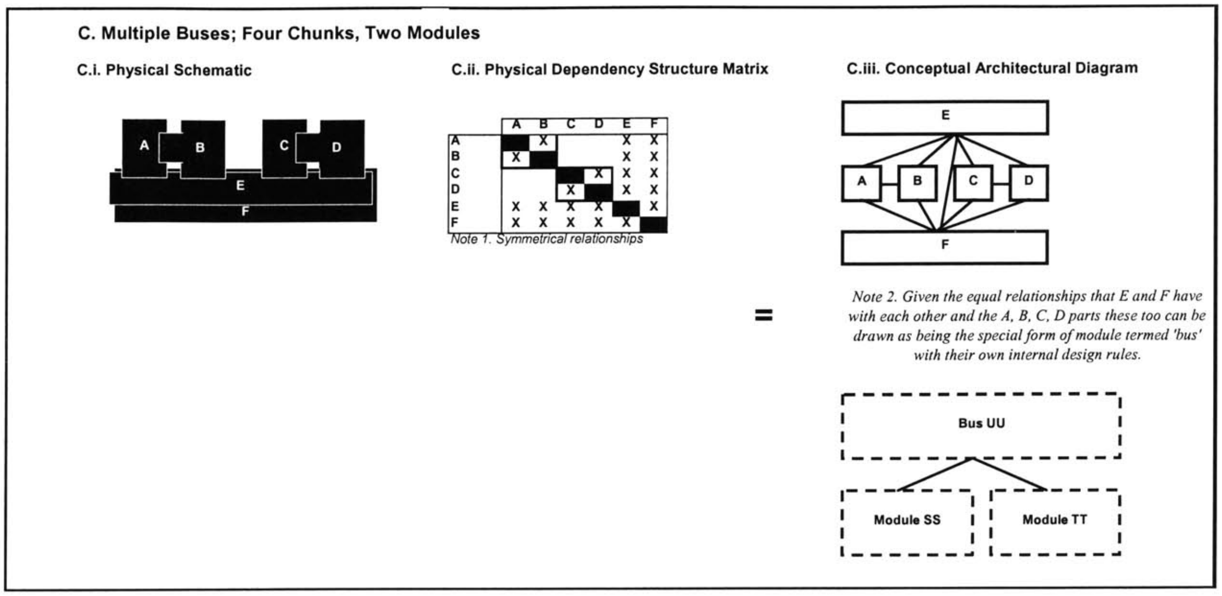


Figure 39: Bus Module With Four Chunks Forming Two Modules

In this instance the design rules governing modules SS and TT are different than those governing bus module UU. At this stage the value orientation of the conceptual architectural diagram has still not been specified as the DSM denotes symmetrical relationships between chunks. However in the absence of asymmetry the bus would tend to be more important as its design rules have a greater visibility than those of the modules SS and TT.

Until now all the relationships have been symmetrical and only the first cartoon DSM indicated the sign convention in use. This next cartoon introduces asymmetrical relationships.

11.3.1.4 Multiple Bus Modules; Six Chunks, Three Modules; Asymmetry

There are occasions where asymmetric physical domain relationships occur and this is illustrated by this example where the modular bus VV is more dependent on the modules SS, TT, UU than the reverse. Where this is the case pictorial conventions become more important in indicating the value gradient. In the case of DSMs there are two possible ways to read a matrix (this ones reads clockwise downwards from elements and then across to elements) and in the case of the conceptual architectural diagram it is necessary to explicitly denote the direction of control as even normal conventions are ambiguous.

The right hand diagram, the conceptual architectural diagram, is an exercise in ambiguity. What do arrows or orientation denote about dependency in this example? It is actually the reverse of what many people would have supposed as the bus module VV is dependent on all other modules and not the reverse. So in this instance the arrows flow from the source of power to the sink, and the orientation is with the higher value relationships at the bottom and not the top. This is a good example of the devil being in the detail.

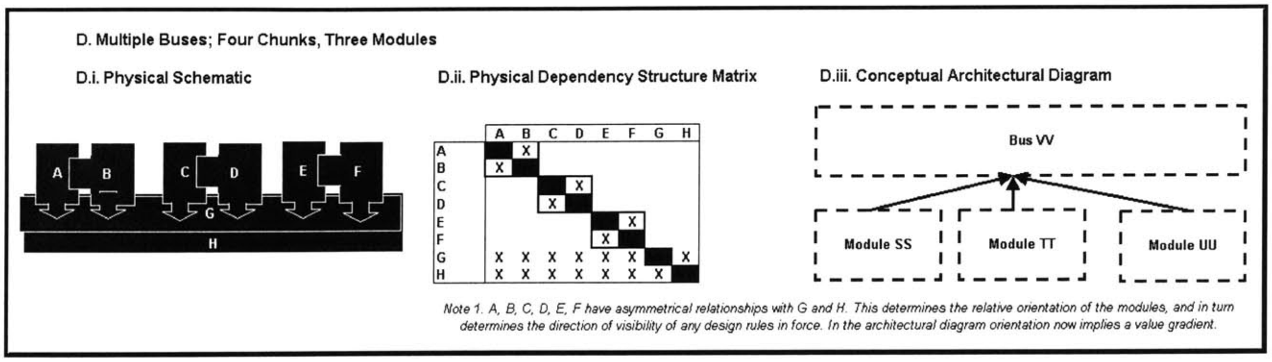


Figure 40: Multiple Bus; Six Chunks Of Three Modules, Asymmetry

11.3.2 Further Concepts

11.3.2.1 Pinning & Holding Away; Imperfection

Many architectures are more complicated than the examples inspected above. Here A, B and C, D can be concatenated into modules; or the sequence A through F be thought of as one super module; or an intermediate module TT be sandwiched between the module SS (comprising A, B) and module UU (comprising E, F); or simply described as being comprised of the primitives A through F with the bus module VV.

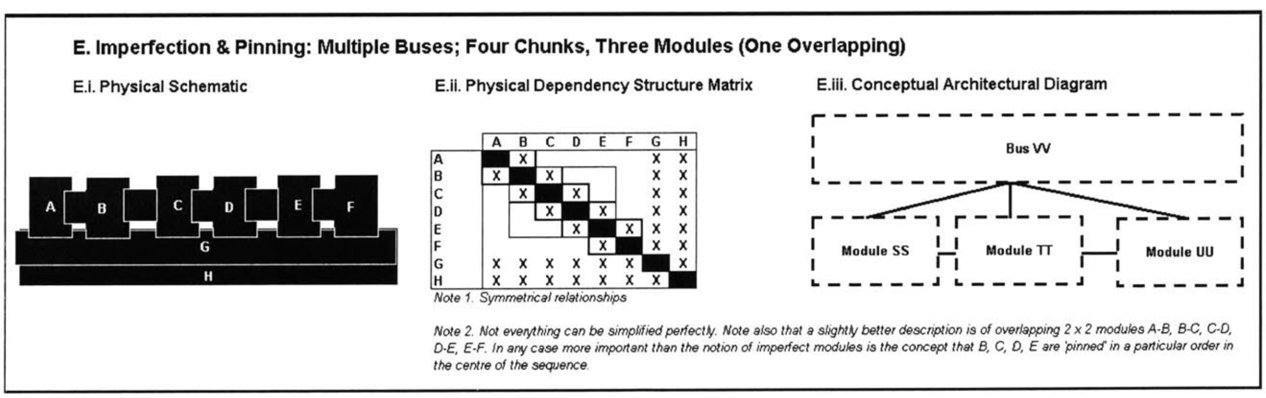


Figure 41: Imperfection; Pinning & Holding Away

None of these descriptions is an ideal description of such an integrated architecture. In an imperfect situation the manner in which the architecture is abstracted reveals the nature of the choices being made and the value system in use to make them. This is either done implicitly if the abstraction is manually executed, or explicitly if an objective function is used. Irrespective of the value system in use there can also be errors of commission or omission in execution, either by the individual or by the automated clustering algorithm.

This example also illustrates the notions of "pinning" and "holding away". Here B is pinned in place between A and C by its relationship with A and C. This is a common and easily appreciated situation in much physical architecture. A less easily appreciated situation is that of "holding away" which is seen here in that element A is held away from element C by element B. The combination of these two notions assists in describing the drawbacks of various clustering arrangements and clustering heuristics. Pinning can only occur to compound elements (modules) that have relationships with two modules, whilst any primitive element can be held away.

11.3.2.2 Path Dependency In Clustering

Consider a triangular arrangement of symmetrical relationships that can be loosely clustered into three similar modules AA, BB, and CC. The DSM for this can only ever show two of the real clusters and must break up the third. The way in which the clustering algorithm operates will be path dependent inasmuch as once it has started to cluster on any two nodes it is unlikely to over reverse out to a different configuration. This situation may occur because of the way in which raw data is presented to the clustering algorithm (for example branch and bound algorithms that are presented with a partially clustered starting point may never branch widely enough to evaluate alternative solutions) or may arise through chance if a perfectly random starting point is presented to an algorithm that makes an initial random guess. In the example below the cluster CC has been broken up even though it is identical in all respects to the other two clusters.

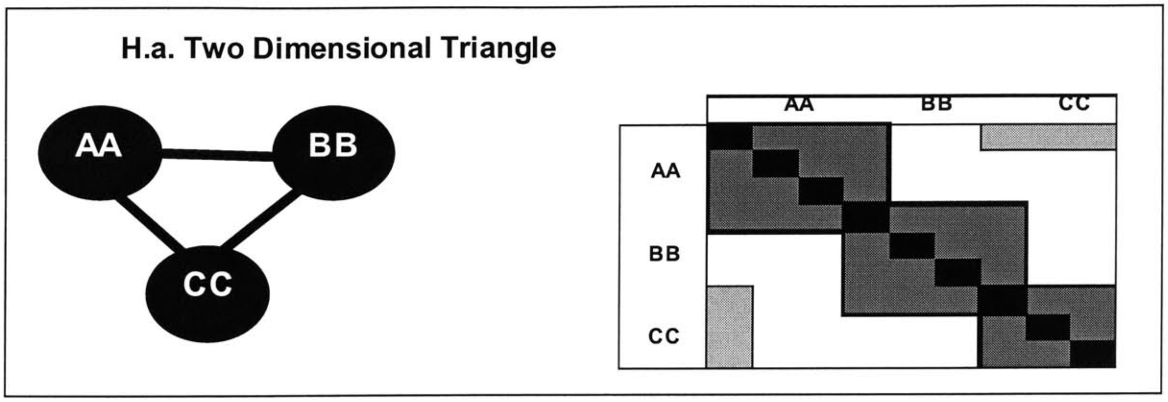


Figure 42: Triangular Clusters Illustrating Path Dependency

This situation arises because of a scarcity of dimensions. It is simply difficult to represent the two dimensional triangle in the one-dimensional space of a DSM. Theoretically one could cluster in multiple dimensions (whilst still remaining within the physical domain) and there may be merit in examining the possibilities this offers. A more important practical point in the short term is to test algorithms for over-stability as one would prefer algorithms that search widely for optimal clustering solutions yet are stable and robust. This topic will be returned to later.

This example also introduces the point that the DSM wraps along two pairs of edges, rather like a torus. This is not something that is commented on in the literature on functional domain DSMs as in those there is an implicit progression from upper left to lower right (it's not truly a time progression as progress can go backwards even as time goes forward; Browning, 1998) but is certainly relevant in a physical DSM.

11.3.2.3 Auxiliary or Weak or Subsidiary Buses

It is possible that elements within a module are pinned in place in the centre of a sequence, or are primitives held away from the main bus by a sequence that is pinned to the main bus, yet have bus-like characteristics. If this is the case the clustering algorithm will be forced to make a choice between disrupting modules adjacent to the main bus or forgoing the clustering opportunity. In the latter situation weak buses occur – computer science literature occasionally refers to the concept of a auxiliary bus or subsidiary bus which is why the alternative descriptions are noted.

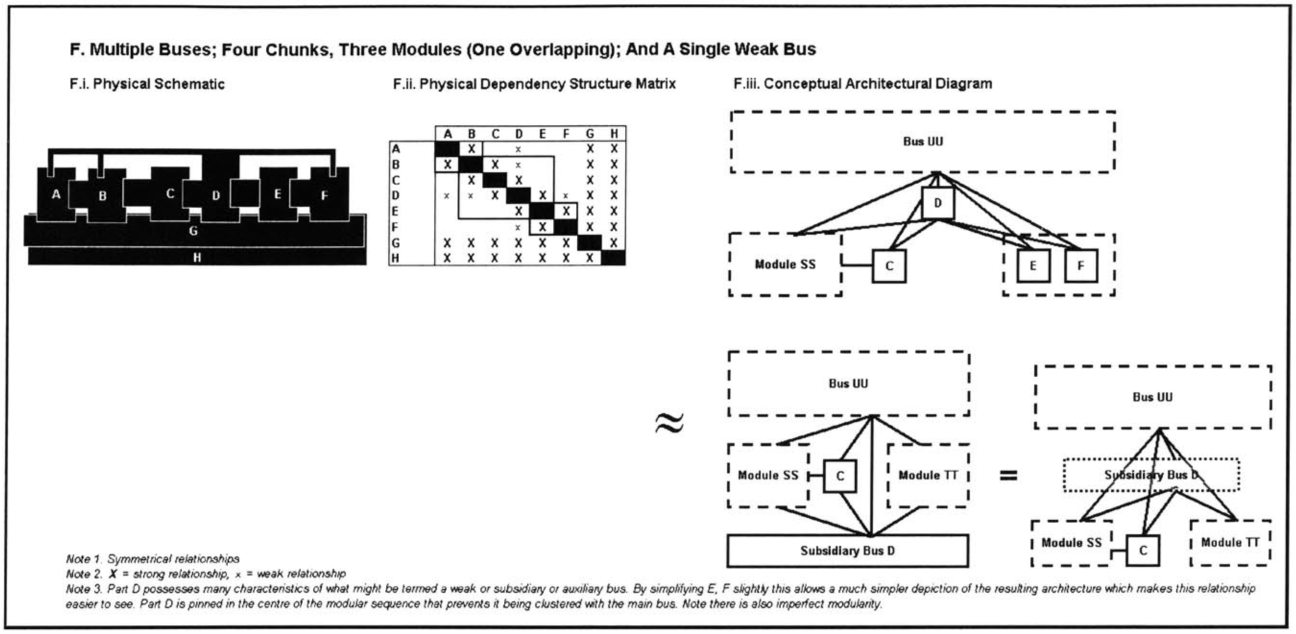


Figure 43: Auxiliary or Weak or Subsidiary Buses

This concept is illustrated in this cartoon where the primitive element D is pinned firmly in place in the middle of a sequence yet has weak relationships with much of the system. In the absence of the strong pins to the adjacent elements the weak system-wide relationships would cause it naturally to be incorporated into the main bus, however this migration cannot take place and so the weak bus remains stranded. In a real world system this would represent an element whose design has not only to take into account the upstream and downstream relationships with C and E, and the bus UU, but which also has to reach an accommodation with A, B, and F. Unsurprisingly any designer or maintainer of such an element will find it difficult to keep track of and optimise all the possible causes and effects. This difficulty in visualising the problem space is evident in the unsatisfactory conceptual architectural diagrams on the right.

In some respect auxiliary buses arise because the 'busness' of the element is less strong than either the modularity of the element (if it is pinned in a module), or the modularity of the elements that are holding it away. Thus the notional objective function is being asked to choose between two undesirable options – disrupt a module or force a bus out into an auxiliary bus and is guided in its choice by a comparison between the degree of perfection of the outcome. The choice will often also be path-dependent as, if a non-exhaustive search is conducted, the relative goodness of the choices will be determined by prior choices that have chosen to emphasise one alternative over another.

11.3.3 Real Levels Of Complexity

So far a series of very simple cartoons have been examined. At his point it is worth introducing slightly more realistic levels of complexity, then inspecting a real world example to see if some of the features under discussion are in fact present, and then returning to consider simple examples once more.

11.3.3.1 A Complex Cartoon

The cartoon below is intended to represent any semi-integrated real world product and is only titled 'industrial gas turbine' because it will shortly be compared with one and it might as equally be titled a chemical processing plant or an automobile.

It brings together a number of the concepts that have been outlined in the discussion above. There are a series of elements pinned to one another in a sequence and which form some sort of difficult-to-describe (i.e. imperfect) module; a weak bus pinned in place inside this sequence; a relatively clean modular bus; a relatively clean module; and a single primitive.

In the upper DSM of this situation the single primitive is shown in the centre. Any good clustering algorithm will recognise that this primitive is relatively indifferent to location and so it should simply be moved so as to optimise the location of the other elements. In this instance the primitive should be ejected to the upper left corner where it disrupts none of the modules.

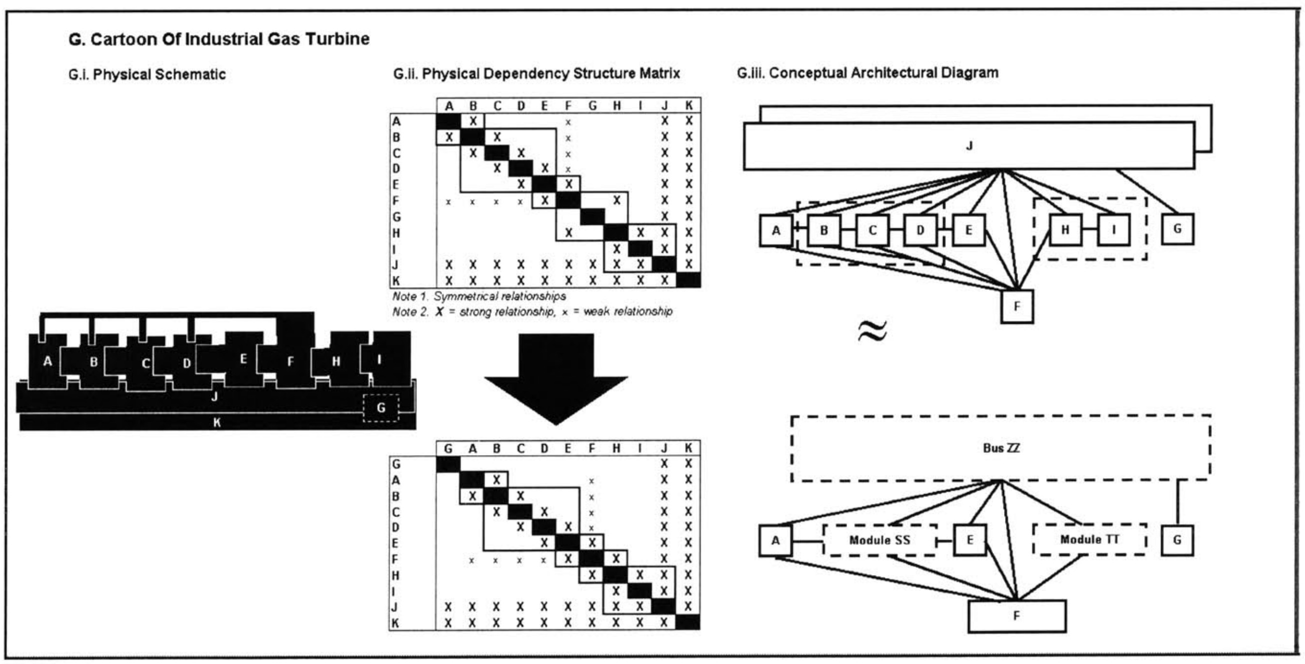


Figure 44: Cartoon Of Industrial Gas Turbine To Illustrate Stylised Real World Issues

As before it is difficult to describe this architecture by means of a network diagram yet this system barely counts as being complex given that it only comprises eleven primitive elements. Whilst these eleven primitives can be clustered usefully into slightly fewer groups, which eases the problem of description, it is difficult to objectively determine which of the possible groupings is best. It will first be necessary to codify the value system before being able to make progress.

11.3.3.2 Features Seen In The Real Gas Turbine DSM; More Manual Solutions

Turning now to the physical domain DSM of a typical industrial gas turbine (albeit the synthetic one previously considered) after manual clustering the same 31 primitive elements all the features noted above emerge. This manual clustering used the last best version (Figure 36) as its starting point but then employed the improved understanding of the issues that was gained from exploring the cartoons to improve once more.

This DSM as shown in Figure 45 exhibits weighted strength asymmetric relationships and this is most evident in the manner in which the bus module is populated. The below diagonal portion of the bus module is more heavily populated (both when looking at the simple binary existence of relationships and when looking at the strengths of the relationships) than the above diagonal portion of the bus module. Under the convention in use this asymmetry indicates that this bus module is relatively dependent on the remainder of the gas turbine system rather than the system being dependent on it.

A sequence of five non-bus modules is identifiable. In this diagram they are given identifying names: heat & noise; turbine island; core; air clean-up; electrical. All but one of these modules are pinned together and one is pinned to the main bus module. The unpinned electrical module exhibits an auxiliary bus characteristic (outlined by the dashed oval) but is apparently held away from the bus module by the pinned sequence.

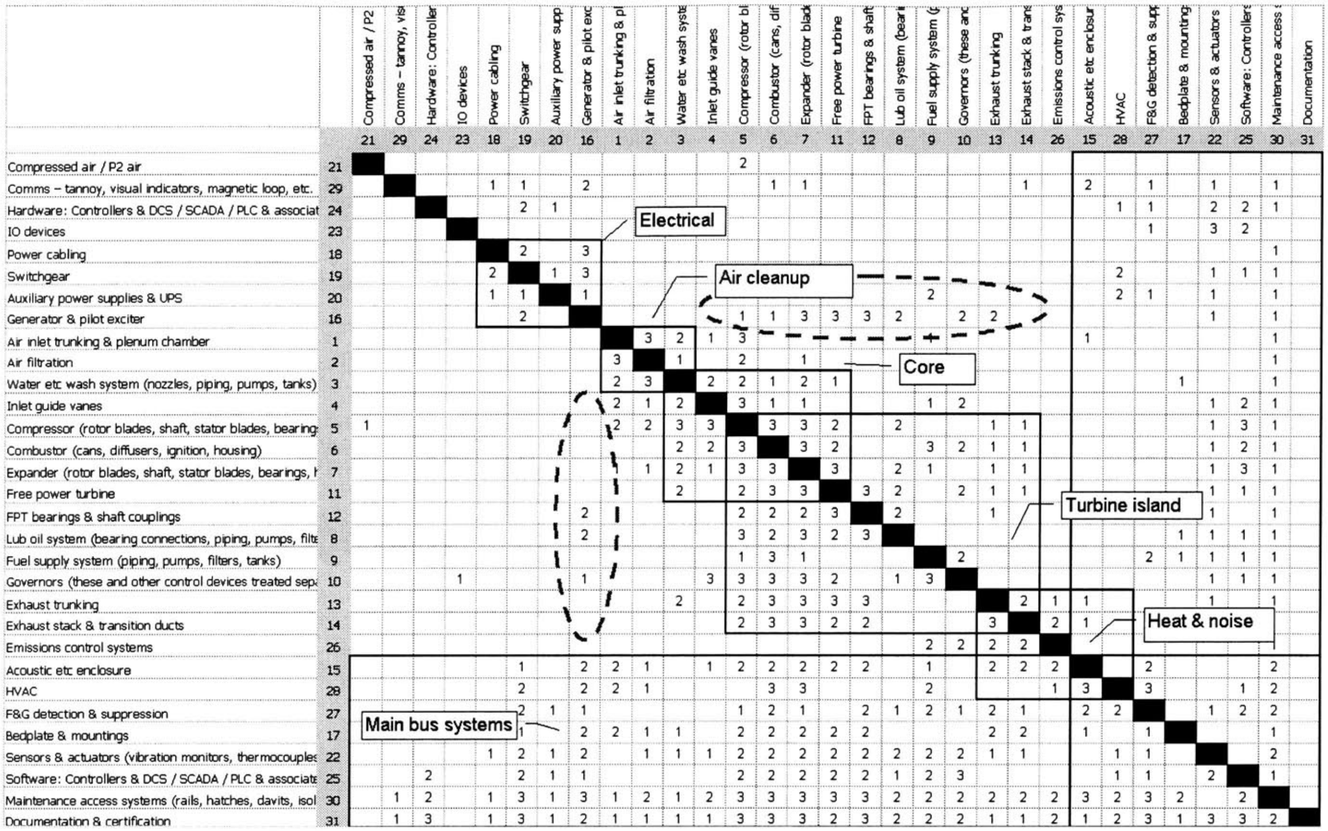


Figure 45: Improved Manually Clustered DSM Of Industrial Gas Turbine

Some of these modules are relatively fully populated but others less so, and there are varying degrees of asymmetry in the modules. Some of the modules are larger than others. At this point it is valid to enquire whether this is the best clustering arrangement. The simple answer is no as, by inspection the alternative solution shown in Figure 46 below is better.

This second solution can be shown to be better by inspection as it changes no sequences within modules but simply adjusts the sequence of some of the primitives in the bus module so as to allow the unpinned electrical module to nestle against the bus module. This obviously reduces the size of any notional penalty term associated with the relationships denoted by the auxiliary bus being held away from the main bus, which is why it is "better by inspection". This second solution employs the torosity of the DSM to yield the better solution.

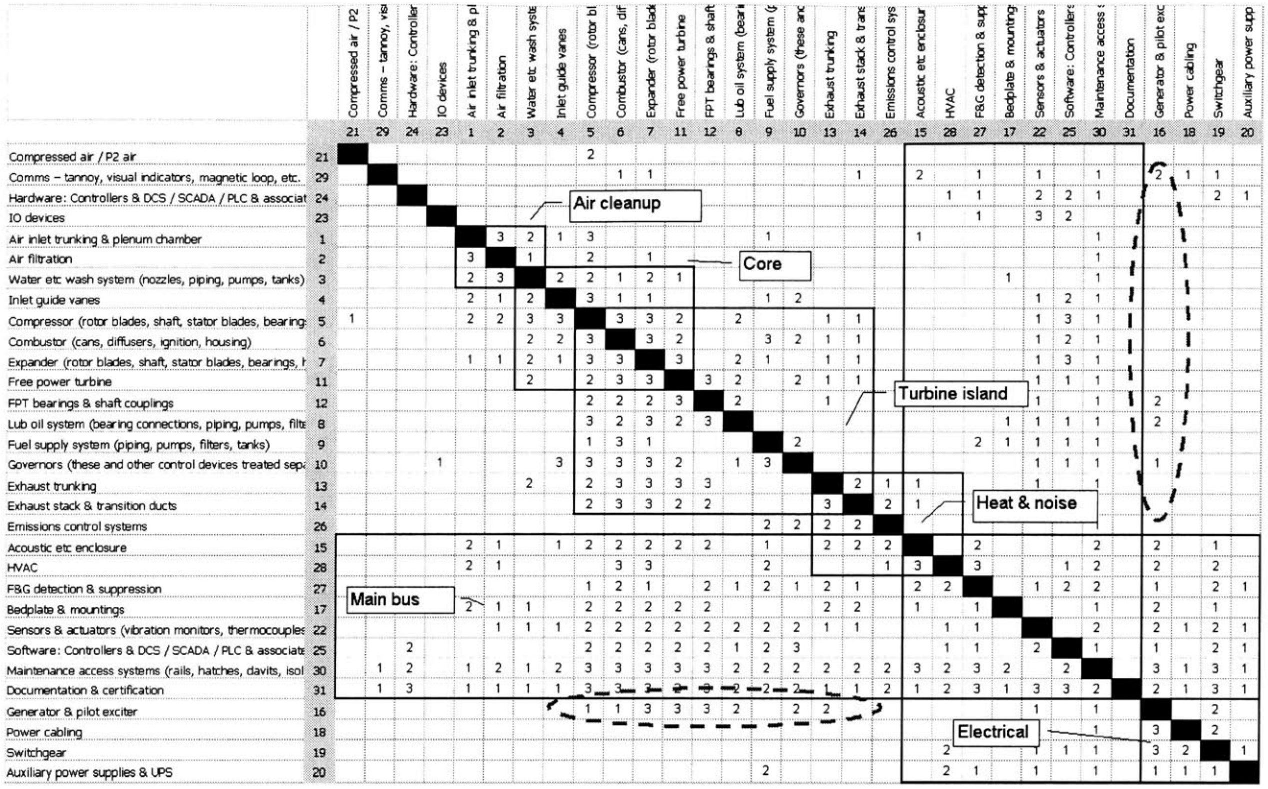


Figure 46: Even Better Manually Clustered DSM Of Industrial Gas Turbine

This alternative solution well illustrates the important point that in the physical domain there is no reason why the bus module is best parked in the lower right or upper left corners. Inspection of all the literature shows that in the physical domain these are the normal parking locations and it appears that this is a carryover from the task (or process or activity) domain DSMs where there is good reason to prefer this layout. Exploitation of this parking freedom is possible because the upper left element of the diagonal is adjacent to the lower right element and this leads on to further consideration of the idea of dimensionality, but first there are two more points to be made.

Firstly it may still be valid to use the normal convention if the number and arrangement of modules permits it. This is because it is easiest to inspect, and because the physical domain often maps closely to the task domain so arranging it in this way eases the process of comparison. This point will be returned to later when discussing inter-domain mapping.

Secondly a series of runs of automated clustering algorithms with a range of possible clustering control parameters failed to reveal this alternative clustering solution. This suggests that the underlying clustering heuristics are less rich than the problems they are being applied to. In order to develop better clustering algorithms and metrics it is necessary to fully understand the principles of clustering and the language of representation. Illustrative examples of the results obtained by different automated approaches are in the next section.

Lastly it is worth pointing out something that may be important. There are only 6 modules or clusters (including the bus) in this suggested arrangement of the DSM of the gas turbine. This is very important as we began this chapter with the notion that there might be an intermediate level of decomposition of the 31 heterogeneous elements that in some way resolved the tension with an optimal decomposition to ~4 modules. It appears that some progress is being made, although it is too early to tell if it is illusory.

11.4 Applying Automated Algorithms To The Weighted Gas Turbine DSM

11.4.1 Origin Of The Clustering Algorithms

Alexander & Manheim (1962) and Alexander (1963) formulated early clustering algorithms and the underlying principles are explained in Alexander (1964). These decompositional algorithms work in either a top down or a bottom up manner, either by cleaving along the line of least resistance or by aggregating along the path of greatest return. In both cases the relevant metric is some function of the number of linkages that cross a partition boundary. In Alexander's work this objective function was implemented with a simple hill climbing search strategy whereby the system was either successively partitioned into ever-smaller sets or successively aggregated into ever larger sets, one element at a time.

This approach was conceived before DSMs were in vogue but at a time when graph theory was becoming popular at MIT. It therefore went all the way back to fundamentals and represented design problems as a graph, G(M,L) of the set L links associated with the variables of M. Normally the domain of M was defined as being that of properties (which I would term functions or parameters depending on usage) and the resultant intermediate answer was then used to guide selection of the final design in the form domain (which I would term physical). Alexander was very careful to point out that neither representation objectively describes either the problem or the solution itself but instead both are intermediate representations used by the designer. Alexander not only made use of graph theory to develop the objective function, but also incorporated information theory by defining the information content per Shannon-Wiener (1949).

Following on from Idicula (1995) Fernandez (1998) used the same approach as Alexander's bottom up aggregation. Each element is placed in an individual set and bids evaluated from all the other sets (clusters). If any cluster is able to make a bid that is better than the current base case then the element is moved inside the cluster. The objective function is therefore a trade-off between the costs of being inside a cluster and the overall benefit. Unlike Alexander's highly theoretical approach to developing the objective function Fernandez's objective function builds upon a series of more empirical work emanating from the product design community (indeed Fernandez's algorithm originates from Idicula (1995) both whom were students of Eppinger) which in turn was based upon more generalised clustering work conducted for a variety of industries and applications. Fernandez was concerned with clustering teams together to work on a product under development and the dominant heuristic in the objective function is that element interaction costs increase non-linearly with the number of teams (clusters), probably in a quadratic manner (a user-defined variable power is used '*pow_cc*'). If this were all the system would naturally fragment to single element clusters and so against this is balanced a cost that is dependent on the numbers of clusters. The result is a suggested number of clusters and a list of elements in each cluster. Important features of these results are that:

1. Clusters can overlap, i.e. possess common member elements.
2. The number of clusters is an endogenous output not an exogenous input (equally this could be stated as the preferred size of a cluster is an output rather than an input).
3. Within a cluster there is no significance in the order of elements.
4. Between clusters there is no significance in the order of clusters, except where they have common members.

The clustering algorithm of Idicula is innovative inasmuch as point 1 above is generally not permitted and point 2 above is generally an exogenous inputs. Since most real world problems have an uncertain decomposition, and since hierarchical decompositions with open topologies are rare this was a useful advance. Fernandez's contribution was to rewrite the C code in a more modular format, link it into Excel as an add-in, refine the user-friendliness greatly, and carry out a lot of performance testing work. Finally Fernandez improved the search strategy over both Alexander and Idicula by incorporating a simulated annealing algorithm to avoid sticking in a local optimum.

Thebeau (2001) extended Fernandez's work and repackaged it into stand-alone MatLab code. The objective function was tuned so as to minimise inclusion of elements in more than one cluster (which I term pinning) and the simulated annealing algorithm was refined so as to minimise instability and the likelihood of returning a worse result than the best intermediate one encountered. These two changes speeded up the execution of the Fernandez algorithm and made it more suitable for handling larger DSMs up to 50 elements or so. The objective function tuning was achieved by introducing a penalty term for membership of multiple clusters – in theory multiple membership is still achievable, in practice it is seldom so. Storing the best intermediate answer and returning to it if necessary fixed the issue of the final answer being worse than intermediate ones. As with Fernandez, Thebeau permitted varied inter-element interaction strengths, indeed he points out that this improves the solution stability and explains why. Finally Thebeau applied this algorithm to a 50 element real case (unlike Fernandez who used much smaller and simpler test cases) and noted that it was unable to cope with system wide integrative elements. These disrupted the objective function and in practice were being near randomly scattered around clusters. In order to alleviate this provision was made for manually holding these elements out of the automated process and reinserting them at the end as lower right elements. Thebeau terms these the 'system' elements whilst I term them the 'bus' elements. As with Fernandez's work it is necessary to tune the algorithm control parameters for each case so as to obtain the best result.

11.4.2 Results Obtained With Automated Clustering Of Weighted Relationships

Since the Thebeau clustering algorithm represents the best tool available it was tested on the fully weighted physical DSM of the gas turbine. Four results are presented here to indicate the typical outcomes for different control parameter settings. In all of these trials the seed DSM was held constant as shown in the following figure. In these DSMs the greater the visual impact of the tick mark the stronger the relationship denoted. The element numbers refer to the key shown in the DSM of Figure 45 & Figure 46.

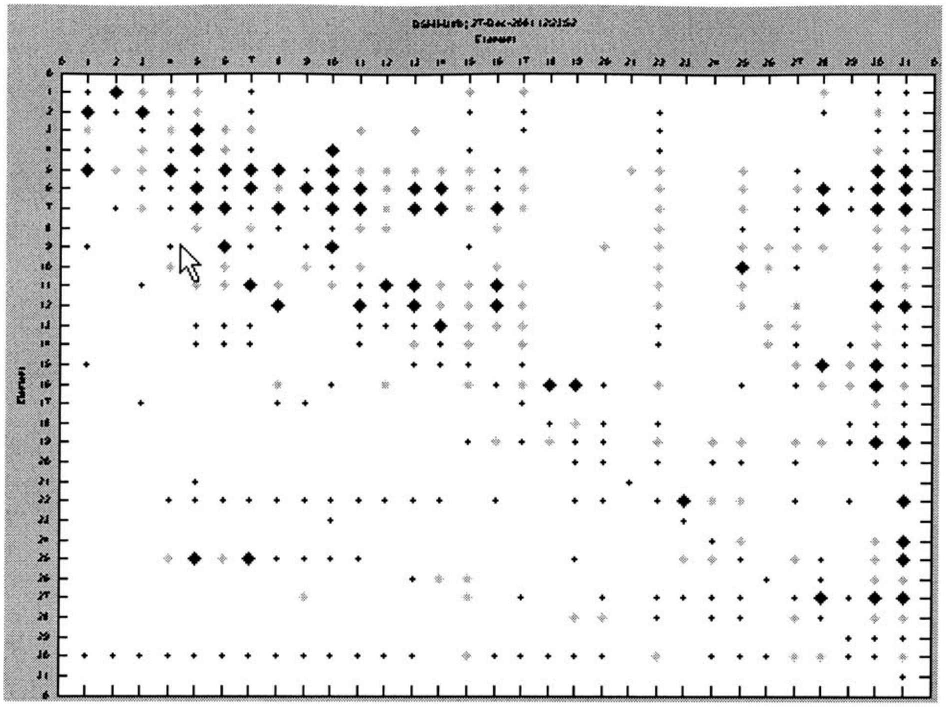
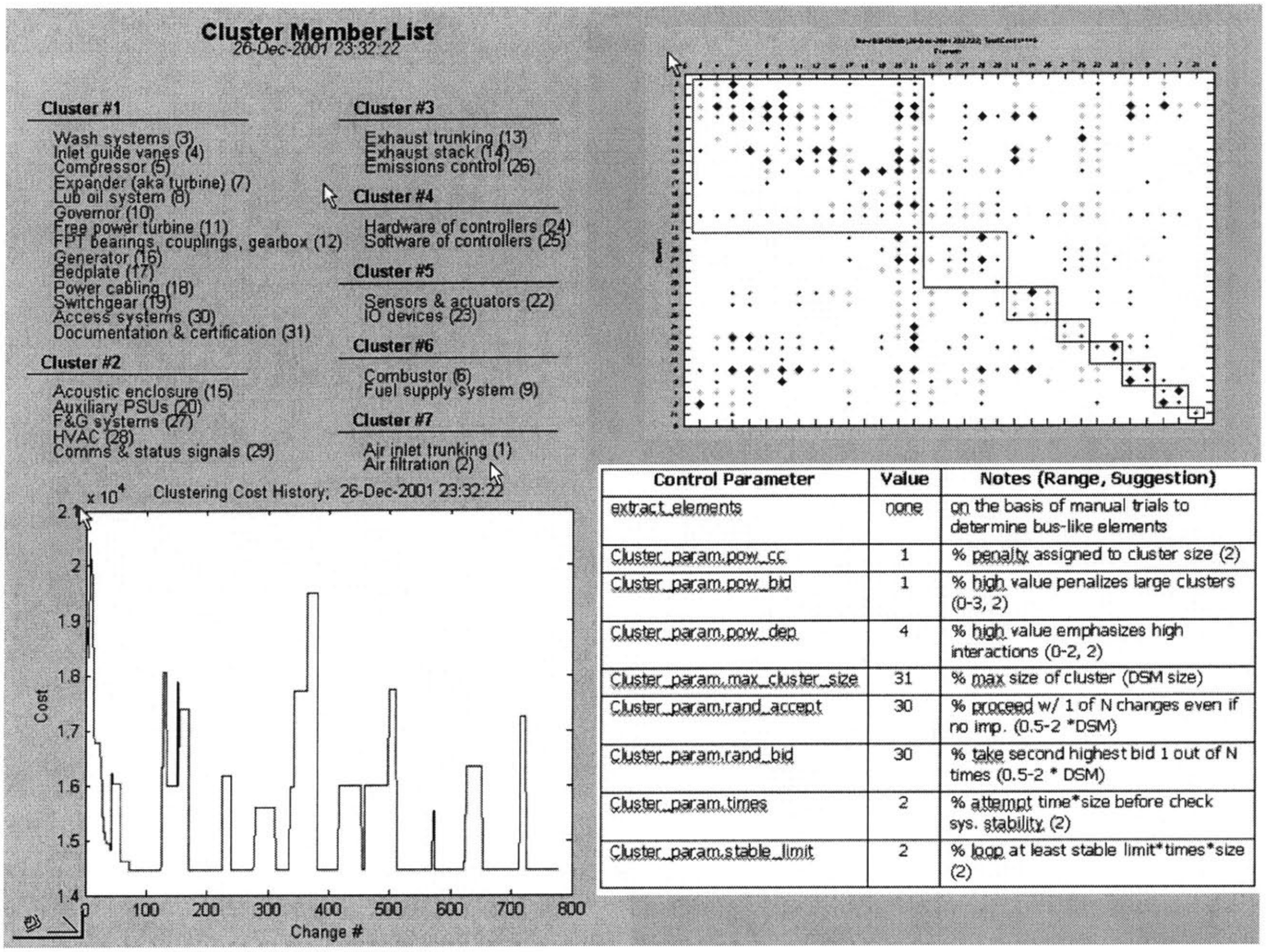


Figure 47: Seed DSM Of Gas Turbine Used For Thebeau Clustering Algorithm

Each of the trial inputs and outputs is represented in a common summary format with four quadrants. On the bottom right is the list of control parameters. On the top right is the solution DSM with clusters delineated by blue boxes. On the top left is the list of elements of the proposed clusters – normally truncated so that the singleton 'clusters' are not shown. Finally on the bottom left is the objective function's cost history showing the search trajectory taken by the simulated annealing algorithm – note that the scale is not and should not be directly compared between runs as the objective function is changed by tuning the control parameters.

11.4.3 First Run With Thebeau Algorithm

In the first run the control parameters were set per Thebeau's final recommendations.



Control Parameter	Value	Notes (Range, Suggestion)
extract_elements	none	on the basis of manual trials to determine bus-like elements
Cluster_param.pow_cc	1	% penalty assigned to cluster size (2)
Cluster_param.pow_bid	1	% high value penalizes large clusters (0-3, 2)
Cluster_param.pow_dep	4	% high value emphasizes high interactions (0-2, 2)
Cluster_param.max_cluster_size	31	% max size of cluster (DSM size)
Cluster_param.rand_accept	30	% proceed w/ 1 of N changes even if no imp. (0.5-2 *DSM)
Cluster_param.rand_bid	30	% take second highest bid 1 out of N times (0.5-2 * DSM)
Cluster_param.times	2	% attempt time*size before check sys. stability (2)
Cluster_param.stable_limit	2	% loop at least stable limit*times*size (2)

Figure 48: First Result With Thebeau Algorithm

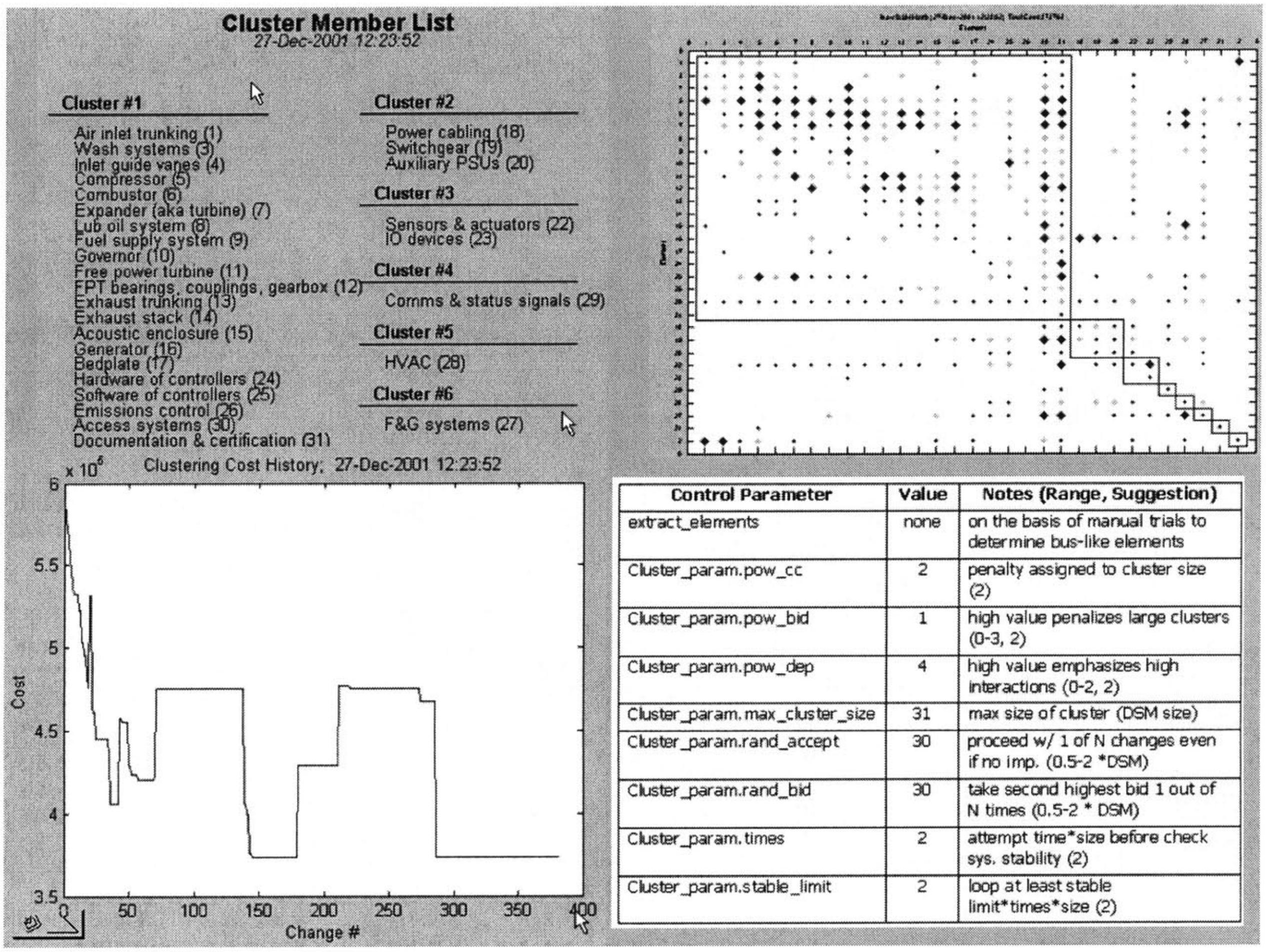
Notice how the objective function's cost history repeatedly rises as the best intermediate solution is set aside whilst a new search is made. It appears that a stable good solution is being found rapidly and reliably.

The first cluster contains fourteen elements, the other seven are much smaller, and none of the eight clusters overlap (a singleton is not shown on the cluster list). By eye it can be seen that some system wide integrating elements (bus elements) are caught up amongst the clusters.

Visually this seems to be a less 'good' solution than the best manual solution. Of course 'goodness' - visual or otherwise – is a rather vague term but in an effort to determine whether it is possible to improve the control parameters were tweaked one by one in the following runs.

11.4.4 Second Run With Thebeau Algorithm

On the second run the *Cluster_param.pow_cc* term was increased from 1 to 2. This is the power term that penalises the size of a cluster in the cost equation: when it is set to one it is a linear function, when it is set to two it is a quadratic, etc. The results of this are shown below:



Control Parameter	Value	Notes (Range, Suggestion)
extract_elements	none	on the basis of manual trials to determine bus-like elements
Cluster_param.pow_cc	2	penalty assigned to cluster size (2)
Cluster_param.pow_bid	1	high value penalizes large clusters (0-3, 2)
Cluster_param.pow_dep	4	high value emphasizes high interactions (0-2, 2)
Cluster_param.max_cluster_size	31	max size of cluster (DSM size)
Cluster_param.rand_accept	30	proceed w/ 1 of N changes even if no imp. (0.5-2 *DSM)
Cluster_param.rand_bid	30	take second highest bid 1 out of N times (0.5-2 * DSM)
Cluster_param.times	2	attempt time*size before check sys. stability (2)
Cluster_param.stable_limit	2	loop at least stable limit*times*size (2)

Figure 49: Second Result With Thebeau Algorithm

It had been expected that this increase would result in the large cluster being broken up but instead the reverse happened and it increased in size. This may be a result of the search becoming trapped in a local optimum or it may be the result of other factors. Three re-runs are revealing as the next figure shows, each with these same control parameters.

On this occasion the objective function values are all comparable. The starting point is always 6 x 10^{-5} and the outcomes are 3.5, 3.7, 3.9 and 3.7 x 10^{-5} respectively. The span of outcomes (0.4 x 10^{-5}) divided by span from the start to the best outcome seen (2.5 x 10^{-5}) is about 16% and corresponds with a wide range of cluster sizes (21 elements, 14, 18, and 21). In practice the Thebeau algorithm's search strategy does not appear as robust as first thought, especially if the objective function control parameters are inadvertently selected so as to suppress the degree of differentiation (information content) in the raw (seed) data.

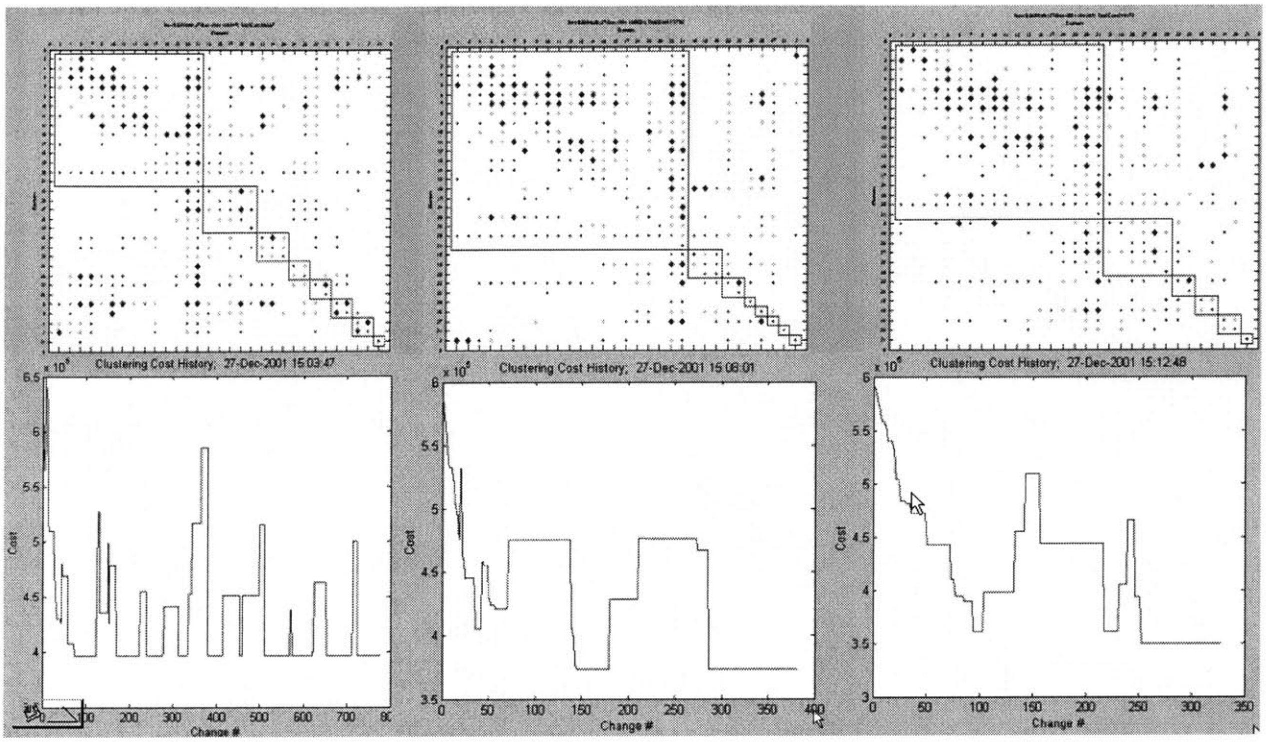
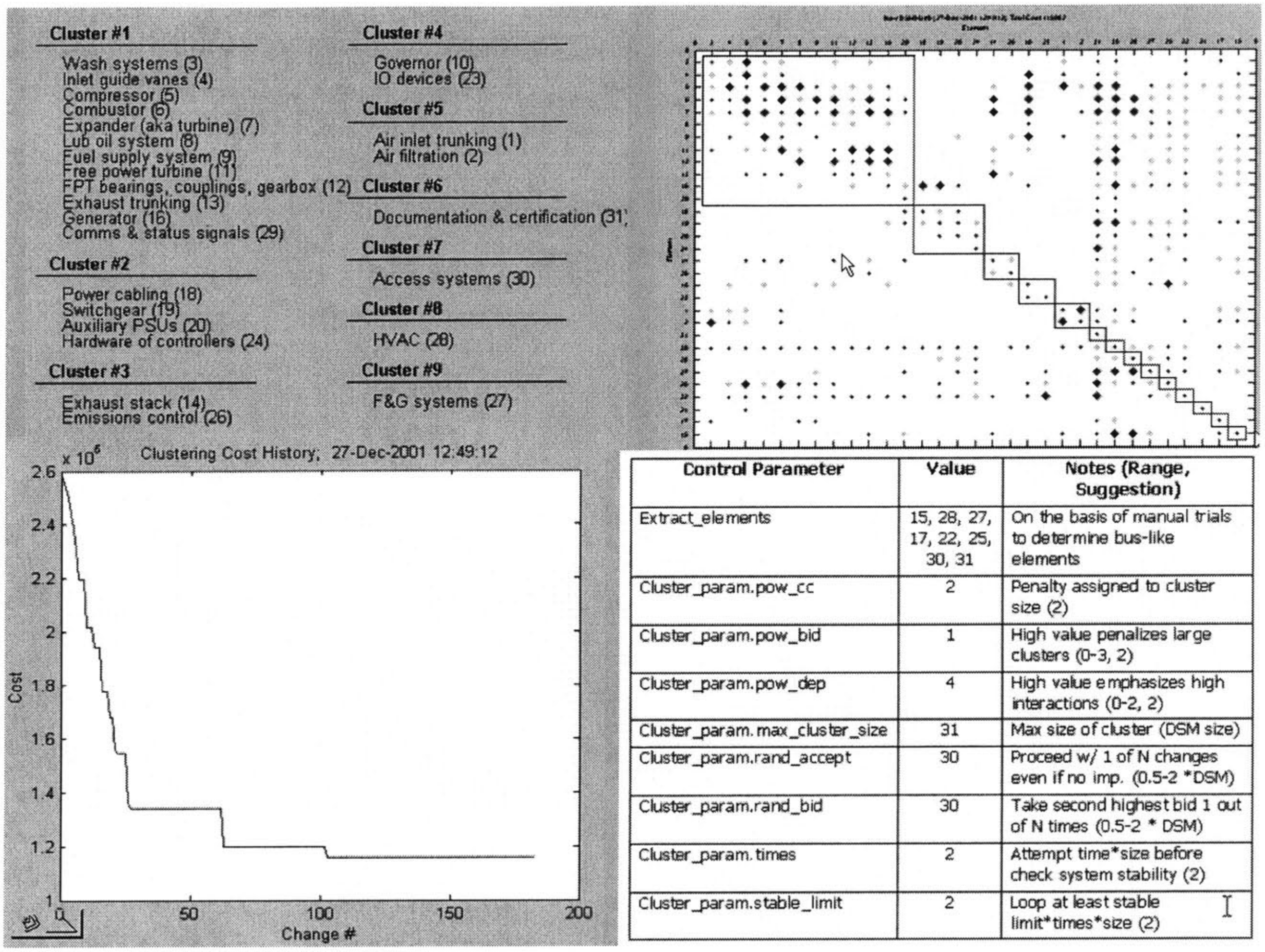


Figure 50: Second Result Revisited With Thebeau Algorithm

One reason why this may be occurring was commented on by Thebeau and is because the system wide integrating elements (buses) are being included in the bidding process. Because these are widely related they will inevitably be selected for inclusion in any particular cluster, and the more perfect their 'busness' is the more random will be their allocation. However because they are so widely related, once allocated to a cluster they create a self-perpetuating path dependency that will influence that cluster's growth. For this reason Thebeau allows them to be manually set aside from the bidding process and simply pasted in afterwards in the order they are identified for set aside.

11.4.5 Third Run With Thebeau Algorithm

In order to test out this a third series of runs was performed with the main bus elements protected from the clustering algorithm. These elements are the ones previously noted in Figure 45, i.e. elements [15, 28, 27, 17, 22, 25, 30, 31] and when they are reinserted they are always reinserted in this order, into the lower right end of the sequence.



Control Parameter	Value	Notes (Range, Suggestion)
Extract_elements	15, 28, 27, 17, 22, 25, 30, 31	On the basis of manual trials to determine bus-like elements
Cluster_param.pow_cc	2	Penalty assigned to cluster size (2)
Cluster_param.pow_bid	1	High value penalizes large clusters (0-3, 2)
Cluster_param.pow_dep	4	High value emphasizes high interactions (0-2, 2)
Cluster_param.max_cluster_size	31	Max size of cluster (DSM size)
Cluster_param.rand_accept	30	Proceed w/ 1 of N changes even if no imp. (0.5-2 *DSM)
Cluster_param.rand_bid	30	Take second highest bid 1 out of N times (0.5-2 * DSM)
Cluster_param.times	2	Attempt time*size before check system stability (2)
Cluster_param.stable_limit	2	Loop at least stable limit*times*size (2)

Figure 51: Third Result With Thebeau Algorithm

The bus elements form eight of the 31 elements and so the visual impression is at once tidier. This is helped by a very effective grouping of many of the remaining high strength interactions into a 12-element cluster. Unfortunately the algorithm seems unable to make further progress and as a result almost as many relationships are left unclustered as are clustered. In order to see if this was simply bad luck another three re-runs are shown below, and as before a rather disconcertingly wide variation is seen:

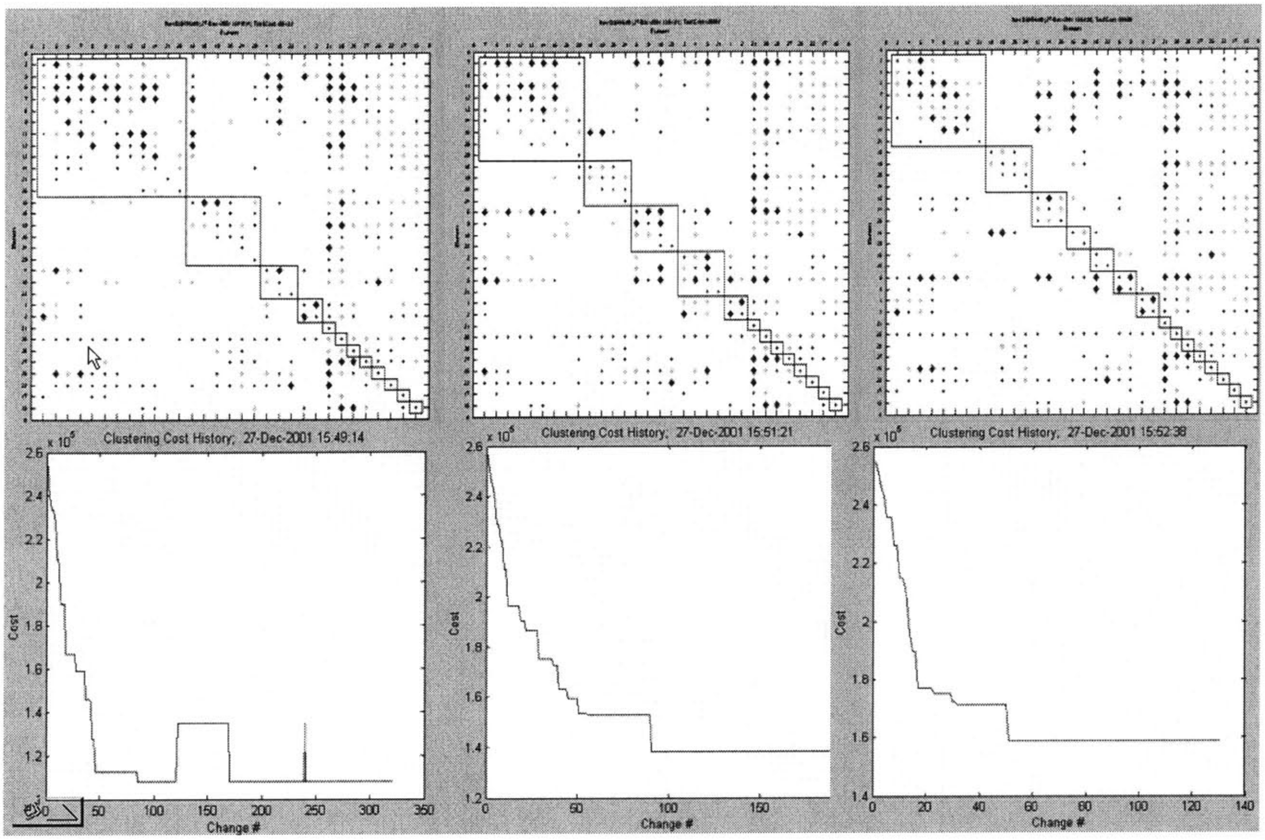


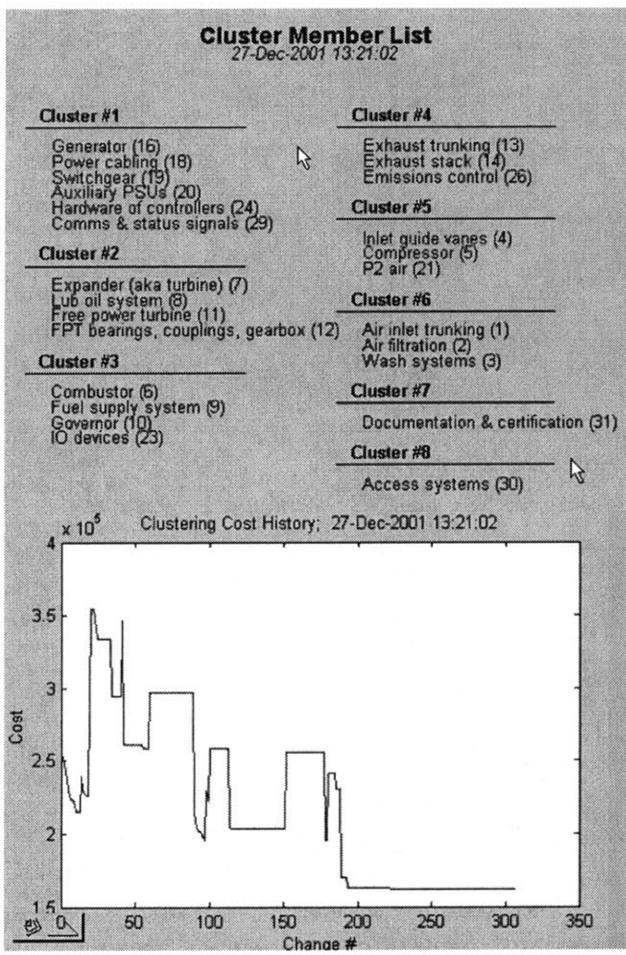
Figure 52: Third Result Revisited With Thebeau Algorithm

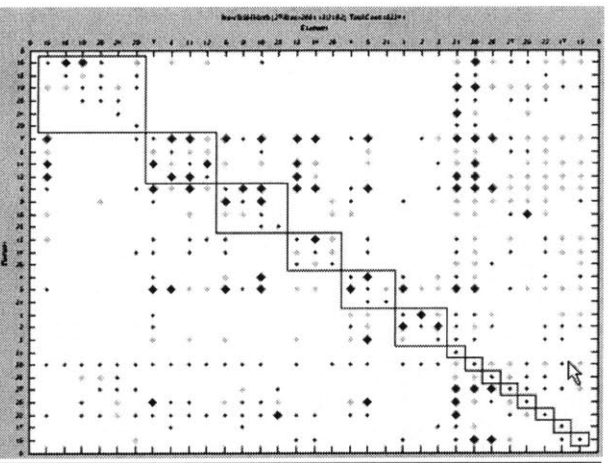
11.4.6 Fourth Run With Thebeau Algorithm

In the previous runs some very large clusters are seen and so on the fourth run the *param_pow_bid* was increased from 1 to 3. This parameter penalises the size of clusters – when it is set at 0 it has no effect. It certainly works as advertised and the 23 manoeuvrable elements are now more evenly distributed amongst six clusters but once again there are a lot of unclustered relationships on display.

As before three re-runs are shown to gauge the effectiveness and repeatability. On this occasion it is not as easy to gauge these two factors visually – apart that is from observing the ratio of clustered to unclustered relationships, which appears pretty constant. The reason for the increase in difficulty is that the order of clusters seems almost accidental once clusters are the same size (it is not accidental at all, but the very low relief in the value landscape is swamped by the randomness in the bidding process). The order of elements within clusters is similarly semi-random and this means one cannot simply scan between the three (or four) runs. Instead it becomes necessary to pay attention to the details in both the DSM and the final outcome of the cost history, and whilst some combinations of elements are seen to be fairly rigid others are not and quite a lot of variation occurs in the result.

Apart from the randomness that is going on in the bidding process it is also worth noting some things that are missing. The eight bus elements that are set aside are given a sequence and are imperfect. However their order does not affect the coalescence of clusters at all: the information content in the bus's imperfection is lost. Indeed the same occurs within the clusters – once clustered the sequence is of no importance, as proximity does not matter. Also no pinned clusters have been seen in any of the results shown. Indeed it was not possible to force pinning to occur even when the control parameters were adjusted as far as possible to encourage it (although it has to be admitted that not all parameter combinations were tried).





Control Parameter	Value	Notes (Range, Suggestion)
Extract_elements	15, 28, 27, 17, 22, 25, 30, 31	On the basis of manual trials to determine bus-like elements
Cluster_param.pow_cc	2	Penalty assigned to cluster size (2)
Cluster_param.pow_bid	3	High value penalizes large clusters (0-3, 2)
Cluster_param.pow_dep	4	High value emphasizes high interactions (0-2, 2)
Cluster_param.max_cluster_size	31	Max size of cluster (DSM size)
Cluster_param.rand_accept	30	Proceed w/ 1 of N changes even if no imp. (0.5-2 *DSM)
Cluster_param.rand_bid	30	Take second highest bid 1 out of N times (0.5-2 * DSM)
Cluster_param.times	2	Attempt time*size before check system stability (2)
Cluster_param.stable_limit	2	Loop at least stable limit*times*size (2)

Figure 53: Fourth Result With Thebeau Algorithm

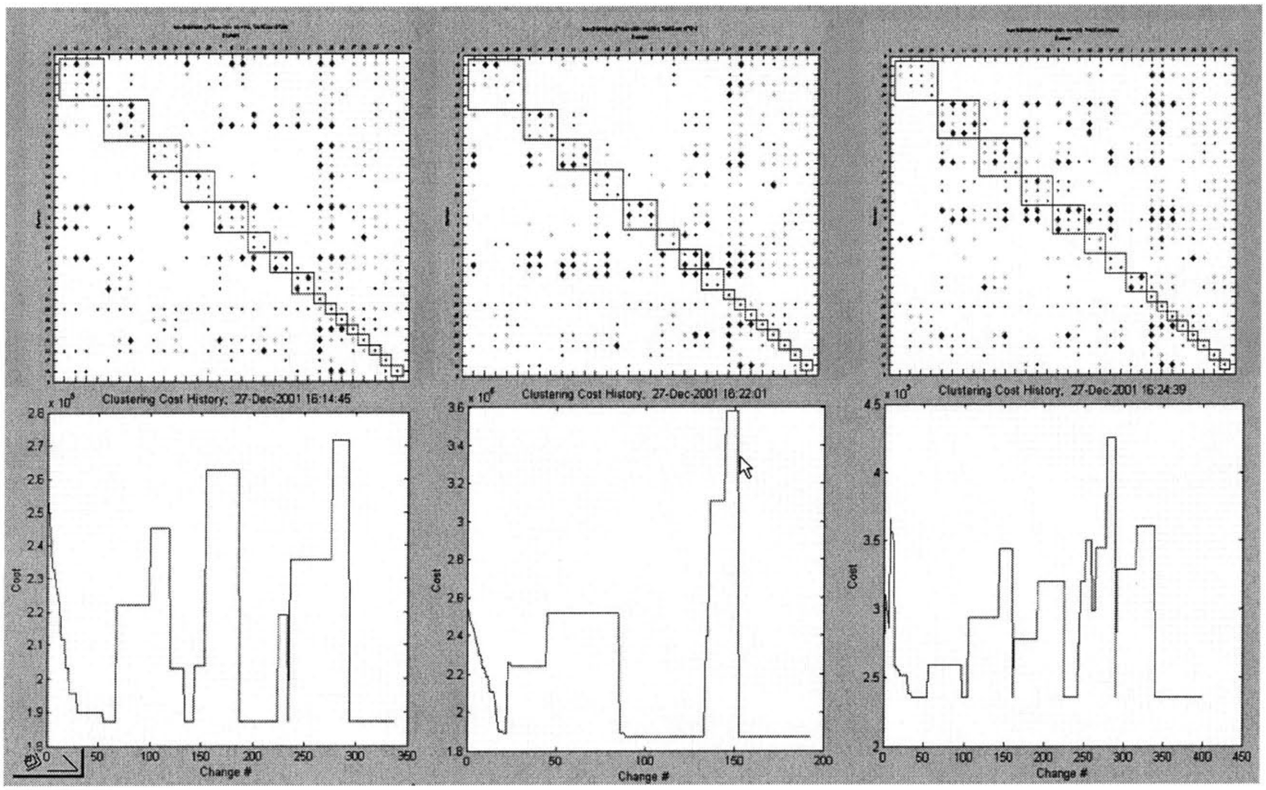


Figure 54: Fourth Result Revisited With Thebeau Algorithm

11.4.7 Comments On The Results Obtained With The Thebeau Algorithm

First a reminder of where we are. At the end of the last chapter the 31 heterogeneous elemental sub-systems of the gas turbine had been shown to possess four or so different characteristic types of option value. In parallel analysis of the gas turbine as if it were comprised of homogenous sub-systems revealed that about four sub-systems were to be expected to maximise value. In the course of this chapter it has so far been shown that an intermediate level of decomposition, semi-manual / semi-automatic techniques suggest that the gas turbine may decompose fairly neatly into about six principal clusters. These six clusters bear some degree of resemblance to the four characteristic types of option value seen, however it has been seen that manual decomposition is problematic and so this may be an undesirable artefact of the method used. Investigation into the underlying characteristics assisted in developing the understanding necessary to achieve the semi-manual decomposition.

In the previous section a fully automated clustering algorithm was presented with the same data set. By tuning the control parameters it was possible to force the algorithm to present about the same number of clusters as with the manual method. However this was not that 'natural' a result as 25% of the elements only ended up in the bus cluster through manual pre-selection, and even the degree of forcing required to prevent the remainder clustering mostly into one mega-cluster did not yield a reliable and repeatable result. Whilst the automated method used is perhaps the best known, and may be satisfactory for identifying extremely 'perfect' modular architectures it is not a 'fire and forget' tool to be confidently used in handling integrated or semi-integrated architectures.

Specific problems with the algorithm are:

1. It does not recognise and handle buses.
2. In practice it does not reveal pinned modules.
3. This means that inter and intra-cluster element sequences are both random. It only allocates elements to clusters.
4. It does not yield consistent results for highly or semi-integrated architectures.

Whilst there are forcing techniques to work around these limitations these are simply manual intervention at one stage of remove. It would be nice to have a better, automated, tool to use.

11.5 The Way Forwards: Why Not Just Tinker With The Algorithms

It is extremely tempting to dive head first into the existing clustering algorithms and tinker with them, creating all sorts of integer and non-integer objective functions and logical tests with the intent of improving the result. Even though I believe this to be a mistake it is useful to explore it to a limited extent.

11.5.1 Recognising Buses

The most obvious place to start is with the bus: why not automate the task of recognising bus elements. Indeed three obvious tests spring to mind – the average strength of relationships to or from an element, the variance in those relationships, and the simple number of non-zero relationships. These three characteristics are shown in Figure 55 and Figure 56 for the manually sequenced DSM elements of the gas turbine as used to seed the Thebeau trials.

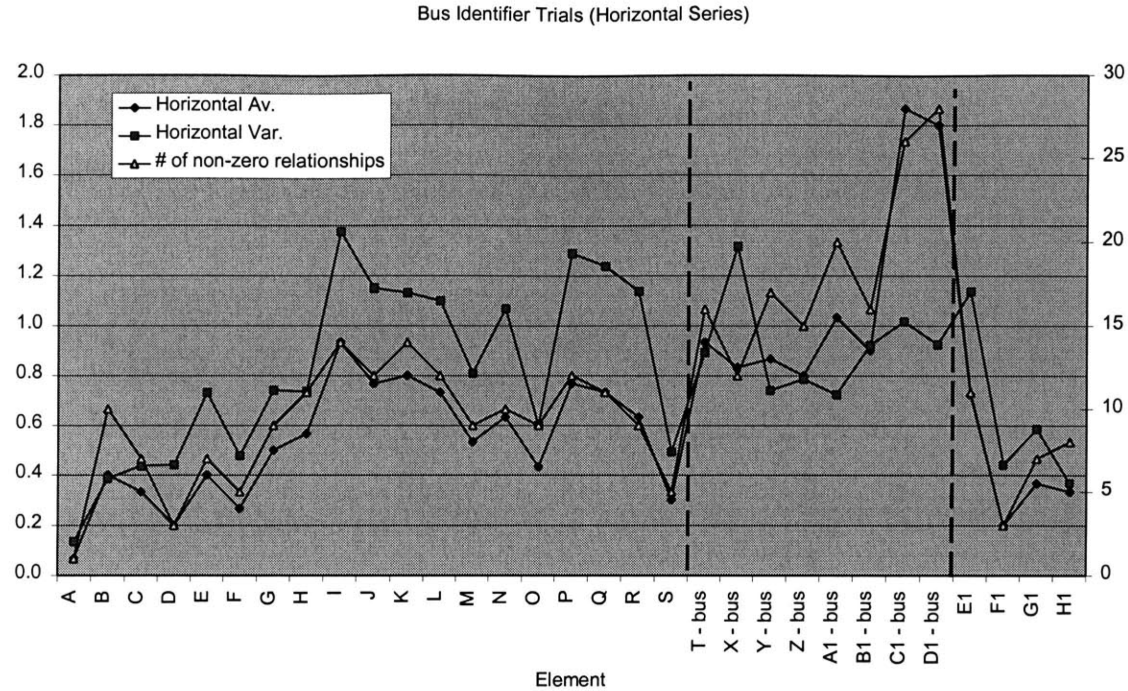


Figure 55: Bus Identifier Trials - Horizontal Series Summation

In the first two figures the horizontal series summation corresponds to the relationships from something (inputs), whilst the vertical series correspond to the relationships to something (outputs). In each the manually proposed bus elements are marked out, and in the horizontal series it would appear that a simple test on the number of non-zero relationships being greater than 50% of the number of elements in the DSM would reveal all but one of the suggested elements of the bus without throwing up any false positives. Looking at these results one might even propose removing element X (which happens to be the HVAC, also known as item 28) from the bus and re-clustering the resultant sequence. However this is an asymmetric DSM so whatever is chosen must also work the other way and such a simple rule only works for 3 of the 8 elements when viewed vertically.

The problem arises partly from the asymmetry and partly from the dominance of a large cluster in the I-M sequence (it is actual the core or gas generator). Unfortunately this large cluster has an associated auxiliary bus that cannot break loose because of the extremely high strength relationships within the gas generator (they are indeed highly coupled units). Because of this auxiliary bus even an adaptive algorithm that uses some form of Pareto test to set an adaptive filter would fail.

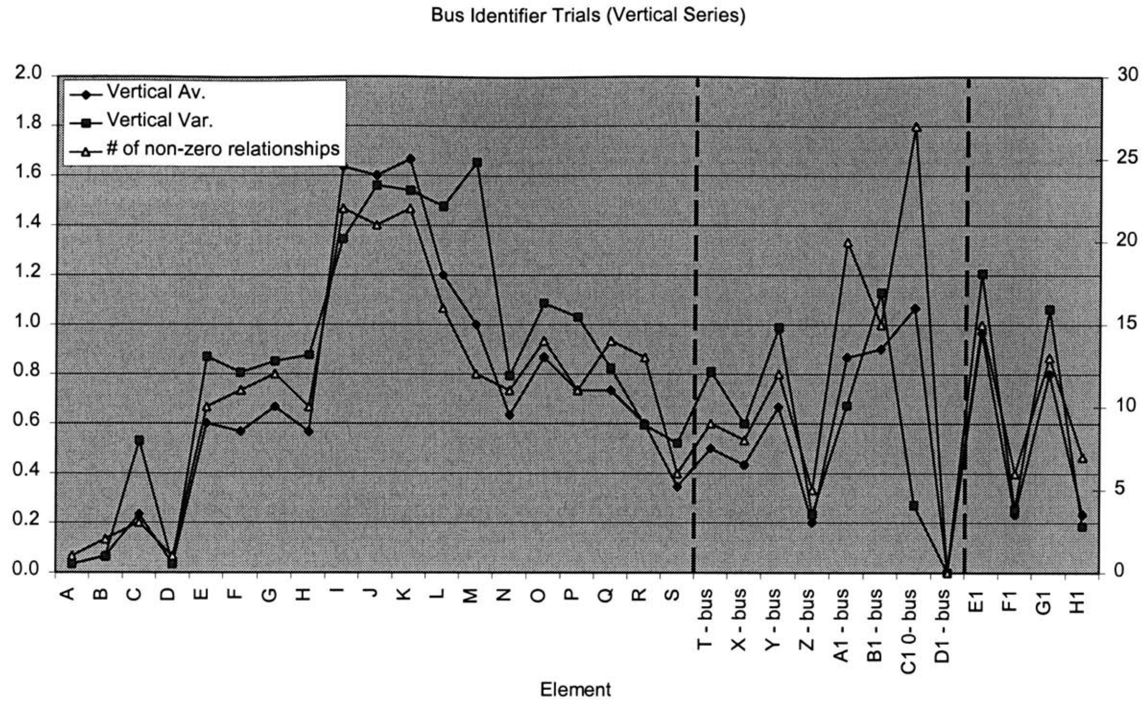


Figure 56: Bus Identifier Trials - Vertical Series Summation

This forces consideration of a combination test. The average strength of relationships is not very useful as this can be skewed by locally intense relationships so instead the variance can be used. A perfect bus element would have zero variance and an imperfect bus a low variance. Whilst this test cannot be used in isolation since this characteristic will be exhibited by non-bus elements it can be tested for in combination. An easy cut-off point to implement is simply the mean variance, and likewise the obvious cut-off for the number of non-zero relationships is the mean number of non-zero relationships.

These can easily be implemented as a truth table as shown below in Table 7: Truth Table Of Possible Bus Tests with a maximum score of four from the four tests (two each for horizontal and vertical). The result is that if a cut-off of say 3 of 4 is used then 50% of the manually selected bus elements would be chosen and none of the others.

In seeking to improve this it appears to be the asymmetry that is causing the problem. By combining the two scores it might be possible to improve the signature and this is shown in Figure 57 below. There are two options for the combined variance – either use a simple sum, or more properly calculate the variance of the series: both are shown and the latter yields the best result.

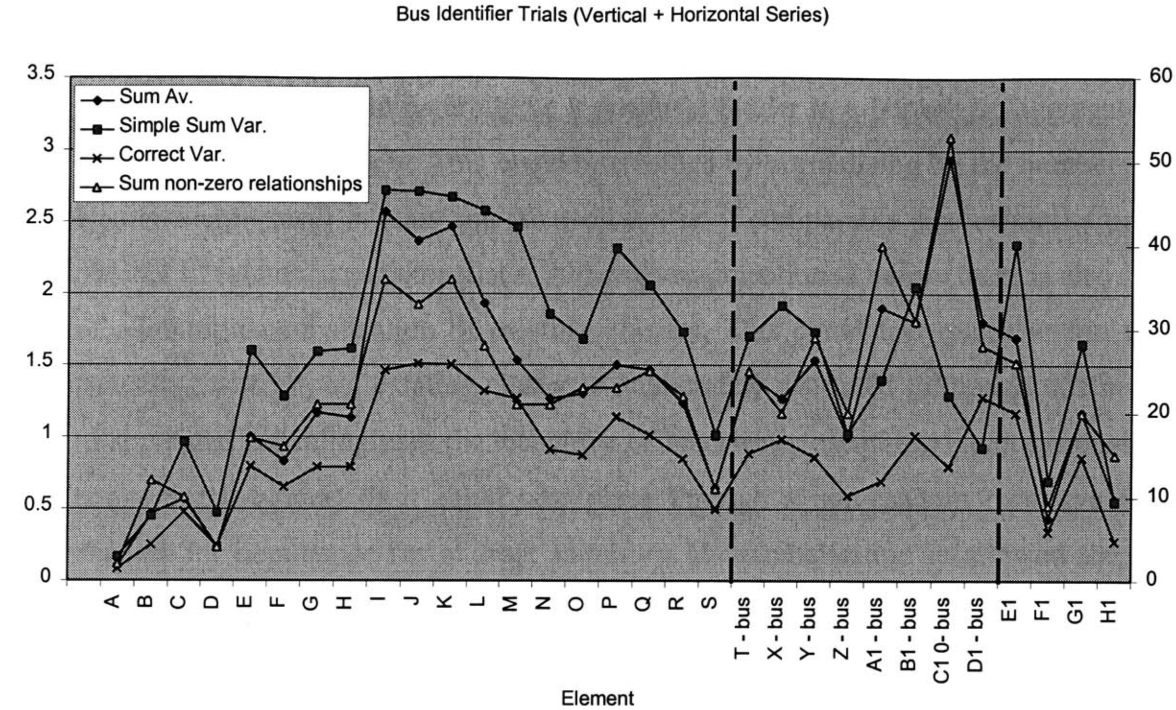


Figure 57: Bus Identifier Trials - Horizontal + Vertical Series Summation

The combined truth table is also displayed. The simple sum of these has a maximum score of six and as before four out of the eight manually selected elements clear a threshold of 4 / 6 with no false positives. Whilst the discrimination of positives has become slightly better the yield rate has not improved.

Table 7: Truth Table Of Possible Bus Tests

Element	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T - bus	X - bus	Y - bus	Z - bus	A1 - bus	B1 - bus	C1 - bus	D1 - bus	E1	F1	G1	H1
Vertical var < mean var.	1	1	1	1														1	1		1		1	1		1	1		1		1
Vertical # of non zero > av.							1		1	1	1	1	1		1		1	1				1		1	1	1		1		1	
Horizontal var < mean var.	1	1	1	1	1	1	1	1					1		1				1			1	1	1					1	1	1
Horizontal # of non zero > av.									1	1	1	1				1				1	1	1	1	1	1	1	1				
Score / 4	2	2	2	2	1	1	2	1	2	2	2	2	2		2	1	1	2	2	1	2	3	3	4	2	3	2	1	2	2	2
Correct sum var < mean var.	1	1	1	1	1	1	1	1											1				1	1		1			1		1
Sum # of non zero > av.									1	1	1	1			1	1	1			1		1		1	1	1	1	1			
Score / 6	3	3	3	3	2	2	3	2	3	3	3	3	2		3	2	2	2	3	2	2	4	4	6	3	5	3	2	3	2	3

11.5.1.1 Practical Implementation Issues For The Bus Identification Scheme

If this bus identification scheme were to be implemented there are some further choices to be made. The most obvious approach is simply to run these tests up-front of the remainder of the Thebeau algorithm and to park the identified bus elements in a segregated non-participatory cluster as at present. A second approach is to allow this cluster to participate in the bidding process, albeit with a pre-formed nucleus. There is a danger that this would create path-dependency, as it would naturally be a powerful bidder in a bottom up aggregation algorithm as it would be overweight. This could be avoided by normalising by the number of pre-seeded elements until (say) at least one other cluster is of comparable size or until a pre-determined number of rounds have taken place. Although not mentioned before there is also the possibility of allowing cannibalisation of existing clusters. This could take place within the simulated annealing whereby an existing cluster is deliberately disrupted (although this implementation of simulated annealing uses the technique of incorrectly assigning the occasional element to a non-winning bidder) or it could take place through a non-random recursive fight between clusters for dominance (or at least stability). If cannibalisation is allowed then the issue is whether the bus's core should have a privileged protection. Overall I suggest that the cluster should participate in bidding, that cannibalisation be permitted, but that the bus nucleus should be protected, and that when necessary the bus nucleus's size advantage be negated by normalisation.

So, superficially, it appears that improvements can be made at least with this example. The issue is whether these improvements can only be made in retrospect, and whether they are indeed meaningful improvements or merely reveal ignorance of the underlying structure.

11.5.2 The Problem With Tinkering

Tinkering in this fashion is a lazy substitute for thinking. At best all one can ever hope to do is to reproduce ones existing value system in the algorithm in use, and at worst one reproduces an imperfect representation of the value system in an overly complex manner that defies understanding. If tinkering is felt to be desirable then the best approach is to train neural nets: at least the tinkering process will be transparent and reproducible. The problem with simply toying with truth tests; substituting a linear relationship for a quadratic; or any other similar approach is that in the absence of investigation of the underlying rationales one is bounded by misunderstood and poorly developed historical heuristics that may not be applicable in future contexts.

However if we use tinkering as a way of exploring our understanding, codifying the value system, determining the underlying principles, and extending them then some purpose is served. This has been the purpose in the work done so far and it appears better to continue in this vein than to merely kludge together a heuristically derived algorithm that merely serves in this instance.

Having said this the question is where to start again. I choose to start with overlapping modules, which I term pinning, which leads to a discussion of one form of dimensionality.

11.6 Dimensionality, Pinning, Boundary Objects

The purpose of a DSM is to enable observation and measurement of structure that is not easily obvious or easily measurable. A DSM can do this admirably – as was said for example when describing a DSM about the Saab JAS 939 Gripen combat aircraft "until this DSM was created[34] no-one knew how the aircraft was organised" (Danilovic & Borjesson, 2001). However because a DSM does this so well it is easy to be seduced by its results. Therefore it is also necessary to be cautious in interpreting these results because structures of a certain complexity cannot be easily described in a conventional DSM.

The problems with applying automated clustering algorithms to non-ideal situations is that they find it very hard to extract the relevant information from the data, and then to convey it to the user. This is most obvious in their poor handling of pinned modules, of buses, and of path-dependent situations. At the heart of all three of these is the issue of what really is a cluster (or module) and what is a relationship between elements. In order to investigate this further we shall return to cartoons.

In the following sequence of cartoons clusters of many high strength relationships between many primitive elements are described by the letters BB, AA, etc. These clusters are related by fewer or weaker inter-cluster relationships than the intra-cluster relationships and this is represented by the small overlaps (pins) that occur and represent interfaces. The density of colouration also represents interface strength and so, for example, the relationship of CC with AA is a low-density region as a result of the way the chosen ordering of the elements in the DSM spreads the relationship over a wider area of the DSM.

[34] It was actually a transform matrix that mapped the Gripen provider organisation onto the primary customer's organisation not a true DSM, but the sentiment of the power of the visual representation is common.

11.6.1 Dimensions & Topology

The word dimension is used in two very different ways in this thesis. Sometimes it is used to describe aspects of information in a domain, e.g. in the physical domain there are dimensions of mass, energy, momentum, etc. It is also used in the more traditional sense to describe geometries, e.g. one dimensional lines, two dimensional planes, three dimensional objects, four dimensional things, etc. In this section the latter, more traditional, meaning of dimension is in use. Wherever it is unclear in this thesis I apologise.

11.6.1.1 Two Dimensional Structures

In the planar triangular cluster AA, BB, CC that we saw before there is a simple and clear structure. The equivalent DSM is able to map the AA-BB clusters in a way that is easily interpretable. As previously mentioned the manner in which this is clustered (interpreted) in a DSM will depend on the extent to which the clustering algorithm is path dependent. Thus it could just as easily have been AA or BB clusters that were broken up by being positioned where CC is in this figure as shown below.

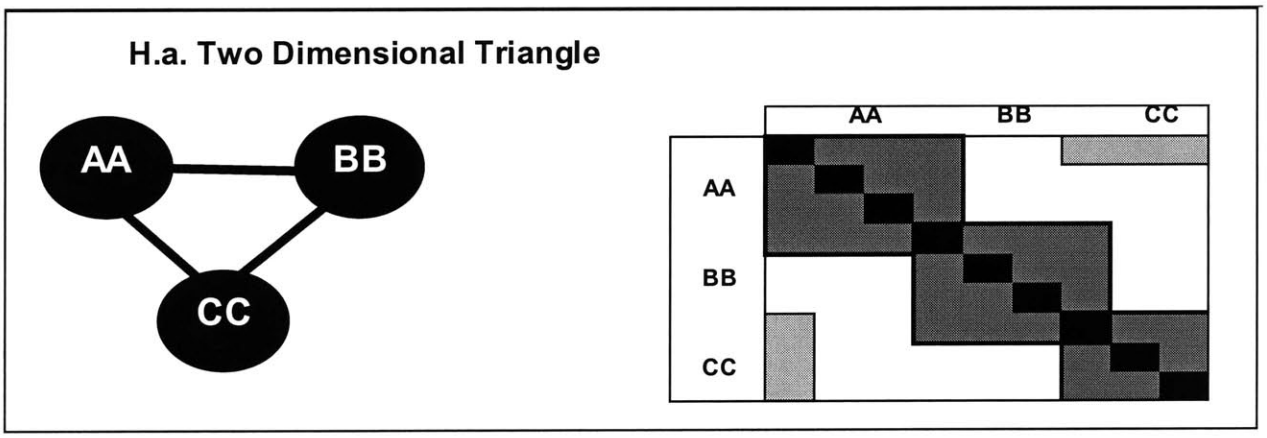


Figure 58: Triangular System

This is relevant because normally the DSM is what is used to interpret the underlying structure. The correct question to be asking is "***If one is only aware of the information within the DSM, can one see what the real structure of the thing is ?***" as otherwise one may make erroneous simplifications or introduce false complexity. So in this example the real structure might falsely be interpreted as being two heavy modules and a light module, and it could simply be random chance that determines which is perceived as being the lightweight module. In this example it seems reasonable that the clear simple structure could be seen by inspection[35], so let us turn to more interesting examples.

The example below is a planar quadrilateral ring structure with four modular clusters -AA-BB-CC-DD- and the corresponding DSM in which the grey shading denotes the relative link density. Once again without knowledge of the real structure could the simplicity have been seen?

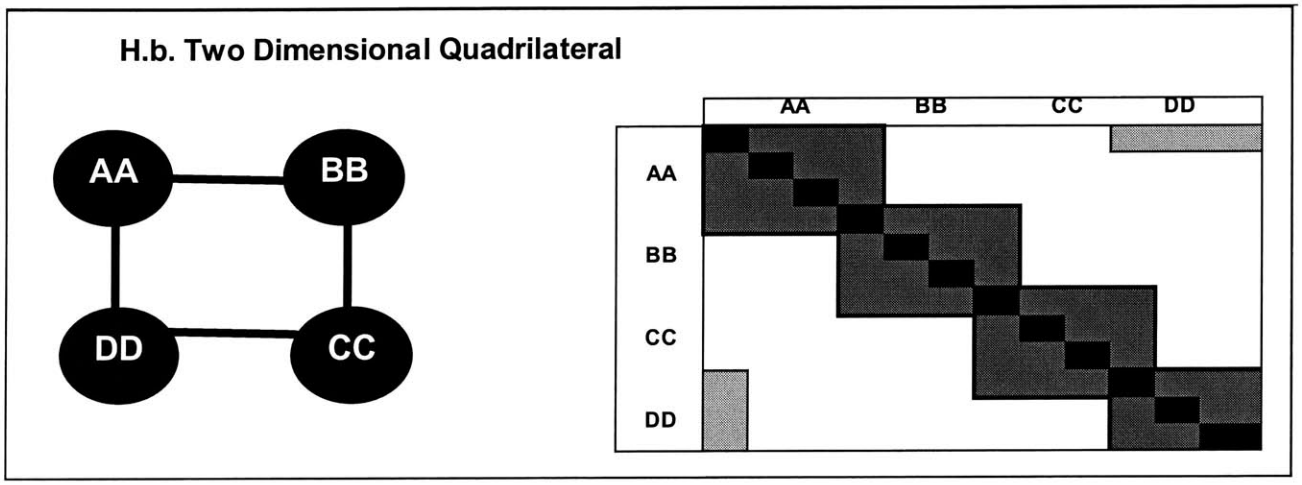


Figure 59: Quadrilateral Cluster

Again it seems reasonable that it could have been puzzled out by inspection. After all AA-BB-CC are clearly a pinned chain of modules and so the only difficulty is in seeing the best way to handle DD. A similar point can be made about any ring structure where each module cluster has only two relationships.

11.6.1.2 Three Dimensional Structures

Things become less clear when structures are most easily observed in more than two dimensions. This does not mean that they cannot be reduced to two dimensions, merely that the two dimensional interpretation of them appears more complex than the underlying reality.

The simplest three-dimensional structure is a tetrahedron or three-dimensional pyramid, depicted in the figure below showing four equal clusters, each with dense internal relationships and weaker (or sparser) external relationships.

[35] Perhaps it is not so reasonable to assume that this structure will be observed. Think about how frequently even simple organisations in a stable environment are reorganised.

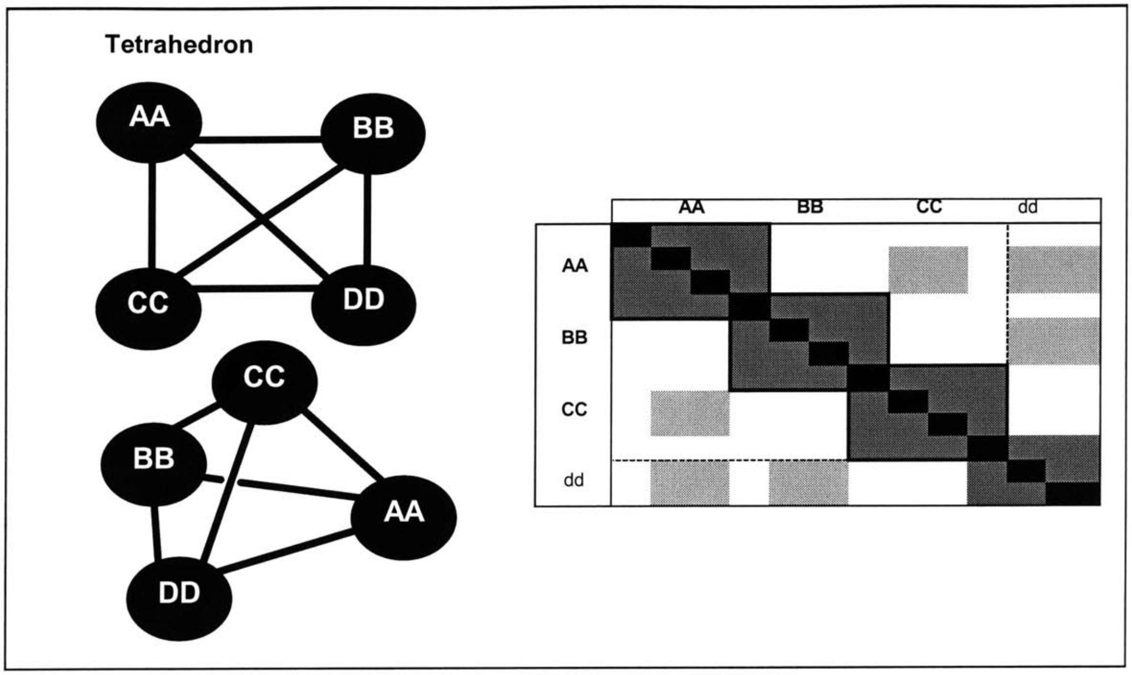


Figure 60: Tetrahedron Of Clusters

As before if all the clusters are perfectly equal it is purely a matter of chance how any clustering algorithm would present an answer. In the example shown the cluster DD is the one that is visually disrupted most by being presented last in the sequence. This has the effect of spreading its inter-cluster relationships over a wider spatial area, which is depicted in the macro-scale DSM as being lower density blocks of grey[36]. To an untrained observer this might be thought to be a bus structure where cluster DD is the unique possessor of system wide integrating functions and some semi-random cross-linking occurs in the zone AA-CC. Indeed it is only with knowledge of exactly what the roles of the elements are in each cluster that a human would have been able to intuit that DD should be the bus as opposed to any other cluster. Often this intuition will have been right, but if there is any dynamicism in the architecture one can easily see how historical value judgements as to what is more or less important can be prone to rapid disruption as a new entrant (with a new perspective) realises the dislocating impact of any creeping changes in underlying structure. In these situations the

[36] The exact locations and densities of the lesser blocks of grey depend on the precise structure of the architecture. Since these drawings are relatively macro scale depictions the precision of the placement is somewhat illusory and other selections would have been acceptable. Micro scale depictions are shown later to investigate the nuances, however the basic point regarding the ease of misunderstanding an architecture remains valid.

value judgements inherent in assignation of a particular grouping to the 'bus' become a historical liability[37].

Given the importance of reducing complexity and minimising artificial boundaries in an architecture (irrespective of domain) it is somewhat surprising that so few architectural diagrams (e.g. hierarchical product decompositions or organisation diagrams) use anything other than planar representations. This tetrahedron is a perfectly simple elegant little structure when perceived in three dimensions, yet looks terribly complicated when seen in only two.

It does not take much for matters to become even more complicated as the following example shows with the addition of a mere fifth cluster. Even though the underlying structure remains simple the DSM is much harder to interpret.

[37] One has only to consider the automobile industry and the role played by, first the engine (initially a dominant element), then the more accurate understanding of the importance of 'powertrain' (a dominant cluster) and now the realisation that other 'systems' with more emergent properties (e.g. NVH, cockpit, media) are important. As yet these other systems are being relegated to a supporting bus-like role by historical industry and company politics, yet the customer purchase decision is being increasingly dominated by them. The logic is that they should be moved to centre-stage as a visually dominant cluster (i.e. an expression of the internal recognition of their importance in company politics and industry strategy) yet on all organisation charts (and industry DSMs) in circulation they remain in a bus-like supporting role. And if proximity is important then note that the DSM's torosity places the upper leftmost dominant module adjacent to the lower rightmost integrating bus.

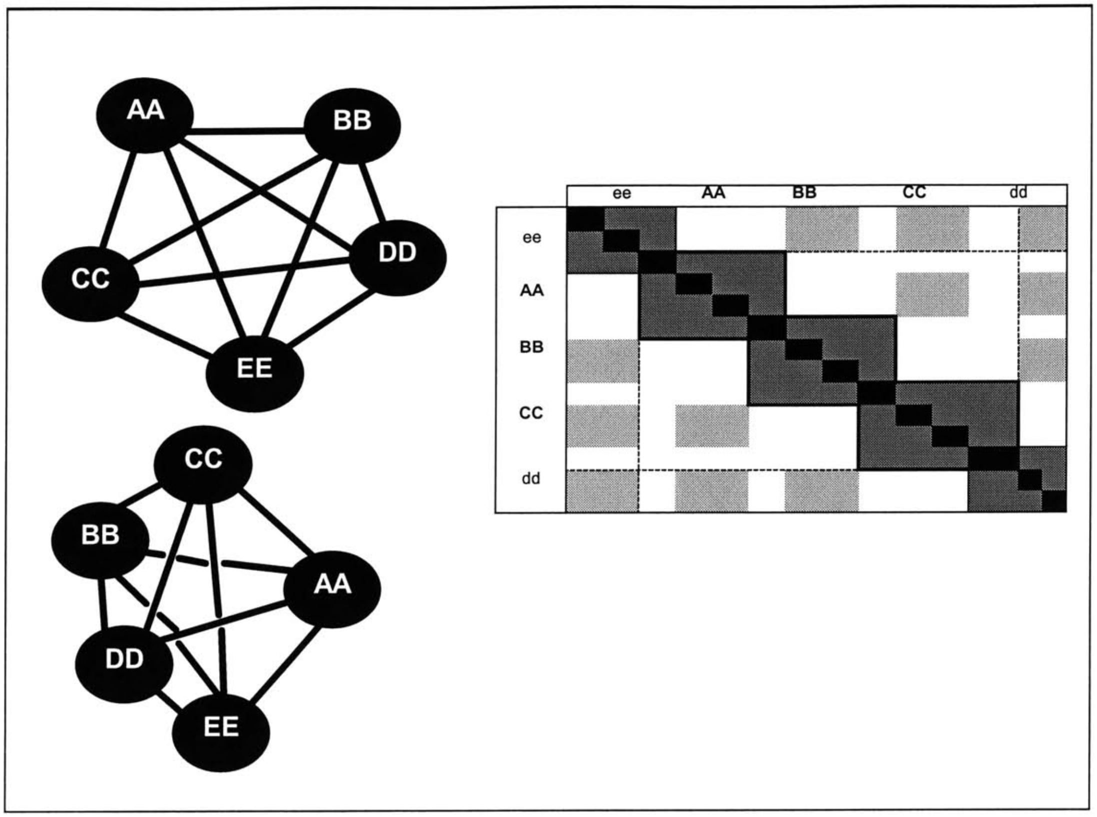


Figure 61: Five Clusters In A Hectagon

As before it is extremely easy to wish a structure into a DSM that is not there. In this instance many observers might remark that there are two sets of system-wide integrating elements, emanating from opposing corners. Such a structure is known to be extremely important in task domain DSMs as the upper left corner represents the set of architectural decisions (or strategic choices) whilst the lower right corner typically represents the test & integrate tasks, and this feature may have equivalents in other domains.

11.6.1.3 Open & Closed Topologies

Irrespective of what structure an observer might artificially impose on this five cluster system, if they only had the DSM results to go by, it is worth noting that if the number of clusters is the relevant metric then this barely makes it onto the bottom end of the standard scale of complexity. After all it has only five clusters, which is at the lower end of the rule of 7 +/- 2. Equally it cannot be the number of inter-cluster relationships per cluster as there are only four of these. Indeed the only metric that exceeds 9 is the total number of inter-cluster relationships, which is 10.

In large part the apparent complexity arises from the closed topology of the system[38]. If the topology had been open it would have been trivial to build a nested hierarchy in the DSM. Yet we have seen an example of a closed topology that was easy to represent in a DSM and scarcely less easy to interpret – namely the flat ring structures. The issue only arises when a closed topology exists that is irreducible below three dimensions, as it cannot then be adequately conveyed in the fundamentally one-dimensional DSM (which after all is a linear construct).

Mathematicians have spent centuries pondering over geometry and dimensions; in this thesis I shall not go any further than these comments. My intuition is that multi-dimensional DSMs could be constructed so as to capture and solve multi-dimensional structures, and that one needs a DSM with at most one less dimension than the structure being observed. Quite how this should be done I do not know. As always the real issue is that models are useful only so long as they are less complex than the real thing and as soon as they become as complex as the real thing then one may as well discard the model. Therefore current DSMs ought to be able to 'solve' (i.e. represent) open topologies and closed two-dimensional topologies. The upper limit is probably that a DSM with a non-distorting algorithm and a knowledgeable interpreter should be able to recognise and sketch closed three-dimensional topologies.

11.6.1.4 Application To The Gas Turbine Architecture

It is instructive to see if these DSM patterns of simple multidimensional architectures assist in interpreting the DSM of the gas turbine. On the left below is the DSM of the gas turbine in the same sequence as the 'best' manual interpretation seen to date. To aid comparison the 'electrical' cluster has been rolled around the torus to the upper left. This rolling does not affect the sequence. To the right of this DSM is exactly the same information plotted as a contour map where the colours indicate a greater density of relationships on a linear strength-weighted basis.

[38] I am grateful to Ed Crawley and Dan Whitney who each independently gave me the correct terminology for this point.

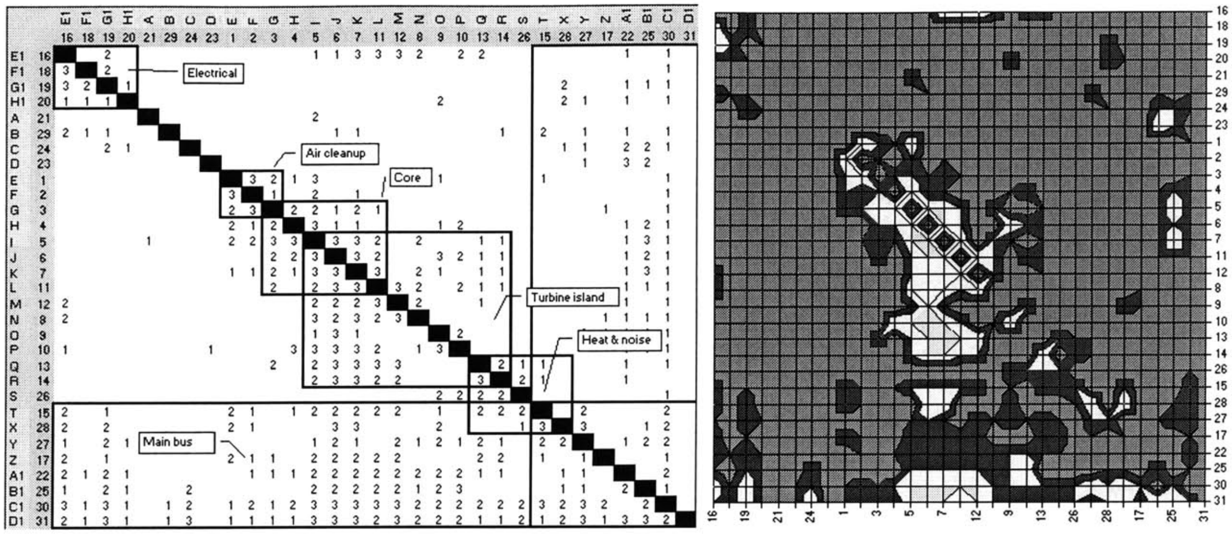


Figure 62: Gas Turbine DSM and Equivalent Contour Map

By simply using the same cluster boundaries (and names: a dangerous practice) it is possible to draw the following diagram in which cluster size is approximately represented by the bubble diameter, and the inter-cluster relationship strengths are approximately indicated by line thickness. Although this is a very naïve representation it can help clarify thinking. For example on the basis of this diagram it would appear that the gas turbine is a closed two-dimensional topology, i.e. it should be just about manageable to analyse. It also suggests that the bus elements might be more significant in the architecture than had been hitherto supposed. This may explain why the discrimination of the 'bus' as a bus was so difficult even with the benefit of hindsight: perhaps the 'bus' is barely a bus after all.

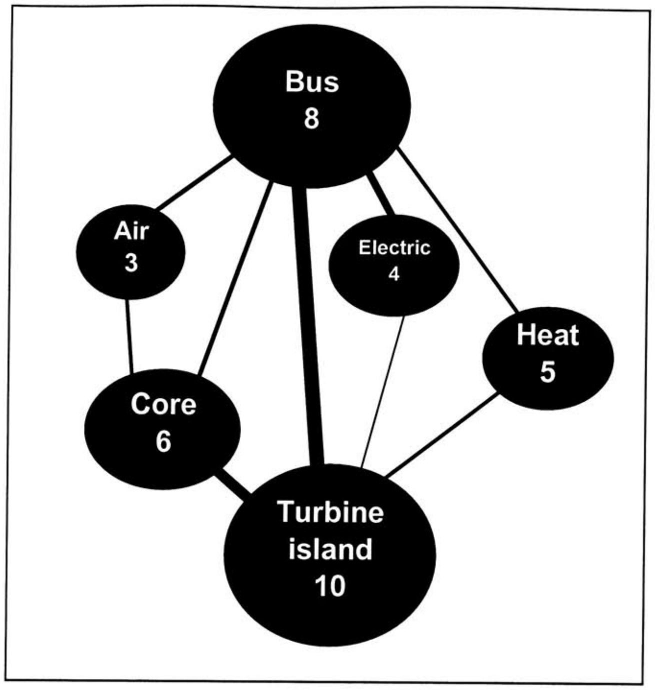


Figure 63: Naïve Gas Turbine System Diagram

Even if the diagram is created on the basis of a good sequence what is not clear in this diagram is the meaning of a cluster or a cluster boundary. For example two-thirds of the core overlaps with (is pinned against) the turbine island so what possible meaning can it have to divide them out ? The answer to this lies in the definition of a cluster boundary. Similarly what denotes the size of a cluster and are some clusters better formed than others ? Once again the solution must lie in the definition of a cluster boundary, as this is what discriminates between the inside and outside.

11.6.2 Boundaries, Relationships, Internal & External Rules

The simplest possible system of clusters is a one-dimensional arrangement of two clusters. An example of this is shown below and is formed from eleven elements 'a' through 'k' and their associated relationships. Two six-element clusters share a common element 'f' whilst within each cluster all elements are fully connected with bi-directional relationships. This therefore is a fully symmetric arrangement.

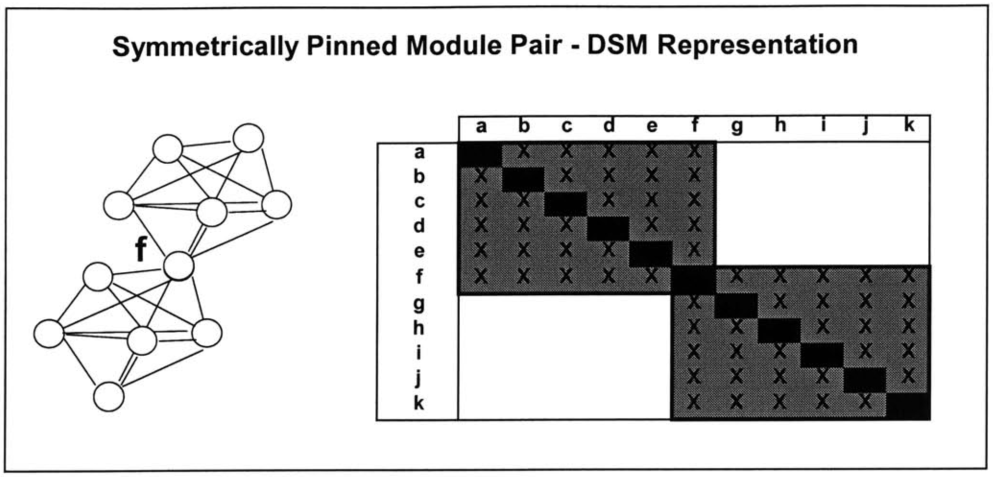


Figure 64: Symmetric Module Pair – DSM Representation

The population of the DSM with tick marks is trivial and the drawing of boundaries seemingly equally so. As they have been drawn it suggests that the two equal perfect clusters are pinned by the element 'f'. However there are other ways of visually depicting this pair of clusters as shown below:

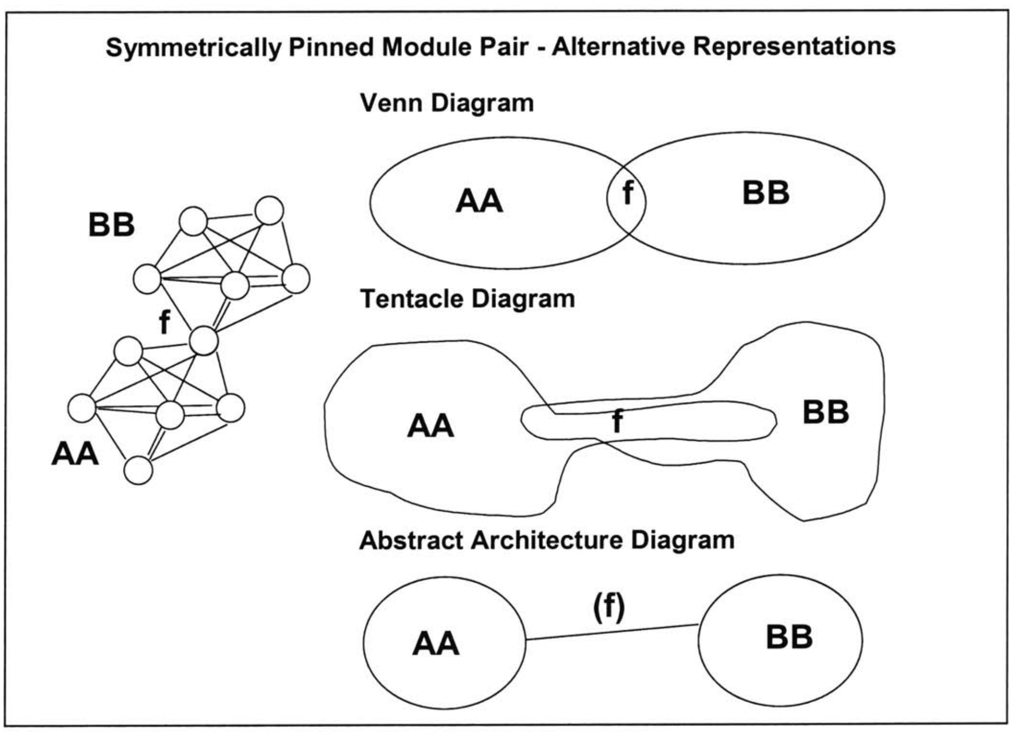


Figure 65: Symmetrically Pinned Module Pair - Alternative Representations

The Venn diagram perspective makes it clear that element 'f' belongs to both set AA and set BB. The tentacle diagram and the more abstract architecture diagram versions are interesting inasmuch as they depict element 'f' as being a bridge between AA and BB – the tentacle diagram is a way of suggesting that 'f' is both a bridge and member of each set whilst the abstract diagram suggests that 'f' is only a bridge and not a set member. This same issue is also finessed in the decomposition below.

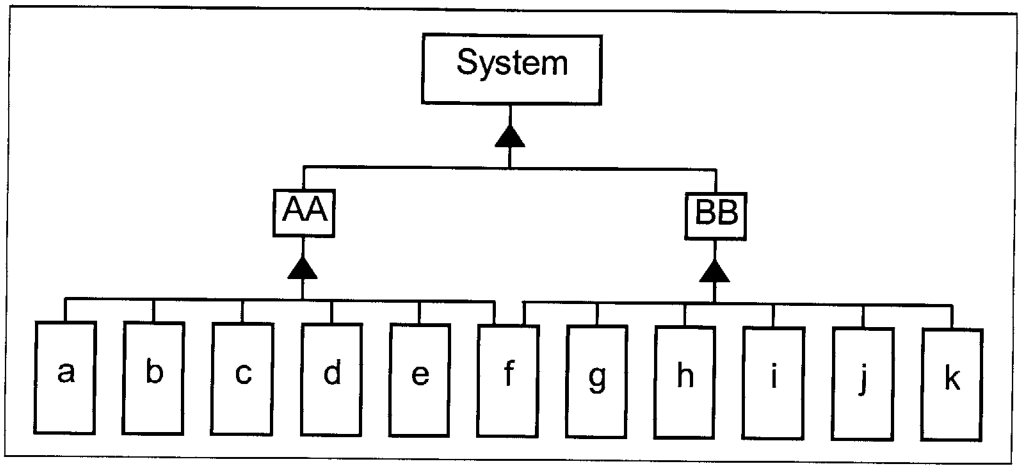


Figure 66: Symmetrically Pinned Module Pair – Two Set Decomposition

Since 'f' is clearly a member of a set an alternative decomposition is to single it out as being a member of a unique set, as shown in the next decompositional illustration below. This alternative is ambiguous about whether 'f' should be thought of as being at the same decompositional level as the other elements.

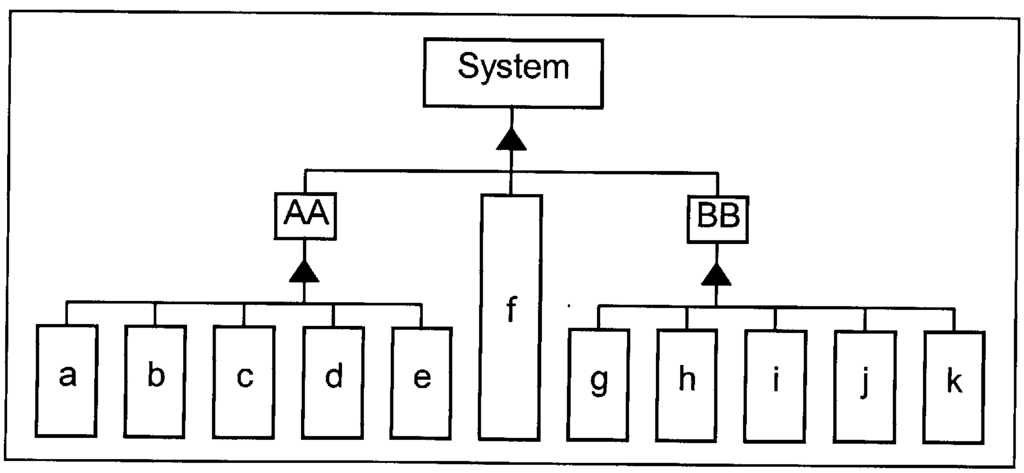


Figure 67: Symmetrically Pinned Module Pair - Alternative Two Set Decomposition

It could equally well have shown 'f' as being the only element into which a set CC was decomposed.

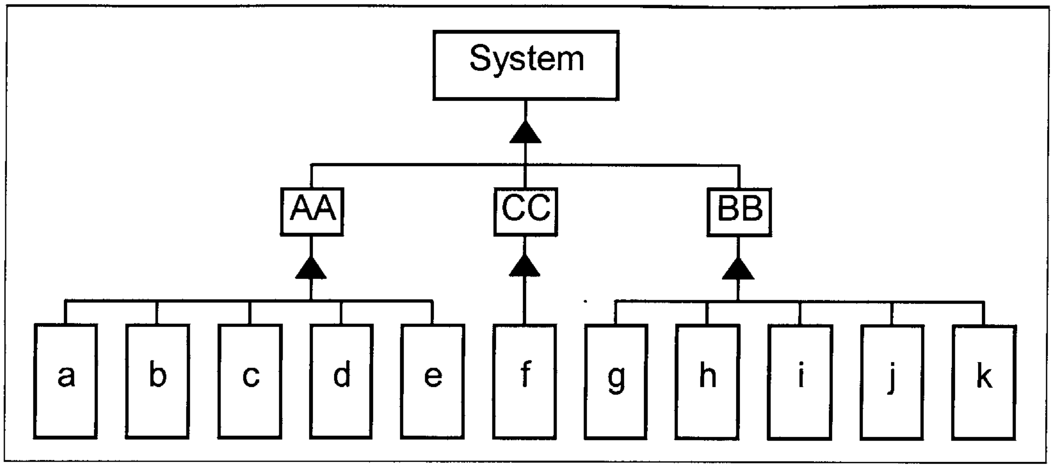


Figure 68: Symmetrically Pinned Module Pair - Three Set Decomposition

This suggests three degrees of freedom that characterise these alternative depictions:

- 'f' is or is not a member of set AA.
- 'f' is or is not a member of set BB.
- 'f' is or is not a member of set CC.

These three degrees of freedom yield eight different ways of drawing the DSM boundaries, so maybe the choice is neither trivial nor obvious.

By removing relationships it is possible to strip away complexity and, at the point where the problem disappears perhaps the truth becomes obvious. A first step is to make some of the relationships asymmetrical as shown in the figure below where the five relationships that 'f' possesses with one cluster become one directional. This is shown in both the DSM form and various other manners.

As before it is possible to characterise the boundary locations in terms of 'f'. Given that boundaries in a DSM must delineate a square (this after all is what makes a flat DSM function; in a three-dimensional DSM this would be a cube) there remain eight possible combinations for drawing the boundaries.

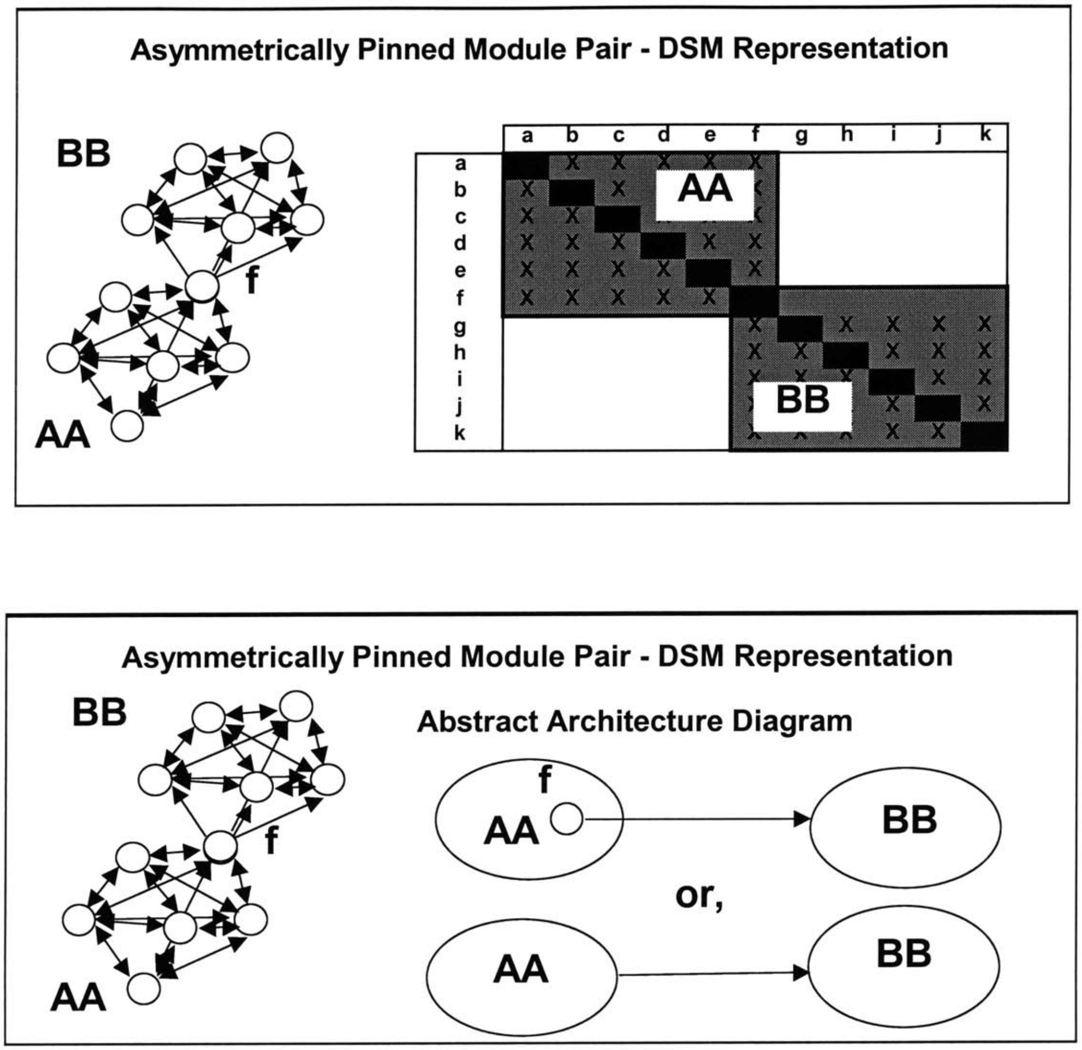


Figure 69: Asymmetrical Cluster Pair - DSM & Alternative Representations

Of course the four depictions in which set CC exists would seldom be observed in the DSM. This is because in most depictions of a DSM the diagonal row is marked out in black and so delineating an element as being a set in itself would not be noticed and is not often suggested. Similarly the issue of what value to place in the diagonal is seldom considered – an element must relate to itself (Or must it ? Can an element be a pure relationship ?) and so in the standard binary DSM the default value for the diagonal is '1'. In a weighted DSM it is possible to conceive of other options for the strength of an element's relationship with itself as not all elements need be as internally coherent especially if the DSM is not being drawn at the ultimate level of decomposition.

At the risk of being overly naïve it appears important to remark that in the DSM shown there is empty space within the cluster boundary. In truth the only discriminating factor between these eight set combinations is the amount of empty space within the boundary and the amount of unbounded relationships. What does the boundary mean ? What is the difference between being inside a boundary as opposed to outside ?

The boundary marks off a sequence as being special. Because a DSM is square any sequence delineation mark must be square, but the mark itself is merely a sequence identifier. The mark contains information – it is a shorthand definition of the internal contents. Indeed everything within the set can be collapsed to that mark provided the mark is properly coded. This is why the sequence does matter within a set as it allows the information content of the set to be compressed more or less efficiently (except in the special cases where the set content is a predictable pattern that simply needs scaling, of which the two most extreme forms are the null set and the full set [0, 0, 0 … 0] and [1, 1, 1 … 1] respectively). Likewise the information content outside the boundary can be collapsed to the boundary mark. One can think of the boundary as defining the set's internal and external design rules. In a DSM the zones of influence of the internal and external design rules are predictably located as the following diagram outlines for a single cluster.

The greater the codification of the design rules of a cluster the stronger becomes the influence of a boundary and the less meaningful become the contents of the cluster itself as all the essential inter-element relationships are contained within the design rules. There are two sorts of design rules in play – natural ones and human imposed ones. It is impossible to change the natural design rules and that is why so few diagrams show them, however in an utterly decomposed system with no superfluous human-imposed design these are in fact the micro-scale design rules of the individual elements. At a macro level they tend to be dwarfed by the more capricious human-imposed design rules. Of course not all design rules are equally strong no matter how much they are codified[39]. At some level of abstraction it is possible to have macro level 'pure' rules without physical embodiment in the system under consideration and this leads to the logical conclusion that boundaries are therefore objects in themselves.

[39] This was famously demonstrated by the medieval King Canute of England who, in order to teach a lesson to his court, is said to have placed his throne on a beach and commanded the tide not to swamp him. His highly codified rule was no match for the strength of the natural rule of the moon's gravity acting on the sea. This is an example of collision between natural and human rules, but human on human rules also collide as is discussed later.

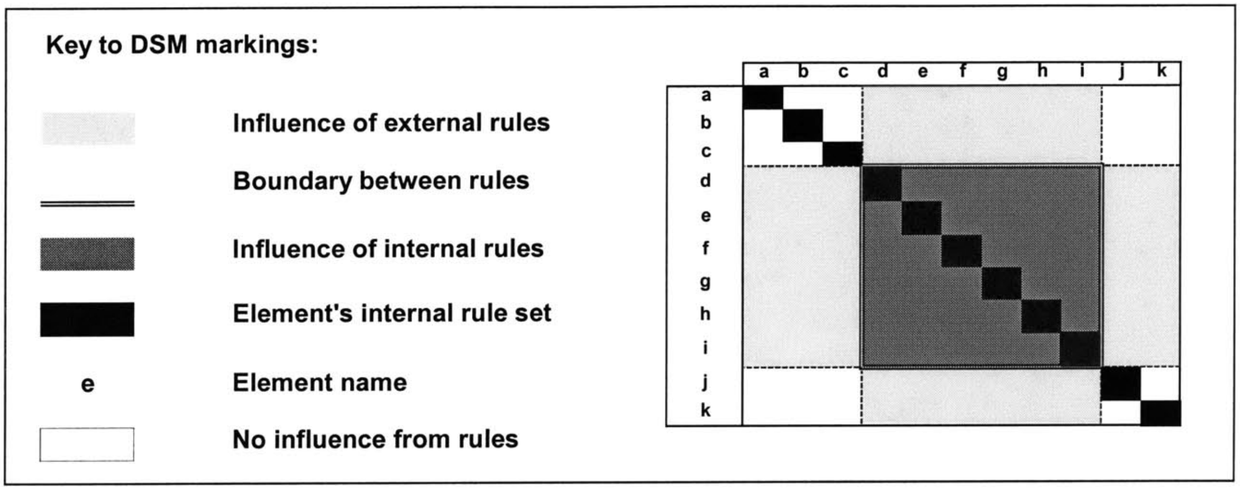


Figure 70: Internal & External Design Rules

If, in the limit, boundaries are objects in themselves it is important to understand whether they are objects in less extreme circumstances, as this will affect how architectures can be constructed. Let us consider only the internal impact of rules for a moment. The computer I am using has a 802.11b wireless LAN PCMIA card. This card can be thought of as an element that is subject to two sets of human rules. On the one hand it must obey the rules for a PCMIA card (the physical form factor, the bus communication protocols, etc.) and on the other hand it must obey the rules for 802.11b wireless LANs (signal strength, frequency, network communications protocol, etc.). In a very real sense the card in this computer is pinned between these two sets of highly codified design rules. The computer's designers recognised this and dealt with it by making wireless LAN connection via the card module as they were unable to predict the market take-up of wireless LAN's in general and individual protocols in particular. Thus the card is a highly constrained module that is nearly perfectly described by two overlapping rule sets. In the decomposition of my computer and the associated network the card would neatly decompose out, wrapped by these two sets of rules which have little impact on the remainder of the system's design (i.e. the 802.11b rules affect relatively little else in my computer whilst the PCMIA rules affect nothing at all in the network end of the wireless LAN). It would appear as if the rules are so closely wrapped around the module as to have a very low visibility to the rest of the system. What happens if the designers now observe the rapid market take-up of wireless LANs and choose to modify the design ? The obvious thing to do is to integrate the card into the computer thereby creating a better antennae location (better signal strength, lower power consumption, frees up a card slot, gets rid of the damage-prone card antennae) and allowing consolidation of silicon (better signal processing, lower manufacturing costs, reduced interconnects) which allows them to predetermine the market share for wireless LANs that they will capture. This has the effect of erasing the DSM object marked 'PCMIA card design rules' and thereby allowing the balance of the higher level PC internal design rules to become more visible to the card's components which in turn has the effect of forcing redistribution so as to maximise the new value[40]. So the boundary has two visibility effects: it protects the internal design rules from affecting the external world, and shields the internal world from other external design rules. The converse is also true: boundaries can both project their own external design rules and attenuate the reception of other external design rules.

Just because boundaries can have these effects does not mean they have to. Returning to the gas turbine the small hydraulic system can demonstrate the issues. Simplifying enormously by ignoring redundancy and segregation, etc. this can be reduced to a four-element system: a **pump** forces **oil** to the **bearings** (for cooling, lubrication, and debris removal) and to the **hydraulic** control system (which uses it as a power fluid in various actuators). An extremely crude DSM of this is shown below with no boundaries whatsoever. This DSM represents the minimal effect that each element need have on the other, i.e. the extent of the natural design rules.

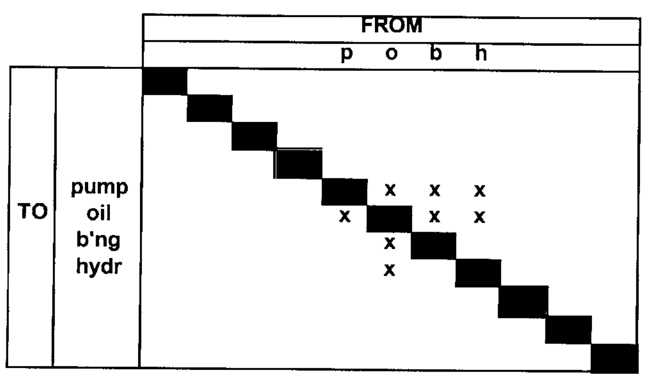


Figure 71: Extent Of Natural Design Rules For Hydraulics

40 This change also affects the market take-up trajectory of successor wireless protocols 801.11 'g' and 'a' as users cannot update an integrated design as fast as a modular one. From a societal value perspective it is not clear which approach is better, but the market participants strenuously seek to affect these different design & standards choices for individual gain.

In this natural state the turbine operator user can happily use whatever oil meets the essential specifications of the system (viscosity, additives, etc.) and would normally base their purchase decision on price and availability. Likewise the user can replace worn pumps with equivalent pumps from any manufacturer. However turbine manufacturers seek to leverage their situation to their own advantage and will re-brand commodity oil for sale at a higher price. In order to force this on a customer they will erect a boundary around 'their' hydraulic system and say it must only be used with 'their' oil. Likewise they will seek to force pump and bearing replacements with 'their' spares at similarly high prices. There are various strategies for erecting and maintaining these or more necessary boundaries.

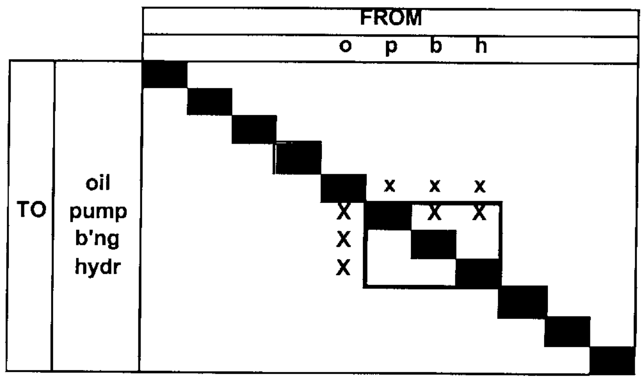


Figure 72: An Example Of A Human-Imposed Boundary

This is an example of a human imposed design rule on the oil, emanating from erection of a controlling boundary around the pump-bearing-hydraulics control system. The boundary has an amplifying effect on rule transmissions to both inside and outside and represents the manufacturer's 'rule object'.

If this is indeed true then there are two ways of handling boundaries in clustering algorithms for man-made systems. On the one hand they could be defined as directional attenuator-amplifiers and on the other hand one might insert new elements with their own relationships. Either approach should work just as well as both will have the effect of distorting the smooth value landscape of the DSM. The attenuator-amplifier approach may be more difficult to execute but is easier to visualise. At present (setting aside the inter and intra-cluster sequence independence which current algorithms ignore) clustering algorithms essentially drive high value relationships towards the diagonal as if under some sort of force field. As shown in this chapter the force field should be the total attraction of relationships for each other, and in this way both the local and global sequences matter. The diagonal becomes the point of attraction not because of some artificially imposed external force field as depicted in the 3-D diagram below, but rather because of the attractive power of the strong relationships that the elements possess with themselves on the diagonal and which can never be broken up by re-sequencing.

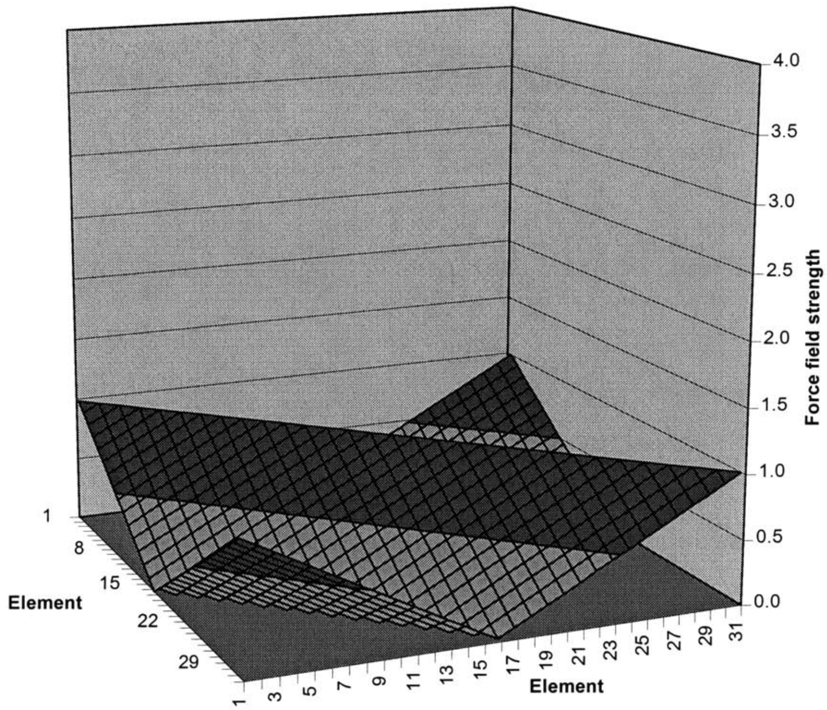


Figure 73: Illustration Of An Externally Imposed Force Field

Even though it is wrong, this notional force field does illustrate a problem that clustering algorithms need to overcome. This particular force field declines linearly with distance from the diagonal, but correctly it notes that the DSM is a torus. Thus if any given non-zero relationship were to be cast into the DSM the buoyancy forces might carry it to an extreme corner rather than the diagonal stripe – and would not be incorrect in doing so. If two non-zero relationships were cast in then if they rose to the strip rather than the extreme corner, once on the stripe they would attract each other along the stripe and finally cluster at a random point. Whilst this particular problem does not occur absent an artificially imposed force-field, it can occur if the projected field of a boundary object has a similar shape, or if the clustering algorithms are (correctly) responsive to the torus shape. To illustrate this latter problem consider a clustering algorithm that sums the forces on the off-diagonal relationship spike in the next diagram. Because the resultant force is zero (after summing right around the torus) the spike will never migrate to the diagonal, and likewise once on the diagonal it has a symmetric force that prevents it migrating to the locally strong on-diagonal intra-element relationships. See next diagram.

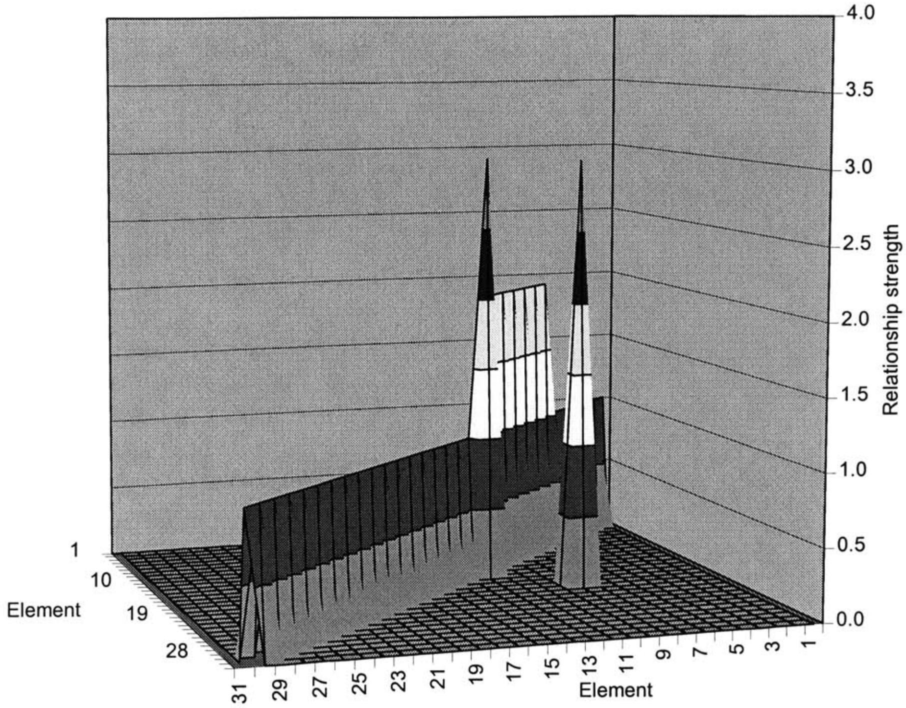


Figure 74: Illustration Of The Symmetry Problem

This symmetry arises because of the transparency of the network to inter-element forces. It is necessary to create asymmetry to enable successful clustering. The bottom-up algorithms do this by weighting the objective function differently for elements within a cluster boundary than outside it, as if the floor value of a cluster were being shifted (by a step function). This is why it is these algorithms are so highly unstable – it is a one-way trip for an element into a cluster. Implicit in this is the notion that distance matters in the network as, if it did not there would be no reason to weigh elements differently dependent on their location. The notion that distance matters can be applied more subtly, by treating the network as if some degree of visibility impairment occurs with distance. This is slightly better than simply imposing some artificial blanket force field across the network, as each element will be treated individually. It would however be path dependent – elements will be most affected by local relationships – and so more attention will be needed to negate these effects (a manageable problem). A welcome aspect of this approach is that one need not artificially unravel the DSM torus.

Such a penalty is equivalent to the concept of resistance, R in an electrical network and there are mathematical expressions for solving such networks. Such an approach can be extended to become dynamic by the introduction of concepts equivalent to inductance and capacitance to create networks analogous to LRC networks that will naturally oscillate and decay. This decay would naturally be accelerated by injudicious clustering and new energy would be injected into the system as technical potential is realised – maximised of course by judicious clustering into beneficial architectures.

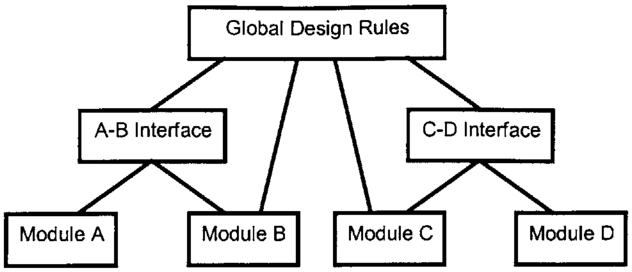
The only problem with this is that there is no apparent justification for applying the concept of off-diagonal distance in the physical domain. For this reason I shall set it aside until later, and in the meantime turn to the related subject of visibility itself and the extent to which it is possible to move away from considering only simplistic nested hierarchies.

11.7 More Notes On Visibility: Information Hiding & Design Hierarchies

11.7.1 Literature On Visibility

Baldwin & Clark (2000) take the stance that for modularisation to work in practice architects must partition design parameters into two categories: visible information and hidden information. Only visible information may interact outside a module and so a good modular design will contain interface specifications that serve to decouple hidden information from visible information so as to allow designers the maximum flexibility.

Baldwin & Clark explain that the relationship between hidden and visible information can be represented in a design hierarchy as well as in the complementary DSM. The left side of Figure 75 reproduces an example design hierarchy that they give, and the right side depicts two equally consistent alternative sequences of the equivalent DSM that I have constructed. In this example design hierarchy there are three levels of visibility: global design rules (at the top); locally visible intermediate level design rules (interface rules in this example); and intra-module design rules (at the bottom, which they term hidden modules).



TO \ FROM	GD	A-B	C-D	A	B	C	D
Global design rules							
A-B interface	1						
C-D interface	1						
Module A		1					
Module B	1	1					
Module C	1		1				
Module D			1				

TO \ FROM	GD	A-B	B	A	C-D	C	D
Global design rules							
A-B interface	1						
Module B	1	1					
Module A		1					
C-D interface	1						
Module C	1				1		
Module D					1		

Figure 75: Three Level Design Hierarchy (B&C Figure 3.5) and DSM Equivalent

With a three level hierarchy a design rule may be directly visible to a module below it (i.e. a line links the two boxes, sometimes bypassing a level in which case it is a redundant link in some respects) or indirectly visible via an intermediate level rule. Thus in this example the global design rules are visible to all of the system, each set of interface rules is visible to 3/7 of the system, and each set of module rules is visible to 1/7 of the system.

This approach to visibility is of course only valid whilst the system is a nested hierarchy where information (design rules) only flows downwards. In order for this to be achievable the system level design rules must be – for all practical purposes – scale independent. If such a nested hierarchy exists only one element may be at the apex and so it is interesting that Baldwin & Clark give examples where multiple sets of rules are visible to all elements (e.g. the computer workstation example has 3 elements with 100% visibility, i.e. these three must form a triangular ring structure at the apex in order to remain consistent) as this appears to contradict the hierarchical assumption.

Eppinger, Nukala, and Whitney (1997) discuss the same issue of information flow from a more generalised standpoint. Whilst Baldwin & Clark largely assume complete knowledge in determining the design rules, and complete effectiveness in promulgating them, it is known that time-consuming design iteration does occur in practice and such iterative processes may be modelled using signal flow graphs (Eppinger et al; 1997). Each individual task element is described by a quantity known as the branch transmission that is composed of the time taken to complete a task (and resources etc.) and the probability of rework. Such signal flow graphs are describing the task domain. In the context of calculating the visibility of a system Baldwin & Clark are essentially assuming that the probability of rework is negligible and this is an assumption I have followed in the notes below[41].

11.7.2 Application Issues

In order to calculate the option value created by an operator it is important to determine how visible is an element's design rules.

11.7.2.1 Identify Missing Task Domain Elements

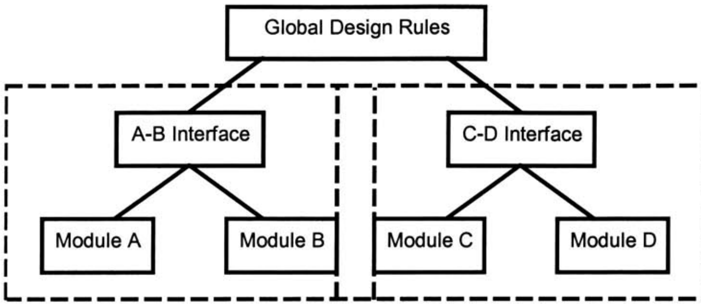
Firstly every box in a hierarchy diagram is an element. However if one only looks in the physical domain these design rules may not be immediately obvious as they are implicit, not explicit. This is not to say that they are not present in the physical domain, rather it is to say that they may not be literally physically touchable. In the example above (Figure 75) whilst the four modules are physically present (e.g. lines of software code or more literal objects such as engines and gearboxes) it is not obvious if the interface rules have a literal presence (e.g. a mating flange) or whether this needs to be inferred. This uncertainty is even more apparent at the level of the global design rules. So simply counting up the number of literal physical boxes may not tell us the real number of design elements. This suggests that information can be lost in going from the task domain to the physical domain as there may be insufficient explicit information in the physical domain to recreate the task domain.

11.7.2.2 Calculate The Visibility

Assuming that all the boxes can be recreated and the DSM written it is necessary to determine the visibility. In order to do this it is first best to re-sequence the DSM to force all the tick marks below the diagonal. Then each stripe or band should form a decompositional sequence delineating the intermediate level architectural subsystems. In a three level system it only remains to fill in any missing relationship gaps in the global design rules (which arise from indirect as opposed to direct relationships), add in each element's design relationship with

[41] I rather suspect that this is one of the crucial over-simplifications that allow modularity to be treated so insouciantly by some academics. In practice determining the design rules sufficiently to allow ~100% process yields is very difficulty and not even Toyota has achieved it. Whitney comments on this with "the VLSI industry would say the auto industry hasn't tried anywhere near hard enough" (speaking at MIT, 2001).

itself on the diagonal, and then sum the columns as shown in the Figure 76 below for the revised Baldwin & Clark example seen previously.

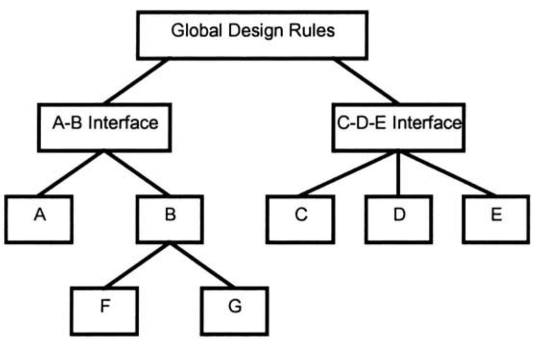


FROM

TO	GD	A-B	B	A	C-D	C	D
Global design rules	1						
A-B interface	1	1					
Module B	1	1	1				
Module A	1	1		1			
C-D interface	1				1		
Module C	1				1	1	
Module D	1				1		1
Absolute visibility	7	3	1	1	3	1	1
Relative visibility	1.0	0.4	0.1	0.1	0.4	0.1	0.1

Figure 76: Calculating Visibility In A Three Level Nested Hierarchy

The same process can be applied to any number of levels provided that a nested hierarchy exists; obviously DSM bands become more complicated as systems have more levels. A four level example is shown in Figure 77 below.



FROM

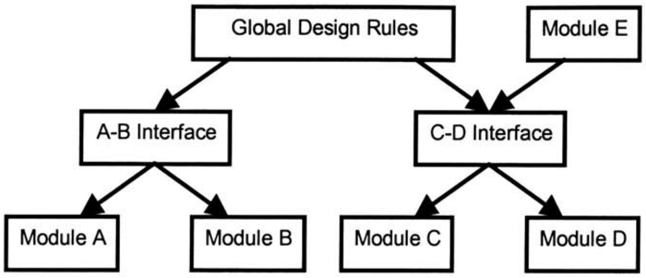
TO	GD	AB	CDE	A	B	C	D	E	F	G
Global design rules										
A-B interface	1									
C-D-E interface	1									
Module A		1								
Module B		1								
Module C			1							
Module D			1							
Module E			1							
Module F					1					
Module G					1					

FROM

TO	GD	AB	A	B	F	G	CD	C	D	E
Global design rules	1									
A-B interface	1	1								
Module A	1	1	1							
Module B	1	1		1						
Module F	1	1		1	1					
Module G	1	1		1		1				
C-D-E interface	1						1			
Module C	1						1	1		
Module D	1						1		1	
Module E	1						1			1
Absolute visibility	10	5	1	3	1	1	4	1	1	1
Relative visibility	1.0	0.5	0.1	0.3	0.1	0.1	0.4	0.1	0.1	0.1

Figure 77: Calculating Visibility In A Four Level Nested Hierarchy

The same rules can also be applied to a system with multiple top-level design rules ***provided that they are mutually shielded from each other*** as shown in the example below. This shielding has the effect that no single task block has 100% visibility.

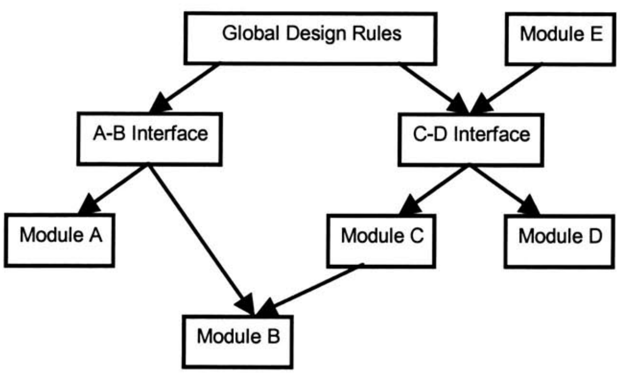


TO \ FROM	GD	E	A-B	B	A	C-D	C	D
Global design rules	1							
Module E		1						
A-B interface	1		1					
Module B	1		1	1				
Module A	1		1		1			
C-D interface	1	1				1		
Module C	1	1				1	1	
Module D	1	1				1		1

Absolute visibility	7	4	3	1	1	3	1	1
Relative visibility	0.9	0.5	0.4	0.1	0.1	0.4	0.1	0.1

Figure 78: Calculating Visibility In A System With Partitioned Top-Level Design Rules

It is possible to cross-link design rules further down the system and still calculate visibility in the system as the next example demonstrates. When such a linkage occurs, provided that information cascades down through the system in one direction only and no recursion occurs, it will always be possible to calculate a deterministic answer.

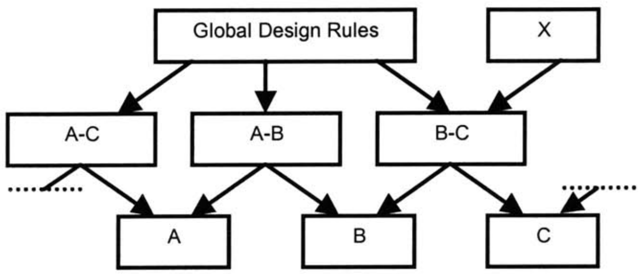


TO \ FROM	GD	E	C-D	C	D	A-B	B	A
Global design rules	1							
Module E		1						
C-D interface	1	1	1					
Module C	1	1	1	1				
Module D	1	1	1		1			
A-B interface	1					1		
Module B	1	1	1	1		1	1	
Module A	1					1		1

Absolute visibility	7	5	4	2	1	3	1	1
Relative visibility	0.9	0.6	0.5	0.3	0.1	0.4	0.1	0.1

Figure 79: Calculating Visibility In A System Cross-Linked At The Base

The following example demonstrates how this can be extended to fully cross-linked ring structures:

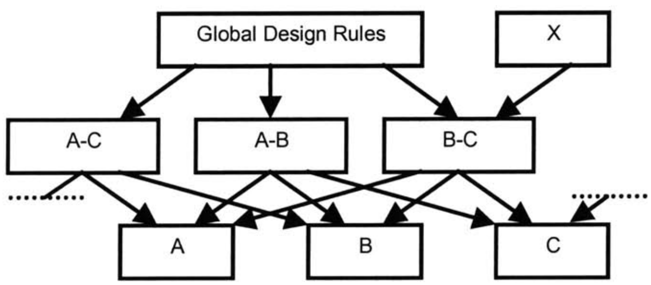


	FROM							
TO	GD	X	BC	AC	AB	C	A	B
Global design rules	1							
X		1						
B-C	1	1	1					
A-C	1			1				
A-B	1				1			
C	1	1	1	1		1		
A	1			1	1		1	
B	1	1	1		1			1

Absolute visibility	7	4	3	3	3	1	1	1
Relative visibility	0.9	0.5	0.4	0.4	0.4	0.1	0.1	0.1

Figure 80: Calculating Visibility In A Ring Structure

And the logical conclusion of this is to non-recursively fully cross-link the lowest level:

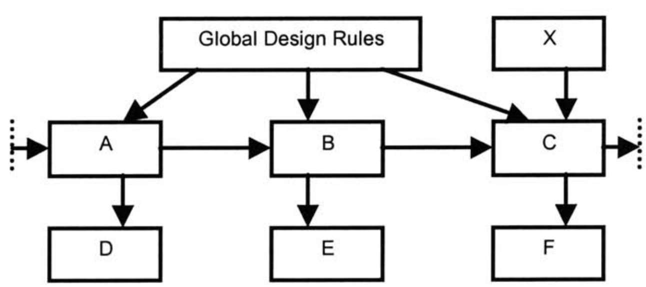


	FROM							
TO	GD	X	BC	AC	AB	C	A	B
Global design rules	1							
X		1						
B-C	1	1	1					
A-C	1			1				
A-B	1				1			
C	1	1	1	1	1	1		
A	1	1	1	1	1		1	
B	1	1	1	1	1			1

Absolute visibility	7	5	4	4	4	1	1	1
Relative visibility	0.9	0.6	0.5	0.5	0.5	0.1	0.1	0.1

Figure 81: Calculating Visibility In A Structure With Full Low-Level Linkage

Then of course as soon as recursive horizontal cross-linking occurs it exposes every element beneath the level of recursion to the entire higher-level rule set as shown in the example below:

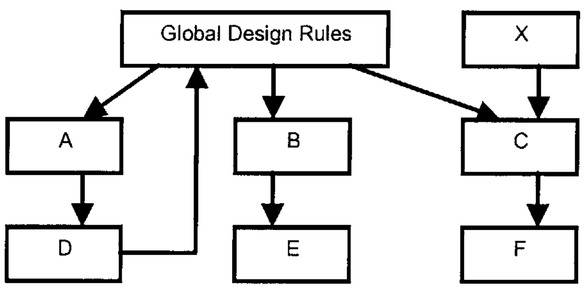


	FROM							
TO	GD	X	A	B	C	D	E	F
Global design rules	1							
X		1						
A	1	1	1	1	1			
B	1	1	1	1	1			
C	1	1	1	1	1			
D	1	1	1	1	1	1		
E	1	1	1	1	1		1	
F	1	1	1	1	1			1

Absolute visibility	7	7	6	6	6	1	1	1
Relative visibility	0.9	0.9	0.8	0.8	0.8	0.1	0.1	0.1

Figure 82: Calculating Visibility In A Structure With Horizontal Recursion

Likewise as soon as vertical recursion occurs the entire rule set of that branch (down until the recursion is bounded) is effectively raised to the level at which the recursion starts, i.e. the existence of a feedback mark in a block has the effect of fully populating the relevant area of the DSM. In small systems this has the effect of tipping the system on its side, and it almost always will lift hidden information to a more visible location. The 'power method[42]' may be used to explore the reach of such feedback loops although, unless some degree of visibility impairment (i.e. resistance) is assumed, there is no logical reason to stop raising the DSM to the n-th power and thereby fully populating the feedbacks to their limit.



TO \ FROM	GD	A	D	X	B	E	C	F
Global design rules	1		1					
A	1	1						
D	1	1	1					
X				1				
B	1	1	1		1			
E	1	1	1		1	1		
C	1	1	1	1			1	
F	1	1	1	1			1	1

Absolute visibility	7	6	6	3	2	1	2	1
Relative visibility	0.9	0.8	0.8	0.4	0.3	0.1	0.3	0.1

Figure 83: Calculating Visibility In A Structure With Vertical Recursion

Such recursion need not have intermediate elements to occur – bi-directional rule exposure between two elements is sufficient.

11.7.2.3 Weighted Relationships & The Heart Of The Relationship Matter

The use of simple binary relationship tick marks skirts the issue of what rule is being related between elements. For example in Figure 83 above it is sensible to ask exactly what rules apply to element F. Obviously the visible rules of X are imposed upon F, and so too are the global design rules (and via recursion A and D). However it is not clear whether C itself imposes additional rules upon F. This is relevant because the designer of hidden module F needs to know what constrains the design space.

Mathematically the set of rules for F is the intersection of the set of rules of X, GDR, and C. But if C is simply the union of GDR and X a different design space results than if C is a sub-

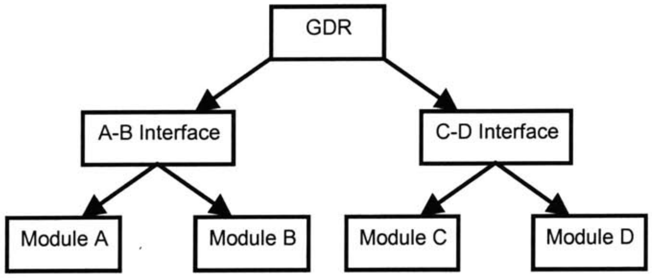
[42] See http://web.mit.edu/dsm/Tutorial/partitioning_powers.htm for a detailed explanation.

set of X union GDR. Indeed the rules that C introduces may not have been considered in either the X or GDR rule set. Revisiting the analogy of the wireless LAN card in the computer clearly the LAN card must obey the intersection of the 802.11b rule set and the PCMIA rule set, however the issue is whether the computer designer has in some way reduced the design flexibility by making the computer ('C') less flexible than the full set of 802.11b and PCMIA rules, and whether the computer designer has perhaps overlain it with some other unanticipated rule (e.g. the computer's carrying case may not have sufficient clearance for the card's protruding antennae).

This is an unresolved issue in most DSMs. Any that are going to be used for calculating visibility need to explain what the contents of each box are as assumptions lead to ambiguity. Clearly the data explosion this creates is likely to be a challenge for any academic investigation of non-trivial products. For my purposes I shall presume that if an intermediate element is introduced then something is occurring as otherwise it would have been redundant. In future when drawing hierarchy diagrams it may be useful to reduce the ambiguity.

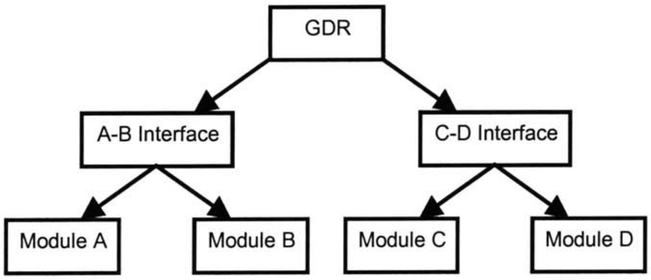
Weighted relationships can be used to denote either the strength of imposition of a rule set, or the quantity of the rule set. For example the gas turbine DSM notionally and intuitively considered transmission of four dimensions: energy, mass, spatial relationships, and pure information. Because of the intuitive ambiguity used in developing the DSM it was not clear whether a score of 3 indicated that many dimensions worth of rules were in play, or whether only one dimension's rules was being related but in a highly constrained way.

Again this is important to specify more clearly if using a DSM to analyse visibility. If it is specified then it may be possible to keep track of changes in visibility in a structure as the examples below show with the same network in each case, in which a relationship can score at most 3. The left-hand branch has increasingly tightly defined design rules as one drills down to the hidden modules which is indicative of a situation where the intermediate level rules are more constraining than the higher level rules. The right hand branch exhibits less constrained rules as one drills down: this may illustrate a situation where a fully rigorous application of the design rules is not taking place, either erroneously or because they were over-specified. This could occur if the rules had more than one dimension only one of which was applicable to the right-hand branch and thereby this illustrates the risk of creating (and depicting) too integrated a high-level rule set.



TO \ FROM	GDR	A-B	B	A	C-D	C	D
GDR	1						
A-B interface	2	1					
Module B	2	3	1				
Module A	2	3		1			
C-D interface	2				1		
Module C	2				1	1	
Module D	2				1		1

Absolute visibility	13	7	1	1	3	1	1
Relative visibility	1.9	1.0	0.1	0.1	0.4	0.1	0.1



TO \ FROM	GDR	A-B	B	A	C-D	C	D
GDR	2						
A-B interface	2	3					
Module B	2	3	1				
Module A	2	3		1			
C-D interface	2				1		
Module C	2				1	1	
Module D	2				1		1

Absolute visibility	14	9	1	1	3	1	1
Relative visibility	2.0	1.3	0.1	0.1	0.4	0.1	0.1

Absolute visibility	14	9	1	1	3	1	1
Norm. Rel. Vis.	0.7	1.3	0.1	0.1	0.4	0.1	0.1

Figure 84: Calculating Visibility With Weighted Relationships

Numerical analysis of these examples also reveals ambiguity. In the upper set of calculations the diagonal is kept at '1' as in the binary version. In the lower set of calculations the diagonal expresses the intrinsic strength of that rule set. There is then a second way of calculating the relative visibility, as it is now possible to normalise it by dividing through by the maximum strength possible.

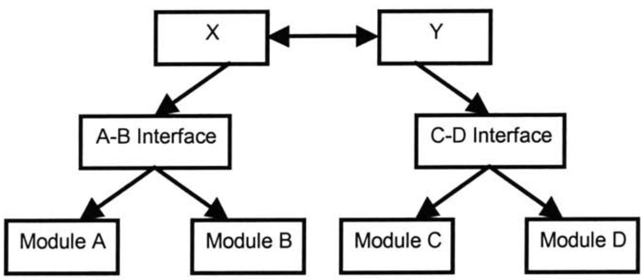
Rather depressingly it begins to appear as if truly meaningful quantitative analysis using DSMs will require extremely full information along the lines of:

$$[\text{Rule X}] = \begin{bmatrix} parameter\_name & etc & etc \\ used\_or\_not & 1 & 0 \\ value & etc & etc \end{bmatrix}$$

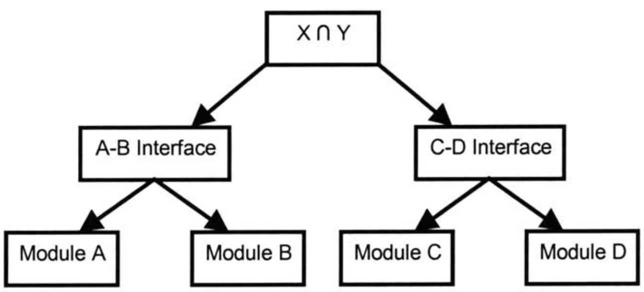
else the DSMs will be too ambiguous for non-contentious analysis. However there may still be merit in a qualitative analysis of the intuitively populated weighted DSMs. This will require careful normalisation of the visibility factors.

11.7.2.4 Visibility Of A Cluster

By assuming that all relationship rules are equivalent and binary, clusters can be handled. In this example two sets of global design rules (X and Y) are mutually visible as shown in the adjacent DSM and the upper drawing. This means that the relevant design rule for all lower level elements can be reduced to X intersection Y as shown in the lower level drawing and DSM. Of course this lower level drawing and DSM are exactly the situation illustrated in many depictions of systems at intermediate levels of decomposition where each element represents multiple primitive elements. This is why intermediate levels of decomposition (and abstraction) are at once useful and dangerous: useful because they can strip out the clutter of complicated detail, and dangerous because they can at the same time strip out meaningful complexity to suggest an overly simple situation.



TO \ FROM	X	Y	A-B	B	A	C-D	C	D
X	1	1						
Y	1	1						
A-B interface	1	1	1					
Module B	1	1	1	1				
Module A	1	1	1		1			
C-D interface	1	1				1		
Module C	1	1				1	1	
Module D	1	1				1		1
Absolute visibility	8	8	3	1	1	3	1	1
Relative visibility	1.0	1.0	0.4	0.1	0.1	0.4	0.1	0.1

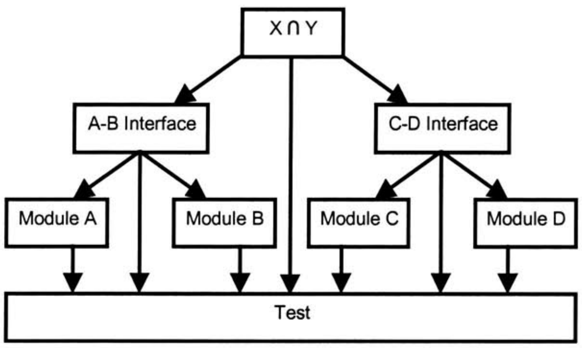


TO \ FROM	X ∩ Y	A-B	B	A	C-D	C	D
X ∩ Y	1						
A-B interface	1	1					
Module B	1	1	1				
Module A	1	1		1			
C-D interface	1				1		
Module C	1				1	1	
Module D	1				1		1
Absolute visibility	7	3	1	1	3	1	1
Relative visibility	1.0	0.4	0.1	0.1	0.4	0.1	0.1

Figure 85: Calculating Visibility & Collapsing Clusters

11.7.2.5 Visibility & Dependence

So far visibility has only been considered from the rather top down perspective of the source of the rule(s). From this consideration the more visible an element is the more impact a change to its contents will be. This of course need not be interpreted as being more 'powerful' as the use of such emotive language may not be warranted in the context (e.g. commanding officers often feel more trapped than empowered): a more neutral analogy might be to describe highly visible elements as being highly responsible as one needs to carefully consider unanticipated consequences of a change. However visibility can also be viewed from the bottom up perspective of the element being acted upon by the rules. These lower level elements (at least in non-recursive structures) are highly dependent whilst being relatively invisible. This dependency is at once at blessing and a constraint for, provided they continue to obey their bounding rule-set, they can evolve without fear of unanticipated consequences.



TO \ FROM	X ∩ Y	A-B	B	A	C-D	C	D	Test	Abs.	Rel.
X ∩ Y	1								1	0.1
A-B interface	1	1							2	0.3
Module B	1	1	1						3	0.4
Module A	1	1		1					3	0.4
C-D interface	1				1				2	0.3
Module C	1				1	1			3	0.4
Module D	1				1		1		3	0.4
Test	1	1	1	1	1	1	1	1	8	1.0
Absolute visibility	8	4	2	2	4	2	2	1		
Relative visibility	1.0	0.5	0.3	0.3	0.5	0.3	0.3	0.1		

Figure 86: Calculating Visibility & Dependence

Dependency can be quantified by taking the horizontal sum of the relationships (i.e. the number of inputs, rather than the number of outputs) as shown in the figure above in which a final test & integrate element has been introduced – this is the sort of system diagram characteristic of a three stage modular design. The scatter plot of visibility against dependence is then another way to characterise the architecture of the system as the figure below shows for this example system.

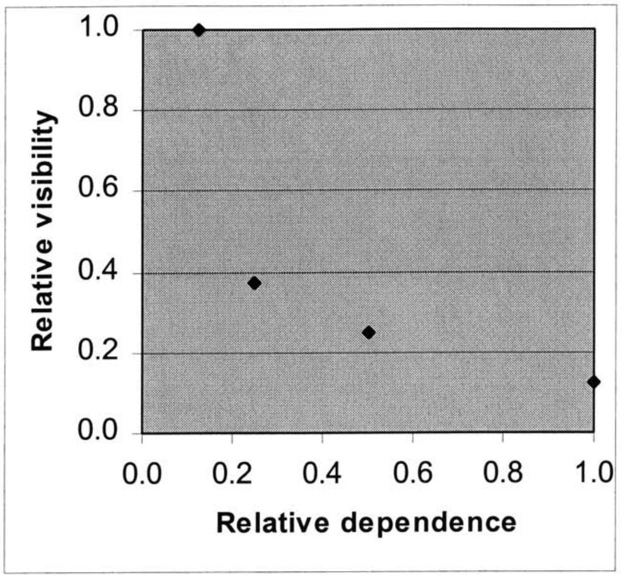
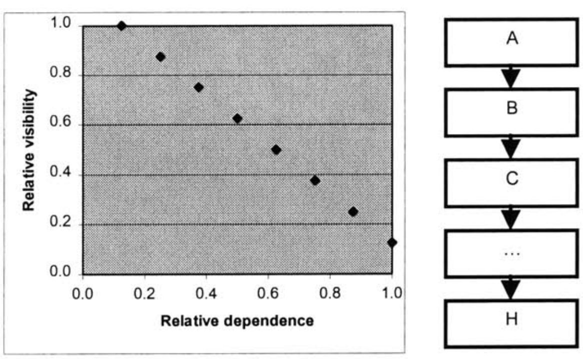


Figure 87: Visibility & Dependence Signature Of A Three Stage Modular System

In this particular example a number of the data points overlie each other. It is interesting to consider the characteristic signatures of other architectures of which the simplest is shown below. This is a straightforward linear hierarchical system and it yields a linear diagonal signature. The offset from the 0,1 – 1,0 is an artefact of the inclusion of the element's relationship with itself in the calculation.



TO \ FROM	A	B	C	D	E	F	G	H	Abs.	Rel.
A	1								1	0.1
B	1	1							2	0.3
C	1	1	1						3	0.4
D	1	1	1	1					4	0.5
E	1	1	1	1	1				5	0.6
F	1	1	1	1	1	1			6	0.8
G	1	1	1	1	1	1	1		7	0.9
H	1	1	1	1	1	1	1	1	8	1.0
Absolute visibility	8	7	6	5	4	3	2	1		
Relative visibility	1.0	0.9	0.8	0.6	0.5	0.4	0.3	0.1		

Figure 88: Visibility & Dependence Signature Of A Linear Hierarchical System

11.7.2.6 Characterising The Visibility-Dependence Signature

The visibility – dependence signature of a system can be described by the following characteristics, which are illustrated in the six diagrams on the next page, Figure 91.

Homogeneity

The degree of scatter or dispersion in the data points illustrates the degree of homogeneity or heterogeneity in the system. In assessing this visually one needs to exercise some caution, as identical data will only show as one point. A homogenous system is one where all elements are equally linked – these are fairly 'neutral' architectures, i.e. it is by definition for a sub set of elements to dominate the system. Homogenous systems will therefore tend to be closed topologies – such as two or three dimensional ring structures.

Connectivity

Not all networks are equally connected and this is observable in these diagrams by the relative location of the centre of gravity. The closer the centre of gravity is to the origin the lower the degree of connectivity. An issue to be aware of is that the placing of the indirect relationships into the DSM will mask the degree of intrinsic connectivity in the system and so it is worth considering the result that is obtained without indirect relationships.

Layering

Multiple tightly clustered groups or horizontal lines indicate that a layered structure exists. Depending on the degree of connectiveness & topology this may be a layer-cake decomposition or it may be a more of a concentric spherical decomposition. Layering eases modularisation. The topmost layer may only be represented by one data point and the number of data points tends to grow as one descends the layers. If the DSM has been constructed at a given level of decomposition in the physical domain, and not had higher-level rule elements added in, these rule layers might not be represented even though they do exist.

Unilateral & Bilateral Relationships

The extent to which the network has one-way unilateral relationships as opposed to bilateral relationships will determine the slope of the trend line in the data. This is because bilateral relationships are essentially 'democratic' – the more an element is visible the more it is dependent, whilst unilateral relationships lie along the orthogonal trend line. The reverse also holds true – the slope of the trend line is a quantifiable measure of the degree of bilateral versus unilateral relationships in a system. From a DSM perspective unilateral systems can be re-sequenced to force relationships below the diagonal, whilst for a bilateral system this is not true as the DSM will be symmetric around the diagonal.

Decompositional Cleanliness

If relatively unilateral relationships dominate the data, then the cleaner the system decomposes the steeper the (negative) gradient of the trend line. This is because clean decompositions have few relationships that cross the decompositional boundaries and so the visibility of a given level of decomposition is minimised. This effect holds true irrespective of the relationship count (e.g. 5, 7, 9 splits at each level, etc.) although that will affect the relative density of data points in different locations on the chart.

Ratio Analysis

The relative density of data points on the graph can be observed by taking the ratio of visibility to dependence for each element and then plotting the frequency distribution, and correlation with visibility or dependence as the two examples below indicate[43].

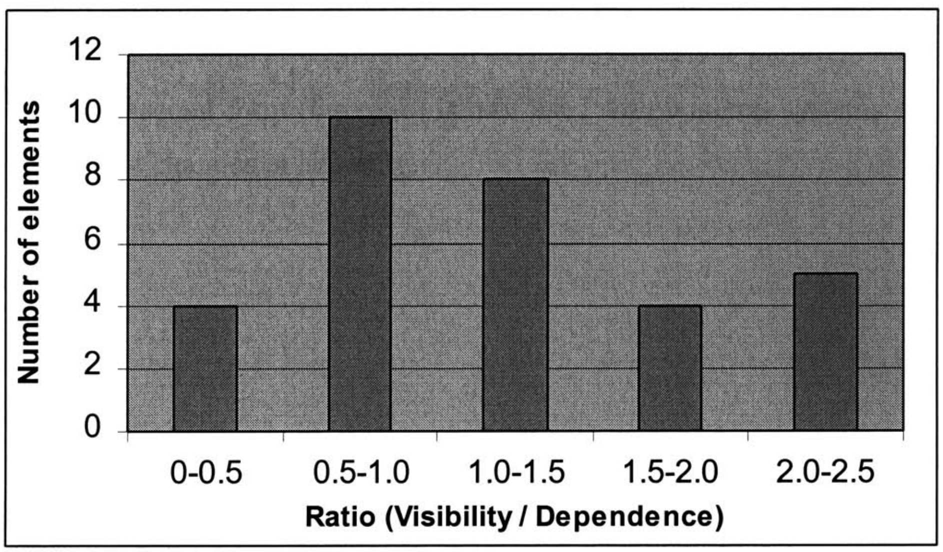


Figure 89: Example Frequency Distribution Of Visibility / Dependence Ratio

[43] The data in these two examples is that of the weighted gas turbine DSM without insertion of the implicit indirect relationships. So whilst it is incomplete it is a useful data set to illustrate intermediate characteristics.

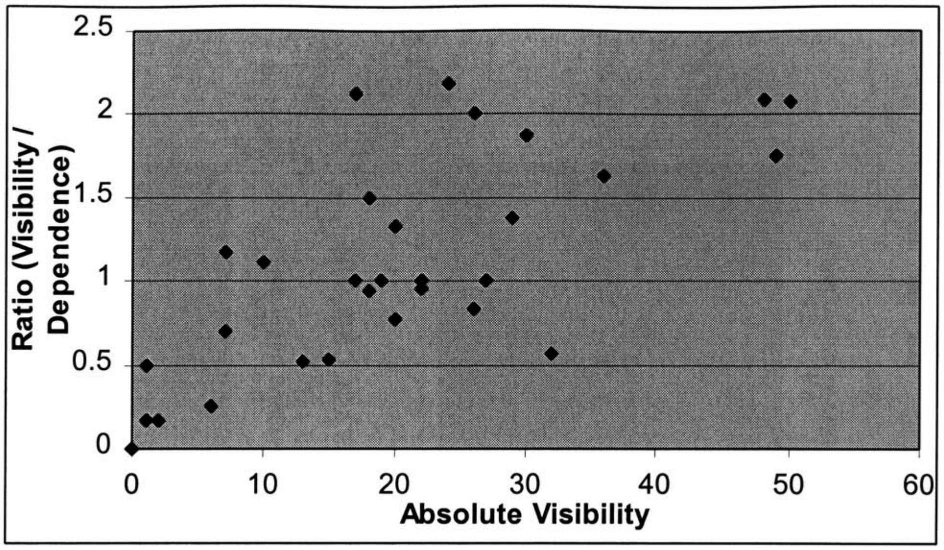


Figure 90: Example Visibility / Dependence Ratio versus Visibility

The frequency distribution graph is best used in unilateral systems where it tends to reveal the numbers of relationships at each level. In bilateral systems it indicates the dispersion around the mean. The second form (the ratio) is best used with bilateral systems where it reveals any depth-dependent changes in laterality.

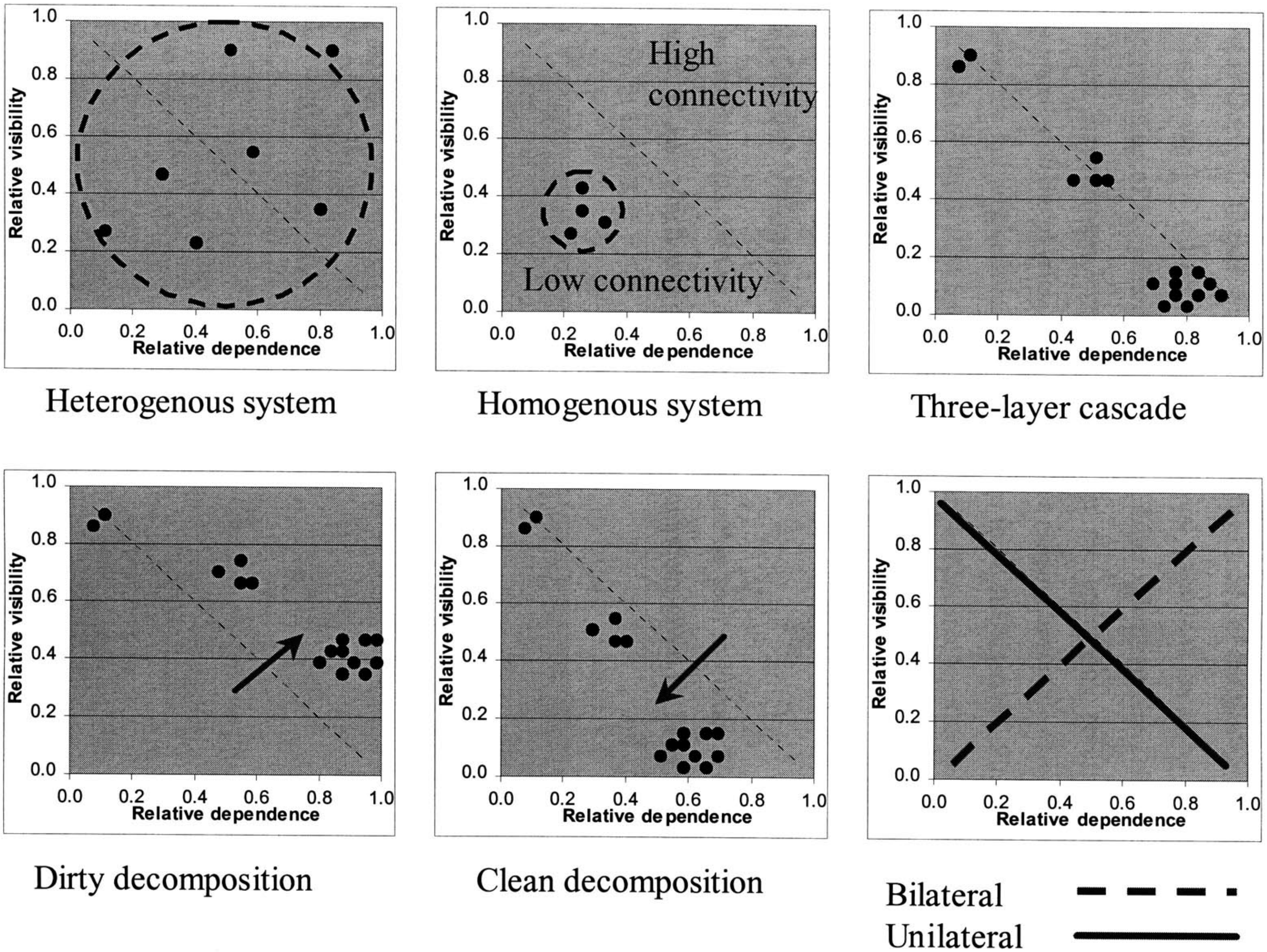


Figure 91: Characteristic Types Of Visibility-Dependence Signatures

12 REVISED VALUATION OF THE GAS TURBINE GENERATOR SET ARCHITECTURE

In this chapter a further series of valuations are made of the gas turbine's architecture using different ways of calculating visibility. Initial valuations focus on applying the improved understanding of the principles of visibility. Later valuations show the impact of imposing a different intermediate level architecture on the underlying reality. The task domain is compared with the physical domain and some missing intangible elements are identified. This allows the full architectural system signature to be characterised. Finally a proposal is made that seeks to justify joint optimisation of multiple domains, together with a suggested objective function for maximising societal value that may resolve the apparent contradictions between what is 'best' in different domains.

12.1 Initial Valuation

The starting point for this chapter is the valuation seen in Figure 27 which is reproduced as Figure 92 and which was based on the assumptions given in Table 4 and the element data shown in Table 3.

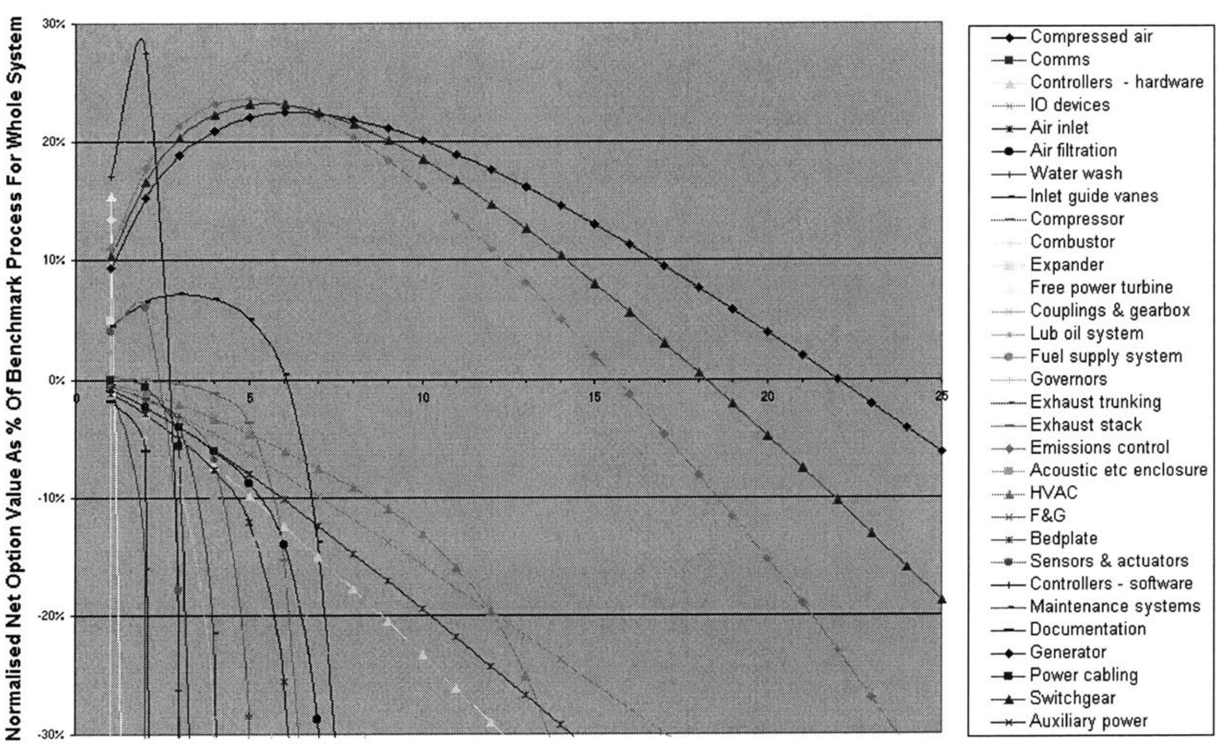


Figure 92: Reproduction of NOV of Genset With Updated Task Domain Assumptions

12.2 Valuation Using Raw DSM Data

Until now the element valuation has used a single directly estimated value for each elements' visibility to the system. As described in previous sections the DSM contains visibility information and even in it's raw form this is likely to be more reliable than the single-point estimate[44]. Taking the weighted asymmetric DSM shown in Figure 46 and summing the relationship strengths vertically we obtain the raw visibility. In performing this summation the assumption is made that the diagonal is of zero weight as the purpose of determining visibility for an option value calculation is for use as external visibility only. Clearly this yields a different set of numbers than the single point estimate – this after all is the reason for doing it. In order to compare the results and minimise disruption when inserting them into the option value calculation it is best to scale them and linear scaling is used in the absence of a rationale for anything else. Comparison of the two data sets shows a linear R^2 value of 0.23 so the estimated values have some agreement with the raw DSM values, which is to be expected, whilst being sufficiently different as to merit attention.

It is a simple matter to insert these new visibility numbers into the option value calculation and the results are shown in Figure 94 below. The overall patterns have not changed: there are still four recognisable types although the increased heterogeneity in the curves has made selection of type boundaries slightly more contentious, and broadly each element is still in the same type as before.

12.2.1 Numerical Result

By taking the maximum for each element it is possible to quantify the value of the system:

Table 8: Table Of Selected Option Value Results For 31-Element System

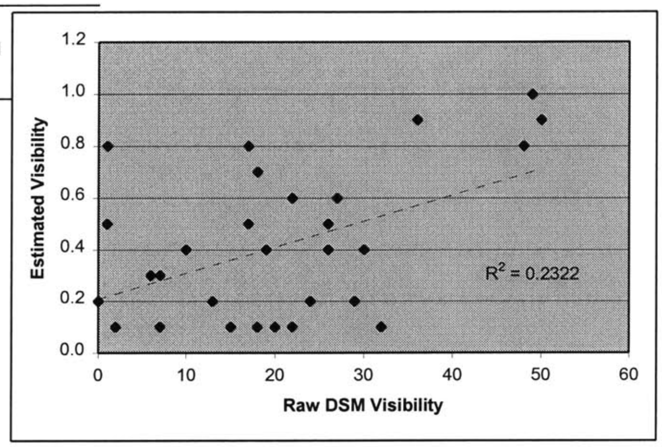
Item	Optimum number of trials	Result
Inlet guide vanes	3	0.07
Compressor	1	0.11
Combustor	1	0.13
Etc.		
...		

[44] All other factors being equal the expectation error of one estimate is greater than the expectation mean error of many.

...		
Etc.	Etc.	Etc.
System Value		**1.87**

Element	Raw DSM Visibility	Estimated Visibility	Scaled Raw DSM Visibility
Compressed air	1	0.8	0.02
Comms	2	0.1	0.04
Controllers - hardware	7	0.3	0.14
IO devices	1	0.5	0.02
Air inlet	18	0.7	0.37
Air filtration	17	0.8	0.35
Water wash	20	0.1	0.41
Inlet guide vanes	17	0.5	0.35
Compressor	49	1	1.01
Combustor	48	0.8	0.98
Expander	50	0.9	1.03
Free power turbine	36	0.9	0.74
Couplings & gearbox	30	0.4	0.62
Lub oil system	19	0.4	0.39
Fuel supply system	26	0.5	0.53
Governors	22	0.6	0.45
Exhaust trunking	22	0.1	0.45
Exhaust stack	18	0.1	0.37
Emissions control	10	0.4	0.21
Acoustic etc enclosure	15	0.1	0.31
HVAC	13	0.2	0.27
F&G	20	0.1	0.41
Bedplate	6	0.3	0.12
Sensors & actuators	26	0.4	0.53
Controllers - software	27	0.6	0.55
Maintenance systems	32	0.1	0.66
Documentation	29	0.2	0.59
Generator	7	0.3	0.14
Power cabling	24	0.2	0.49
Switchgear	7	0.1	0.14
Auxiliary power	0	0.2	0.00
	619	12.7	12.7

SF	48.74016



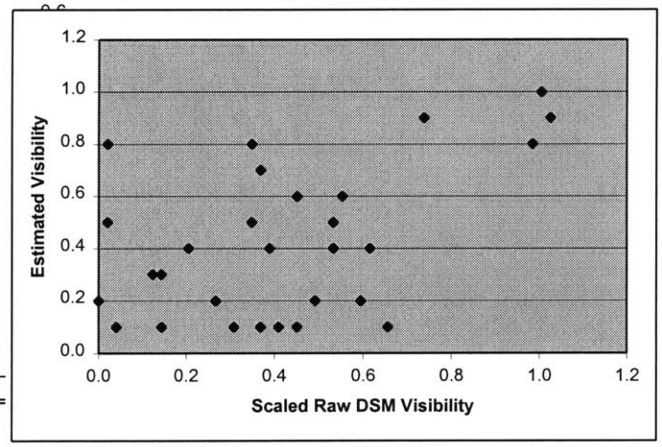


Figure 93: Visibility Of The Raw Gas Turbine DSM

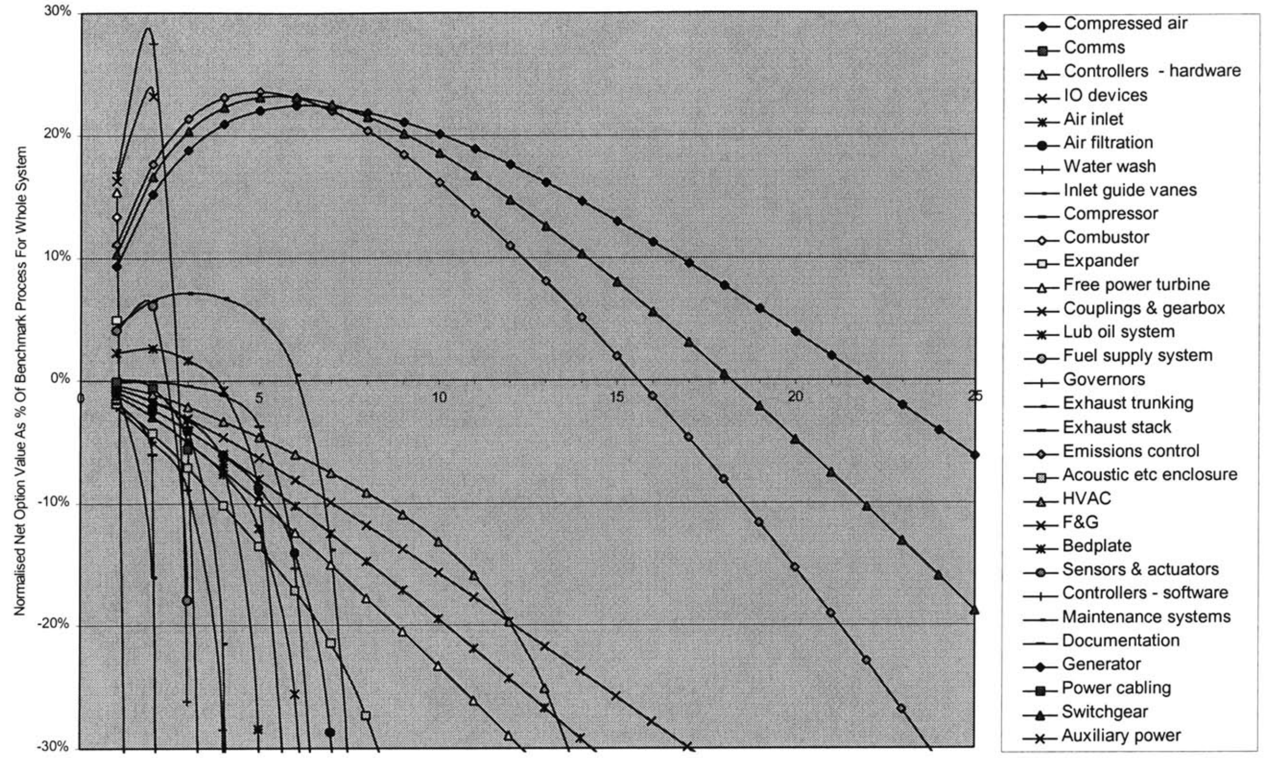


Figure 94: Gas Turbine Option Value Using Raw DSM Visibility

12.3 Valuation Using Cleansed DSM Visibility

12.3.1 Insert The Missing Elements Into the DSM

The next step is to insert the missing elements into the DSM so that the full picture may be seen. It is easier to do this in the task domain than in the physical domain, as this is where the missing assumptions are most visible. The outcome of this is illustrated in the DSM on the next page (Figure 95) in which the centre of the DSM is unchanged as it reflects the fundamental isomorphism of stage two.

In the DSM centre the physical domain elements possess a 1:1 mapping with the stage two task domain elements and so the element name should now be interpreted as "design …." rather than " … itself", e.g. the task of "design air inlet" rather than the assembly of the "air inlet itself". These central elements represent for the most part the second stage of work on hidden modules and the essential sequence has been left unchanged except that the gas generator core and free power turbine have been moved to the upper left and the electrical output system inserted in its place.

12.3.1.1 The Missing Elements

Fourteen new task elements have been introduced into the DSM. Eleven of these new elements represent the stage one tasks of formulating the design rules and these are placed first in the sequence, and three represent the stage three tasks of testing and integrating and are placed last in the sequence. These fourteen elements are my interpretation of the missing elements of the design process of a generic generator set together with my interpretation of the overall relationships[45]. Because these fourteen elements are a new introduction they are outlined next.

The relationships pertaining to stage one and stage three elements are indicated in the DSM by a tick mark ('X'). These tick marks are less firm than the relationships indicated in the central stage two area which is why, for the time being, they have not been assigned numerical weights. Also the exact location of the tick marks is less certain – for the time being it is the trend that is important rather than the details.

12.3.1.1.1 Stage One: Formulate Design Rules

The first four elements of stage one determine the scale of the product. This is important because integrated or semi-integrated complex systems do not scale in the same way as fully modular systems. Whilst engineers often possess design rules for predicting emergent properties these may operate only over a limited span. Thus it is first necessary to determine what is the fundamental 'point' of interest in the design space. In the case of the genset the main issues are the nominal power output, the nature of the load (e.g. base load, intermittent peaking duty, or mixed), the time to market, and any other notable customer requirements (e.g. hot/high duty, max. installation weight requirements, maintainability, etc.). These then serve as the inputs into the application of scale dependent rules that determine the optimum cycle (e.g. simple cycle, combined cycle), configuration (e.g. direct drive of free power turbine? Gearbox? Compressor & turbine stages? Etc) and related trade studies (e.g. derivative or new

[45] United Technologies Corporation's Pratt & Whitney division generously shared some confidential internal data regarding the early stages of the design process for a current gas turbine for intended use in power generation. In part because this gas turbine is of an extremely different size to the ones I am accustomed to, and in part because I have never worked with Pratt & Whitney engines before, this insight allowed me to generalise the design process at the appropriate level of abstraction – something I would not have been able to do on the basis of simply drilling deeper into comparable 10 MWe turbine design processes. The titles I have given for these fourteen missing task elements are not necessarily those used by Pratt & Whitney.

build? Dedicated turbines for 50Hz and 60Hz markets, or slippage, or static frequency converters? Etc). The last four elements are where a lot of the supplementary detail is put in place around the basic product decomposition that results from the selection of an optimum configuration: these details are the design rules that determine the permissible boundaries and target parameters for all of the hidden modules. So overall first stage "formulate design rules" should be divided into three subsections for semi-integrated products:

a) Determine scale.

b) Apply scale dependent rules to determine decomposition.

c) Apply decomposition dependent rules to determine module design rules.

12.3.1.1.2 Stage Two: Design Sub Systems

The central box of the DSM represents the second stage of relatively independent (parallel) designs of the sub-systems. This experimentation effort is sometimes internal to the lead firm (the prime contractor, or systems integrator, or original equipment manufacturer) and sometimes external. It is also sometimes explicit ("the best design effort wins") and sometimes implicit ("select the most appropriate off-the-shelf solution").

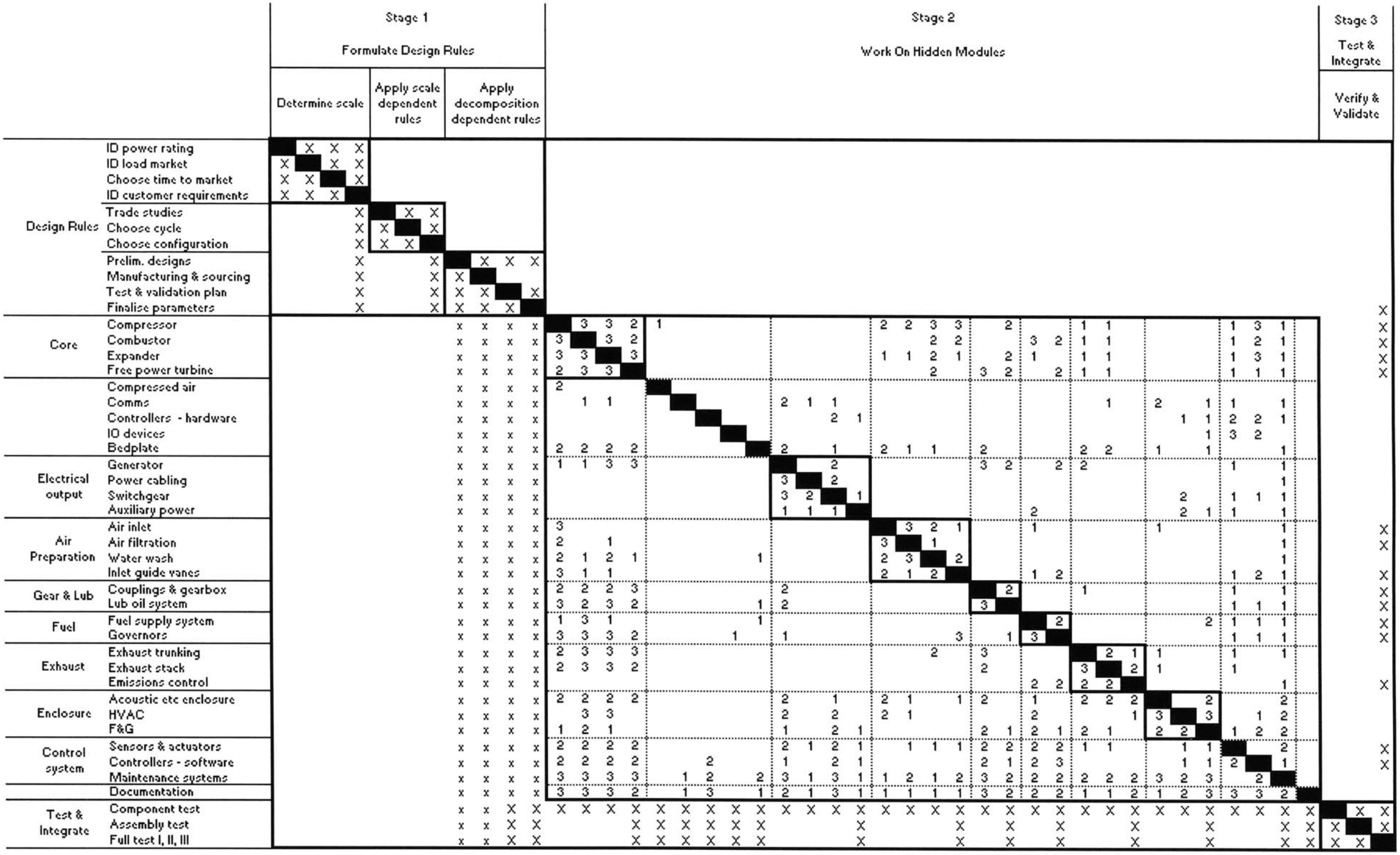


Figure 95: Complete DSM Of Genset In Task Domain

The terminology of these first three sub-stages varies from firm to firm and industry to industry: stage 1a is often called market opportunity analysis; 1b is generally referred to as conceptual design; and 1c as preliminary design.

12.3.1.1.3 Stage Three: Test & Integrate

The three task elements that comprise stage three represent the testing that takes place of any engineered product of significant value and complexity. Whilst most management literature refers to this as "test & integration" most engineering literature uses the terms "verification & validation". Verification is the process of confirming that the product meets the design requirements (Is the product built right?) and validation is the process of confirming that the customer approves of the design parameters (Has the right product been built?). This is a three stage aggregative process of testing individual components (e.g. turbine blades, lines of software code), then assemblies (e.g. compressor discs, code modules), and then increasingly more demanding final product tests (cold sea level tests to 25 ppm emissions, hot & high tests to 8 ppm emissions). These tests are a significant expense of time and money, rarely taking less than a year and perhaps taking as much as three years if significant rework is required (e.g. the Rolls Royce Trent required a ~40 months of extended development and testing to resolve combustor problems). The possibility of rework as a feedback relationship is indicated in the rightmost column of Figure 95.

12.4 Inter-Domain Mapping: Isomorphism Revisited

A series of work conducted by Eppinger and his students as summarised in Eppinger & Salminen (2001) and exemplified by Rowles (1999) comments on the apparent similarity between DSMs in the physical, task, and organisation domains as illustrated in Figure 96 below. As yet no one system has been traced exhaustively through all three domains and so Eppinger has yet to conclude these investigations, however it is this series of observations that prompted the Baldwin & Clark notes on the fundamental isomorphism between the task and physical domains (although they denote these two domains by the terminology Task Structure Matrix, TSM and Design Structure Matrix, DSM).

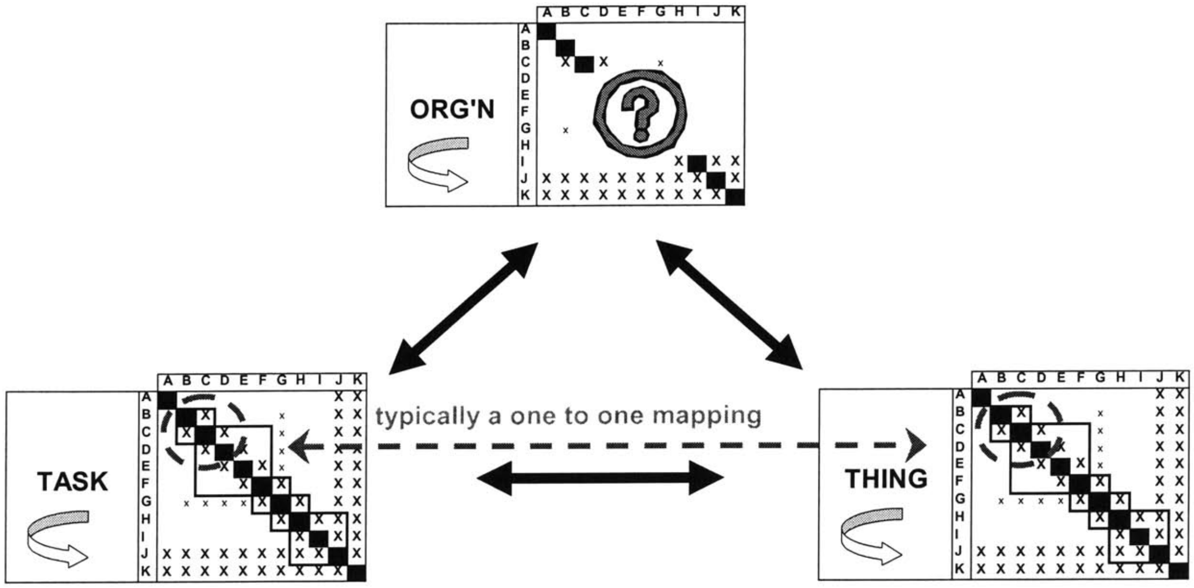


Figure 96: Hypothesised Inter-Domain Isomorphism (after Eppinger & Salminen, 2001)

My thoughts on the matter are that the data set on which these observations are being made is being bounded by the observed physical domain elements. For example the Rowles, 1999 work compares the organisation domain (Pratt & Whitney's integrated product teams) with the decomposition of the ***tangible*** aero engine's system elements in the physical domain. By definition this omits any ***intangible*** physical domain elements.

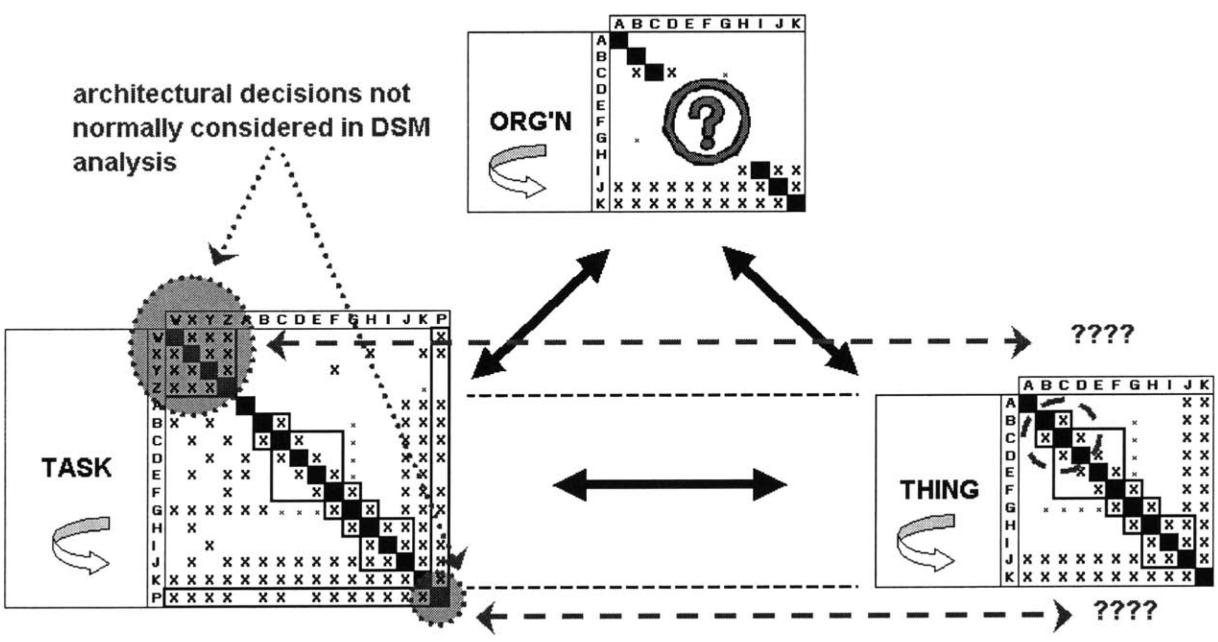


Figure 97: Alternative Hypothesis For Inter-Domain Mapping

This situation is illustrated in the revised figure above, which indicates a possible location of intangible elements as being in stage one and stage three of the design process. This is certainly what appears to be occurring in the task – physical domain mapping in the generic gas turbine genset example discussed in this thesis, and also is the case in the wireless LAN card example that has been referred to, where the physical domain decomposition would observe; the PC, the card, and the wireless router but not observe; the PCMIA and 802.11b standards. In this instance there may still be a one-to-one mapping between domains but one would need to insert the apparently intangible elements that may not currently be observed – for example there is a physical domain interpretation of the 802.11b standard as a stack of paper produced by an IEEE committee, it is merely that it wasn't obvious by inspecting the PC itself. It may also be necessary to call out and insert lower level rule sets that pertain to individual modules or elements but this will be dictated by the individual product decomposition.

Another point about this is that 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 supplier-purchaser contracts this qualitatively explains why so many supply chains carry the seeds of their own destruction.

12.5 Element Valuation

Remarkably even though new elements have been included (to make 45 elements in total) 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. This is because in this industry there is normally nothing to 'sell' at the level of intangible products. In this industry to recoup the cost of intangible elements participants need to retain a physical presence in some form or another. 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 the product remains so highly integrated this is unlikely to occur.

12.6 System Signature & Architectural Hierarchy Diagram

In order to consider the signature of the gas turbine's architecture the tick marks, that represent stage one and three relationships, are first replaced with weighted values. These weightings are once again directionally correct but not necessarily precisely so. After placing the direct relationships it is next necessary to place the indirect ones; it was decided to insert these for the stage one rules but not for stage two and stage three rules. The reason for not inserting indirect relationships for stage three rules is that they should represent intergenerational learning unless a critical design error has been made, e.g. the Rolls Royce Tyne combustor problems are an example of a critical design error that required immediate large scale rework. The reason for not inserting indirect relationships for stage two rules is rather more intuitive: because of the multi-dimensional circular relationships the entire stage two would become fully populated if these indirect relationships were to be inserted. This would cause a loss of information as to the underlying structure, which is the opposite of what is desired. The raw scores were used with the diagonal weight ignored and the result is shown in Figure 98 below with the source 45-element DSM shown in Figure 100 on the following page.

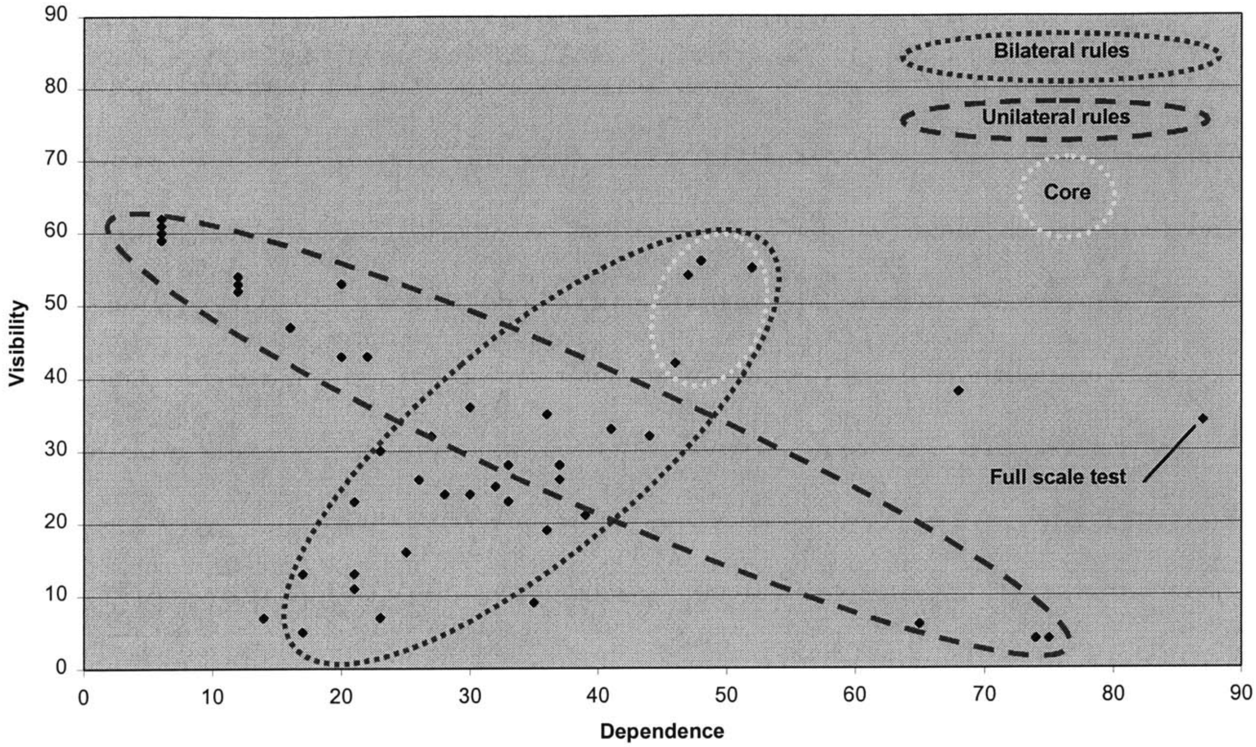


Figure 98: Gas Turbine Architecture Signature With Indirect Stage One Relationships

This figure is merely another way of looking at the architecture's structure – it need not show anything new but it may reveal trends that are hidden in the messy detail of the DSM. Bearing in mind the archetypes discussed in section 11.7.2.6 the characteristics may be categorised as follows:

Table 9: Characteristics Of Gas Turbine Architecture Signature

Characteristic	Archetype Comment
Heterogeneity	Intermediate
Connectivity	Intermediate
Layering	Clear layering at the highest and lowest levels; messier in the middle.
Laterality	Exhibits both archetypes. The core is an effective extension of the bilateral rule set into the high level unilateral space; as if stage one and stage two were pinned together by the fundamental importance of the gas generator & free power turbine.
Cleanliness	The human-imposed unilateral relationships decompose cleanly.
Ratio Analysis, c.f. Figure 99	Not directly characterisable because of the mixed laterality, however a few elements dominate the design space with high visibility ratios. With the exception of full scale testing to reveal undesirable emergent properties, these elements are intangible stage one rules, which is another way of

	saying that the early design decisions matter most.

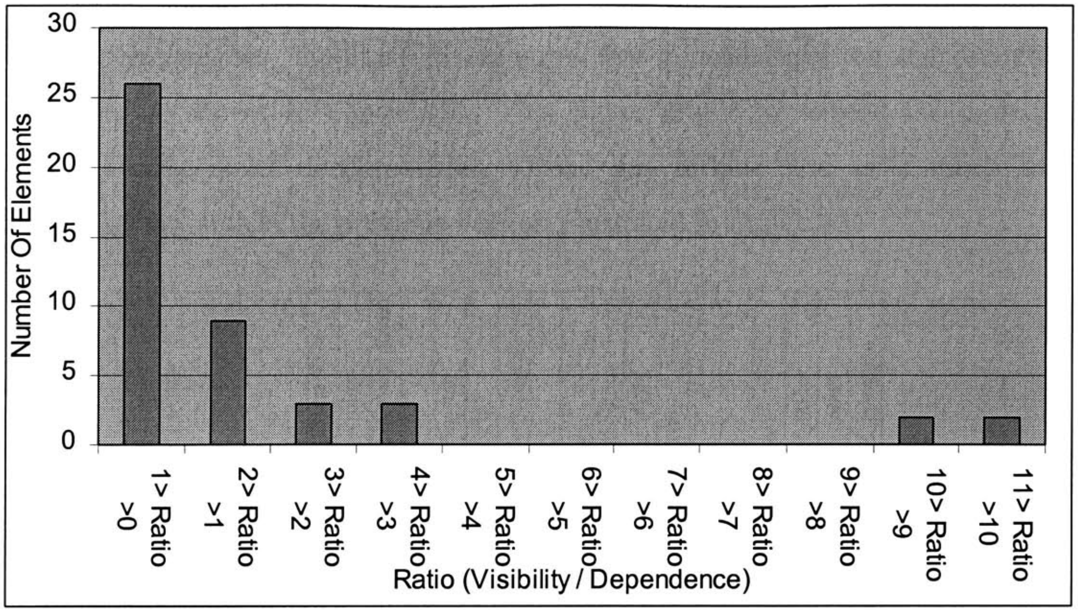


Figure 99: Ratio Analysis Of Genset DSM With Intermediate Stage One Relationships

There is a striking differentiation between the human-imposed unilateral design rules (humanity choosing to do things a particular way) and the nature dependent bilateral design rules. Clearly the system performance will greatly depend on the extent to which the human rules are able to determine the optimal system decomposition by defining the correct inter-element feedback parameters prior to their emergence in full scale testing. Also interesting is the way in which the gas turbine's core of the gas generator and free power turbine[46] have high absolute visibility in an uncommonly unilateral manner for hidden modules. This is of course realised by genset designers who focus almost exclusively on defining the relevant gas generator design parameters in stage one.

In an attempt to sketch the architecture the obvious clusters are manually identified on these DSMs that has the effect of reducing the 45 elements to 18, of which 6 are still primitive singletons. This equates approximately to the second level of decomposition (i.e. 0, 1). These obvious clusters have been deliberately identified without overlaps so as to ease the task of

[46] That the core of this genset includes a free power turbine is an artefact of the configuration chosen. If a direct drive configuration had been selected the free power turbine would not exist and the core would have been simply the gas generator.

computing relationships, and the summed relationships are shown in the second level DSM below. This version shows only direct relationships. By ignoring the singletons it is then possible to sketch out the hierarchy diagram of the gas turbine where circle size indicates the sum of the hidden relationships, and line size and arrow size indicate direction and strength of visible relationships as shown in Figure 101. This figure might be termed the molecular structure of this second level decomposition of the gas turbine and is a much more sophisticated representation than the previous version illustrated in Figure 63.

	1a. Determine scale	1b. Scale rules	1c. Decomposition rules	Core	Compressed air	Comms	Controllers - hardware	IO devices	Bedplate	Electrical output	Air preparation	Gear & Lub	Fuel	Exhaust	Enclosure	Control system	Documentation	3. Test & integrate
1a. Determine scale	24																	
1b. Scale rules	21	11																
1c. Decomposition rules		22	20															2
Core			21	33														8
Compressed air			5	2														
Comms			4	2										1	3	1	1	
Controllers - hardware			4												2	4	1	
IO devices			4												1	5		
Bedplate			4	8						4	4	2		4	2		1	
Electrical output			18	8						16		5	4	2	5	8		
Air preparation			18	17					1		22		4		1	7		6
Gear & Lub			8	19					1	4		5		1		5		4
Fuel			10	16				1	1	1	3	1	5		2	6		4
Exhaust			12	21							2	5	4	12	2	4		2
Enclosure			12	18						11	7	5	6	10	12	10		
Control system			13	28		1	4		2	18	9	12	13	8	12	7		4
Documentation			4	11		1	3		1	7	4	5	4	4	6	8		
3. Test & integrate			18	24	6	5	4	4	3	24	24	12	12	18	18	18	6	12

Figure 100: Reduced Form 18-Element DSM

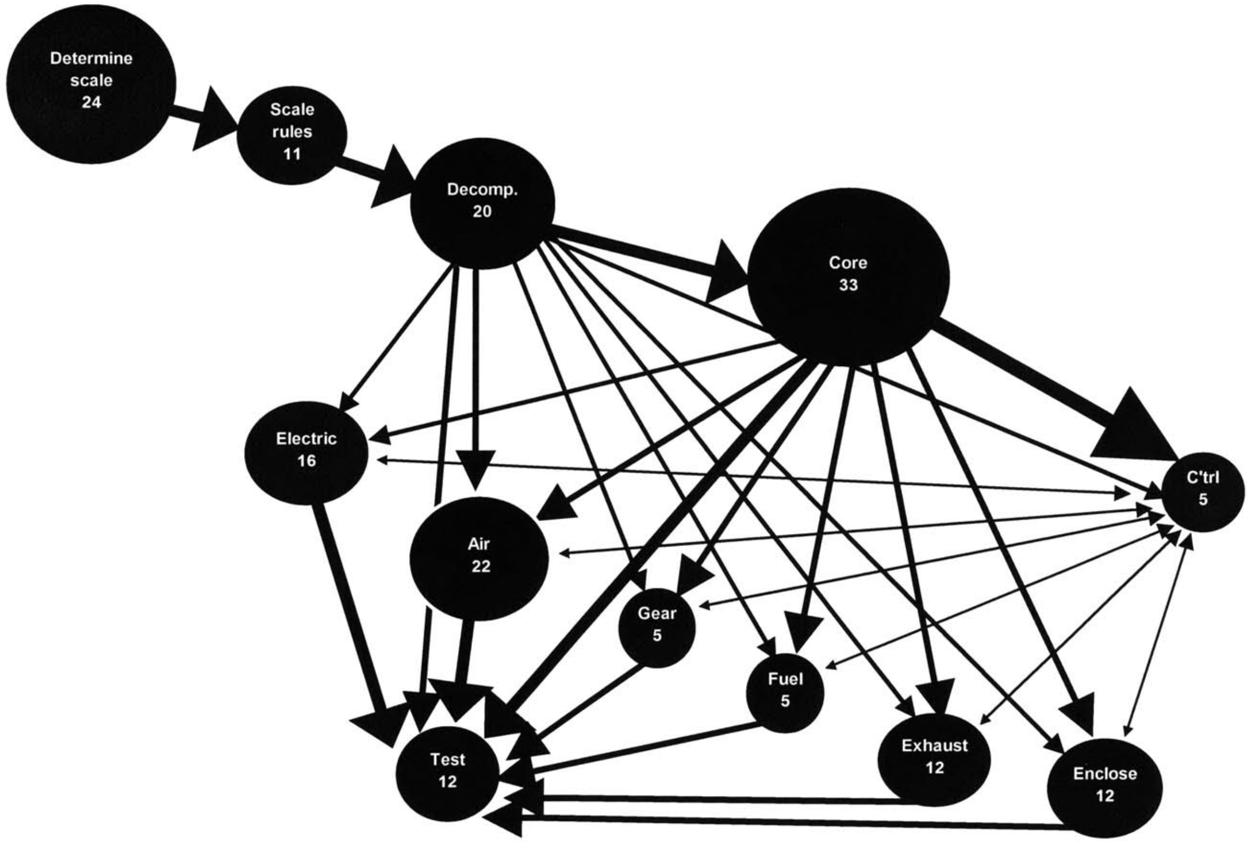


Figure 101: 'Molecular' Hierarchy Diagram Of Reduced Form Gas Turbine Decomposition

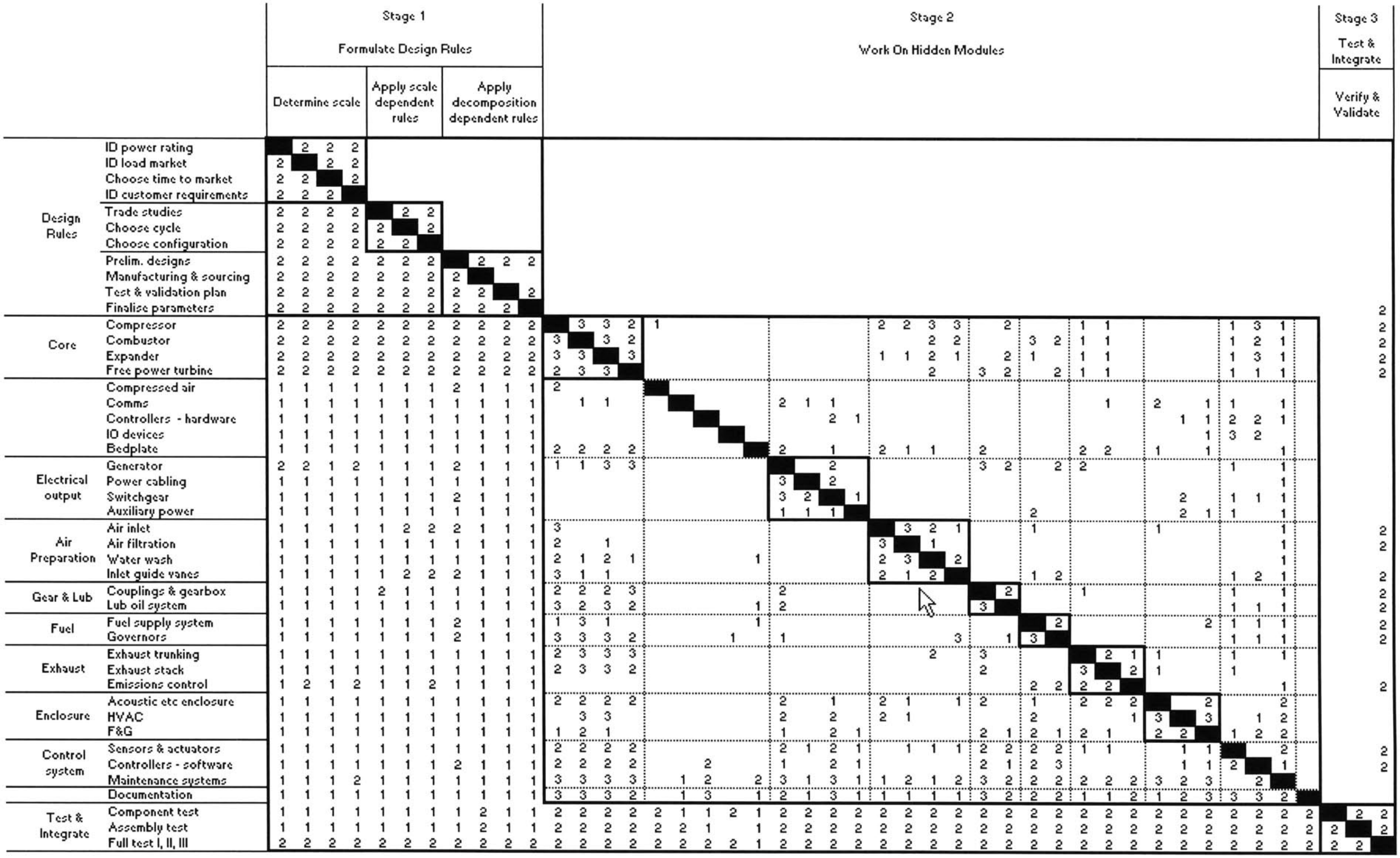


Figure 102: Gas Turbine DSM With Stage One And Three Weighted Intermediate Relationships

12.7 Valuation Arising From Clustering

If a few elements are highly visible to each other and less so to the remainder of the system then it may be worth drafting a set of design rules detailing the interface between this sub-system (module[47]) and the remainder. Until now all the valuations have assumed that this interface occurs at the third level of decomposition, which resulted in a system net option value of 1.87 per Table 8. In order to show how the same type of valuation might be applied at a higher level the clusters identified in the DSM above and the accompanying hierarchical diagram have been valued.

Because of the ambiguity in this DSM the following rules were used to perform this valuation:

12.7.1 External Visibility

The simple sum in a vertical direction of all stage two relationships outwith the module boundary. If less ambiguous relationship markers had been used it would have been better to sum all non-common relationships (i.e. AuB – AnB) as this approach does risk some double counting. The assumption is being made that the module design rules are perfect, i.e. relationships within the module are completely invisible from the outside – this could be amended in a more sophisticated analysis.

This summation is only of directly visible relationships. From a practical perspective this is because if the network is transparent the indirect relationships will saturate the network. From a theoretical perspective this suggests that all the relevant information is contained in the direct relationships and that the indirect relationships are therefore content-less redundant noise.

12.7.2 Standard Deviation

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

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

[47] It is a matter of definition and convention whether these sub-systems are called clusters, modules, meta-modules, or whatever. It largely depends on how high in the decomposition one is – clearly in the genset example the 31 elements are already fairly substantial modules in their own right and so it may be best to use the term meta-module in this particular instance. This is an example of the fractal nature of architectural structures.

This notes that the element standard deviation already in use is weighted to account for technical potential. Since all the means are zero this typically results in a greater upside for experiments.

12.7.3 Module Size

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

> ***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 either be used as it stands with a minimum value set as the sum of the elements, or an alternative and more fundamental formulation introduced.

12.7.4 Coefficients

The same co-efficients are used as in the lower level calculation – in this case the same as in the 31-element calculation which resulted in a system NOV of 1.87. This has the effect of allowing like for like comparison.

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 (or fully modularised – depending on one's point of view) system and since the underlying system remains constant at 31 elements this should not change.

12.7.5 Result

The results of this arrangement of elements are shown in the curves and summary table below.

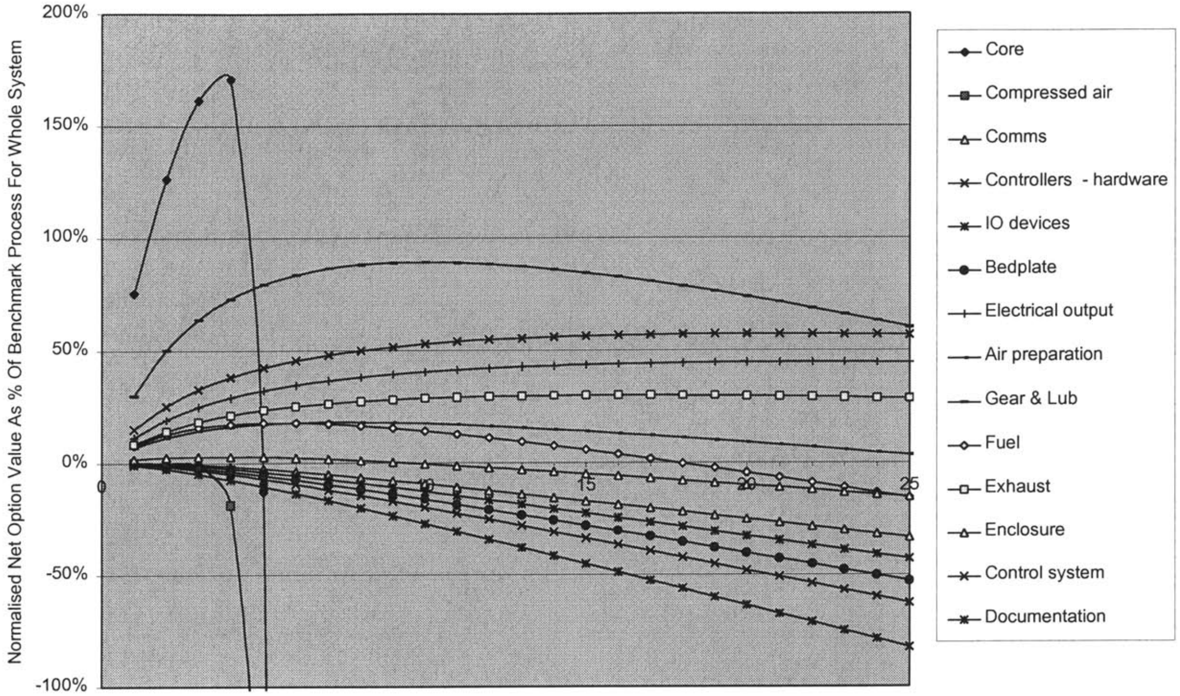


Figure 103: Gas Turbine Option Value Curves For Intermediate Decomposition

Table 10: Gas Turbine Option Value For Intermediate Decomposition

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 (from this decomposition)			**4.31**

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 on the basis of the work described in section 11.3.3.2 has hidden sufficient detail that the increased costs of the larger modules are outweighed by the increased benefits. Of course not all cluster combinations will be equally effective and some may even destroy value, and 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. Note that such an algorithm not only has to specify the sequence but also has to specify the boundary locations, and that in principle this need not be restricted to a flat DSM but could be extended to allow tunnelling in another dimension. Furthermore there is no fundamental reason why the algorithm could not yield a result that allows overlapping (pinned) modules, or must disregard the wrapping opportunities afforded by the torus shape of the DSM. It is only for convenience that I confine myself to discussing planar (non-tunnelling) clusters as the only viable form of module, however it is reasonable to ask the question as to how likely people are to be able to understand and act on any more complex analyses.

This shows how it has been possible to relate the organisation of a relatively integrated product or system's architecture to its expectation value on the basis of the fundamental principles of option values and design rules. Whilst some simplifications and approximations have been made the principles of valuation as suggested by Baldwin & Clark have been respected and extended to an integrated architecture without resorting to heuristics.

It is interesting that this valuation did not use the concept of dependence. This is because it was a societal valuation and dependence only comes into play when considering entity valuations in a multi-player situation as will be discussed in the next chapter.

12.8 Multi-Domain Optimisation

As shown earlier in this chapter and as Figure 97 illustrates 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, 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. Thus 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 the diagonal.

12.8.1 Assumptions

In the task domain the system is best considered to be a series of feedback loops with probabilities and consequences of rework. The order (sequence) in which the elements are arranged determines the network's response. This is a non-linear effect.

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.

There is information in the physical domain that assists in clustering of elements into modules. There is information in the task domain that assists in placing elements in an efficient sequence. Perhaps there is information in other domains (such as the organisational) that assist in other ways (such as for siting personnel).

The domains do not have to be exactly the same. Indeed if they are, one should simply run all the objective functions on the same data set.

Arranging tasks below the diagonal optimises the task domain. Depending on circumstances this may be with an 'as soon as possible' (ASAP) sequence or with an 'as late as possible' (ALAP) sequence. Above diagonal marks indicate the possibility of feedback occurring, either from one generation of design to another or within the same generation in case of serious errors in estimating the initial design point.

Arranging the relationships as far as possible within a cluster boundary optimises the physical domain. In the simplistic example shown the cluster boundary is drawn around any diagonally-centred group with a ~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 that 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).

12.8.2 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 below represents a pretty good compromise. This is relatively harmonious as the B:C module can now independent of the D:G module provided the B>G relationship is respected[48]. 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 to 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. Obtaining exactly this compromise would require the two objective functions to be correctly balanced, as over-emphasising one domain is the same as ignoring the information in the other domain.

[48] 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 the classic 2 x 2 open x proprietary standards grid.

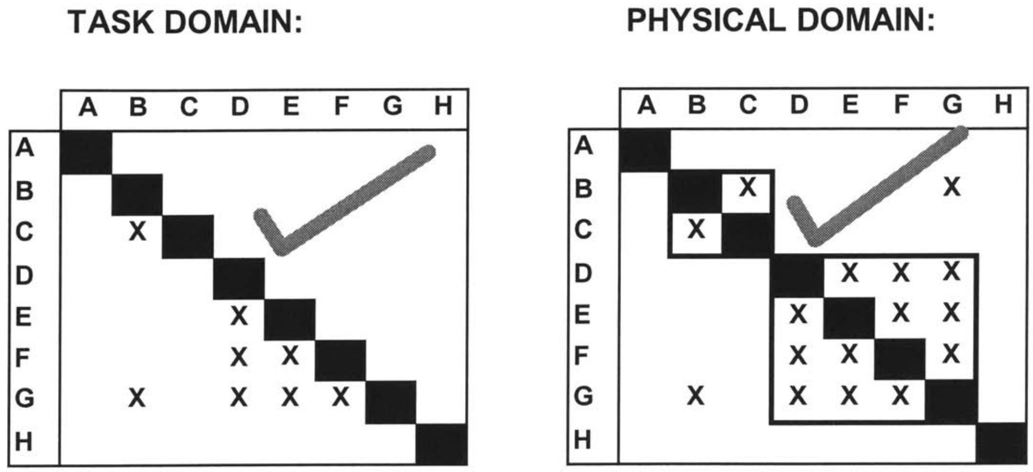


Figure 104: Good Optimisation Of Two Domains

For example 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 microphone & speaker impedances, operating voltages and currents, and pin/socket configurations. In the physical domain B<>G represents the actual electrical and mechanical mates.

However if the information contained in the physical domain had not been known then efficient sequencing of the task domain by an ASAP algorithm could easily have yielded something like this:

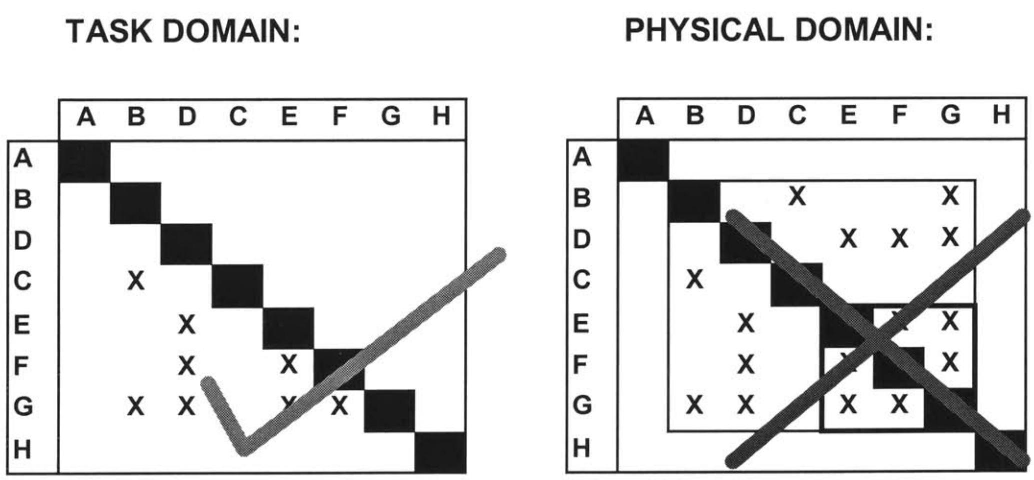


Figure 105: Poor Optimisation From Concentrating On Task Domain

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, perhaps representing the power supply or something, 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 this:

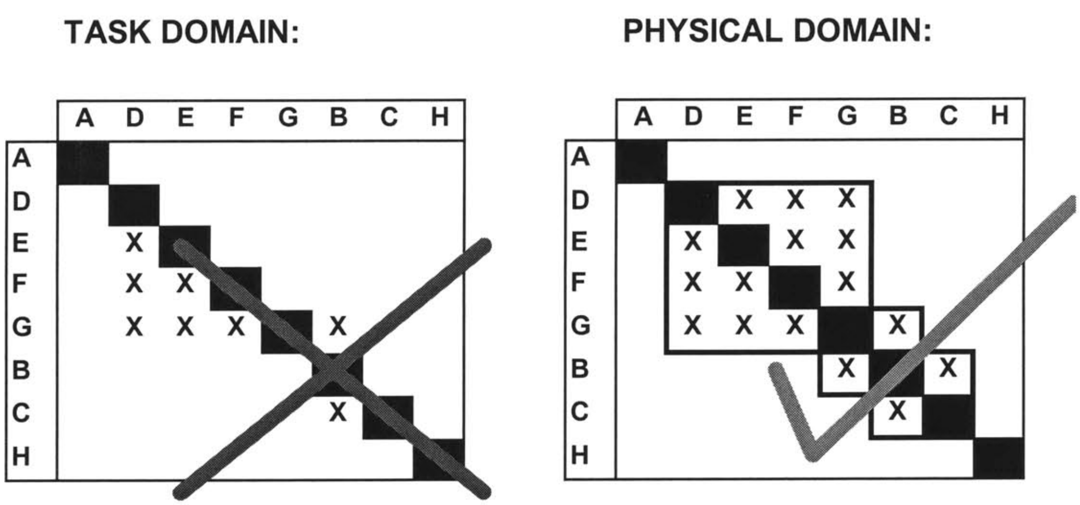


Figure 106: Poor Optimisation From Concentrating On Physical Domain

Whilst this might be thought to be good for the physical product it causes an iterative design loop around G>B. In the example of the telephone design 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.

These issues do not pose a serious 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, within design such as between 'engineering' and 'styling'). It is in these more serious situations that blind evolution is most undesirable.

Application of all the different objective functions to single-domain data may qualitatively suggest areas of tension. Ideally one should have all relationships mapped in all domains and individual domain optimisations identified, then a global optimum selected. 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 improvement.

As other domains are added in – especially the organisational and individual, perhaps also the functional and/or parameter, each with their own objective function and rules on required sequence relationships, the overall effect is to create a highly complex value surface. This is what we know to be the case in real life so directionally it is probably sound to research this more[49].

12.8.3 Outline Multi-Domain Societal Value Objective Function

If multi-domain optimisation is a useful concept then the objective function will be of the form:

$$\textbf{\textit{Max.}}\ \alpha_1 \textbf{\textit{. (Value task domain)}} + \alpha_2 \textbf{\textit{. (Value physical domain)}} + \ldots + \alpha_{j.} \textbf{\textit{(Value j'th domain)}}$$

Where α_j etc. represent the co-efficients of the j'th domain's contribution to value, and $\alpha_1 + \alpha_2 \ldots + \alpha_j = 1$. At present all objective functions set all α_j to 0 except for the domain they are being applied to which is set to 1.

At present the societal value of the physical domain is maximised by objective functions of the form of Eq. 11.1 and 11.2 which can be crudely applied using the same series of assumptions as is used to develop Table 10. To the best of our knowledge the societal value of the task domain is

[49] My consideration of this was based on the intuition that the physical domain is affected by off-diagonal distance, and by borders as these are what everybody immediately comments about or draws onto any DSM they see. This is what led me to think in terms of networks etc. and thence to graph theory (via Dan Whitney), although I didn't realise at the time that the effects might be taking place in different domains.

maximised by activity networks with uncertain levels of rework (Browning, 1998; Browning & Eppinger, 2000; Eppinger et al, 1997). In isolation optimising the task domain will primarily be an issue of permitting better partitioned modules to evolve at a faster cycle time than poorly partitioned systems – this obviously allows for more rapid encounters with sources of novelty as a result of the higher intensity learning experience. As yet the objective functions for the organisational domain are less clear but those based around the various proximity measures, e.g. the inverse inter-person distance squared rule, seem most appropriate although the foundation of this rule (Allen, 1977) predates widespread use of modern high-bandwidth electronic communications to enable dispersed and decentralised teams. The combination of the spatial dimension of the organisational domain with the task domain will need testing against the predictions of technical communication in organisations (Morelli, 1993; Morelli et al, 1995). Other proximity measures are discussed in Fine et al. (1995). Similarly the combination of the individual domain with the organisational domain may be of use in assessing efficient allocation of human capital.

The search strategy to implement an objective function of this nature is not discussed here, however it need not restrict itself to searching for planar modular sequence clusters, nor for exclusive (i.e. unpinned) modules. The latter should be present (or not) as a natural outcome of their value, although they are more likely to be present if non-planar (i.e. tunnelling) modules are permitted. The strongest arguments for ruling out non-planar modules is that a) they may be more difficult to implement and b) they may not be understood by most users and so it may be counter-productive to try and explain them to decision makers.

This suggested objective function only describes societal value. As has been hinted at, this is often different than the value to a firm and this is something that is discussed in the next chapter.

13 EVOLUTIONARY DYNAMICS AND THE IMPLICATIONS FOR THE FIRM

This chapter discusses the difference between creating value and capturing value and contrasts societal and actor value. It is suggested that visibility determines societal value created whereas the balance between visibility and dependency determines the ability to defend and/or capture value. The way in which multiple actors can compress S-curves is discussed, as is the potential for one product to contain modules or elements with different cycle times. This leads to a discussion of the different sorts of discontinuities that may occur and what characterises them. These discontinuities are then compared with the notion that firm's need dynamic capabilities in order to make successful transitions, and then it is suggested that these discontinuities map fairly well to proposed types of boundaries (e.g. semantics, syntactic, etc.) and boundary processes (e.g. boundary objects, (Carlile, 2001)). This allows for a discussion on bridging methodologies of which the notion of Net Option Value is an example.

13.1 Single Actor S-Curves

It is widely accepted that product lifecycles follows an evolutionary 'S' curve because of a process of innovation and diffusion as described by Porter, 1980; and more specifically this is applicable to technologically-driven products as a technology adoption lifecycle (Fine, 1998; Moore, 1999; Christensen, 1997; etc.). Although not directly mentioned in Baldwin & Clark, 2000 this is consistent with their proposals as is shown in the figure below, which is a detail of Figure 7 shown before. In the version of the figure shown below, the value of performing experiments on the large / hidden module is selected for consideration and the cumulative value of experimentation is also plotted: this latter reveals a S-curve. This particular module's value is maximised by 3 experiments, and performing more than 7 experiments begins to destroy value.

What is especially interesting about Baldwin & Clarks' work is that they propose a plausible and quantifiable causal mechanism that explains the more epidemiological observations and qualitative mechanisms of other researchers in this field. This is in stark contrast to the classic text on analysing competitive strategy using five-force analysis (Porter, 1980), which only quotes one quantifiable measure, that of the rate of sustainable growth of a firm (p.66, 1998 Free Press edition).

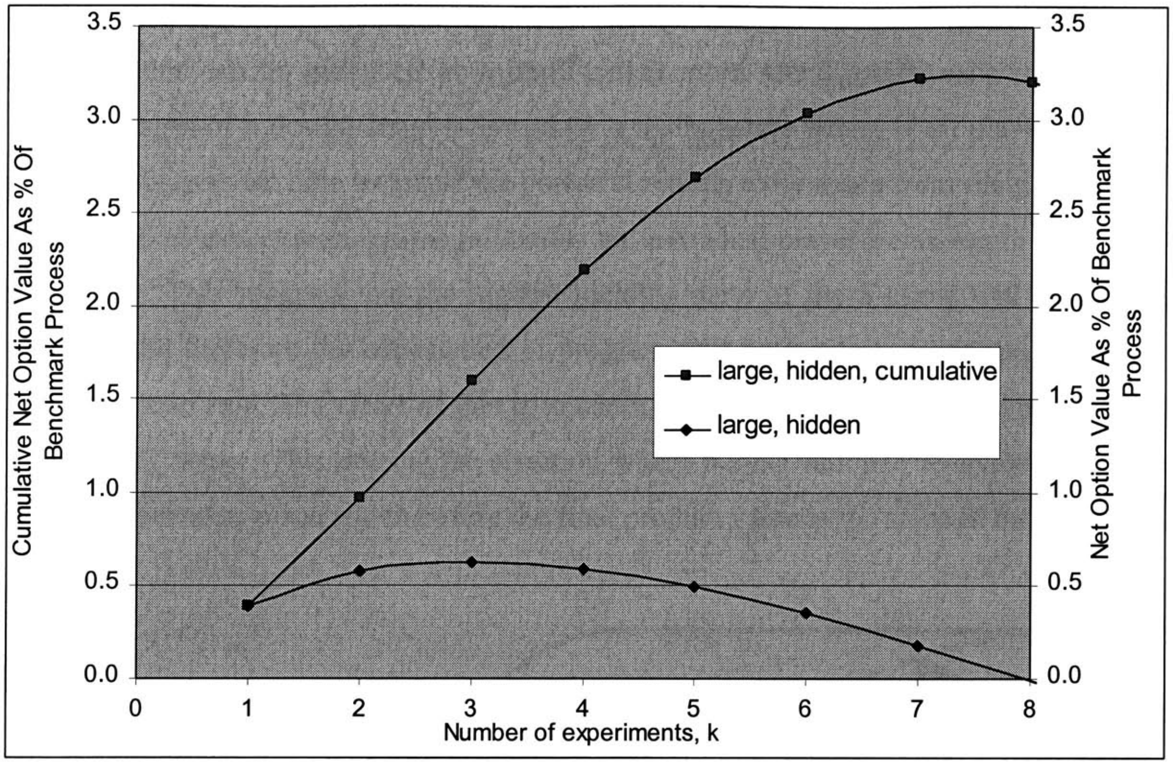


Figure 107: 'S' Curve Resulting From Experimentation On A Hidden Module

The horizontal axis of this figure is the number of experiments, k, and so for this axis to directly correspond to time only one actor would invest serially in subsequent generations of the product, each generation of which took the same time to produce.

13.2 Multi-Actor S-Curves

Accepting the proposition that inter-generational times within a given industry remain approximately constant, as suggested by the work of Fine (1998) and Christensen (1997)[50], then the obvious other question is how many actors will participate. Qualitatively the answer is that

[50] Much of the research in this area has been performed on highly modular electronic & software products. In these industries there are other endogenous factors that cause companies to approximately march in lockstep. It may be that this correlation breaks down as products become more integrated, e.g. the Japanese automobile manufacturers appear to have consistently maintained a different and changing inter-generational time period than the rest of the world (Womack, Jones, Roos, 1990). In such industries it may be that the number of actors is endogenously fixed (at least historically speaking) and therefore the free term in the equation changes from actors to inter-generation time. Whether this is so has obvious implications for those automobile companies that have expressed a desire (not always genuine) to move towards modular / platform products.

the more attractive the investment the more actors will participate. Depending on various assumptions this can be modelled as a multi-player multi-round game, each round of which normally consists of a design stage followed by a manufacture stage. If all players are rational then as more players become aware of the potential returns to be made from the game then more will enter, and as expectation returns fall below an individual player's risk/return threshold then they will exit. This suggests that the highest yielding parts of the S-curve will have the most participants and therefore the expectation of progress will be greatest, thus compressing the S-curve on the time axis. The effect of this is to accentuate the S-curve's 'S-ness' as illustrated in the Figure 108 below. This depicts the situation where perfect rational behaviour occurs as no over-investment takes place and therefore the final product value is the same in both cases.

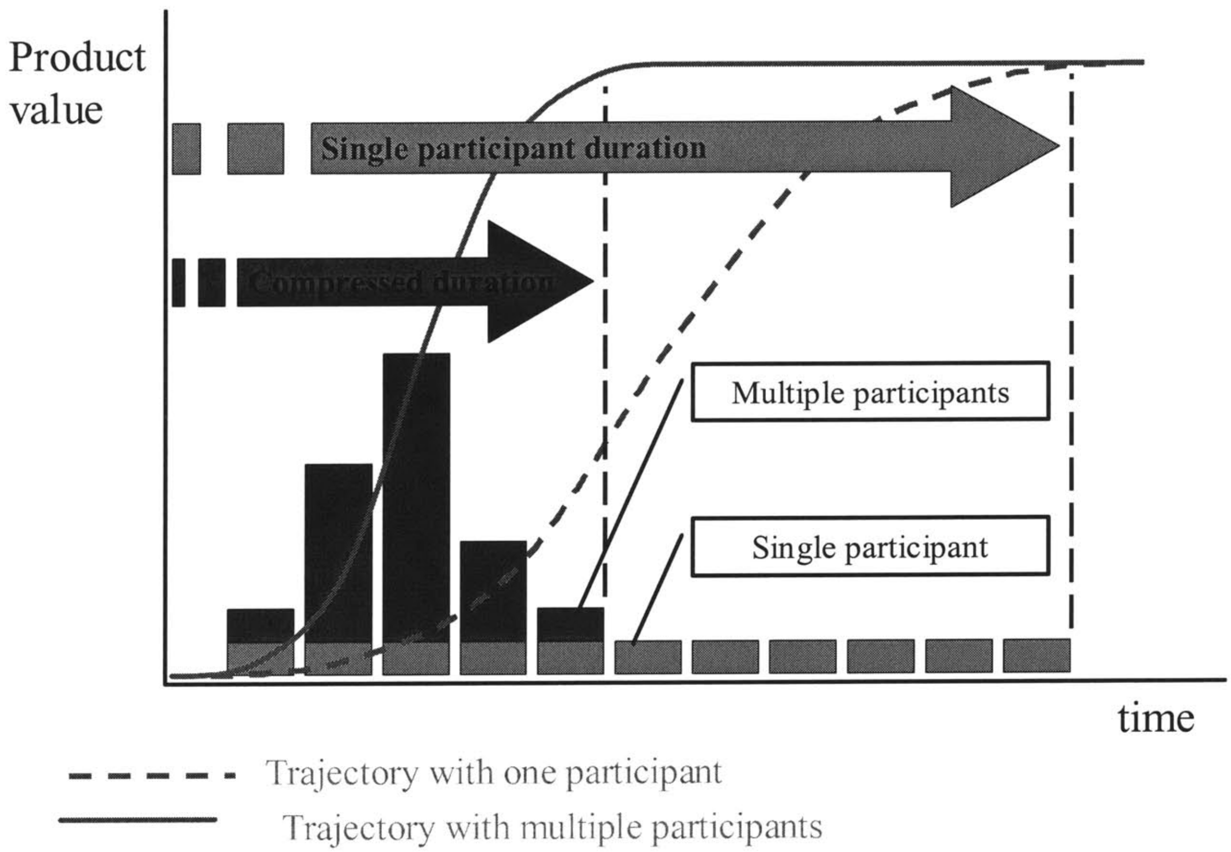


Figure 108: Effect Of Multiple Participants On The S-Curve

Modularisation affects this development trajectory by permitting more rapid progress at the product level. This is because tasks in modules become generally more hidden than the corresponding task in the integrated architecture and therefore greater rates of change are supportable (c.f. Figure 6 for a comparison of hidden and visible modules' progress rates).

However this can only be achieved once the product is well enough understood that the design rules can be formulated, and premature modularisation prior to this point will normally require rework, as inappropriate partitioning of the product will often have occurred. Provided that there are no system level effects, i.e. a modular system has the same maximum performance as an integrated system; this will once again result in the trajectories of the integrated and modular designs plateauing at the same level. This situation is illustrated in the figure below from which it can be seen that there is an optimal transition point. At the transition point the rate of technological process will generally accelerate as more players are enabled to enter the system, and this will further accentuate the overall 'S-ness' shape of the S-curve in an asymmetric manner. To an observer this change may appear as a discontinuity and, because of gaming, it may be worth some players trying to force modularisation onto a product even if it is premature whilst others would wish to delay this transition: these motives would typically correspond to new entrants (especially from adjacent product spaces) and to incumbents respectively. This transition point may correspond to known salient points on the technology adoption curve but further work is required to investigate this.

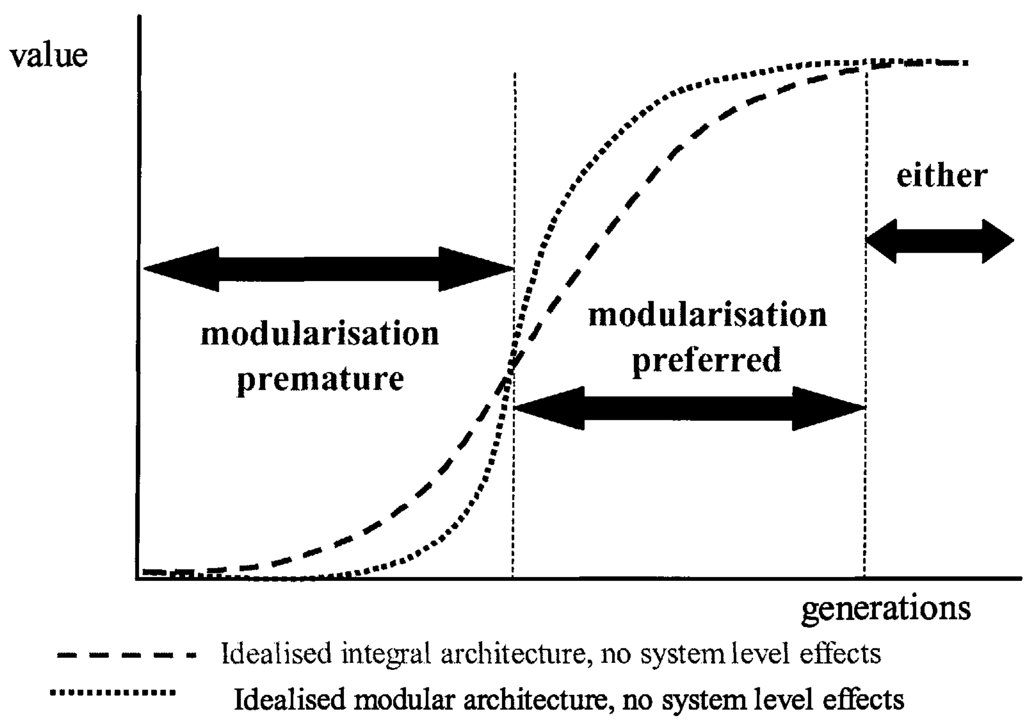


Figure 109: Effect Of Modularisation On The S-Curve

It can be suggested that system level effects cause modularisation to be undesirable in mature products. This situation can arise once the potential inherent in each module has been discovered and exploited because, at this point, the inefficiencies inherent in the design rules are the only remaining source of evolution within that paradigm. This situation is illustrated in the figure below where the architecture re-integrates towards the end of the technology cycle in order to overcome the system level penalty created by the design rules. This would take place at the point where the expectation value to be gained from the next generation of modular development is lower than that to be gained by integration.

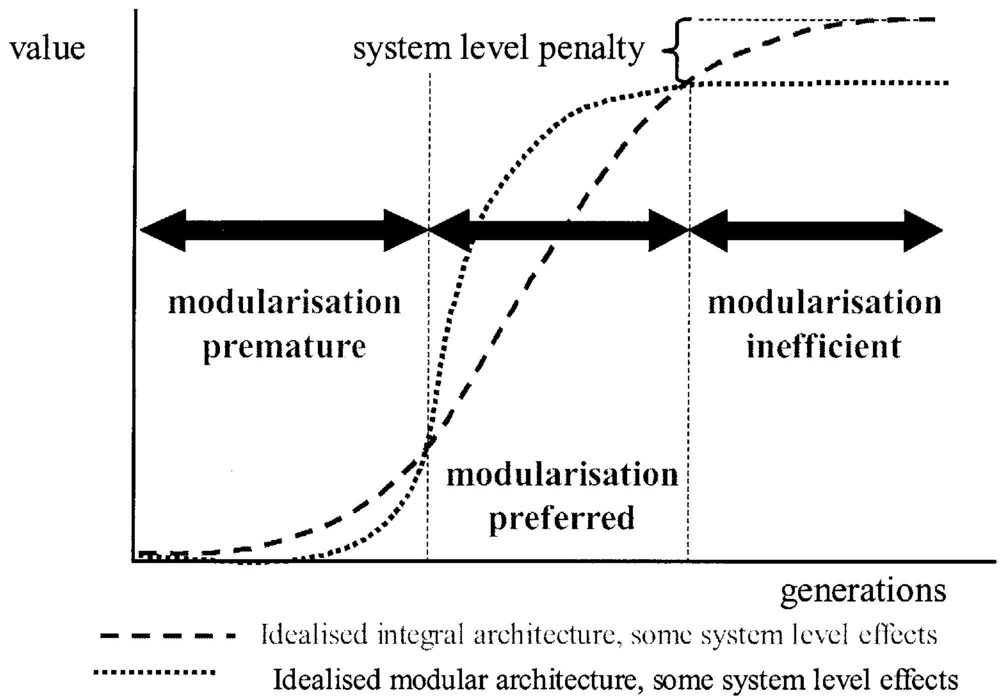


Figure 110: Effect Of Re-Integration On The S-Curve

Once again this point of reintegration will be subject to contention where some competing players may wish to force an end-game shakeout by introducing an integrated product just ahead of their competitors, whilst others would prefer to postpone transition. The greater the degree of contention on this the more difficult it will be to observe these transitions in practice. As before this change may appear as a discontinuity to a participant, especially a losing one.

If the entire S-curve, from an immature integrated architecture to a mature integrated architecture via a modular one is considered to be one technology cycle, these intra-technology discontinuities will tend to be associated with a restructuring of an industry in the manner suggested by Fine (1998). This suggests that the cycle of industrial restructuring that Fine discusses will not be a smooth process but will be more akin to the theories of punctuated equilibrium that evolutionary biologists now consider to be the norm (Eldredge & Gould, 1972). As the industry more rapidly restructures from a horizontal to a vertical integration across these discontinuities the number of participants in a given product will tend to change in a more abrupt manner across the intra-technology discontinuity than at other intra-cycle product generations, however once again this effect may well be masked by some participants forcing the pace.

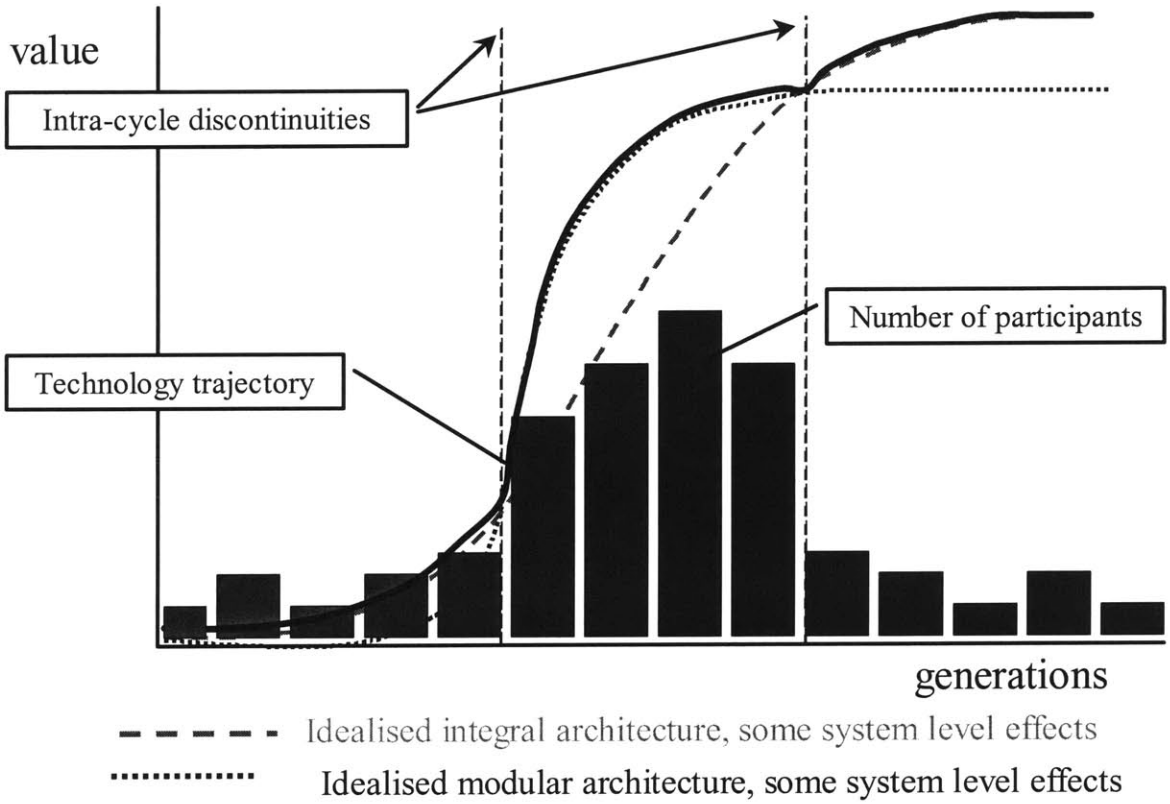


Figure 111: Effect Of Intra-Cycle Discontinuities On The Number Of Participants

13.3 Multiple Technology Cycles

These potential intra-cycle discontinuities need to be carefully watched for if conducting any epidemiological studies of industries in an attempt to observe S-curves and measure industry clockspeeds[51] as the intra-cycle transitions may be confused with the inter-cycle transitions. The inter-cycle transitions occur when a truly novel technology burst onto the scene and both forces and enables even more rapid restructuring. A number of researchers state they have observed these inter-cycle transitions but it is possible that they may occasionally be mistaken; – e.g. Christensen's case study of disc drives would appear to be examples of successive generations of products with a common architecture; the transitions from automobile coachwork firms to integrated manufacturers is an example of a intra-cycle transition; whilst Christensen's case study of the transition from wire rope dragline excavators to hydraulic excavators is a inter-cycle transition. The definition of the inter-cycle transition is the widespread adoption of a disruptive technology (Fine, 1998; Moore 1995; Christensen 1997) and this is generally described as being an interlocking ladder of S-curves (Christensen, 1997; Henderson MIT lectures, 2001). A puzzling feature of these disruptive events is the extent to which the dominant player in one technology cycle is so seldom dominant in the next, or even a mere competent participant. A variety of mechanisms have been proposed and this chapter and this chapter introduces another suggestion. This failure to transition is especially odd when one considers that the dominant players do seem able to cope with intra-cycle disruptions; indeed it is not uncommon to see dominant players adopting successful transition-forcing strategies to shakeout sub-scale competitors, or deliberately electing to use their greater resources to enforce primacy through early follower strategies against weaker transition leaders (and in the interim harvesting additional rent).

Two interlocking S-curves are depicted in Figure 112 below that illustrate this point. It appears relatively common that successful transitions are made by firms across the generational discontinuities of points 1), less common across 2) which represent the intra-cycle transitions in either technology cycle, and least common that a successful transition is made across point 3) which represents the inter-technology cycle discontinuity.

[51] An industry's clockspeed is defined by Fine, 1998 as the time taken for an industry to transition from one point all the way around the cycle and back to the origin. In industries that progress at a constant inter-generational period it is suggested that the clockspeed will also remain approximately constant.

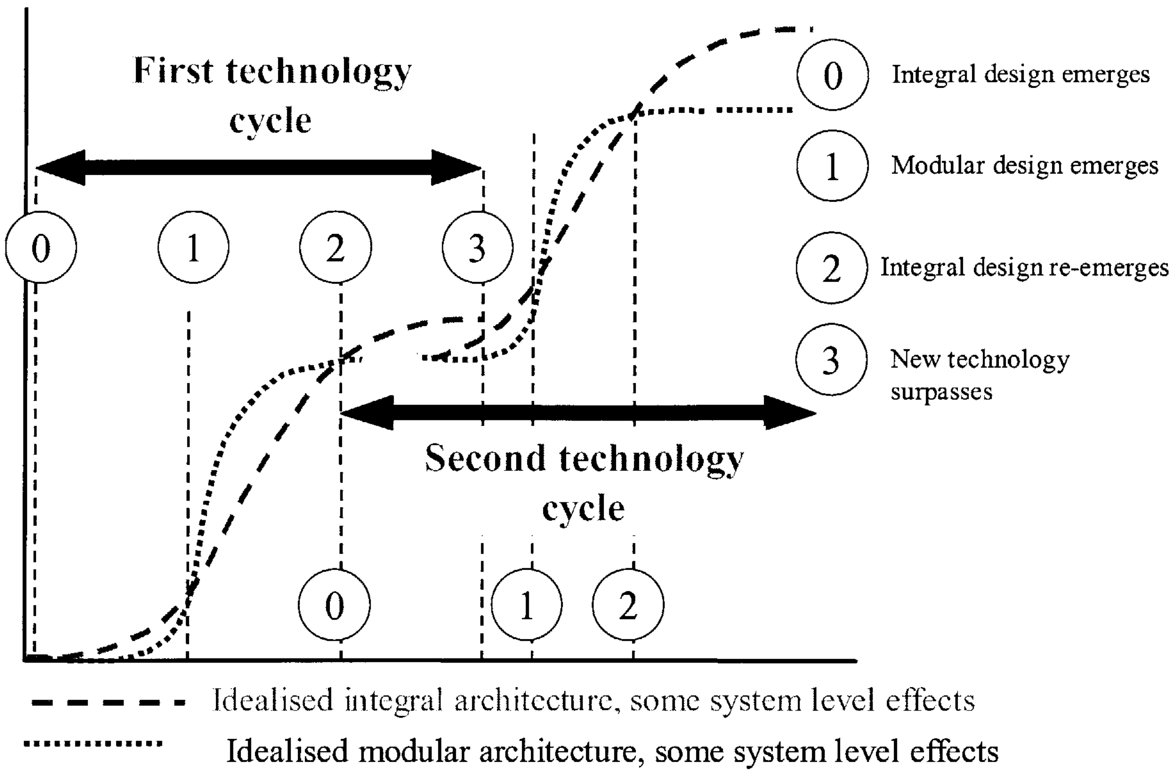


Figure 112: Interlocking S-Curves And Inter-Cycle Discontinuities

Before examining this puzzling question in detail it is worth considering both the case of the semi-integrated product and more details regarding what happens to players during modularisation.

13.4 Qualitative Discussion Of Effect Of Modularisation On Capturing Value

In the case of a simple integral product (i.e. composed of one module) with single stage investment decisions per generation[52], it's developmental trajectory can be generalised by the following diagram which illustrates the situation where the expectation is that a 5-generation game will fully exploit a particular technology; where the first generation has resulted in a product value of X_1 which forms the base level for several players to compete at improving on in

[52] A single stage decision reflects deciding to invest in manufacturing capability concurrent with commencing design, or where manufacturing capability is immaterial (e.g. in software). A two-stage decision per generation reflects the situation where the design output is known before the manufacturing investment decision is made. For a good introduction to game theory see (Gibbons, 1958, 1992).

the second generation; where a normal distribution is centred around X_1 with (for the purpose of this example) four positive results (players 'a', 'b', 'c', and 'd') and two negative results (players 'e' and 'f'). The two negative results will presumably be culled and the best result ('a') will form the base level X_2 for the third generation game. However whilst this diagram discusses the value creation process it does not say much regarding the value capture process. Participants 'a', 'b', 'c', and 'd' will each have an opportunity to capture value during the period of time it takes for the next generation of products to be developed. Depending on the extent to which they are able to execute perfectly they will maximise the potential inherent in their version of product by riding the market's price & performance elasticity curves to optimise their individual profits and/or deny profits to their competitors. In this situation if all other factors are equal and if all actors are rational and do not collude then 'a' will capture more value than 'b', and 'b' more value than 'c', etc[53]. The extent to which any or all of these capture a surplus (or whether it is transferred to the user) and what the absolute value actually is depends on whether it is in fact a one or multi-stage game, on the nature of that game, on the quality of execution, and the market's elasticity curves. This is a subject that Baldwin & Clark, 2000 outline using the four extreme cases possible in a two-stage game as an illustrative example. In this thesis I am more interested in what happens when a product modularises and the relative attractiveness of particular modules.

[53] As a complicating factor the products from the previous generation may be able to participate as well.

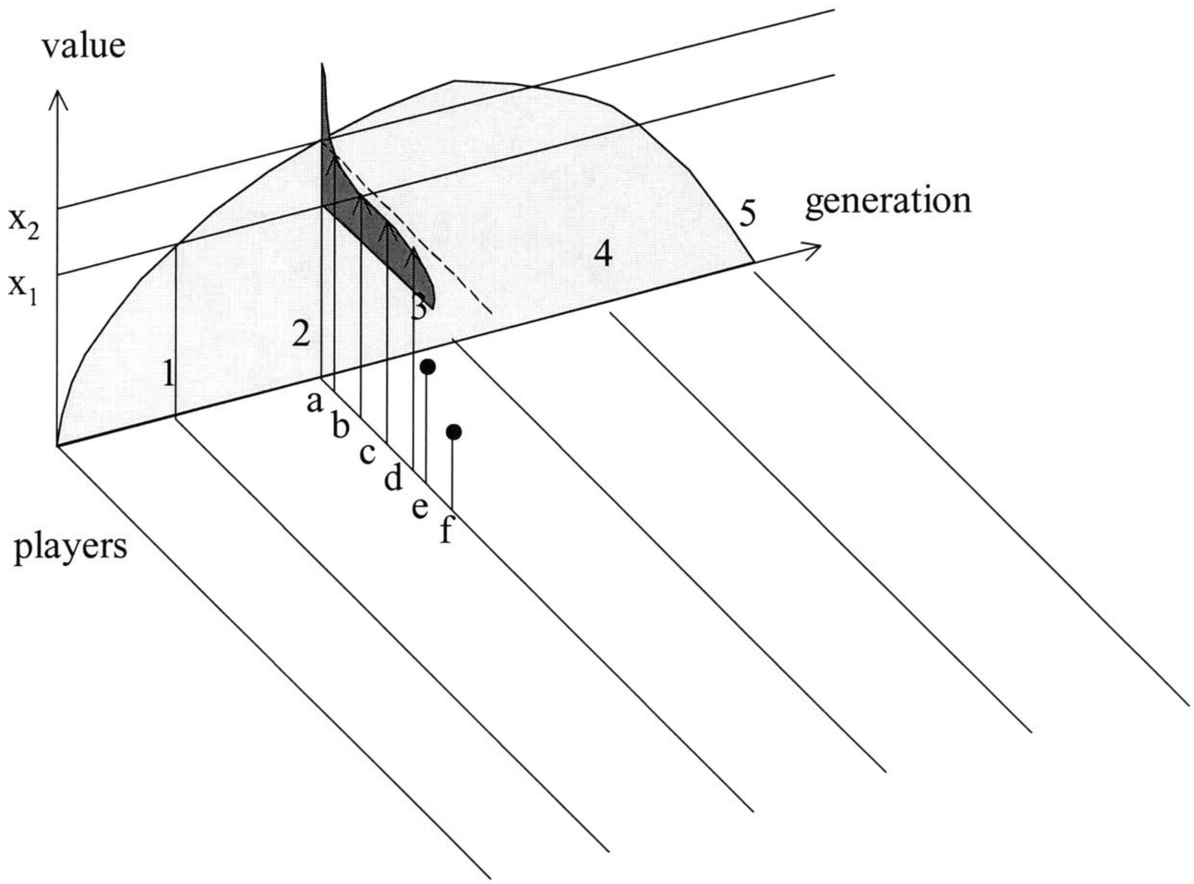


Figure 113: Outcome Of A Generation In A Single Module Multi-Player Game

Apart from the tactical execution problem of capturing value from the current (second) generation, actors will be making decisions about investing in the next (third) generation of designs. These may be incumbent participants (a-d), incumbent non-participants (e-f), or new entrants. If actors are rational then they will choose depending on their individual expectation of capturing value in the next round and the risk associated with attempting to do so. Depending on how the design rules of this integrated product are structured, and the nature of the underlying technology, participants may start from a level position or may be handicapped by their position at the end of the last round. This handicapping represents a path-dependency and might be caused by e.g. intellectual property gained in the last round, but should not be affected by the financial success of the last round except inasmuch as the rate for market finance will generally be risk-adjusted to reflect the probability of (predominantly technical) success – something that is often based on historical performance. In an efficient market the continual entrance and exit of players will tend to depress the expectation rate of return to the background level for risks of that level. In this respect, if all other factors are equal, larger products are riskier than smaller ones, as the stake is higher because of the greater design effort required, and so the reduced expectation will create a corresponding reduction in the number of participants. This is one reason why, if handicapping occurs, successful actors in one stage will seek to enlarge the market's conception of the normal size of the product as it asymmetrically increases the risk for new entrants and thereby improves the incumbent's expectation return[54] by decreasing the expectation competition.

As products increase in size, and as the underlying design rules become better understood, a position is reached where it becomes attractive to split the integral product into one or more modules. Following on from the argument above and by considering the situation described in the previous Figure 7 (reproduced below) if all other factors are equal larger modules will tend to attract more competition, and the opportunities for cost-effective improvement will be exhausted earlier.

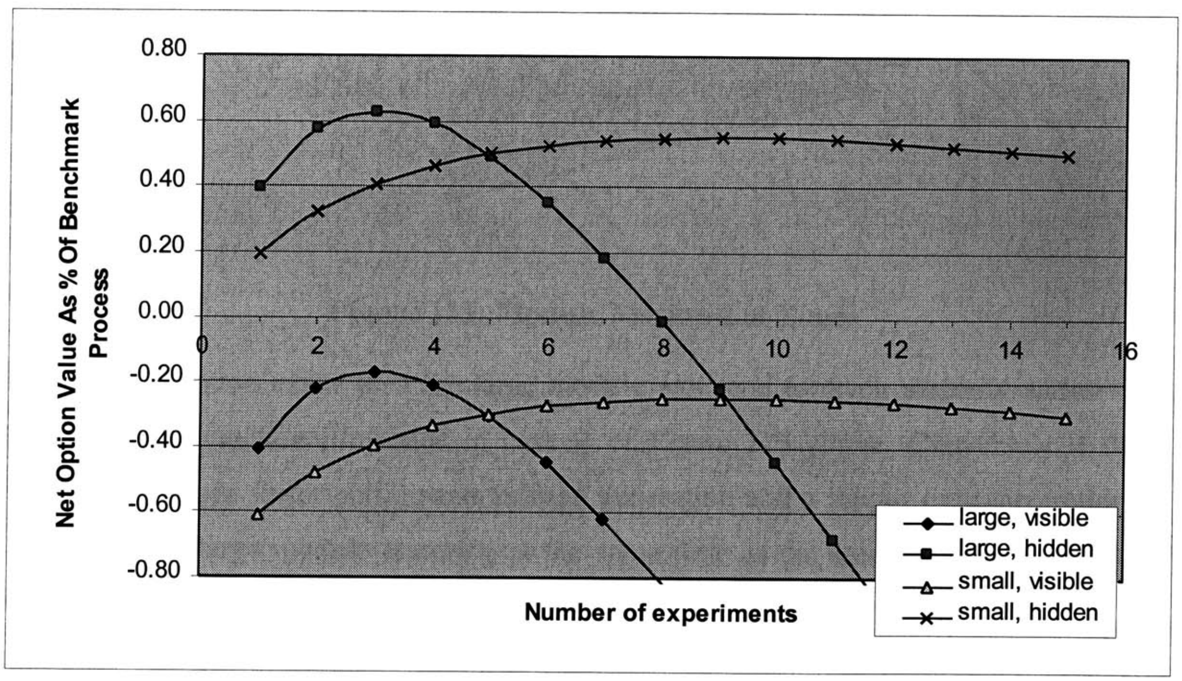


Figure 114: Net Option Value For Different Combinations Of Size And Visibility (rpt.)

[54] This is one explanation why e.g. software packages bloat. There are other reasons, some of which will be discussed.

Market inefficiencies (e.g. a lack of technically capable participants in large module design) and handicapping will tend to be more material in larger modules, however it qualitatively likely that the larger modules will take longer to design, i.e. have a longer inter-generational period.

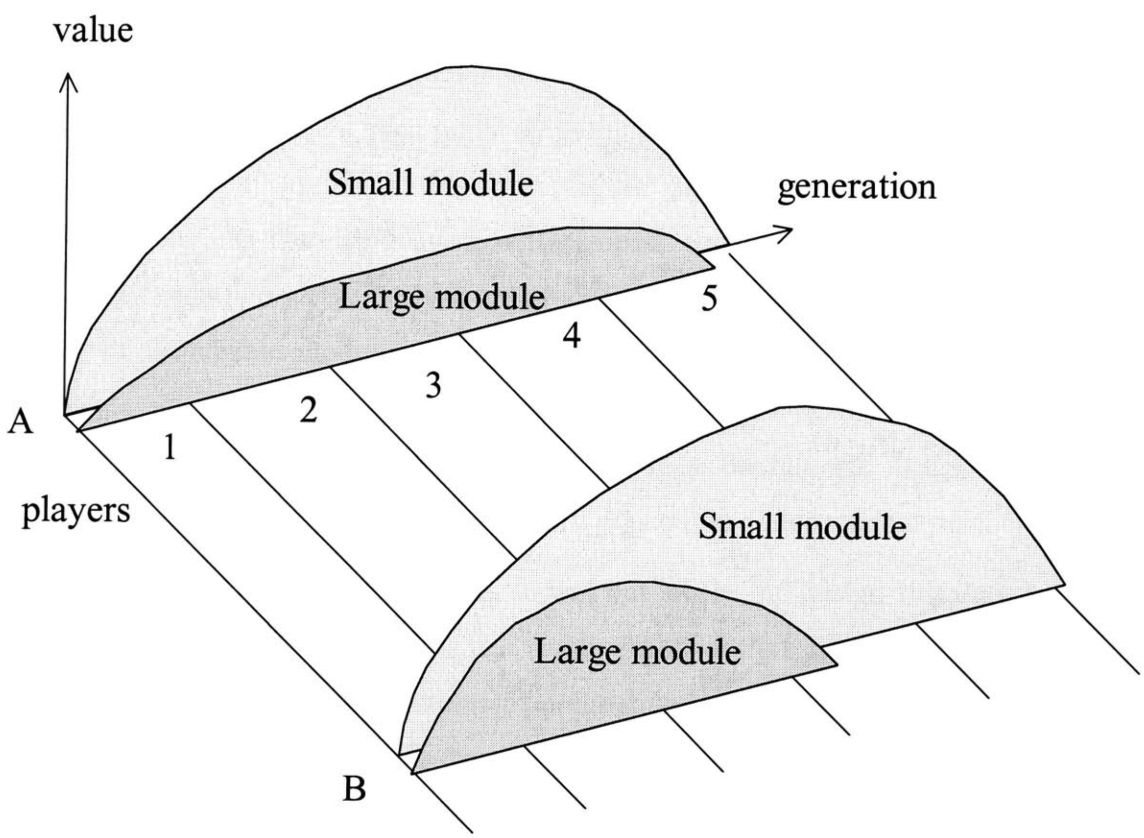


Figure 115: Module Maturation Times

If the former situation exists then the large module will tend to reach maturity earlier than the small module, a situation illustrated in case B of Figure 115 above. Conversely if the latter situations exists then fewer participants in each generation and a longer inter-generation period will result in the large module maturing at the same time as the smaller module, or later. The expectation cumulative improvement will be the same in both cases.

13.5 Why Invest In Highly Visible Modules ?

The other thing that Figure 114 shows is that the more visibility a module has, the lower the incentive to invest. Irrespective of the nature of the game there is no surplus value available for capture if the NOV of an experiment is negative. So in this situation why would any players participate in improving a highly visible module, or why might the system's design rules be set up to create such an unattractive module in the first place?

Let us first consider the situation where a system comprises just two modules with approximately equal maturation times, and of approximately equal sizes but one is highly visible and the other is relatively hidden, i.e. the hidden module is unilaterally dependent on the visible module. If the design rules of the system have been perfectly delineated prior to the zero'eth generation (i.e. if there is no a priori handicapping arising from players' participation in the integrated portion of the system's evolution) then all players will compete equally in the first round. Let us say arbitrarily[55] that only two viable designs are produced for the visible module and that no further generations of the visible module are economic. Again arbitrarily let us assume that several viable designs are produced for the hidden module but that a number of further rounds (generations) are economically attractive. In this situation because the design rules have been perfectly delineated these two modules games will play out in a independent manner.

Now let us consider the same situation as above, but where the design rules are not perfectly delineated. In this case the owners of the two viable visible designs have a competitive advantage in the second and subsequent generations of competition for the hidden module's value. This advantage can take the form of a time-based advantage in learning the new design rules for the subsequent rounds, or can even take the form of deliberate manipulation of the design rules to prefer the owner's proposed solution over competitors' solutions. In this situation the owners of the visible modules have essentially shifted their probability distribution function for hidden module designs relative to the other competitors and thus are able to capture rent from possession of the visible designs. This rent provides an incentive for increased competition to win the visible module's ownership over and above that which would have been forecast by consideration of a disassociated game. In the limit a monopoly owner of the uppermost visible module will harvest all the surplus value from the lower level modules thereby provoking intense competition for a replacement. Unfortunately path dependency can make it rather difficult to unseat an incumbent monopolist[56].

[55] The actual numbers don't matter too much – it is the relative numbers that are important.

[56] Microsoft's dominance of the PC operating system is the classic example. Once captured not only did it give a competitive advantage in the series of fights over dominant applications packages, but it also is not surprising that

In a more complex situation intermediate level modules can act as barriers to the visibility of the highest visibility modules. Consider a unilateral three level system of three equal size modules. As the visibility of the module decreases the number of participants in the competition for that module's value increases: thus the top module may have only one or two viable designs competing over only one or two generations, the middle module may have five or six designs competing over a few generations, and the lowest module may have a dozen designs competing over several generations. In this situation the owners of the lowest level modules can collude to avoid the capture of the intermediate level by the topmost level. Such collusions are an unstable equilibrium outcome for each round of the game. They are most likely to break down if the uppermost layer is a monopoly, as the monopolist has greatest power over the selection of designs in the intermediate layer, and in allocating surplus amongst the alternatives. The extreme form of collusion is for the lowest level participants to agree on a common solution to the high level module – this is of course the situation where otherwise ferocious competitors agree on open non-proprietary standards which are highly visible to the system. Another way intermediate level modules can act as a barrier is if they are equally dependent on multiple higher-level modules, each of which are competed for by the same participants. Since the intermediate level module must comply with the higher level ones the high level competitors become mutually hostage to each other, and the lowest level competition therefore becomes disassociated. This situation occurs if lower level modules are dependent upon multiple higher-level modules.

When bilateral relationships occur a situation exists where visible modules are at the same time dependent on each other. If different players dominate each module then an unstable equilibrium will exist. The stable equilibrium is when each player has a viable design for each module, which in the limit becomes either only one player with a modular design, or multiple players each with an integrated design. This is the situation that exists with the gas generator core of the gas turbine genset, as the compressor, combustor, and turbine are each interdependent. In this situation whilst intense technological pressures make it attractive to move towards relatively modular designs the risk of becoming captive to a competitor's design rules results in extremely well

---

the size has consistently evolved along the maximum limit of what the user would accept, with for example DOS integrating with Windows and with other key elements. Despite excessive rent extraction (for such a highly visible element) it well illustrates how it's pathway has given it the resources to resist attempts to reduce its monopoly.

choreographed alliances[57]: my compressor with your casing; my combustor with your blades. This also explains how gas turbine vendors are able to survive when they are actually manufacturing (as opposed to assembling) the least attractive modules in the entire system: they are using their dominance of the highly visible modules to extract rent from the hidden but dependent elements in the system.

In a unilateral system the bus function of integration & testing is in an interesting situation. This test & integrate function and the more tangible elements with bus-like characteristics such as control systems have poor visibility and tend to be highly fragmented as competition inevitably causes many entrants. In order to survive bus players will seek to integrate yet purchasers (who control highly visible elements) will impose clear inter-operability standards so that they can mix and match inter-generational improvements[58]. If there are genuinely many players in the bus it is possible for the visible elements to avoid a locus of control emerging in the bus. However because the bus will by definition be dependent on multiple higher-level modules it will tend to be a disassociated game from which the visible elements are less likely to be able to harvest unwarranted rent – a situation that will be exacerbated if the bus becomes dependent on standards that are outwith the span of control of the product. Yet it is seldom as clear-cut a unilateral situation as this as the bus elements typically have weak outputs to other elements within the current generation, or perform the inter-generational feedback role. In this instance abdication of the bus and employment of a strategy of rent extraction purely by control of the visible components is a more risky strategy. Whilst the game is well known this risk can be managed, however as the basis of the game changes so too does the degree of risk.

[57] Not all alliance participants understand the issues equally well – rather famously GE has extracted most of the profit from the joint venture with Snecma. Another view on this is that different participants may possess different value systems: GE's is for economic profit whilst the French government has consistently pursued other objectives as well.

[58] Bus players will to a certain extent welcome this as they can employ these standards across multiple products, platforms, and industries thereby gaining economies of scale and scope. Examples would include standards such as CAD formats, Profibus and Fieldbus instrumentation, and standardised interconnection fittings.

13.6 Risk At The Discontinuities

The nature of the game changes at each discontinuity. These may be either inter or intra-cycle discontinuities. As previously noted the incumbents appear more successful in overcoming intra-cycle discontinuities than inter-cycle discontinuities. There are three levels of difference observable in this system, which in order of increasing challenge are:

- Inter-generation transitions which should not represent a discontinuity unless they coincide with;
- Intra-cycle discontinuities which are associated with the transition from integrated to modular designs or vice versa, and;
- Inter-cycle discontinuities, which are associated with the introduction of a transforming technology or architecture.

These three levels of difference correspond well with the "three-T model" of knowledge boundaries in new product design (Carlile, 2000 & 2001; Carlile & Rebentisch, 2001) where three boundaries are observed:

Table 11: Comparison Of Transitions With Knowledge Boundaries

Degree of Difference	'Sharman' Model	3-T Model
Low, known	Inter-generation transitions	Syntactic boundary "transfer mode"
Intermediate	Intra-cycle discontinuities	Semantic boundary "translation mode"
High, novel	Inter-cycle discontinuities	Pragmatic boundary "transformation mode"

The 3-T model points out that at a boundary the outcome is "either to stick with the old … or engage in a process of transforming the old knowledge to create new knowledge" (Carlile, 2002). These three boundaries represent step changes in the mechanisms (infrastructures) that firms require to successfully deal with the challenge of creating new knowledge as opposed to "sticking with the old" and inevitably withering: they represent qualitatively different infrastructures rather than simply being quantitatively different. At the level of the inter-generational competition any firm that succeeds for more than one round[59] must have developed the internal capability of representing the knowledge inherent in the old product and transferring it to the new product in an augmented manner. This gives the firm the dynamic capability (Teece, Pisano, Shuen; 1997; Teece, 1998) to sustain some degree of competitive advantage within this technology and product whilst the basic architecture of the product remains approximately constant (i.e. either integral or modular). The temporal span of this dynamic capability will depend on the inter-generational clockspeed of the product / module, e.g. commercial aerospace operates at a far slower cycle speed than personal computers – this scaling issue needs to be accounted for when attempting to correlate the dynamic capabilities of firms from disparate industries with particular success factors. If the product architecture is integrated the firm will, of necessity, maintain a wide range of internal capabilities however if the product architecture is more modular then the firm is faced with decisions about how great a range of capabilities to maintain. It is obvious that the firm needs to understand the modules it directly participates in but since capabilities are also costs it will be under pressure to reduce its range in other areas. Conceptually there are three or four areas of interest, which can be related to the modules that the firm chooses to participate in:

1) Direct participation through direct competition in these modules.

2) Indirect participation:

   a) Attempts to harvest value by leveraging visibility over dependent modules.

   b) Attempts to evade rent demands originating from higher visibility modules.

3) No participation; these modules have zero direct or indirect visibility or dependency and are only related through perfect design rules.

[59] One should not denigrate this ability – most start up ventures fail to develop even to this point of capability and have a short life as one trick ponies.

Whilst a product is deep within the inter-generational transitions the firm ought to be able to restrict its investment to 1) and sufficient of 2a) and 2b) that it can maximise its return[60], and none of 3). However as the product comes temporally closer to an intra-cycle discontinuity it needs to widen its capability range. At the time this will seem difficult: the firm will need to invest in something unknown in order to sense and discover whether it might happen, and if the trajectory can be beneficially influenced. When transiting from an integrated to a modular architecture this is a process of reduction in knowledge (which is always easier and perhaps may already be underway spontaneously as the product's scale forces increased sub-contracting) about things (the physical domain) and an increase in knowledge about the very different dynamics of interaction in a modular system (i.e. acquisition of a new process knowledge in the task and organisational domain). At this point the firm's extensive knowledge about the physical domain becomes a handicap in seeing the much wider range of options that become available in the other domains. This is a semantic problem, as the new knowledge needs to be integrated with the current knowledge in order to become useful. When transiting from the modular product to the integrated product the reverse is a challenge: a much simpler process needs to be learned but much more physical domain knowledge needs to be internalised. In approaching these inter-generational transitions the firm is therefore challenged to burden itself with additional capabilities just at the moment that things are already going well but scale is growing out of control (integrated > modular), or just when scale is coming under control but an economic shakeout is occurring (modular > integrated) and at neither time will it seem like the obvious

[60] In the short term the extent to which 2a) and 2b) are carried out within or outside the firm reflect the theory of the firm as having an efficient frontier of competence (Coase, 1937; Teece, 1998). All I am doing is pointing out that in order to manage risk in an efficient manner the boundary of the firm's efficient frontier must be pushed out, during and in advance of crucial transitions, beyond that which can be supported over the longer term. Unfortunately whilst this may be optimal over the longer term it can lead to the firm becoming non-viable in the short term if the competitors do not understand and react accordingly. At a societal level this can lead to localised industrial clusters endlessly postponing a mutually beneficial transition and then failing when they encounter an industrial cluster that has successfully transitioned, e.g. the mutual suicide of the now defunct British shipbuilding industry and, arguably, the US automobile industry of the late 1970s and early 1980s. This has to be one of the stronger arguments against extreme focus on short-term results in selecting the basis of capital allocation policies. Note that in both examples it was not just the industry that restructured it was also the product. There is a symbiotic relationship between product and industry and valid predictive strategy can occur by analysing the product in the small.

thing to do, hence the fact that failures do occur. The better a firm is at managing its existing paradigm the more averse it will be to changing paradigm, hence the increased risk for incumbent top-dogs.

It is probably not enough for a company simply to delve back into history and ask how the industry restructured last time. Even if not one change has happened in the underlying architecture, provided the underlying architecture is multidimensional, there will be multiple possible ways of interpreting the system and reorganising. In these situations just the normal amount of randomness in trajectory will cause Fine's double helix to cycle around to a different yet equally consistent interpretations of viable industry configurations. This is another illustration that path dependency matters albeit in this case the clustering algorithm is the invisible hand of the evolutionary process rather than an artificially modelled algorithm. Thus even if nothing is changing the firm must engage in active sensing rather than consider the situation to essentially be a static historical replay. This also can explain why a continual cycle of restructuring may occur without there seeming to be a material motivation, e.g. from discipline to product oriented organisations, or from vertically to horizontally integrated industries. Similarly the randomness in the process should not be taken for granted – decisions to participate or not by firms will affect the probability of coalescence around any one of the viable interpretations of the structure as the firm, product, and market co-evolve (this is another interpretation of "waiting until a market is large enough to be interesting"; Christensen, 2000.).

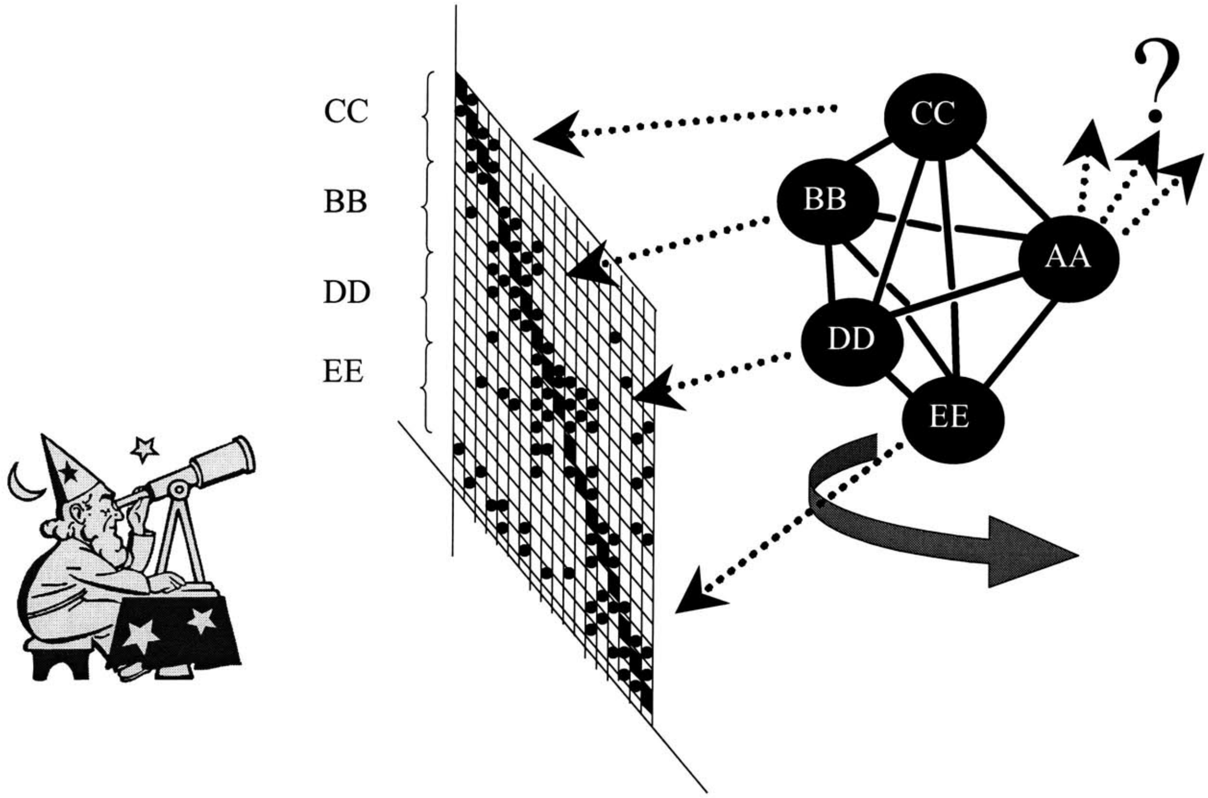


Figure 116: Multi-Dimensional Architectures Can Yield Radically Different Yet Equally Consistent Interpretations

This need to engage in active sensing is especially true for situations where a strong bus exists in the architecture as here there is a far greater risk that the essential multi-dimensionality of an architecture may have been misinterpreted, with a nodal cluster of dominant modules being mistakenly relegated to a 'passive' supporting role as a bus. Even if little stochastic variation occurs in the normal cycle of horizontal-vertical restructuring this opportunity is least likely to be noted by the owners of the prevailing dominant elements, as they will be most captive to the existing paradigm and its supporting language. This is a very good reason for owners of dominant elements to remain active participants in the typical bus elements of control systems[61] and test & integration functions, as in this way they will both reduce the risk of being surprised by a novel basis of the intra-generational game, and obtain a disproportionate return on their investment in terms of obtaining an insight into the role of other elements in which they may not

[61] See Prencipe, 2000 for an example of how various firms have taken different approaches to this, with different results.

otherwise be active indirect participants. However this is also a good example of where a firm (or its resident individuals) that has become accustomed to setting investment thresholds on the basis of the returns to be obtained from dominant modules will screen out the longer term risk-reduction to be obtained from these investments as the short term returns are too low, or in the extreme case may even be seen as pure costs. In case this is thought of as being an overly pessimistic view I can only say that I have never picked up a budget and seen 'test and integration' associated with a positive return on investment, and that it has always appeared as a cost item rather than being a value-added learning opportunity. In the absence of risk-reducing investment in any buses that offer the potential to form the nodes around which new dominant clusters might coalesce, the incumbent owners will be extremely surprised when even small changes in the value surface of the current architecture causes the industry to snap across to a new stable state in an abrupt manner as the first participant to surface the tacit knowledge and manage the semantic process triggers change. The more homogenous the system and the lower energy the environment the greater the propensity for such apparently surprisingly fast restructuring to occur, in a manner analogous to watching a super-cooled liquid crystallise instantaneously.

The rather odd result that the societally less valuable visible modules from the perspective of creating value, are highly attractive points of control from the perspective of capturing value plays out somewhat differently over the third type of transition: that of the inter-technology cycle or pragmatic boundary. In general the societal value of an architecture will be maximised by the product decomposition that most efficiently hides information. However individual players will seek decompositions that maximise their ability to extract rent by monopolising high level nodes of visible information (to influence the outcome of lower level nodes) and participating in sufficient lower level nodes (to extract additional rent in a direct or indirect manner). Thus players in a node will find their ability to defend their rent compromised by the degree of dependence, and their ability to demand additional rent determined by their degree of visibility. In less simple situations the degree of concentration (and the ratio) of visibility and dependence become important[62]. A simple situation is depicted in Figure 117 below where one module captures a disproportionate share of the value as if it were dominant in the architectural hierarchy over subsidiary modules. This is shown in both the disaggregated entity form and the cumulative aggregated societal form. Because it is less relevant the issue of whether the cycle commences and finishes in integrated architectures is set aside.

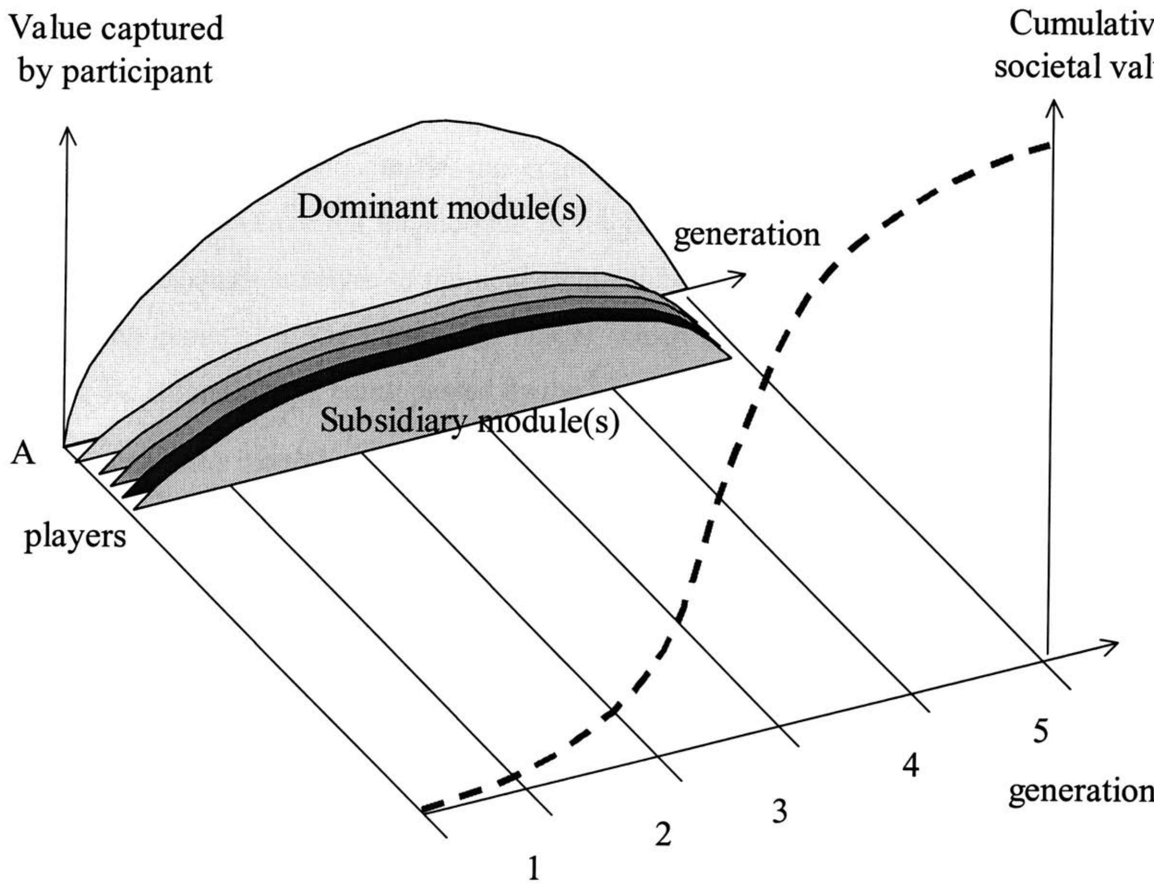


Figure 117: One Cycle S-Curve

During the cycle the subsidiary modules will continually attempt to defend themselves from attempts by the owners of dominant modules to extract value. Because they do not have the power (defined by the ration of visibility/dependence) to resist this inside the product's system boundary they will, through a rational process of resource allocation under constraint, seek

[62] As a result of this the optimisation algorithm for maximising societal value by clustering elements (in a DSM) is far simpler than the equivalent algorithm for capturing entity value, as the latter must take into account accrued handicaps and any learning effects about past and future strategies of other entities.

opportunities outside the system. So long as these opportunities do not impinge on the market for the product such a rational strategy is to the benefit of all players. Equally the dominant modules will rationally invest in improving and continuing their dominance of the system.

At some point an alternative technical solution will emerge to the same customer need. At that moment the market for fulfilling the customer need is being fully and efficiently harvested by the dominant modules who have all of their dynamic capability invested in controlling their ecosystem. However the subsidiary modules have no such vested interest, indeed they would be only too glad to break away into a new architecture where they have a new opportunity to establish different path dependencies that offer greater upside. In this new architecture there may even be further opportunities for modular development of the same subsidiary modules as first they accommodate themselves to the new architecture's design rules and then exploit it. This is the situation depicted in the Figure 118 below, which shows the two-cycle S-curve that results from the old technology A being passed by the new technology B.

In this situation the incumbent dominant elements or modules can be summarised as having been captured by their paradigm, however this is a much-abused simplification that merits explanation. Unless they have accumulated near limitless resources (a la Microsoft) their investment decisions will be dominated by the need to maintain control of their existing system and any slackening of their efforts will result in short-term revenue impairment as, either their competitors for control of the dominant modules take advantage, or as the subsidiary modules take advantage. Furthermore it is unlikely that this will have happened in the middle of the technology S-curve for them and so there will almost certainly be above average competitive pressures in evidence as a cycle of consolidation takes place around an integrative intra-cycle discontinuity.

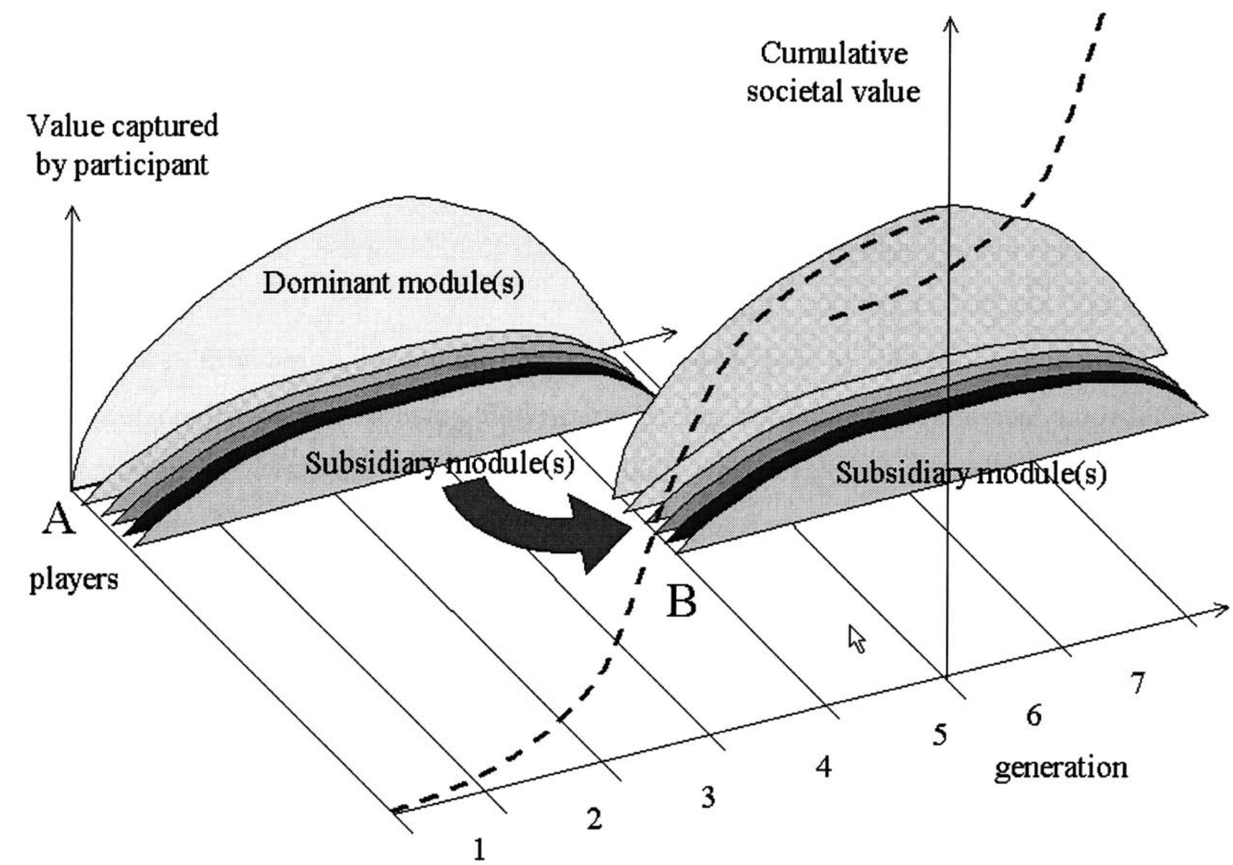


Figure 118: Two-Cycle S-Curve

In all fairness in this situation they will be struggling to maintain sufficient dynamic capability to even manage the existing relatively well known semantic translation and will be hard pressed to have the vision and capability to undertake an even more taxing (for them) pragmatic transformation across a far more risky inter-technology cycle boundary. In contrast the Quisling subsidiary modules are faced with a very well understood syntactic transfer process that they have already mastered through selling their low margin wares into many markets, whilst the entering technology is fully focussed on the syntactic product challenge.

This 'new' technology may not be that new at all. It may simply have been biding its time waiting for the right conditions to occur. These conditions may have inadvertently been created by the 'old' technology pushing the performance of crucial common subsidiary modules or the scale of the demand to the correct point, or may have been created in a parallel market. An example of the latter may be occurring in the solar photovoltaic market where silicon-based solar panels have been serial failures as an investment proposition, but have finally reached a cost-competitive position for niche distributed power applications on the back of advances in silicon technology gained by experience in large volume microprocessor manufacturing, and may reach a position of wider cost-competitiveness on the back of R&D in biotechnology. An example of the former comes from the gas turbine market when the incumbent steam boiler and steam turbine manufacturers for naval propulsion were unable to meet the challenge of creating a marinised gas turbine. This was not for want of trying by companies such as Vickers over a twenty year period, but in the end the challenge was overcome by the aero-engine manufacturers such as Rolls-Royce but utilising large parts of the common technological knowledge and supplier base and with the advantage of by then well-developed capabilities in the parallel aviation field. However until many of these advantages have been made the entrant technology may simply stutter along and users, seeing a satisfactory rate of technology development in the current solution, will see no reason to switch and adopt the new. The circumstances in which adoption curves themselves can resemble punctuated equilibria have been developed (Loch & Huberman, 1999) and the points made here is the different ones of why the old technology player finds it so difficult to pre-emptively adopt the new (an issue of insufficient resources to acquire the necessary dynamic capability), and why the introduction & growth rate of the new is perhaps more rapid than can be explained by the simple exogenous adoption rate (an issue of subsidiary modules having lower barriers to adoption and then accelerating endogenous progress).

The new architecture may not simply be the old one with a new dominant cluster. It may not require all of the old subsidiary modules, it may promote others, and it may require additional subsidiary elements. An example of this may be found in four cycles of technology-driven architectures for medium sized propeller-driven surface warship propulsion in the last 150 years[63]: steam driven reciprocating propulsion; then steam driven turbines; then direct drive gas turbines; then gas turbines driving indirectly through electrical generators and motors. Each of these has affected the propulsion architecture differently, for example: the introduction of gas turbines also created a need for more complex reversing gearboxes or controllable pitch

[63] This is a summary of the dominant designs in the class of medium sized war vessels requiring relatively rapid acceleration to high speeds. Other dominant designs have catered for larger and smaller vessels, and a variety of hybrid vessels have emerged during periods of ferment, e.g. COSAG / CODAG during the period of experimentation with gas turbines; or for specialist needs, e.g. the diesel-electric designs of the long-range German pocket battleships.

propellers; the introduction of steam turbines eliminated reciprocating engines as a dominant element but retained the dominance of the boiler, whilst the introduction of the gas turbine eliminated both the steam turbine and the boiler. The introduction of the new high visibility electrical elements into the design also open up the pathway to the replacement or demotion of gas turbines as other technologies become viable[64].

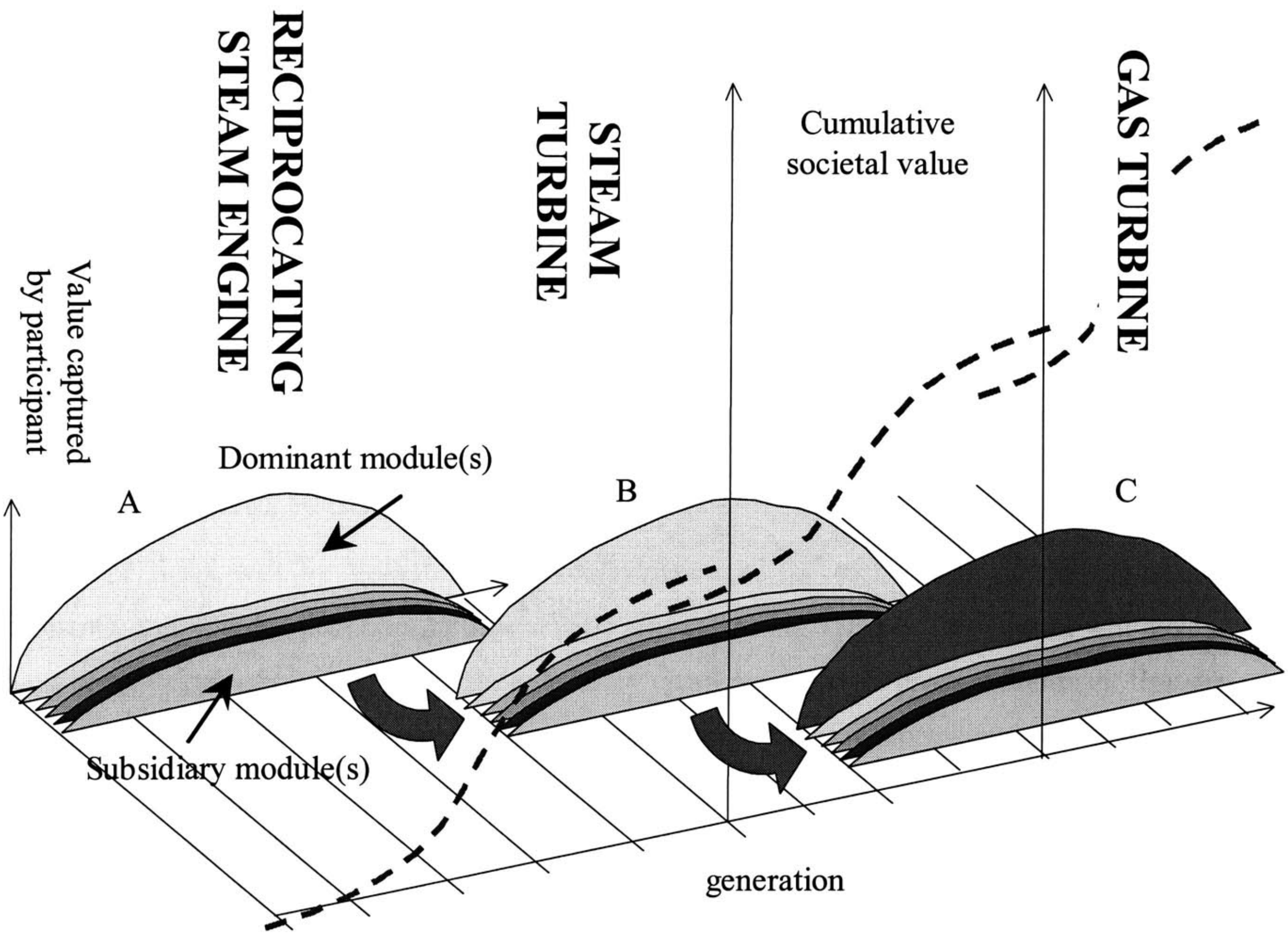


Figure 119: Successive S-Curves For Medium Sized Warship Propulsion

These dynamic changes at the inter-cycle transition make modelling of the situation extremely difficult. Not only do the constituent fundamental elements of the architecture change but the performance contribution of an element in the old architecture may not be the same as the

[64] By inserting the top level module electrical modules it becomes possible to decouple surge capacity from long range endurance capacity in future cycles, e.g. by inserting rapid release storage devices such as fuel cells, superconductor coils, flywheels, etc.

performance contribution in the new, system level effects may be unknown, and there is an added degree of uncertainty related to the market's technology adoption timing and rate. This is exactly the problem the dominant firms face – they more than any need to answer these questions.

Most organisations are resource constrained. Qualitatively the point of greatest return on investment for risk reduction appears to be to invest in bus-like elements that stand the greatest chance of having a material change in their visibility/dependence relationship under a new architecture. These elements are a) most exposed to feedback from other elements that reveal the accumulating tension in the existing value system, b) most likely to play a pivotal role in a new architecture which gives the opportunity to acquire value directly or at least acquire time to make catch-up investments and c) therefore may avoid the need to make a wide range of investments in potential new technologies in a premature manner. The case for investing in the bus is akin to the value of information calculations commonly conducted in determining whether investments should be made in tests to minimise uncertainty.

13.7 People and Processes

Dominant firms will by definition have competence in the technology of the controlling elements, and in order to maximise the parasitic rent-extracting value of this control will have to have some degree of competence in the subsidiary elements. A single product dominant firm can therefore be viewed as hosting concentric layers of competence wrapped around the core. As soon as the firm attempts to manage risk by investing in capability in a novel and potentially dominant second core technology it becomes a bi-polar system. Of course there may well be internal debate about which novel technology will pose the greatest risk and so multiple seed investments may be made for risk-mitigation purposes. At this point even the simplest one-product firm has become to some extent a technology conglomerate, a situation that worsens if the firm is resident to a disparate array of products. This of course is a classic recipe for a management nightmare.

The simplistic approach is to say that the strategic decision makers in the firm should be technically knowledgeable. This is another way of saying that the senior management of a technology conglomerate should be predominantly engineers and not accountants, and in answer to the riposte that most senior managers of these organisations initially trained as engineers the point might be made that in order to succeed they will inevitably have become apparatchiks who now only recognise the common language of short-term finance.

This simplistic approach is unfortunate in the way it polarises the debate, as many technical firms have prospered under accountants and many others have vanished under engineers. It is true that there is still no commonly accepted metric for comparing technical value and this is why the common currency of exchange is accounting but this is not to say that some notional balance needs to be redressed by having an accounting-based process conducted by an engineering – sourced product. What it suggests rather is that any organisation needs to develop tools to recognise the gaps in their capabilities and to bridge them before they distort the decision making process. This thesis has outlined one possible method for doing so which becomes in essence a way of representing technical potential in a commonly understood value system; there are others. Clearly the approach suggested in this thesis is extremely immature. It is not yet capable of being applied in a semi-automated manner or on a large scale, and it has not been tested to any significant degree. However it appears directionally interesting and warrants further study and development.

14 CONCLUSIONS AND FURTHER WORK

About half way through calculating the value of the gas turbine in the physical domain I made a note to myself that these sums are insanely complex, and that I wasn't even trying to differentiate between societal or actor value, nor to quantify multi-round games, nor to consider the other evolutionary operators (porting, etc.), nor to consider multiple domains. At that point I doubted whether this approach would yield anything of value. However these calculations are exactly what are taking place in the real world except that nature doesn't do sums – it is called evolution. Yet any sentient designer is trying thinking through a similar process even if only in a localised and / or intuitive manner. It is because of the hideous complexity of the calculation that so few individuals are able to a priori architect complex systems - and even if they can, they are unable to explain what they are doing and why. In the absence of a unifying structure we have resorted to the only mutually recognisable language we know – that of accountants – and in doing so created problems by focussing firms in on the things they understand and can describe to the detriment of their ability to cope with the risky unknown. So inevitably society has continued to stumble down the blind watchmaker's road with the unseen hand of the market as the only guide, which is an essentially retrospective approach.

As a result of this test case I think we can do better, and I am in general happy that:

- Baldwin & Clarks' theories can be applied in a qualitative manner to real cases and seem to provide a directionally sound causal mechanism.
- There is merit in using option value as a metric for judging different strategies.
- That strategically useful option values can be calculated in the small for architectures, sometimes without the need to take a macroscopic view.
- That option values can be calculated for integrated and semi-integrated architectures as well as modular architectures.
- That DSMs can contribute to making a better calculation.
- That taking a multi-domain view adds value, as there is information in the different domains.

- That until 'fire and forget' quantitative tools have matured there is still value in these concepts, provided that people with the right mix of skills and experience apply them.
- That the difference between visibility and dependency appears crucial in understanding the dynamics of multi-player games.
- That one must be very cautious in assessing highly dimensional architectures, especially if they are relatively balanced and subject to alternative interpretations, and that in these cases one is often best served by investing in buses to minimise risks of being blindsided, especially when a test & integration bus is pinned to the current source of architectural control.
- That DSMs are prone to mis-interpretation in highly dimensional architectures.

However I think that actually 'doing it' is still a non-trivial task and the following challenges remain:

- The process of calculating single domain societal value needs automating.
- Multiple domain objective functions need to be tested and suitable search strategies developed.
- The process of calculating visibility and dependence needs further thought in systems with a high degree of bilateral links.
- That simulators need developing for calculating actor values, and optimal strategies for games.
- That the other operators (porting, etc.) need adding in so as to be able to evaluate dynamically evolving architectures.
- That it is worth considering whether there is more evidence for the three levels of transition in S-curves, and whether these are indeed related to knowledge boundaries and dynamic capabilities.
- That more case studies need to be conducted and tested – it may be better to investigate slowly evolving situations so as to really dig to the bottom of the causal mechanisms.

Returning to my origin – that of considering the energy industry's evolutionary pathway – I hesitate to present anything more than a qualified opinion on the basis of this work. It appears that the growth of the centralised electrical generating and transmission technologies as dominant modules are now open to reassessment in the light of increased network interconnection technologies and distributed generation. Thus only small changes in technical readiness have the potential to prompt a rapid architectural restructuring of a fundamentally different order of magnitude than the endless horizontal / vertical restructuring of generation / transmission / distribution. The drivers for change are visible: population growth, global industrialisation & post-industrial economies, environmental concerns, resource longevity and availability – but must be balanced against the not inconsiderable forces with a vested interest in preserving the current equilibrium. However it is not only the electrical system's architecture that should concern us as the transportation sector will also contribute to the range of options in a more coupled manner than in the last century. Whilst vested interests may be able to restrain or direct the evolutionary pathway in either the electrical or the transportation system it is unlikely that they will be able to prevent cross-over technologies from entering the scene. Those players (new entrant or incumbent) who create the internal capabilities to understand these pivotal technologies will have an enhanced opportunity to interpret and leverage any new joint architecture and thereby create and lock in longer term sustained competitive advantage.

From the perspective of a manufacturer of medium scale fixed generating assets, quite apart from the obvious issue of considering novel generating technologies, I would be tempted to invest in technologies for actively managing network interconnections, especially at a small scale and in a remote manner – a number of technologies fit this description. These are sufficiently close to the existing business to add short-term value (or at least not destroy capital) yet sufficiently novel and pervasive to expose the business to new sources of learning with at worst the risk minimisation this implies, and at best the opportunity to influence and pre-empt the architectural pathway.

15 REFERENCES

Alexander, C. 1964. "**Notes On The Synthesis Of Form**", Harvard University Press, ISBN 0-674-62751-2.

Alexander, C. and Manheim, M., "**HIDECS 2: A Computer Program For The Hierarchical Decomposition Of A Set With An Associated Graph**", MIT Civil Engineering Systems Laboratory Publication Number 160, 1962 and Alexander, C., "**HIDECS 3: Four Computer Programs For The Hierarchical Decomposition Of Systems Which Have An Associated Linear Graph**", MIT Civil Engineering Systems Laboratory Research Report R63-27, 1963.

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

Baldwin, C.Y. & Clark, K.B. "**Sun Wars: Competition within a Modular Cluster, 1985-1990**", in "*Competing In The Age Of Digital Convergence*" 123-157, editor Yoffie, D.B.; Harvard Business School Press; ISBN: 0875847269; 1997.

Baldwin, C.Y. & Clark, K.B. 1999. "**Design Rules**", MIT Press, Cambridge, ISBN 0-262-02466-7.

Bowman, E.H., Moskowitz, G.T., "**Real Options Analysis and Strategic Decision Making**", *Organisation Science*, Vol. 12, 2001.

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

Browning, T.R., Eppinger, S.D., "**Modelling The Impact Of Process Architecture On Cost And Schedule Risk In Product Development**", MIT Sloan Working Paper Number 4050, revised April 2000.

Boppe, C.; "**Systems Engineering, ESD.33**"; MIT classes, 2001.

Carlile, P.R., "**An Integrative Framework Of Managing Knowledge Across Boundaries**", MIT/Sloan Working Paper, 2001.

Carlile, P.R., "**The Dynamics Of Knowledge And Competitive Advantage**", MIT/Sloan Working Paper, 2002.

Carlile, P. 2002. "**A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development**". Forthcoming in *Organization Science*, 2002.

Carlile, P.R., and Rebentisch, E.S., "**Into The Black Box: The Knowledge Transformation Cycle**", MIT/Sloan Working Paper, 2001.

Christensen, C.M. 1997. "**The Innovator's Dilemma**", HarperBusiness, New York, ISBN 0-06-662069-4.

Coase, R., "**The Nature of the Firm**", *Economica*, 1937.

Crawley, E.; "**Systems Architecting, ESD.34**"; MIT classes, 2001.

Danilovic, M., and Borjesson, H., "**Managing The Multi-Project Environment**" and "**Participatory Dependence Structure Matrix Approach**", Proceedings Of Third Dependency Structure Matrix Workshop, MIT, October 2001.

Eldredge, N., Gould, S.J., "**Punctuated Equilibria: An Alternative to Phyletic Gradualism**", *Models In Paleobiology*, Freeman, San Francisco, 1972.

Eppinger, S.D., Nukala, M.V., Whitney, D.E, "**Generalised Models Of Design Iteration Using Signal Flow Graphs**", *Research In Engineering Design* 1997:9, Springer-Verlag.

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

Fernandez[65], C.I.G, "**Integration Analysis Of Product Architecture To Support Effective Team Location**", MSc Thesis, MIT, 1998.

Fine, C.H. 1998. "**Clockspeed**", Perseus, New York, ISBN 0-7382-0153-7.

Fine, C.H. & Whitney, D.E.; 1996; "**Is The Make-Buy Decision Process A Core Competence?**"; MIT Center for Technology, Policy, and Industrial Development Working Paper, February 1996; published in: *Logistics in the Information Age;* Moreno Muffatto and Kulwant Pawar (eds.); Servizi Grafici Editoriali, Padova, Italy, 1999, pp. 31-63.

[65] I follow the English convention of referring to Carlos Iñaki Gutierrez Fernandez as "Fernandez" so as to allow other researchers to trace the references, however I note this is not the correct convention for a Spanish name.

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.

Fixson, K.F., 2001, "**Three Perspectives On Modularity – A Literature Review Of A Product Concept For Assembled Hardware Products**", MIT Engineering Systems Division, Working Paper, 2001.

Gibbons, R. 1992. "**Game Theory For Applied Economists**", Princeton University Press, New Jersey, ISBN 0-691-00395-5.

Greene, W.H., "**Econometric Analysis**"; 4th edition; Prentice Hall; ISBN: 0130132977; 1999

Hauser, J.R. and Clausing, D. 1998. "**The House Of Quality**", *Harvard Business Review*, May-June 1988.

Idicula, J.; "**Planning For Concurrent Engineering**"; Thesis draft; Nanyang Technological University; March 1995.

Jacobson, M.Z. and Masters, G.M. 2001; "**Exploiting Wind Versus Coal**"; *Science* 239, 1438.

Kennedy, P., "**A Guide To Econometrics**"; 4th edition; MIT Press; ISBN 0262611406; 1998.

Loch, C.H., Huberman, B.A., "**A Punctuated-Equilibrium Model Of Technology Diffusion**", *Management Science*, Vol. 45-2, 1999.

Mascoli, G.J.; "**A Systems Engineering Approach to Aero Engine Development in a Highly Distributed Engineering and Manufacturing Environment**"; MIT MSc (SDM) thesis; 1999.

Mikkola, J.H., 1999, "**Modularisation in Black-Box Design: Implications For Supplier-Buyer Partnerships**", DRUID Winter Conference 1999.

Mikkola, J.H., 2000, "**Modularisation Assessment Of Product Architecture**", DRUID Winter Conference, 2000.

Mikkola, J.H., 2001, "**Modularity And Interface Management: The Case Of Schindler Elevators**", INCOSE, 2001.

Miller, G.A., "**The Magical Number Seven, Plus Or Minus Two: Some Limits On Our Capacity For Processing Information**", *Psychological Review*, 63, 1956.

Moore, G.A., and McKenna, R., "**Crossing the Chasm: Marketing and Selling High-Tech Products to Mainstream Customers**", HarperBusiness; ISBN: 0066620023, rev. ed. 1999.

Morelli, M.D., "**Evaluating Information Transfer In Product Development**", MSc Thesis, MIT-Sloan, 1993.

Morelli, M.D., Eppinger, S.D., Gulati, R.K., "**Predicting Technical Communication In Product Development Organizations",** IEEE Transactions On Engineering Management, Vol. 42., August 1995.

Moy, H.M., 2000, "**Commercial Gas Turbine Engine Platform Strategy and Design**", MSc Thesis in Engineering & Management, MIT.

Pimmler, T.U. and Eppinger, S.D.; "**Integration Analysis of Product Decompositions**", Proceedings of the ASME Sixth International Conference on Design Theory and Methodology, Minneapolis, MN, Sept., 1994. Also, M.I.T. Sloan School of Management, Cambridge, MA, Working Paper no. 3690-94-MS, May 1994.

Porter, M.E. 1980. "**Competitive Strategy**", Free Press, New York, ISBN 0-648-84148-7.

Porter, M.E. 1985. "**Competitive Advantage**", Free Press, New York, ISBN 0-648-84146-0.

Prencipe, A. "**Breadth and depth of technological capabilities in CoPS: the case of the aircraft engine control system**", *Research Policy* 29, 2000, 895-911, Elsevier.

Rechtin, E., and Maier, M.W.; 1991; "**The Art Of Systems Architecting**"; CRC Press; ISBN-0-8493-0440-7; first ed. 1991, second ed. 2000.

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

Schilling, M.A.; "**Modularity In Multiple Disciplines**"; NYU Stern Business School Working Paper; May 2001.

Senge, PM. 1994. "**The Fifth Discipline**", Currency Doubleday, New York, ISBN 0-385-26095-4.

Shannon, C.E. and Weaver, W.; "**The Mathematical Theory Of Communication**"; *Urbana Ill.;* 1949.

Sterman, J.D., "**Business Dynamics: Systems Thinking and Modelling for a Complex World**"; Irwin Professional Pub; ISBN: 007238915X; 2000.

Suh, N.P., "**Axiomatic Design: Advances and Applications**"; ISBN: 0195134664; Oxford University Press, 2001.

Sullivan, J.P., "**The Relationship Between Organizational Architecture, Product Architecture, And Product Complexity**", MIT MSc thesis, 1998

Teece, D.J. 1998. "**Capturing Value From Knowledge Assets: The New Economy, Markets For Know-How, And Intangible Assets**", *California Management Review*, Volume 40, Number 3, University Of California.

Teece, D. J., G. Pisano and A. Shuen. 1997. "**Dynamic Capability and Strategic Management"** *Strategic Management Journal*, 18 (7), pp. 509-533.

Ulrich, K.T., and Eppinger, S.D., "**Product Design and Development**", McGraw-Hill, ISBN-0-07-229647-X, 1995, 2000.

UN, 2000. "**World Energy Assessment: energy and the challenge of sustainability**", United Nations Development Programme, United Nations Department Of Economic Social Affairs, World Energy Council, World Energy Council, ISBN 92-1-126126-0.

Whitney, D.E. 1996. "**Why Mechanical Design Cannot Be Like VLSI Design**", MIT.

Womack, J.P., Jones, T.J., Roos, D. 1990. "**The Machine That Changed The World**", Rawson Associates, New York, ISBN 0-06-097417-6.

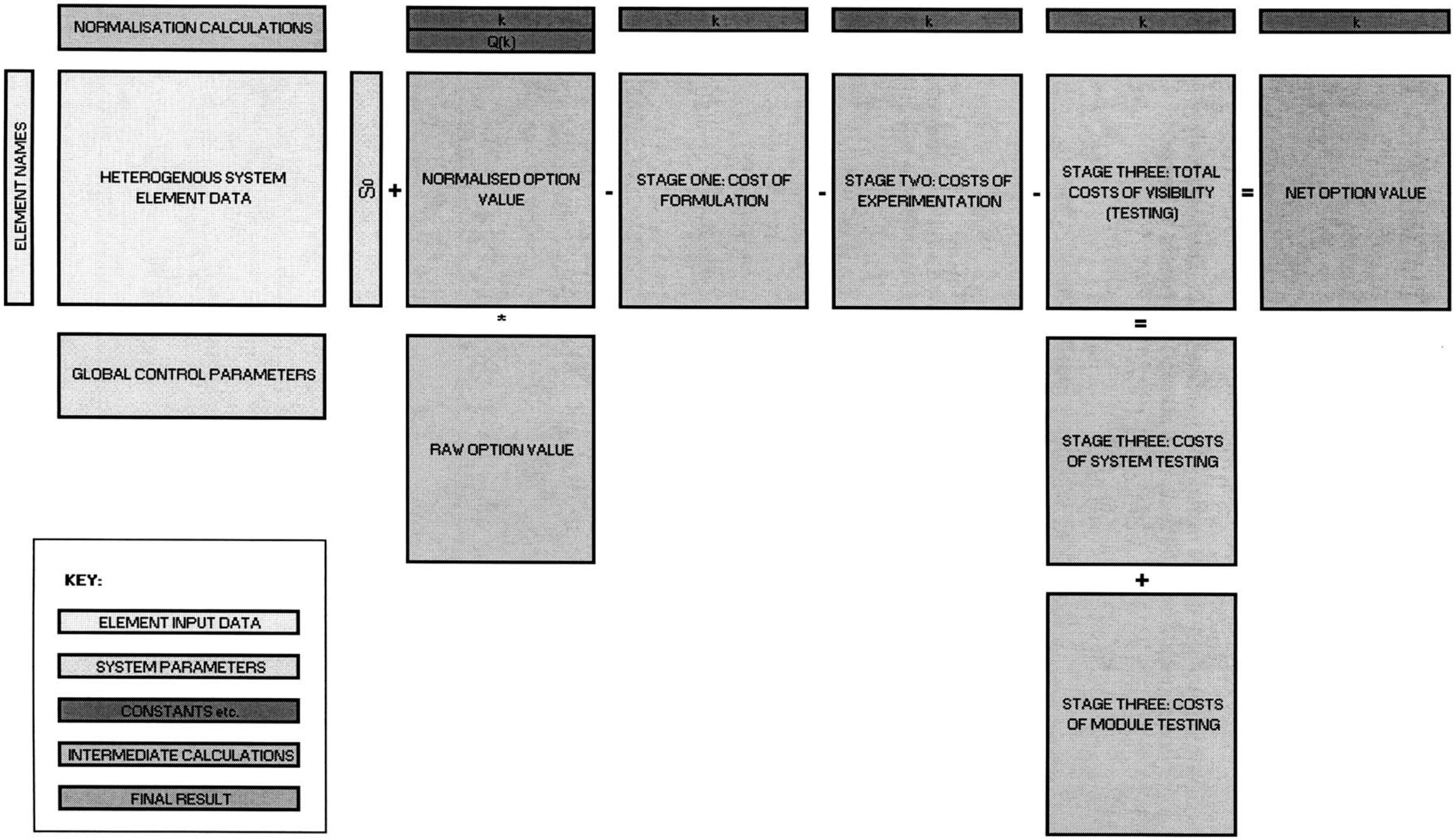
http://mit.edu/dsm/ "**The MIT Design Structure Matrix (DSM) Home Page**", 2001.

APPENDIX A: VALUES FOR Q(K)

k	Q(k)	j					
		1	2	3	4	5	6
1	0.3989	0.399	0.564	0.691	0.798	0.892	0.977
2	0.6810	0.681	0.963	1.180	1.362	1.523	1.668
3	0.8881	0.888	1.256	1.538	1.776	1.986	2.175
4	1.0458	1.046	1.479	1.811	2.092	2.338	2.562
5	1.1697	1.170	1.654	2.026	2.339	2.616	2.865
6	1.2701	1.270	1.796	2.200	2.540	2.840	3.111
7	1.3534	1.353	1.914	2.344	2.707	3.026	3.315
8	1.4242	1.424	2.014	2.467	2.848	3.185	3.489
9	1.4853	1.485	2.101	2.573	2.971	3.321	3.638
10	1.5389	1.539	2.176	2.665	3.078	3.441	3.770
11	1.5865	1.587	2.244	2.748	3.173	3.548	3.886
12	1.6293	1.629	2.304	2.822	3.259	3.643	3.991
13	1.6680	1.668	2.359	2.889	3.336	3.730	4.086
14	1.7034	1.703	2.409	2.950	3.407	3.809	4.172
15	1.7359	1.736	2.455	3.007	3.472	3.882	4.252
16	1.7660	1.766	2.498	3.059	3.532	3.949	4.326
17	1.7939	1.794	2.537	3.107	3.588	4.011	4.394
18	1.8200	1.820	2.574	3.152	3.640	4.070	4.458
19	1.8445	1.845	2.609	3.195	3.689	4.124	4.518
20	1.8675	1.868	2.641	3.235	3.735	4.176	4.574
21	1.8892	1.889	2.672	3.272	3.778	4.224	4.628
22	1.9097	1.910	2.701	3.308	3.819	4.270	4.678
23	1.9292	1.929	2.728	3.341	3.858	4.314	4.726
24	1.9477	1.948	2.754	3.374	3.895	4.355	4.771
25	1.9653	1.965	2.779	3.404	3.931	4.395	4.814
26	1.9822	1.982	2.803	3.433	3.964	4.432	4.855
27	1.9983	1.998	2.826	3.461	3.997	4.468	4.895
28	2.0137	2.014	2.848	3.488	4.027	4.503	4.933
29	2.0285	2.029	2.869	3.513	4.057	4.536	4.969
30	2.0428	2.043	2.889	3.538	4.086	4.568	5.004
31	2.0565	2.057	2.908	3.562	4.113	4.598	5.037
32	2.0697	2.070	2.927	3.585	4.139	4.628	5.070
33	2.0824	2.082	2.945	3.607	4.165	4.656	5.101
34	2.0947	2.095	2.962	3.628	4.189	4.684	5.131
35	2.1066	2.107	2.979	3.649	4.213	4.711	5.160
36	2.1181	2.118	2.995	3.669	4.236	4.736	5.188
37	2.1293	2.129	3.011	3.688	4.259	4.761	5.216
38	2.1401	2.140	3.027	3.707	4.280	4.785	5.242
39	2.1506	2.151	3.041	3.725	4.301	4.809	5.268
40	2.1608	2.161	3.056	3.743	4.322	4.832	5.293
41	2.1707	2.171	3.070	3.760	4.341	4.854	5.317
42	2.1803	2.180	3.083	3.776	4.361	4.875	5.341
43	2.1897	2.190	3.097	3.793	4.379	4.896	5.364
44	2.1988	2.199	3.110	3.808	4.398	4.917	5.386
45	2.2077	2.208	3.122	3.824	4.415	4.937	5.408
46	2.2164	2.216	3.134	3.839	4.433	4.956	5.429
47	2.2249	2.225	3.146	3.854	4.450	4.975	5.450
48	2.2331	2.233	3.158	3.868	4.466	4.993	5.470
49	2.2412	2.241	3.170	3.882	4.482	5.011	5.490
50	2.2491	2.249	3.181	3.896	4.498	5.029	5.509

Values of Q(k) calculated using a Mathcad program written by Carliss Y. Baldwin, © 1996.

APPENDIX B: SPREADSHEET LAYOUT



This is the basic spreadsheet layout used for the NOV calculations. Although conceptually simple the actual tables become quite large, hence the inclusion of this much-simplified cartoon. Replacing heterogeneous data with homogenous permits system optimisation.

APPENDIX C: EXECUTIVE SUMMARY

Valuing Architecture For Strategic Purposes

By

David M Sharman

Submitted to the System Design and Management Program in Partial Fulfilment of Requirements for the Degree of Masters of Science in Engineering and Management

at the

Massachusetts Institute of Technology

February 2002

Signature of Author:

___

David M. Sharman
System Design and Management Program
February 2002

Certified by:

___

Paul Carlile
Thesis Supervisor
Professor, Sloan School of Management

Certified by:

___

Ali A. Yassine
Thesis Supervisor
Research Scientist, Centre for Technology, Policy, and Industrial Development

Accepted by:

___

Steven D. Eppinger
LFM/SDM Co-Director
GM LFM Professor of Management Science and Engineering Systems

Accepted by:

___

Paul A. Lagace
LFM/SDM Co-Director
Professor of Aeronautics & Astronautics and Engineering Systems

Problem Statement

Even though there exist a number of instances where the selection of one total architecture (product, organisation, etc.) over another has clearly influenced both the societal value created and the net value captured by any given economic entity, there exist few tools to objectively analyse the best architecture for a given problem. This is essentially a strategic question and there are strategic analysis tools available, however these are unsatisfactory in objectively and quantitatively resolving such questions in anything other than an ad-hoc manner.

It has been proposed (Baldwin & Clark, 2000[66]) that the societal value of highly modular architectures can be analysed objectively by the application of real options theory. This thesis attempts to extend this proposal to more integrated architectures. It then outlines a methodology that may allow determination of the value capture question.

Originality Requirement

The worked example of an integrated architecture used to explain the proposed extension of the modular valuation technique of Baldwin & Clark is new.

In exploring a series of problems in measuring the architecture new insights are gained into alternative ways of describing architectures using Dependency Structure Matrices (DSMs) and in the limitations of DSMs. This leads to the definition of new architectural terms, illustrations of alternative solutions using the notion of domains, dimensions, and topologies, and the proposal for new DSM manipulation algorithms. Proposals are made and illustrated regarding quantification of the degree of modularity and integration in architectures.

A holistic integrating explanation of the relationships between DSMs, Quality Functional Deployment matrices (QFDs) and the inter-domain transforms described by axiomatic design's Design Matrices (Suh, 2001[67]) is described.

The introduction of penalty terms into the real options valuation to represent boundaries is new.

The suggestion that architectural control may also be exercised through test and integration competency is new, as is the description of why these are so easily subject to inversion from one to another, and the implications of this.

The proposed extension of societal valuation to determination of value capture applies the dynamic process of multi-stage multi-player game theory in a new context.

The hypothesis is that the application of this method will quantitatively demonstrate, and conceptually link, the relationship between dealing with "knowledge boundaries" (Carlile, 2002[68]) and the more strategic discussions of a firm's dynamics capabilities (Teece, 1997[69]). This demonstration is particularly helpful since it might be able to explain some of the difficulties historically experienced by technology conglomerates in harvesting synergy.

---

[66] Baldwin, C. Y. and Clark, K. B., 2000, "Design Rules, Volume 1: The Power Of Modularity", *MIT Press*.

[67] Suh, N. P., 2001, "Axiomatic Design: Advances And Applications", *Oxford University Press*.

[68] Carlile, P. 2002. "A Pragmatic View of Knowledge and Boundaries: Boundary Objects in New Product Development". Forthcoming in *Organization Science*.

[69] Teece, D. J., G. Pisano and A. Shuen. 1997. "Dynamic Capability and Strategic Management," *Strategic Management Journal*, 18 (7), pp. 509-533.

Content and Conclusion(s)

A representative test case of a small industrial gas turbine generator set ("gen-set") is used throughout this thesis.
There are several stages in this work. A literature review sets the scene and is supplemented by a section dealing with expected difficulties in extending the work of Baldwin & Clark to integrated architectures.

Then the first set of activities focuses around the creation of a DSM for the gen-set which represents the relationships between various elements of the product in the physical domain and will be used to document and evaluate alternative architectures. A brief discussion is given of the relationship between the physical domain and the other relevant domains.

The second set of activities is to identify and evaluate the architecture of the gen-set in the physical domain using the DSM. Observations are made regarding unexpected difficulties in doing this using the currently available algorithms. In order to understand better the underlying reasons for the difficulty in identifying architecture in an algorithmic manner a series of simplified theoretical abstract architectures are presented and evaluated using a variety of different representations. These are compared with the architecture of the gen-set in order to determine whether the practical difficulties can be explained by the suggested theory.

The third set of activities is to create the non-DSM parameters for the gen-set and to take a first cut at implementing the option valuation. After examining the results it is necessary to return to the algorithms for evaluating the DSM and define a series of terms that assist in resolving both the expected and unexpected difficulties. These are then applied to the gen-set using modified algorithms and the results discussed.

Fourthly a discussion is included of the extent to which the question value creation has been (and can be) answered, and of the domain issues in conducting further work. This is followed by a discussion of the proposed methodology for the question of identifying value capture.

Overall this suggests that strategic leadership of technology conglomerates must be by people who possess either the tacit knowledge of the financial, organisational and technical aspects of the business, or who possess explicit tools to bridge the gap. Given that explicit financial tools are available, in the absence of unique individuals the strategic planning process needs to incorporate measures designed to a priori check that the proposed strategies will result in knowledge creation and value capture. The option value of an architecture represents a potential tool for holistically expressing this value and so can be used to substitute explicit language for tacit knowledge.

System Design and Management Principles

The ability to objectively and quantitatively describe any architecture is an important first step in selecting between competing alternatives. Architectures are defined as spanning multiple domains and the principles of decomposition and aggregation are applied in the identified dominant domain. This requires a holistic view of the entire architecture in order to identify the appropriate domain over the relevant life cycle. It also requires a systematic understanding and recording of the relationships between the primitive elements at a lower level of decomposition that are arranged to form an architecture at a higher level.

Engineering & Management Content

In order to create the generic example used as an integrating test case representative of an integrated architecture it is necessary to understand a gas turbine generator set from a multi-disciplinary engineering perspective that includes its entire life cycle. Professional gas turbine manufacturers and operators' insights are incorporated in the development of the synthetic example

In exploring the difficulties in describing the architecture of the gen-set through the medium of DSMs it is necessary to understand real-world engineering practices so as to remain focussed on the relevant issues rather than interesting but irrelevant issues. This allows for the reduction of the salient points to abstract cartoons that may then be analysed to reveal the underlying points more clearly. Without a sound understanding of the underlying engineering principles and processes it would not have been possible to confidently switch between the generic gen-set example and the simplified cartoons, nor to navigate the various domains.

The management content of this work is the creation of a theoretical valuation model that measures the intrinsic value of an architecture from a holistic perspective. By understanding the nature of a firm, the nature of a contract, and the micro-economic issues involved in applying game theory the logical extensions of this theory are proposed.

The theoretical model is crudely validated against real-world data to reveal whether it is directionally worth persevering with for more detailed examples.

As an aside product complexity is measurable as a result of performing this analysis and inter and intra-firm organisational recommendations are made as appropriate.

Statement of Authorship and Originality

The work performed to write this thesis is the author's, and is original.

The proposed extension of societal valuation to value capture by applying multi-player multi-stage game theory was independently developed by this author, however subsequent discussions with Professor Baldwin of HBS revealed that this was also Baldwin & Clark's previously intended extension path.