A Multiple Viewpoint Modular Design Methodology

by

Joanne Stuart Smith

A thesis submitted for the degree of

Doctor of Philosophy

CAD Centre

Department of Design, Manufacture and Engineering Management

University of Strathclyde

Glasgow, Scotland, UK

2002

Acknowledgements

I would like to take this opportunity to express my sincere gratitude to a number of people who have supported me during my PhD study.

First of all, I would like to say thank you to my supervisor, Dr Alex Duffy, for his patient supervision and his persistent use of the question 'why?' I have greatly benefited from his professionalism and drive for excellence. My PhD journey, under the guidance of Dr Duffy, has taught me more about research, professionalism, and personal boundaries than can ever be reflected in the written words of this thesis.

My thanks are due to the EPSRC for their financial support during the first three years of my study.

Many thanks to the Forward Design Group (FDG) within BAE Systems Marine Ltd., where I was based for a considerable portion of the last 3 years, for their provision of both financial support and design expertise. I have gained a great deal of experience from the active involvement of the FDG members in my work and wish to extend my appreciation to each and every one of them. Special thanks go to Mr Charles Nisbet, the FDG manager, who instigated the collaborative work, and Dr Malcolm Robb whom I continually 'pestered' for information.

Many thanks too to Dr Phil Jones for presenting the opportunity to carry out another industrial case study within the Battle Group Thermal Imager (BGTI) team at Thales Optronics Ltd. Special thanks go to Mr Martin Orr, Mr John Monhan, and Mr Kenneth Baxter for their keen interest and involvement in providing information for the case study, completing related workshops and evaluation procedures.

My thanks also go to the research students and staff of the CAD Centre, and the wider DMEM department, for the hard-working but friendly environment that they create. Special thanks go to Dr Ian Whitfield, of the 'CAD Centre Clan', for the generous contribution of his 'programming' skills and his continued enthusiasm in supporting the work presented in this thesis.

My sincere thanks to all my friends for their constant and unconditional support throughout my PhD study. A special mention to Shona and Julie who endeavoured to keep my feet on the ground, my body healthy, and my mind sane with the occasional 10K run, frequent girly chats, and the constant supply of chilled white wine and chocolate.

My heartfelt thanks goes to my dad who, in the absence of my mother, has taken on a dual role in my life. He manages to balance fatherly advice with motherly kindness and I could not express my gratitude to him in words. I can only hope that my path through life reflects the excellent upbringing which he and my mother provided and that I make him proud.

Lastly, my thanks go to Paul who has shown me that true love does not have boundaries, it does not judge, it does not restrict, it does not expect, but rather opens up a world of possibilities and provides a strong foundation from which to embrace these. I may not have embarked on this journey with him, but I would not want to start or complete any other without him.

Abbreviations

Unless stated explicitly, the following abbreviations are used in this thesis.

Abbreviation	Meaning
AMOAN	Actual Mode of Action Network
ANN	Artificial Neural Networks
BGTI	Battle Group Thermal Imager
CA	Components and Assemblies
CAD	Computer Aided Design
CAPU	Central Processor and Control Unit
CBR	Case Based Reasoning
CC	Clustering Criteria
CN	Construction Network
CWK	Current Working Knowledge
DK	Domain Knowledge
DMOAN	Desired Mode of Action Network
DP	Design Parameters
DSM	Dependency Structure Matrix
EDR	Engineering Design Re-use
ES	Expert System
FDP	Functional Dependency Perspective
FR	Functional Requirements
GA	Genetic Algorithm
IDA	Intelligent Design Assistant
ITM	Integrated Technology Mast
K.I.D	Knowledge, Information and Data
KM	Knowledge Modularity
KMn	Knowledge Modularisation

Abbreviation	Meaning
MD	Modular Design
MI	Module Identification
MSM	Module Structure Matrix
MSVM	Modular Structure Viewpoint Models
MVEA	Multi-Viewpoint Evolutionary Approach
PDM	Product Data Management
PDP	Product Development Process
PFA	Product Family Architecture
PLP	Physical Link Perspective
PS	Product Structuring
SDN	Similarity Dependence Network
SDP	Similarity Dependence Perspective
SLS	System Level Specifications
SMA	Service Mode Analysis

Symbols

Unless stated explicitly, the following symbols are used in this thesis.

Symbol	Meaning
Logical Connectives	
Iff	if and only if
$\wedge$	and – connective (conjunction)
$\vee$	or –connective (disjunction)
$\neg$	not-connective (negation)
$\equiv$	equivalent quantifier ('there exists' or 'some' quantifier)
Quantifiers	
$\exists$	existential quantifier
Sets	
{...}	a set of some elements
$\{C_1, C_2, \ldots, C_n\}$	explicit representation of a set that consists of elements $C_1, C_2, \ldots, C_n$
{C}	representation of a set that consists of element type C
General Operators	
$\sum$	sum of
$R(C_i, C_j)$	a relation R from element C_i to C_j

Abstract

Engineering Design Re-use refers to the utilisation of any knowledge gained from the design activity to support future design. As such, Engineering Design Re-use approaches are concerned with the support, exploration and enhancement of design knowledge prior, during and after a design activity. Modular Design is a product structuring principle whereby products are developed with distinct modules for rapid product development, efficient upgrades, and possible re-use (of the physical modules). The benefits of Modular Design centre of a greater capacity for structuring component parts to better manage the relation between market requirements and the designed product. This work explores the capabilities of Modular Design principles to provide improved support for the Engineering Design Re-use concept. The Modular Design principle is extended to structure not only the artefact's components but also their associated knowledge, to support, explore and enhance the knowledge generated during the evolution of the design process.

A novel modular design approach, termed a Multi-Viewpoint Modular Design Methodology, is developed to address identified requirements including; support for evolutionary design knowledge, exploration and identification of inherent modularity and maintenance of the modular solution. The overall concept of the Methodology is to support the designer in evolving a modular artefact whilst utilising the principles of modularity to structure the artefact knowledge to enhance its potential applicability for re-use, the concept is termed *knowledge modularity*. Based on the results of a state of the art review deficiencies of existing approaches are identified including; insufficient support of evolutionary design knowledge, insufficiencies in the modelling, exploration, identification and representation of *knowledge modularity,* limitations in the module identification process. Declarative and procedural knowledge is developed to define a novel Modular Design Methodology to address these deficiencies. As such, the Methodology presents a formalised approach to support the modelling, optimisation and identification of modularity, both within and across viewpoints (*function, working principle* and *structure*) of the product structure, and evolutionary design knowledge. The core phenomena of a *knowledge module* is formalised in terms of the knowledge of design concepts and their dependencies. The formalism supports the identification of inherent modularity. An alternative model, termed the Modular Structure Matrix is developed as part of the Methodology to represent this inherent modularity. In addition, the Methodology has been developed, through a 12-month industrial residency, to address the requirements of practising designers.

The Methodology is applied throughout a design activity to formalise and represent (in a matrix formalism) knowledge of the concepts embodied by a design artefact. The resulting model provides the basis to determine and represent interdependency knowledge between design concepts. The modelled concept and dependency knowledge can be utilised to support a modular analysis of the product structure both within and across design viewpoints. An optimisation and module identification mechanism can then be applied to the model and, based on the dependency data, identify inherent modularity within individual viewpoints of the product structure. Further, a mapping methodology has been developed to support the maintenance of the modular solution, and its associated artefact knowledge, across multiple viewpoints of design. The new methodology can be applied in a cyclic and iterative manner to support modularisation of the artefact design knowledge through the evolution of the design.

A computational implementation has been developed to aid the evaluation of the Methodology. The functionality of the Methodology has been illustrated through two literature based case studies and two industrial implementation evaluations. An implementation and evaluation methodology was formalised through the rationalisation of the activities carried out during the first, and further utilised as the basis to support the

second, industrial implementation. The two literature based studies evaluate the functionality of the methodologies optimisation and module identification mechanisms. These evaluations result in the identification of modular hierarchies that were not evident in the findings of the original publications. In addition, both industrial implementations result in the identification of potential improvements in the design. The evaluations illustrate the functionality of the Methodology in identifying and maintaining modularity, structuring design knowledge, supporting decision-making, learning, and improving design understanding. In addition, the evaluators outlined further potential Methodology application fields such as team design, manufacturing design and technology life-cycle management.

Further the strengths and weaknesses of the Methodology, the computational implementation, and the research methodology utilised to facilitate the work presented in this thesis, are discussed. Finally, future work required to enhance the capabilities of the Multi-Viewpoint MD methodology and the functionality of the computational implementation have been identified, including; the development of more advanced modular clustering criterions, the introduction of constraints and constraint management, and the development of module costing mechanisms/metrics.

Contents

List of Figures

List of Tables

1 Introduction

'Our society, like many advanced economies in the developed world, is experiencing deep change as we move from an industrial age into what is often called the knowledge economy' (Hargreaves 2001). Knowledge is increasingly viewed as a resource central to the success of a company. For example, 80% of UK businesses said that knowledge was either one of their most or their most important asset (Council 2000/2001). As such, knowledge capture, maintenance and management techniques are becoming increasingly important to business success. However, only 16% of businesses in the UK have any 'knowledge sharing initiatives' in place within their company (Council 2000/2001).

Design can be considered a 'knowledge intensive activity' as during the activity of designing knowledge is generated, utilised and evolved to further the product definition. Design is often a gradual and iterative process (Andreasen 1991). Design knowledge is generated and evolves both within and across a number of *viewpoints* of design (*function, working principle,* and *structure*) (Andreasen 1991; Zhang 1998) to define a *completed design model* that realises the specified *design requirements.* Engineering Design Re-use (EDR) is an approach concerned with the utilisation of this generated design knowledge to support future design. Engineering Design Re-use approaches can be applied prior, during and after the design activity. Engineering Design Re-use is a total approach which encompasses the processes of *designing by re-use,* i.e. the re-use of previously acquired concepts in a new design situation, *designing for re-use,* i.e. the identification and extraction of potentially re-usable knowledge fragments during a current design activity, and *exploration of the domain,* i.e. the examination of the design domain to identify, rationalise and extract reusable fragments of knowledge.

A product can only be *designed by re-use* if re-usable sources of knowledge are available through, for example, *designing for re-use* or *exploration of the domain. Designing for re-use* requires that knowledge generated during the current design process, known as *current working knowledge* (CWK), is supported. Based on the nature of the design process itself the *current working knowledge* of a design evolves over time in a gradual and iterative manner. An *evolved product model* represents the embodiment of the *current working knowledge* at a distinct moment in time. *Domain Exploration* is the process whereby the characteristics of a domain are conceptualised from available sources of *domain knowledge* (DK), i.e. past cases and general knowledge. Figure 1.1 illustrates the processes of Engineering Design Re-use, the design knowledge utilised within each process and the context of the work presented in this thesis.

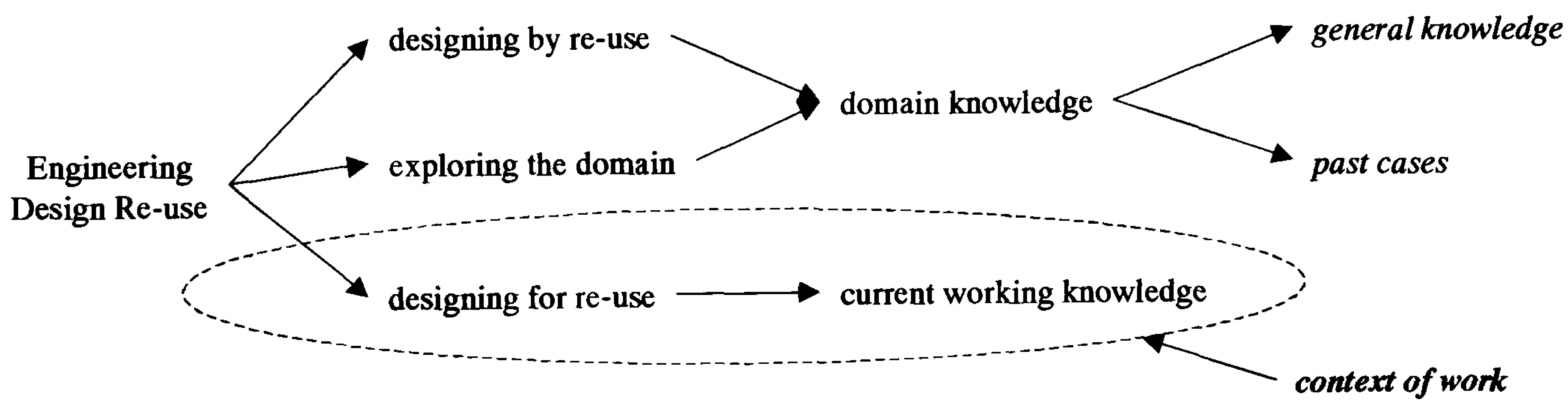


Figure 1.1: The context of the work

Designing for re-use requires a conscious effort to identify, extract, enhance and record design knowledge throughout the evolution of the design. Thus, the evolutionary design knowledge requires to be modelled, represented and analysed by some means to enhance its knowledge content and provide a structured resource for potential re-use.

Product Structuring (PS) concerns the activity whereby the characteristics of a design are defined (Andreasen, Duffy et al. 1995; Erens and Verhulst 1995). Product Structuring, applied to a current design process, can be utilised to support, maintain and enhance the design knowledge resource and facilitate *designing for re-use*. Modular Design (MD) is a product structuring principle that is synonymous with the design of distinct detachable modules for rapid product development, efficient upgrades and potential long term re-use (Gu, Hashemian et al. 1997). The benefits of modular design centre on a greater capacity for structuring both physical parts and product knowledge to better manage the relation between market requirements and the design product. Hence, modular design supports increased utilisation of experiential knowledge for new product development and can thus provide an approach on which to actively support re-use.

Modular design principles have been applied to the structure the *designed product* (Ishii, Lee et al. 1995; Erixon 1996; Gershenson and Prasad 1997; Huang and Kusiak 1998; Muffato 1999; Salheih and Kamrani 1999; Gonzalez-Zugasti and Otto 2000; Jarventausta and Pulkkinen 2001; Otto 2001) and its' associated *production system* (Zhou and Irani; Rogers and Bottaci 1997; He and Kusiak 1998; Miller 1999). Modular design approaches are applied to explore and identify modularity within *individual products* (Ishii, Lee et al. 1995; Erixon 1996; Gershenson and Prasad 1997; Huang and Kusiak 1998; Salheih and Kamrani 1999; Jarventausta and Pulkkinen 2001) or across a *product family* (or generations of a product family) (Muffato 1999; Gonzalez-Zugasti and Otto 2000; Otto 2001). Modular design applied across a product family is aimed at *exploring the domain* to identify common characteristics in the product structures that can be realised as a common *module*. A set of modules common to a product family is often referred to as a *platform*. New product family members are generated through the *selection* and/or *configuration* of modules from the *module platform* (Ouyang, Chenggang et al. 1996; Tseng and Jiao 1997; Juengst and Heinrich 1998; O'Grady and Liang 1998; Siddique and Rosen 2001). However, the focus of this work is on the development of a modular design methodology to support the process of *designing for re-use* and is thus concerned with modularisation of *current working knowledge* generated for an *individual product*. Thus, the work presented in this thesis focuses on the modularisation of artefact design knowledge during the evolution of the design activity.

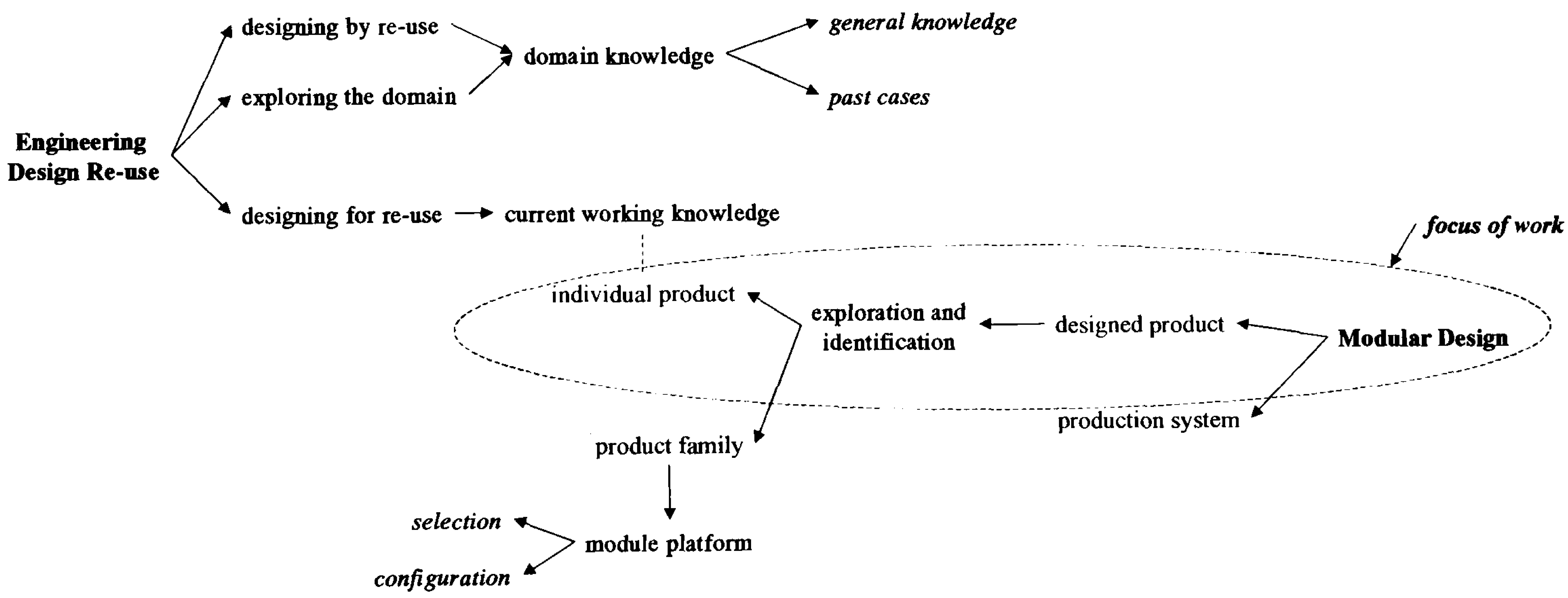


Figure 1.2: The focus of the work

In general modular design principles are applied to the components of a design and as such supports only one *viewpoint* of design, referred to hereafter as the *structure viewpoint*. This current limitation of the modular design principle results in a lack of support for the evolution of the design knowledge and, consequently the modular solution. Further, research has shown that unlike in the field of software design, where the re-use of chunks of code, adders and microprocessors has met with considerable success (Lubars 1991; Jones 1995), the re-use of previous engineering designs in their entirety does not generally meet with the comparable success (Sivaloganathan and Shanin 1998). Research has illustrated that 'patching' the engineering design 'to fit' can require such considerable effort that it cancels out 'so much of the advantage of reusing it that it may be easier to design from scratch' (Mostow, Barley et al. 1993). This is attributed to the Engineering Design Domain dealing with more abstract concepts (MacCallum, Duffy et al. 1995) and that the current formal documentation is generally based on a low level of abstraction that does not represent this abstract knowledge and its evolution through the design process (Finger 1998). It is suggested that the current application of modular design principles to support the *structure viewpoint* only, also fails to explicitly support the more abstract knowledge related to *function* and *working principle* concepts and alternatives to these. As such, it is difficult to facilitate the process of *designing for re-use* based on the utilisation of current modular design principles. In addition, there are a number of barriers to the implementation of modular design in practice (Chang and Ward 1995; Sosale, Hashemiam et al. 1997; Burke and Miller 1998; Huang and Kusiak 1998; Juengst and Heinrich 1998; Miller and Elgard 1998; O'Grady and Liang 1998; Philippi 1998; Miller 1999), including:

- A gap in the research communities understanding of the core phenomenon of modular design itself in relation to their understanding of the benefits it may provide.
- The treatment of modularity in an abstract form in literature, i.e. a lack of formalisation of the phenomena itself.
- Unsatisfactory exploration of the principle in industry.

The above deficiencies have resulted in a lack of design theories and tools in the mechanical world that serve as articulate procedures for modular design practitioners to follow. Due to the lack of articulate modular design procedures and explicit knowledge related to the nature of the designed artefact, there is little support to aid the designers in decision making to define the modular design. Thus, 'approaches are needed to determine modules, represent modularity, optimise modular design and assess the impact of modularity on the design process, manufacturing and management' (Huang and Kusiak 1998). Therefore, in order to facilitate *designing for re-use* based on the principles of modular there is a requirement to:

- Provide formal support for the evolution of the current working knowledge throughout the design process.
- Determine the knowledge requirements to support the exploration and representation of modularity both with and across the differing viewpoints of the design, i.e. the evolution of the modular design.
- Model this evolutionary design knowledge in such a means so as to facilitate the designer in exploring the modularity of the product structure and its impact on various phases of the product lifecycle.
- Formalise the core phenomena of modular design itself and utilise this formalisation to support the designer in determining the modularity of the product structure.

As such, the work presented in this thesis aims to establish an modular design methodology to overcome the limitations of existing approaches to modular design and to satisfy the requirements of modular design for improved Engineering Design Re-use support.

1.1 Motivation

The work presented in this thesis has been motivated by the aim of achieving the associated benefits of a re-use approach to engineering design. The provision of formalised support for the processes of re-use is seen as a key to successfully realising re-use benefits in terms of cost, time, quality and performance. Motivated by the initial establishment of a relationship between the principles of Modular Design (MD) and that of Engineering Design Re-use (EDR) the work defines and develops a holistic modular design methodology.

1.2 Research scope

The work covered in this thesis aims to establish a modular design methodology to support engineering design that overcomes the limitations of existing modular design approaches and satisfies the requirements of Engineering Design Re-use support. The scope of the research with respect to the Engineering Design Process is as follows:

- Product lifecycle

  The design activity phase of the product lifecycle is the main focus of the work, in particular, the evolution of the product definition from the requirement to the solution structure. At this stage of the research the work focuses on supporting the evolution of a modular design and its associated design knowledge. However, the utilisation of the design activity phase to facilitate other product lifecycle objectives (ease of manufacture, assembly, maintenance, refit) is considered as part of this research.

- Application area

  The work is concerned with the development of a generic approach to support the activity of design within the engineering design domain and concerns the modular design of individual product instances. As such, the rationalisation of product families and/or generations of these are not considered as part of this work. Further, the work is concerned with the identification, exploration and maintenance of modularity within the product structure of the evolving design to enhance its knowledge content. The selection and/or configuration of predefined modules are not considered as part of the work presented in this thesis.

  In addition, despite the consideration of lifecycle objectives, the work does not directly support the modular design of the associated production systems.

- Design Domain

  The work focuses on the Engineering Design Domain with the evaluation being carried on two products from that domain: an 'Integrated Technology Mast' of a naval ship, and a 'Battle Group Thermal Imager' mounted in a battle vehicle. Though evaluated on military products the methodology and supporting tool defined throughout this thesis are developed such that they are generic in nature and applicable across the spectrum of the Engineering Design Domain. Other design domains, such as software, industrial or chemical, are not covered by this work.

1.3 Research Methodology

An adaptation of the research methodology developed in the CAD Centre, University of Strathclyde, UK (Duffy and O'Donnell 1998) was utilised to undertake the work presented in this thesis. This methodology was chosen as the basis for conducting the research presented in this thesis for a number of reasons, including;

- It was founded on the requirement to conduct effective research into the Intelligent Design Assistant (IDA) philosophy (MacCallum, Duffy et al. 1987). The Intelligent Design Assistant philosophy refers to the over-riding focus of the research centre where this research was undertaken.
- It was developed to map the specific requirements of conducting research in the engineering design field with an emphasis on industry based design practice.
- Its successful application to a number of previous research studies in the engineering design field (Manfaat, Duffy et al. 1998; Zhang 1998) has verified the methodology as a valid and appropriate approach on which to facilitate research in this area.

Figure 1.3 illustrates the main flow of the methodology.

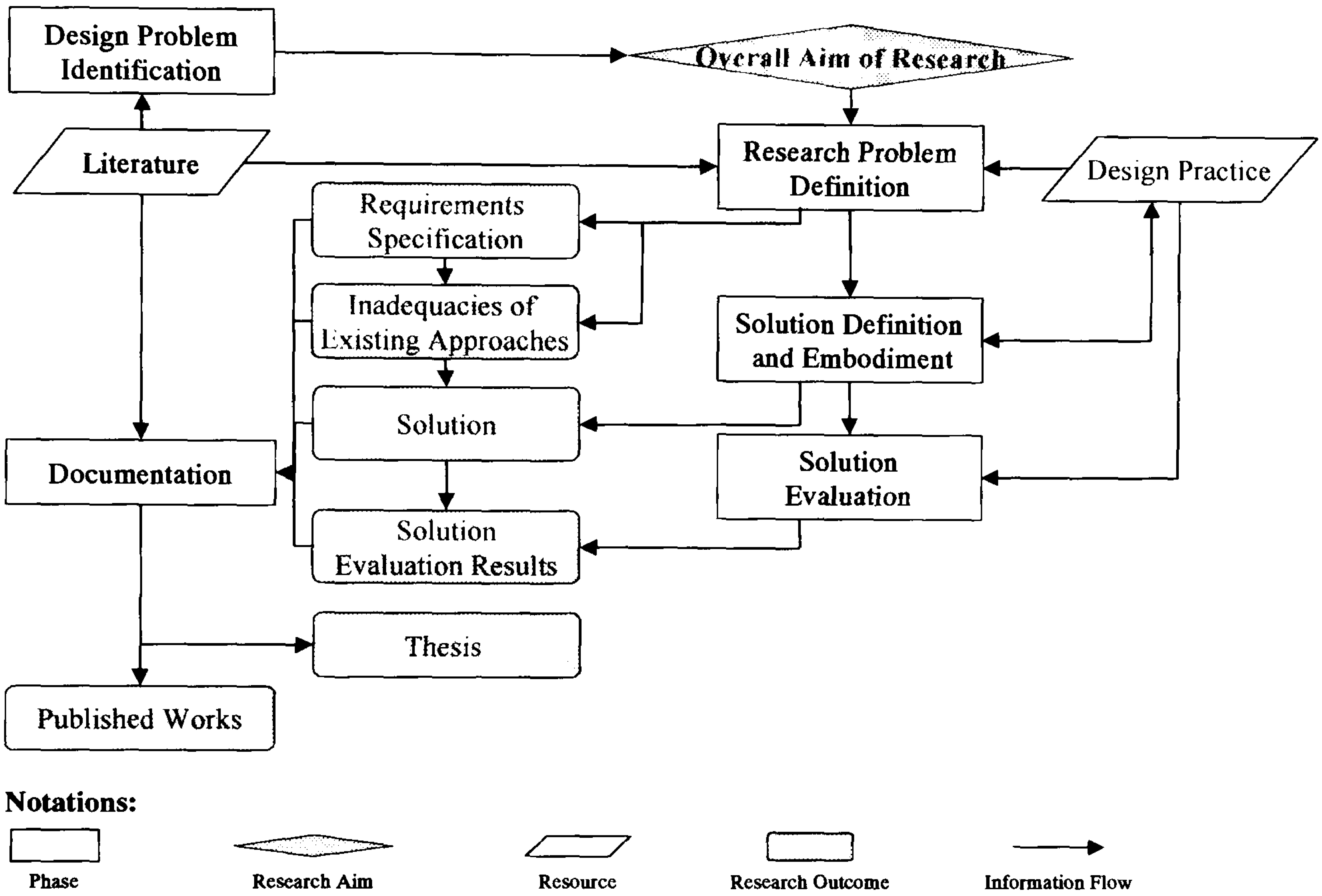


Figure 1.3: Research Methodology (adapted from (Duffy and O'Donnell 1998))

The *Design Problem Identification* phase is conducted, based on literature resources, to identify the overall aim of the research. This overall aim is utilised as the basis to conduct the remaining three phase; *Research Problem Formalisation, Solution Development and Embodiment,* and *Solution Evaluation.* Based on the overall aim of the research the *Research Problem Formalisation* considers both *literature* based research and *design practice* to identify a *Requirements Specification* and to identify the *Inadequacies of Existing Approaches* based on these defined requirements. The *Solution Definition and Embodiment* phase details an approach aimed at overcoming the *Inadequacies of Existing Approaches* and addressing the requirements defined in the *Specification* whilst considering the resulting solutions application to *design practice.* The resulting *Solution* undergoes an *Evaluation* phase where its capabilities with respect to the *Requirements Specification* are verified based on its application to facilitate *design practice.* Throughout the entire research process the outcomes of the research phases are *documented* and utilised to produce *Published Work* to aid in the dissemination of the research finding to engineering design research fields and industries at large. Similar to the process of design, research is an iterative process in nature, however, this is not illustrated in Figure 1.3.

1.4 Aims and objectives

The overall aim of this research is to establish a modular design methodology to support multi-viewpoint modularisation of designs within the engineering design environment. Adopting the methodology described in Figure 1.3 the following objectives are identified:

- To disclose the characteristics of Engineering Design Re-use including;
  - the relation of Engineering Design Re-use to the process of design and the knowledge generated during design,
  - the potential of Engineering Design Re-use to support different phases of the design process and different types of design,
  - the potential benefits of Engineering Design Re-use and the impact of the differing characteristics of Engineering Design Re-use on these, and
  - the limitations of existing Engineering Design Re-use supporting approaches.
- To disclose the characteristics of Modular Design (MD), including;
  - the relation of modular design to the process of Engineering Design Re-use,
  - the relation of modular design to the engineering design process, and
  - the limitations of current modular design approaches with respect to engineering design process and Engineering Design Re-use.
- To identify the requirements for an improved Engineering Design Re-use support mechanism based on the characteristics of modular design and considering the current limitations associated with modular design support for Engineering Design Re-use.
- To develop and define a Modular Design Methodology for Engineering Design Re-use support. The methodology intends to overcome the limitations of existing modular design approaches and satisfy the identified requirements.
- To evaluate the functionality of the developed approach. The approach will be evaluated through its partial realisation within a computational support tool and its implementation within two industry based engineering design processes.
- To analyse the strengths and weaknesses of the developed approach, based on the evaluation results, and considering the requirements for improved modular design support for Engineering Design Re-use.
- To identify avenues of future research based on the current research findings.

1.5 Thesis structure

The remainder of the thesis is structured into three parts, as follows.

PART 1. Research Problem Definition (Chapters 2, 3, 4)

Chapter 2 discloses the characteristics of Engineering Design Re-use with respect to the activity of engineering design. Engineering Design Re-use is discussed with respect to its potential benefits, the knowledge issues which arise from formalising an approach to Engineering Design Re-use, and the applicability of Engineering Design Re-use within the engineering design activity. The gaps in current Engineering Design Re-use support are disclosed and their effect on the potential to achieve Engineering Design Re-use related benefits is highlighted.

Chapter 3 discloses the characteristics of the field of Product Structuring with specific emphasis on Modular Design as it is shown to readily map to the requirements of improved Engineering Design Re-use support. The basic requirements for a modular design methodology to support improved Engineering Design Re-use are outlined.

Chapter 4 critically reviews existing modular design methodologies in order to identify the strengths and weaknesses of current support. Areas for further research are identified based on the highlighted deficiencies of existing methodologies.

PART 2. Solution Definition and Embodiment (Chapters 5, 6)

Chapter 5 outlines the declarative and procedural knowledge that defined the novel Multi-Viewpoint Modular Design Methodology developed as part of the work presented in this thesis. The chapter provides an overview of the main elements, which constitute the modular design methodology, and the envisaged procedure for its application during design.

Chapter 6 details the formalisms and mechanisms embodied within each of the elements of the Multi-Viewpoint Modular Design Methodology.

PART 3. Evaluation, Discussion, Conclusion (Chapters 7, 8, 9, 10)

Chapter 7 details the computational realisation of the main elements of the methodology. The computational implementation is developed to support the evaluation of the proposed methodology.

Chapter 8 details the evaluation of the methodology based on the findings of four studies, two studies are extracted from literature and two industry applications. The two literature-based studies illustrate the methodologies additional functionality with respect to current MD approaches whilst the two industry implementation studies: an Integrated Technology Mast (ITM) for a naval ship, and a Battle Group Thermal Imager (BGTI) for a combat vehicle) evaluate the capabilities of the methodology as a whole.

Chapter 9 discusses the strengths and weaknesses of the proposed methodology based on the evaluation results obtained in Chapter 8 and the observations obtained during the research process. Future work to enhance the capabilities of the proposed methodology and the functionality of the computational support tool is highlighted.

Chapter 10 concludes the thesis outlining the main findings and outcomes of the work presented in this thesis.

Part 1 – Research Problem Definition

2 Engineering Design Re-use

Chapter 1 introduced the concept of EDR and outlined its applicability in the 'knowledge' driven environment in which companies currently operate. The main aim of the following chapter is to disclose the characteristics of EDR and its relation to the process of design and knowledge generated during design. The identified characteristics will be used to highlight the potential of EDR to support different design phases, and types, and the potential benefits of EDR and its processes. Finally, the current gaps in support for EDR will be highlighted to provide a context for the remainder of the work documented within this thesis.

2.1 Background

Gao et al, (Gao, Zeid et al. 1998) estimated that '90% of industrial design activity is based on variant design' and in such a redesign case '70% of the information is re-used from previous solutions' (Khadilkar and Stauffer 1996). Thus, when a new design problem arises, it is frequently solved through modifying an existing design rather than creating a completely new design. Characteristically designers do not 're-invent the wheel' every time a new design instance calls for one, their natural response is to glean from past experience and *re-use* previously acquired knowledge. The concept of *re-using* is inherent within the natural process of design. However, this *re-use* process is ad-hoc and relies on the natural inclination of the individual designer. The origins of formal design re-use practices are found in the realms of software engineering where designers, 'faced with increasing complexity and time-to-market pressures, began to consider re-use as a realistic solution to their problems' (Jones 1995). Similarly, the engineering design domain was faced with increasing product complexity and 'a design process itself constrained by requirements of cost and time' (Ormerod, Mariani et al. 1997). This resulted in an increased focus on the applicability of re-use in the engineering design domain. In the software engineering domain the re-use of large chunks of code became commonplace and re-use libraries holding proven building blocks such as 'low-complexity' adders, to 'high level' microprocessors and customisable cores, were conceived (Lubars 1991; Jones 1995). Taking its lead from the software engineering domain, the initial focus of EDR centred on specific and/or standard parts. However, the engineering design domain deals 'with more abstract concepts' (MacCallum, Duffy et al. 1995) and consequently it is not always possible to re-use a previous engineering design in its entirety (Smith and Duffy 2001). Thus, standard components began to be developed 'to enable both the re-use of the part and the experience associated with that part' (Culley 1998). This concept of *re-use* in the context of engineering design was further extended by Finger (Finger 1998) who stated that 'designers may re-use a prior design in its entirety, ... may re-use an existing shape for a different function, or may re-use a feature from another design'. Within this work, EDR is taken to its natural conclusion and defined as the:

' total approach which supports the utilisation of any knowledge gained from a design activity'.

The key features of this definition are that EDR is:

- considered to be a total approach, i.e. it embodies all *processes* required to utilise generated design knowledge from all phases of the design lifecycle.

- concerned with *knowledge* gained from the design activity, i.e. not just past designs or artefacts[1] as in work by (Culley 1998)
- concerns the *utilisation* of knowledge, i.e. it is concerned with its application to design, as opposed to its regurgitation in design.

2.2 The advantages and disadvantages of supporting EDR

The following section covers the potential advantages and disadvantages of supporting EDR.

2.2.1 Advantages

It is interesting, initially, to consider the current re-use benefits achieved in the field of software design, where formalised *re-use* originated, as a relatively more mature *re-use* research area. For instance, a cost model developed for the software industry by Synopsis Inc. Figure 2.1 highlights the increasing costs of chip design and the widening gap between design costs utilising 'formal' re-use and current practice. The chart shows that chip designers who fail to take advantage of 'formal design re-use' practices face 'unsustainable cost increases, of up to 64 times higher' (Inc 1999).

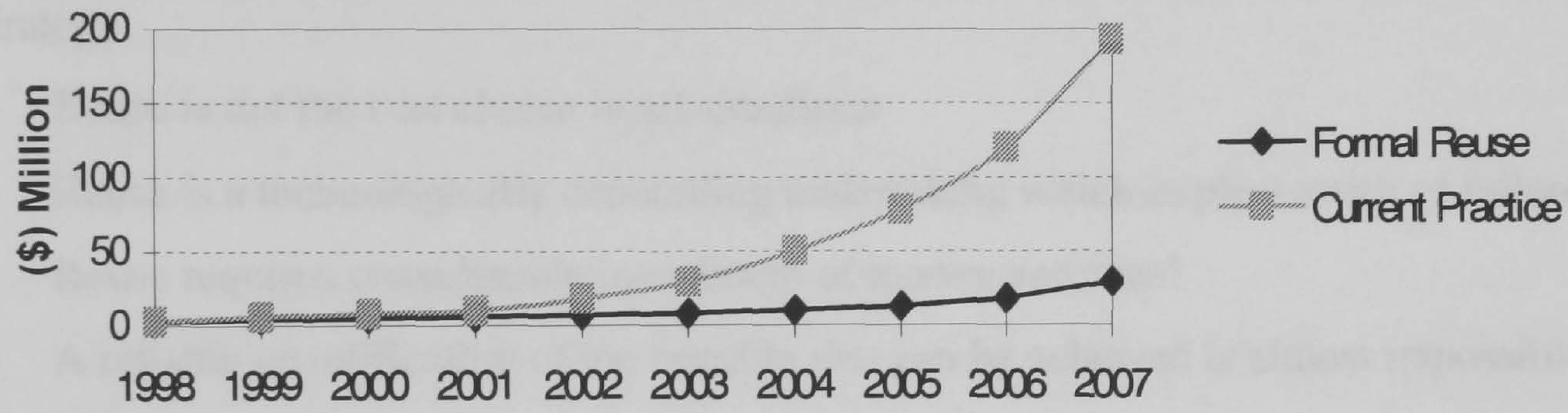


Figure 2.1: Who can afford a $193 million chip? (Inc 1999)

Similarly cost was amongst the main *re-use* benefits analysed in a study in the engineering design sector (Duffy and Ferns 1999). The study also considered benefits in the areas of performance, quality, and time. The study concluded that the potential benefits to an industrial company, of applying an overall re-use approach, far exceeded the benefits they currently received from relying on designers' natural inclination to re-use. Figure 2.2 summarises the benefits analysis finding.

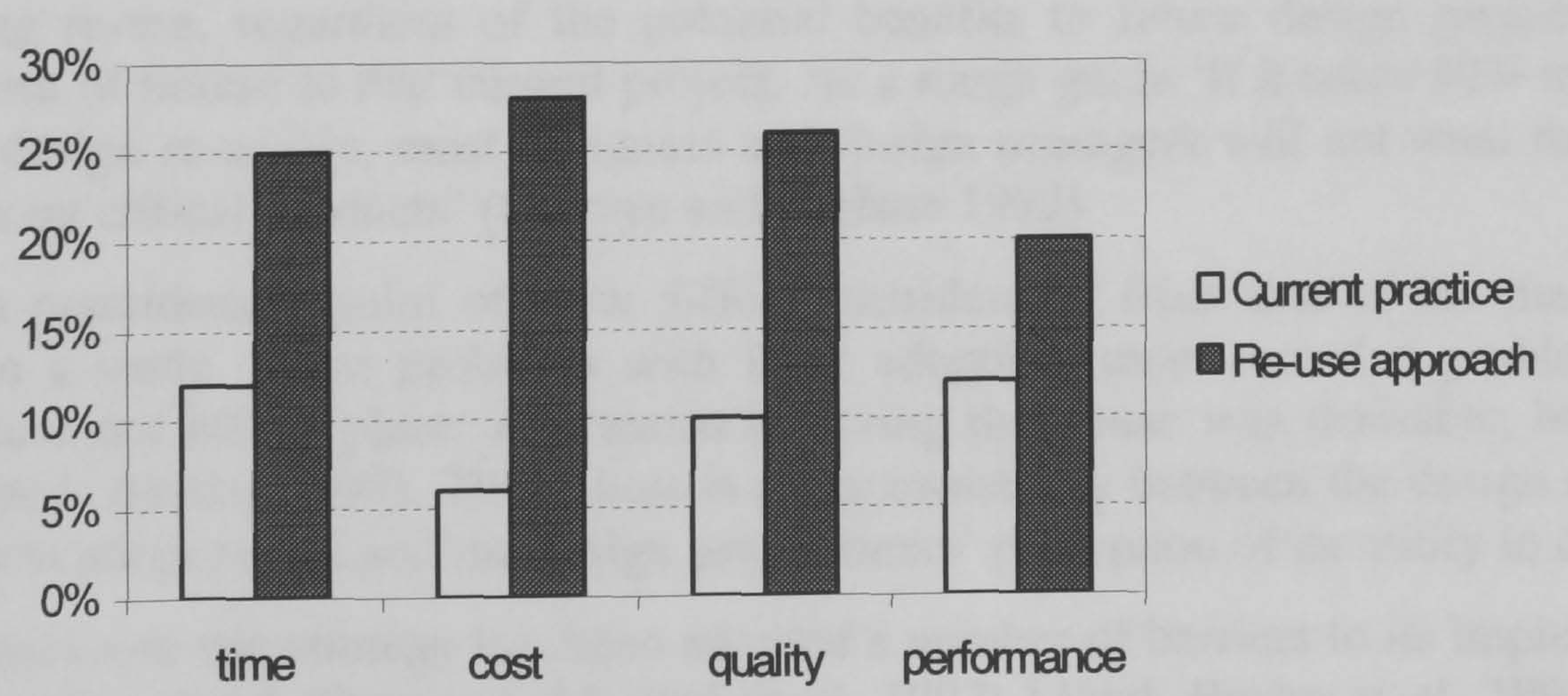


Figure 2.2: Current and foreseen overall Re-use benefits (Duffy and Ferns 1999).

[1] A product in the context of this work defines an artefact or system that represents the output from the activity of engineering design.

It can be seen that the first column in each category (time, cost, quality and performance), shows the benefits received from current 'ad-hoc' re-use practices. These provide greatest benefit to time and performance whilst the costs benefits are almost half at only 6%. The second column in each category indicates the foreseen benefits of a formal approach to 'design re-use' which can be expected to provide overall benefits to time, cost, quality and performance in the region of 20% to 28%. For instance, this can be translated into an improvement in terms of cost benefits, over current practice, of around 360%. The study substantiates that formalised EDR approaches, methodologies and systems can achieve considerable potential benefits.

2.2.2 Disadvantages

Despite the evidence as to the benefits of EDR, 'design organisations seem to find reuse surprisingly difficult' (Busby 1999). The reason for the lack of re-use in the engineering design domain is difficult to ascertain, because as stated by (Busby 1999) 'it is hard to find studies of reuse that investigate it in the organisational environment in which most design is done'. Thus, again the software design re-use experiences are considered as a starting point to examine the re-use adoption prohibitions. For example in a study on software design re-use (Trauter 1998) states that there are four main reasons why companies do not adopt a re-use strategy:

- Reuse is not the best choice in all situations
- Reuse is a technologically demanding undertaking which implies a risk of failure!
- Reuse requires considerable investment of money and time!
- A reliable quantification of the benefits that can be achieved is almost impossible!

This highlights the dilemma faced by design managers in adopting re-use in terms of the benefits versus the risks. The design manager is faced with the knowledge that re-use can yield significant benefits but that it not always appropriate and that it is technologically demanding. Further, the design manager has no reliable quantification as to benefits within their design field. In addition 'producing designs suitable for re-use slows down the design process, increases costs' (Jones 1995) as 'obviously the performance, cost and quality of the parts must be good enough to make design reuse successful' (Girczyc and Carlson 1993). By their nature the benefits of design re-use are often not realised until a later generation of the product. Thus, it could be argued that the design manager may not want to take responsibility for adopting re-use, regardless of the potential benefits to future design projects, due to increased risk of failure to that current project. As a rough guide 'if it takes 30% more effort to make a design re-usable, most designers and design managers will not want to incur the added costs on critical products' (Girczyc and Carlson 1993)

The design practitioners point of view differs considerably from that of the manager, for example, in a study of the problems with EDR adoption, most recorded problems 'were cases of reuse not taking place: informants believing that reuse was desirable, but had not been practised' (Busby 1999). Thus, there is an inconsistency between the design managers' willingness to adopt re-use and the design practitioners' perception of its utility in design.

In cases where a re-use strategy has been adopted a number of barriers to its implementation have been encountered (Ormerod, Mariani et al. 1997; Lloyd, Busby et al. 1998; Trauter 1998; Busby 1999), including:

- Process discontinuities

  Discontinuities in the organisation's design information systems has resulted in disproportionate levels of effort having to be devoted to reuse

- Knowledge issues

  knowledge sharing

  There are considerable discrepancies between the number of design documents available and their usefulness to the re-use process. In addition, differences in how people learn from their experience, and the context of the experience itself, results in expert designer holding different opinions about optimal solutions and the process of achieving these.

  design fixation and lack of innovation

  Design fixation, termed 'overusing' by Lloyd, occurs when designers continually re-use an existing design without revisiting the problem. This can lead to the same problems occurring over and over again when design faults are not documented. In addition, the re-use phenomena where part or all of a solution to a previous design problem is used to develop a solution to a current problem this is often seen by designers as at odds with efforts to introduce innovative ideas into the design process.

Thus, it can be argued that within the engineering design field there can be significant value in re-using knowledge and experience related to existing designs to support future design. However, to overcome many of the problems associated with re-use adoption and implementation it is suggested that a greater understanding of the process and knowledge issues related to re-use at the organisational level is required. In addition, there is a need to formalise EDR approaches, methodologies and systems to support the achievement of the considerable potential benefits attributed to this approach. Thus, providing greater knowledge and formal approaches to support the identification of appropriate re-use application fields and allowing the design manager to undertake re-use implementation with minimised risks.

2.3 Knowledge issues in providing 'formal support'

Based on the previous definition of EDR given in Section 2.1 we consider the issues involved in utilising sources of design knowledge for re-use. Gao (Gao, Zeid et al. 1998) states that because of the 'complexity and the rich knowledge involved, there is significant value in re-using an existing design'. Weighted against the potential benefits (Section 2.2) is the principle that a stored design '99% right for a given task, often takes much more than 1% of the effort needed to create the design in the first place to patch it to fit, cancelling so much of the advantage of reusing it that it may be easier to design from scratch' (Mostow, Barley et al. 1993). Thus, we identify fundamental issues in providing support for a *re-use* approach: (i) modelling and managing, even for a relatively simple product, the complex and rich design related knowledge, and (ii) providing solutions to the problems of partially re-using previous design solutions and their associated knowledge to effectively satisfy new design requirements. Accordingly, to obtain the maximum benefits of formal *re-use* requires that we optimise support for this rich and complex knowledge resource, appreciate the source, nature, and growth of design knowledge, and successfully manage knowledge acquisition, maintenance and utilisation for *re-use*. Thus, supporting *re-use* requires that we can ascertain: what knowledge can be re-used; how it can be maintained to maximise its applicability; and where and when it can be utilised in new design.

2.3.1 Design knowledge in EDR

Design knowledge is often referred to as complex (Gao, Zeid et al. 1998). The notion of complexity is one topic that has received great attention, from mathematicians, scientists, and engineers alike, due to a lack of common definition and concrete theories (Suh 1999)[2] Firstly, we must consider the factors contributing to design knowledge complexity and secondly how these effect its' ability to be re-used. Duffy defines the factors influencing complexity, and their associated issues, through the 'design complexity map' (Duffy 1995). A simplified version of this map is shown in Figure 2.3.[3]

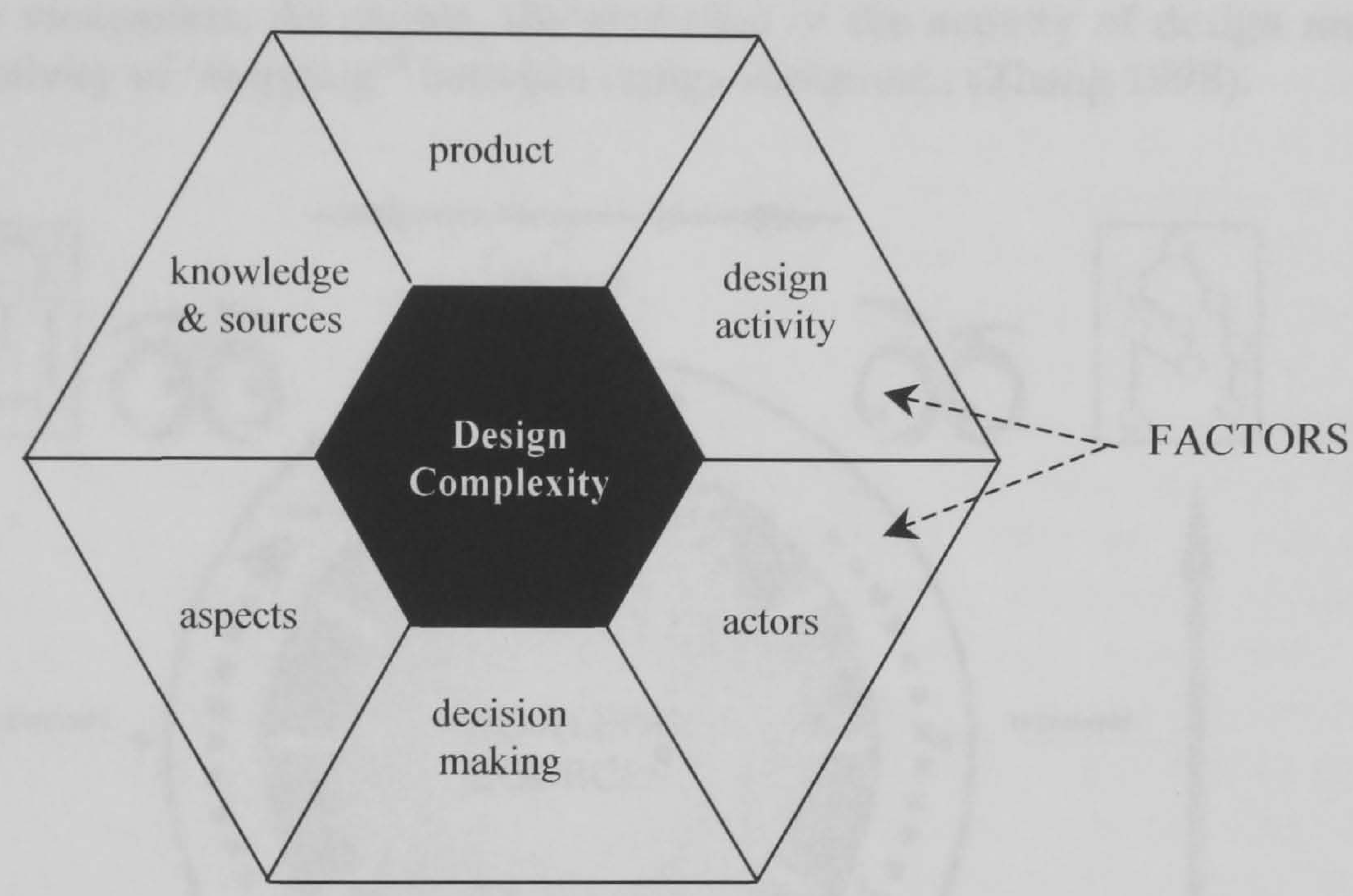


Figure 2.3: Design Complexity Map (Duffy 1995)

We can view complexity in design in such diverse factors as the product being designed, the design activity itself, the actors involved, the decision making process, the aspects impinging on the design, and knowledge and sources used and generated. Moreover, the issues affecting each of these factors further compound complexity. Even the simplest product may be associated with a complex array of factors, which shape the activity of design and consequently the final product definition, and result in a vast accumulation of related design knowledge. The final product definition is dependent on: amongst others, the company organisation; the type of design; the chosen design process, designers, and tools; and external factors out-with the designers control (Duffy 1995). Thus, modelling knowledge for *re-use* is a complex issue in that there are a number of factors, and their associated issues, to take into consideration during this process. A problem further amplified by the differences in terms of characteristics, types, sources, forms and origins of design related knowledge. To further complicate matters, design related knowledge can be considered from many *viewpoints* such as; functional, structural, and behavioural (Gero and Maher 1990; Bhatta and Goel 1992).

[2] A comprehensive review of complexity theories is beyond the scope of this work, however the reader is referred to Suh, N. P. (1999). "A Theory of Complexity, Periodicity and the Design Axioms." Research in Engineering Design **11**: 116-131..

[3] For the full map and a detailed explanation it and its relevance to engineering design knowledge the reader is referred to Duffy, S. M. (1995). The Design Coordination Framework and the Design Complexity Map. Integrated Production Systems, IPS Research Seminar. Fuglsocentret, Fuglso, Institute for Engineering Design,Technical University of Denmark, Lyngby, Denmark..

Brice and Johns (Brice and Johns 1998), Finger (Finger 1998), and Mostow (Mostow, Barley et al. 1993), also emphasise the importance of knowledge related to the 'why' and 'when' of decision making, known as the *rationale* and *history*. Brice and Johns (Brice and Johns 1998) conclude that 'constructive use of design rationale will become an integral part of the design process'.

Figure 2.4 illustrates a knowledge model for re-use in which there is a bi-directional flow between the design activity and the differing knowledge sources (as depicted in Figure 2.3). The bi-directional flow indicates that the designer can both utilise knowledge sources to evolve the design activity and generate new knowledge sources as part of that design activity. The designer may consider design knowledge from different *viewpoints* throughout the evolution of the design activity and consequently may utilise and generate knowledge sources within these *viewpoints*. As shown, the evolution of the activity of design may be represented by the activity of '*mapping*'[4] between design *viewpoints* (Zhang 1998).

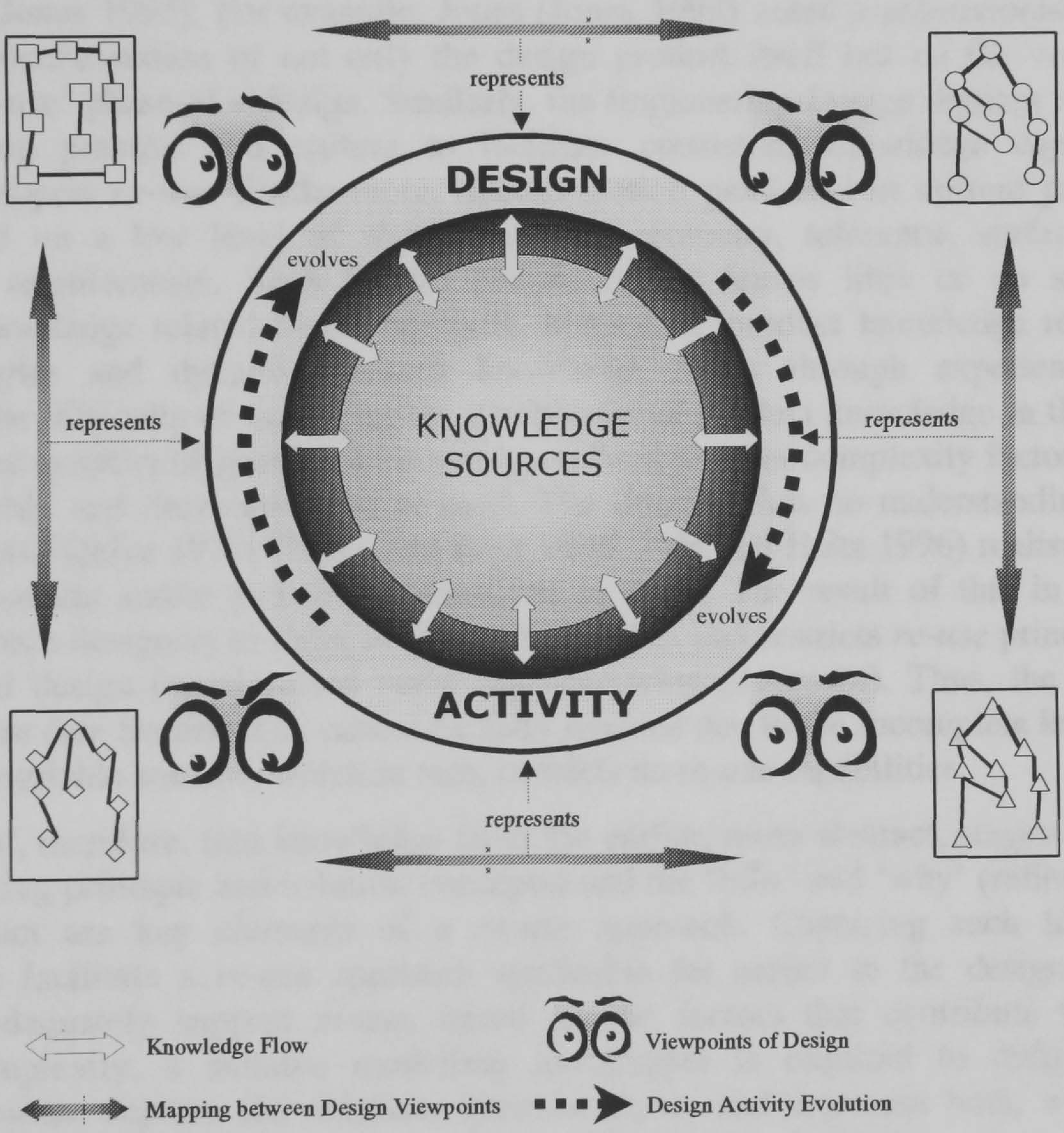


Figure 2.4: A model of design knowledge for re-use

Thus, it can be argued that to adequately support future *re-use* of generated design knowledge a formal modelling mechanism is required to support knowledge generated during the design activity itself. Such a modelling mechanism would provide the designer

[4] The *mapping* activity finds a design concept in one viewpoint say, the structural viewpoint, as the realisation of a concept in another viewpoint, for example, functional i.e. '*mapping*' a concept evolves the working design from one viewpoint to another.

with a greater knowledge resource to utilise to *explore the domain* and support further *designing by re-use.*

Despite our increasing understanding of the utility of consistent capture of design knowledge throughout the design activity, 'typically the only formal documentation is the final set of drawings' (Finger 1998). Additionally, current design tools have the effect, on the practice of engineering design, of emphasising 'those parts of the process that were well understood and/or easily systematised (e.g. detailed design, analysis, machine path planning), while minimising those parts that were less well understood (e.g., problem definition, synthesis and conceptual design)'(Finger 1998). Consequently, the earlier design phases, where more general, abstract knowledge is generated, are not adequately supported. The importance of supporting earlier phases is illustrated when we consider that at the end of the conceptual phase approximately 80% of the lifecycle costs are already committed. Thus, non-capture would negate many of the potential cost benefits of re-use. The requirement for a cultural as well as technical shift when implementing re-use practices has also been noted in the software field (Jones 1995). For example, Jones (Jones 1995) noted a requirement to focus on first-class documentation of not only the design product itself but on the verification process and 'in-use' phase of a design. Similarly, the Engineering Design domain requires a shift in working practice and culture to facilitate consistent knowledge capture and consequently support *re-use*. Furthermore, documentation generated in current practice is generally based on a low level of abstraction e.g. geometry, tolerance, surface finish, manufacturing requirements. Such formal documentation leaves little or no scope for representing knowledge related to the rationale, history, or product knowledge relating to concept principles and dynamic process knowledge learnt through experience. This contributes to the difficulty of managing the complexity of product knowledge in that a vast proportion of the quantity of generated knowledge related to other complexity factors are not captured explicitly and thus cannot be re-used. The designer has no understanding of the solution concepts (Tjalve 1979; Hubka and Eder 1988; Pahl and Beitz 1996) realised by the products components and/or possible alternatives to these. The result of this in a re-use scenario is to force designers to think in terms of specifics and restricts *re-use* principally to support detailed design (standardised parts, manufacturing processes). Thus, the potential benefits of *re-use* (see Section 2.2) cannot be fully realised due to the incomplete knowledge content of the available sources, which in turn, restricts its *re-use* capabilities.

It can be argued, therefore, that knowledge from the earlier, more abstract, stages of design (function, working principle and solution concepts) and the 'how' and 'why' (rationale) of a designed product are key elements of a *re-use* approach. Capturing such high level knowledge can facilitate a *re-use* approach applicable far earlier in the design process. Therefore to adequately support re-use, based on the factors that contribute to design knowledge complexity, a suitable modelling mechanism is required to define *design knowledge elements,* capture the *relations* between these, and represent both, within and across different *sources* and *viewpoints*, whilst modelling the *behaviour* of these as the product definition evolves.

2.3.2 Utilising knowledge in design re-use

Mostow, et al (Mostow, Barley et al. 1993) debate whether re-using a previous design can be justified in terms of the additional effort required to make it 'fit'. However, they concede that despite the difficulties of partial re-use, human designers do 'modify the structure of a design to fit a new application' but maintain that 'this process tends to require considerable expertise'. Evidently, it is not always possible to re-use a previous design in its entirety and modifying a design to fit can involve expertise beyond the scope of explicitly available design knowledge. However, as the utilisation of previously acquired design knowledge in new design is central to the success of a re-use approach there is a need to overcome such problems by supporting and maintaining knowledge acquired through design experience (explicit and implicit) to support its application to, as opposed to its regurgitation in design.

As the designer is seen to adapt previous solutions to solve a new problem by learning from experience, an effective *re-use* approach must encompass elements of this 'learning' process to extend the approach's ability to utilise design knowledge and support re-use of partial solutions (Mostow, Barley et al. 1993). Within this work, (as shown in Figure 2.5) learning is considered to be the process of acquiring new knowledge, the modification of existing knowledge and the generation of new knowledge.

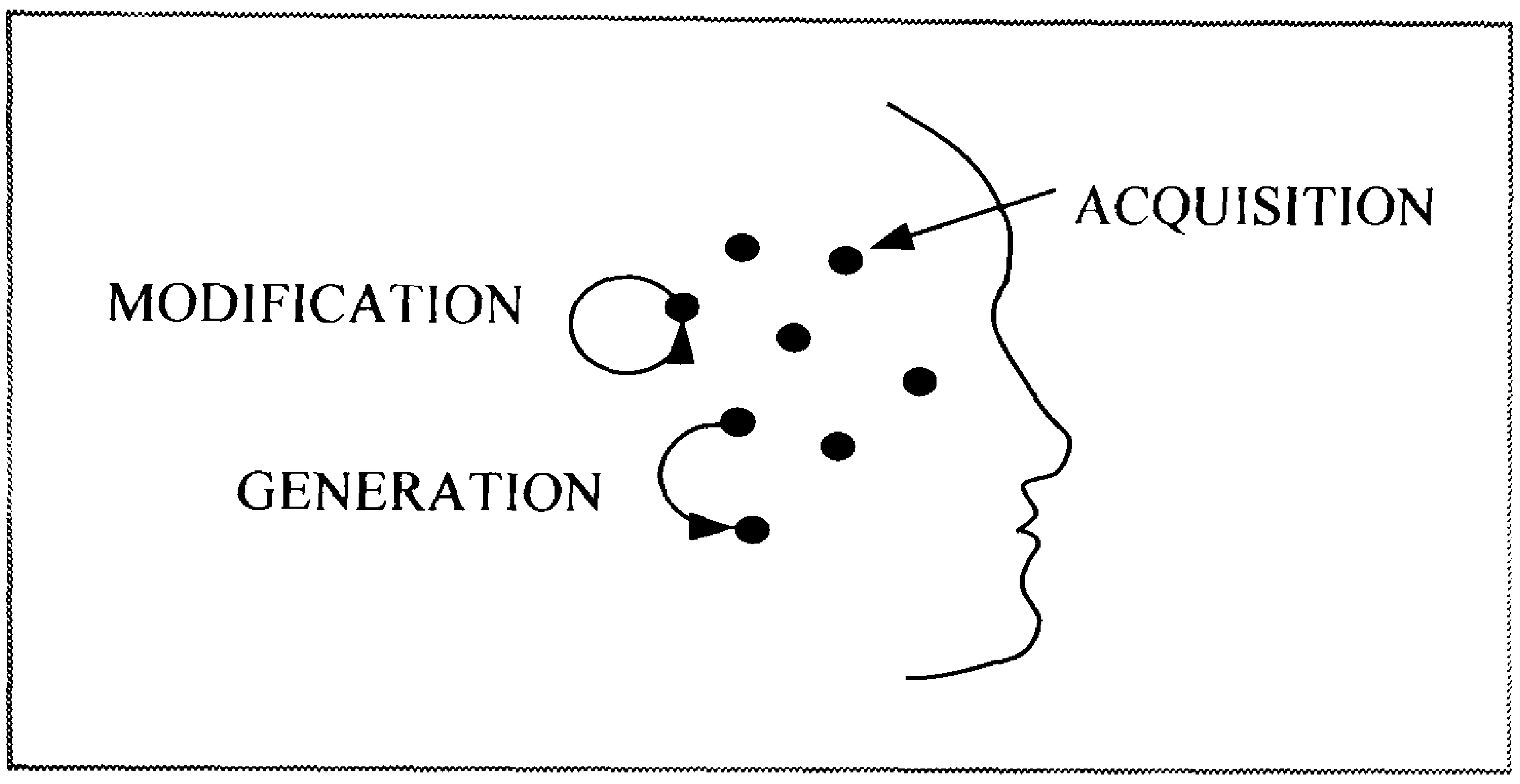


Figure 2.5: The learning activities (Duffy 1997)

Duffy (Duffy 1997) states that 'learning helps to maintain experiential knowledge'. Experiential knowledge represents one of the most powerful resources a designer possesses (Kerr 1993). Activities such as 'abstraction and generalisation' help to maintain knowledge (Manfaat, Duffy et al. 1998). They promote the flexibility of experiences by removing highly specific details and generating more generally applicable knowledge. Such maintenance of experiential design knowledge, through abstraction from the specific to general, supports the dynamic nature of knowledge and prevents knowledge related to design experiences from becoming static and obsolete. Thus, the learning activity extends the utilisation capabilities of knowledge for 'designing by re-use'. Thus, the process of learning: the acquisition, generation and modification of knowledge, alleviates some of the difficulties associated with the application of knowledge from 'a design that is not 100% 'right' to a new situation'.

2.3.3 The applicability of knowledge re-use

EDR has been established as an approach concerned with the provision of support for the capture, management and subsequent utilisation of design knowledge, gained from previous experience, to further new design (Sivaloganathan and Shanin 1998). However, what applicability does such an approach have within the field of design itself?

In the design process

Re-use models or paradigms such as (Fothergill, Lacunza et al. 1994; Henninger 1996; Altmeyer and Noll 1998; Gao, Zeid et al. 1998; Deneux and Wang 2000) consider *re-use* as an approach solely concerned with aiding new *design by re-using* past design solutions. However, the concept can be further extended to include not only this phenomena of *designing by re-using* but also that of *designing for re-using*, whereby the design process and product solutions are specifically managed and created to promote their future *re-use*. Indeed an industry based brainstorming program (Shanin, Andrews et al. 1998) highlighted the need for capturing design information into the design re-use system whilst a design is being carried out. The definition of EDR posited within this work addresses this need as it considers re-use as a total approach which, with the support of well developed tools and methods, can encompass all phases of the design life cycle. Thus, a comprehensive EDR approach should emphasise support, management and utilisation of design knowledge prior, during and after the completion of a design activity.

In different design types

Research (Duffy, Duffy et al. 1995; Sivaloganathan 1998; Busby 1999; Smith 2000) into why designers' re-use experiential knowledge has highlighted the main issues as:

- an avoidance of previous design faults and unnecessary re-invention;
- duplication of design success; and
- the use of known and proven characteristics

This may suggest that re-use is relevant only to variant (repeat order) or adaptive (evolutionary) design as these issues are readily equated to the repetition, improvement and enhancement of an already existing design.

At first glance, the application of re-use to variant and evolutionary design is far more apparent than to that of original (innovative) design. However, 'competitive advantage is now obtained by innovation and creation of knowledge rather than access to financial or material capital' (Preiss 1999). Thus, it is important to determine the capabilities of *re-use* in the 'original' design environment in a bid to optimise the impact of *re-use* benefits in design.

Innovation is deemed to have occurred when either tacit (the collection of data, rules, which lie beyond the realms of explicit knowledge) or explicit knowledge (that which is readily accessible, written down, in computers, etc.) is converted to gain additional explicit knowledge, not previously available (Preiss 1999). Innovation in the design process can be considered as instances where 'new variables are introduced into the design process' and 'the state space is expanded' (Gero and Maher 1990). Thus, the conflict between re-use and innovation arises from the perception of *re-use* as merely an approach to support and maintain previously existing knowledge where innovation requires that new knowledge be created and/or added to the design process. For example; the perception of *re-use* as a facilitator of negative design fixation (a phenomena whereby designers re-use features to which they are regularly exposed), i.e. complacency, re-use of 'bad' design solutions, lack of technology transfer (Smith 2000).

The applicability of re-use to original (innovative) design is better appreciated when, as here, re-use is considered as a process capable of supporting existing knowledge while permitting abstraction and generalisation of this knowledge to generate or modify knowledge. Thus, the processes is synonymous with knowledge maintenance and learning can not only prevent its stagnation or obsolescence, increase its applicability to new design, but also support the *re-use* of design knowledge in original design environments. Further effective generalisation of design knowledge can increase applicability for design not only within a single domain but due to the removal of 'case and domain specific' knowledge it can also be utilised to effectively support the re-use of knowledge across distinct domains.

2.4 Models of Engineering Design Re-use

EDR has been established as a total approach to the activity of design which;

- is concerned with *the utilisation of any knowledge gained from a design activity and not just past designs of products.*

- concerns the provision of support for the acquisition, management and utilisation of sources of design knowledge, to further new design.
- is applicable prior, during and after the design activity.

Based on the above characteristics, the research community requires the formalisation of tools and methods to successfully achieve the associated benefits of EDR. To develop EDR tools requires that we limit the complexity involved in understanding it (Logan 1988). Thus, there is a need for a general model of EDR to clarify and define the approach from a theoretical point of view, similar to that noted by (Takeda, Veerkamp et al. 1990), with respect to models of design. A 'model' is perceived as the 'natural link between the physical world and our interpretation of it' and constitutes a 'representation of concepts that are based on the notion of typicality' (Donaldson and MacCallum 1994), or similarly 'an abstraction of reality' (Logan 1988). It is important to note that, models are dynamic in that they can only reflect our current understanding and hence, they may evolve based on increasing knowledge of the subject area. Thus, a general model of EDR should identify and formally represent generic *re-use* elements and determine the overall structure of an EDR approach based on our current understanding. The purpose of such a model is to 'direct our attention to those parts of the problem which together constitute a single conceptual piece' (Logan 1988) and allow researchers to reduce the complexity of providing support for these parts (Lee 1999).

Having established the need and purpose of a formal model of 'EDR', the following section considers existing *re-use* models and the extent to which they explicate the elements and relations within a contemporary 'EDR' approach as identified in the previous sections (2.1 to 2.3).

Relative to the field of design, where research into appropriate and adequate models is still ongoing, formalisation of EDR is at a relatively early stage. Thus, *re-use* models can be deemed to possess a relative immaturity that increases the difficulty associated with their analysis and categorisation. With this in mind the following presents a review of a representative sample of *re-use* models, which typify those presented by the re-use research community, and not an exhaustive evaluation of all models. A number of research works have modelled different aspects of re-use (Taleb-Bendiab 1992; Fothergill, Lacunza et al. 1994; Duffy, Duffy et al. 1995; Altmeyer and Schurmann 1996; Henninger 1996; Gao, Zeid et al. 1998; Perera and Watson 1998; Deneux and Wang 2000). The following reviews these with respect to their ability to fulfil the requirements of a general model of re-use as discussed in the opening paragraphs of section 2.4.

Case-Based Reasoning (CBR)

Paradigms of case-based reasoning (CBR) (Riesbeck and Shank 1989; Kolonder 1993) feature prominently in *re-use* models (Taleb-Bendiab 1992; Henninger 1996; Gao, Zeid et al. 1998; Perera and Watson 1998). The CBR approach 'formalises a computational model of problem solving based on memory organisation and reminding' and implements 'computer support for designers by applying recall processes to reuse the experience' (Maher and Garza 1997). The CBR paradigm assumes the availability of a large base of past design cases and predominantly considers issues related to computational representation of past design cases, recall and their direct reuse in a new design activity. Thus CBR lacks focus on issues such as knowledge acquisition during the design activity and utilisation as opposed to regurgitation of design knowledge, which devalues its capacity as a general model of *re-use.*

Gao et al (Henninger 1996; Gao, Zeid et al. 1998) attempt to overcome the limitations of CBR as a comprehensive model of *re-use* by combining the case paradigm with that of design plans (Gao, Zeid et al. 1998) and domain models (Henninger 1996). Gao et al (Gao, Zeid et al. 1998) maintains that the addition of a plan supports the need to 'model both the whole design process including design procedure (steps of design) and design rationale (intent of design)'. The focus of the realisation of this model is predominantly memory organisation, case recall and retrieval. Individually, each gives greater consideration to the inclusion of design rationale and modelling aspects of the design domain. However, neither explicitly models issues related to knowledge acquisition, supporting generated knowledge throughout the evolution of a design, nor further utilisation in new design.

Deneux's re-use model (Deneux and Wang 2000), again, concentrates solely on the retrieval process and is based on the requirements of a re-design process. The model formally represents designers' knowledge (experience) as a link between the product and process knowledge that can aid in the retrieval of potential solutions. The approach concentrates less on the underlying processes of re-use retrieval and more on the knowledge requirements for retrieval. Similarly, Altmeyer's re-use model (Altmeyer and Schurmann 1996), focuses on the retrieval and re-use of existing design cases, by formalising a generic design model and an indexing model for past cases. However, unlike many of the above approaches that consider elements of individual case selection and less frequently adaptation, this model focuses on the first step of case retrieval: finding a set of good cases as a starting point for case selection.

Each model represents an aspect of re-use (generally case representation and/or retrieval) from different viewpoints; Henninger has taken a cognitive approach (modelling designer's behaviour), Deneux's has modelled the knowledge based requirements, Altmeyer's prescribes a specific approach to the more generic re-use process (retrieval), and Gao's model focuses on the computational aspects of the retrieval process.

No individual CBR based model represents all the significant aspects of re-use, with most concentrating on a particular viewpoint of the organisation, recall and retrieval related aspects. The above models were developed to support the specific aspect of re-using past cases and, due to their tendency to be defined within the context of a particular approach or domain, they fall short as comprehensive models of re-use.

Holistic models

Duffy (Duffy, Duffy et al. 1995) and Fothergill (Fothergill, Lacunza et al. 1994) have modelled rather more holistic views of design *re-use*. Fothergill's model (Fothergill, Lacunza et al. 1994), further realised within the DEKLARE system (Fothergill, Lacunza et al. 1994; Saucier, Vargas et al. 1994; Vargas, Saucier et al. 1994; Vargas, Saucier et al. 1995), includes 3-stages: design analysis, design description and design advisory. The model considers such elements as function, product and process models, company and product history, design libraries and instantiation, and query of design knowledge. The DEKLARE model exhibits a more overall approach to *re-use* in comparison with the CBR based models. It considers *re-use* from acquisition to recall and retrieval, and supports modelling of design rationale and a variety of aspects related to the exploration and definition of the domain. Again, the approach relies on the availability of a number of past cases and the model is prescriptive in that it relates to a specific [re-use] tool and hence fails to explicitly represent the generic underlying re-use processes.

Duffy et al. (Duffy, Duffy et al. 1995) present a model of EDR which draws upon the foundations of a model from software engineering (Hall 1992) while tailoring it to EDR through the 'identification of existing design practices and through previous research into the use of experience in design' (Duffy, Duffy et al. 1995).

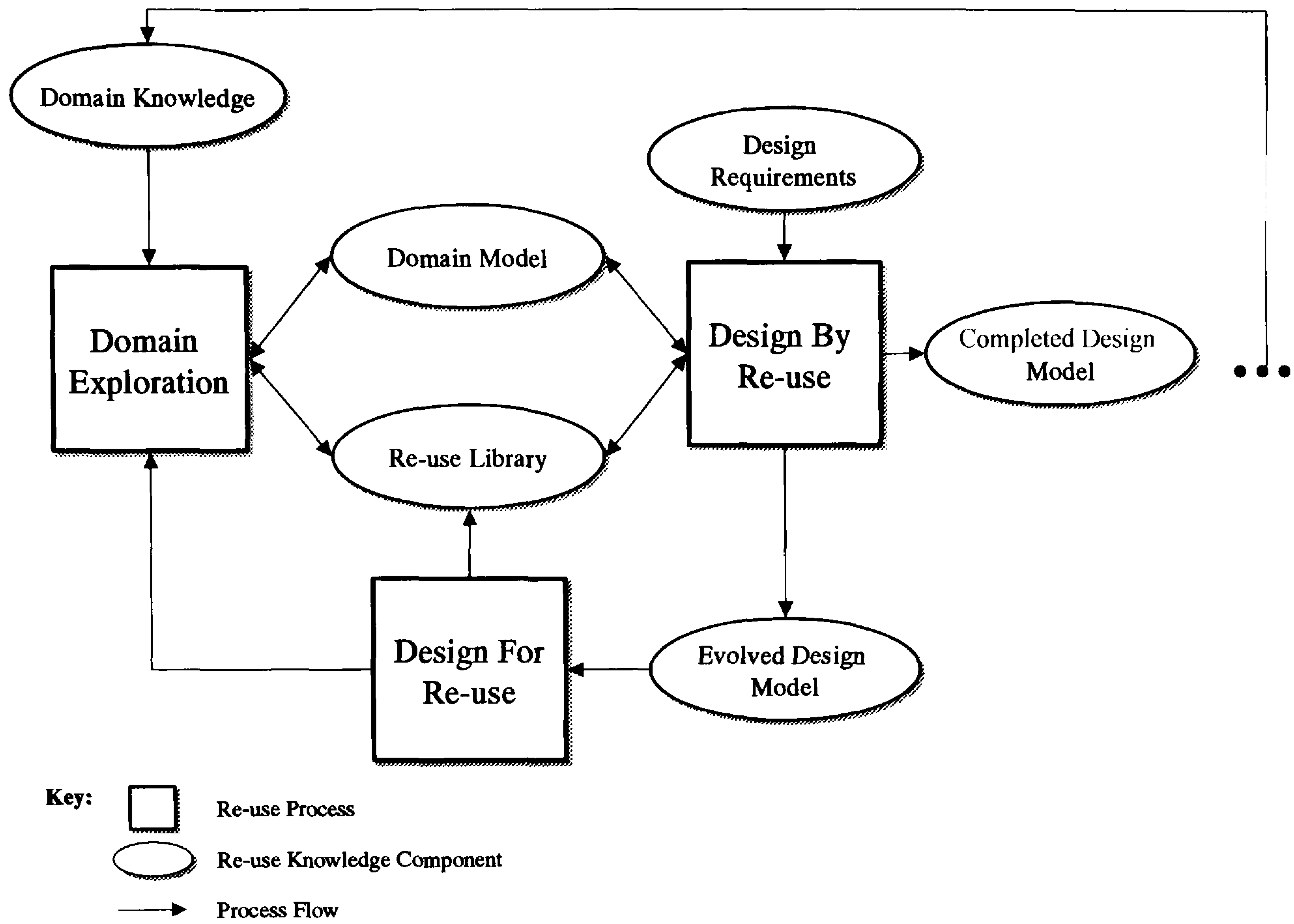


Figure 2.6: Design Re-use Process Model (Duffy, Duffy et al. 1995)

The model includes three processes that relate to the requirement for acquiring, managing and utilisation of knowledge prior, during and after design:

- ***Design by Re-use*** - The re-use of previously acquired concepts in a new design situation. The process, in which previously acquired knowledge is identified, retrieved and applied to the new design problem. *Design by Re-use* can only occur if re-usable resources are available through, for example, *Domain Exploration* and *Design for Re-use*. The retrieval and adaptation aspects of CBR are particular instantiations of *Design by Re-use.*

- ***Domain Exploration*** - The examination of a design domain from which re-usable fragments of knowledge can be identified, rationalised (structured), extracted, stored and subsequently used to develop new designs. The process through which characteristics of a domain are conceptualised, from which re-usable fragments of knowledge can be identified, extracted and stored for subsequent use in design.

- ***Design for Re-use*** - This process is carried out during design itself. Design For Re-use requires a conscious effort towards the identification, extraction and recording of possible reusable knowledge fragments of a current design and the enhancement of their knowledge content, including developed design alternatives, modifications and associated reasoning behind design decisions, for re-use.

In addition, the model includes the interaction (see Figure 2.6) of these processes with six knowledge components generated and/or utilised as part of the design activity itself:

- ***Design Requirements*** – A statement of design need that is used as a basis to stimulate design.

- ***Domain Knowledge*** - Knowledge pertaining to a design domain, e.g. existing product information, past design alternatives, potential solution alternatives. This component signifies the diverse and scattered items of knowledge or information that a designer can employ when designing.

- ***Re-use Library*** - An organised collection and store of knowledge. An important distinction is that a 're-use library' represents an organised store not a 'bin'.
- ***Domain Model*** - A representation of the designers' conceptualisation of the current design problem domain.
- ***Evolved Design Model*** - A statement of an evolved design, which may be at any level of abstraction, of an incomplete design or a final completed design.
- ***Completed Design Model*** - A completed statement fully defining a finished design solution that meets all the design requirements.

The model takes a holistic view of EDR and represents both the overriding processes of re-use and its knowledge requirements. Unlike many of the above models the re-use process is portrayed as ongoing and cyclic. The 'design re-use process model' presents a high level overview of EDR emphasising the processes, knowledge components and models[5]. However, due to its emphasis on the *phenomena of re-use* as opposed to a specific aspect of *reusing knowledge*, such as recall (Maher, Garza et al. 1997; Gao, Zeid et al. 1998), and/or a specific approach, such as computational support (Gao, Zeid et al. 1998), the 'design re-use process model' is deemed by the author to most adequately reflect the EDR characteristics as defined through sections 2.1 to 2.3.

The 'design re-use process model' represents the most holistic model of EDR phenomena presented to the research community to date. As such, it posited that such a formalisation as the 'design re-use process model' provides a cohesive framework within which to ascertain the capabilities and issues in current EDR support.

2.5 Support issues

The focus of the work presented in this thesis is on the development of improved support for EDR. As such, the following section identifies the discrepancies and gaps between current EDR support and the elements outlined in the 'design re-use process model' with the aim of providing a context for the remainder of the work presented in this thesis. The 'design re-use process model' (outlined in Section 2.4) is utilised, as a holistic model of *re-use*, as the basis for a number of reviews of current support for EDR (Duffy, Smith et al. 1998; Smith and Duffy 2001; Smith 2002). The following section discusses the reviews main findings. Elements, of the EDR approach, that currently lack support are identified and the significance of such findings in relation to potential EDR benefits is determined.

2.5.1 Existing support

Critical reviews of support for EDR have been carried out as part of this research work. The scope, content and volume of the combined reviews are too great for them to be included in their entirety as part of this thesis. As such the following section presents their main findings. For a more detailed examination of the aim, scope and findings of the reviews the reader is directed to (Duffy, Smith et al. 1998; Smith and Duffy 2001; Smith 2002).

---

[5] For a more detailed description of the generation and significance of the 'design re-use process model' the reader is directed to Duffy, S. M., A. H. B. Duffy, et al. (1995). A Design Reuse Model. International Conference On Engineering Design, Praha, Duffy, A. H. B. and S. M. Duffy (1996). "Learning for Design Reuse." Artificial Intelligence in Engineering Design, Analysis and Manufacturing **10**: 139-142..

The reviews covered the following areas: computational support, engineering design research; and product structuring. In all cases the research was mapped to the 'design re-use process model' shown in Figure 2.6. The aim of each review was the exploration of how current research in the particular field of investigation met the requirements of a *re-use* approach to engineering design. Though each review resulted in conclusions specific to the area of investigation, the general finding resulted in the identification of an overriding trend in EDR support. All three reviews concluded that the main contribution of current research and computational support was to the process of *Design by Re-use* and in consequence the *Re-use Library* knowledge component. The remaining two processes of *Design for Re-use* and *Domain Exploration* were shown to be lacking support in all three areas. The knowledge component *Evolved Design Model* also exhibited low levels of support in all three reviews. This is not altogether surprising due to its intrinsic link to the *Design for Re-use* process.

The main finding of the reviews was that despite a profusion of approaches with design, which claimed to support re-use, the research community appears to exhibit a significant lack of understanding of the overall process of *re-use* resulting in a compartmentalisation of research efforts. The significance of such compartmentalisation of *re-use* research is only fully appreciated when we consider this finding in relation to the potential benefits of re-use.

2.5.2 Lack of support and EDR benefits

A study by Duffy and Ferns (Duffy and Ferns 1999) utilised the 'design re-use process model' (Figure 2.6) as a basis to ascertain the potential benefits of EDR. The study analysed seven areas of engineering design within a naval shipbuilders; electrical, structural, heating and ventilation, weight assessment, resistance and propulsion, platform and forward design (innovation and future development). Each area was analysed with respect to the processes and components of the 'design re-use process model' to ascertain the associated benefits of applicable current and potential future *re-use* activities. The findings substantiated that the greatest foreseeable benefits were attributable to the process of *Design for Re-use* as shown in Figure 2.7.

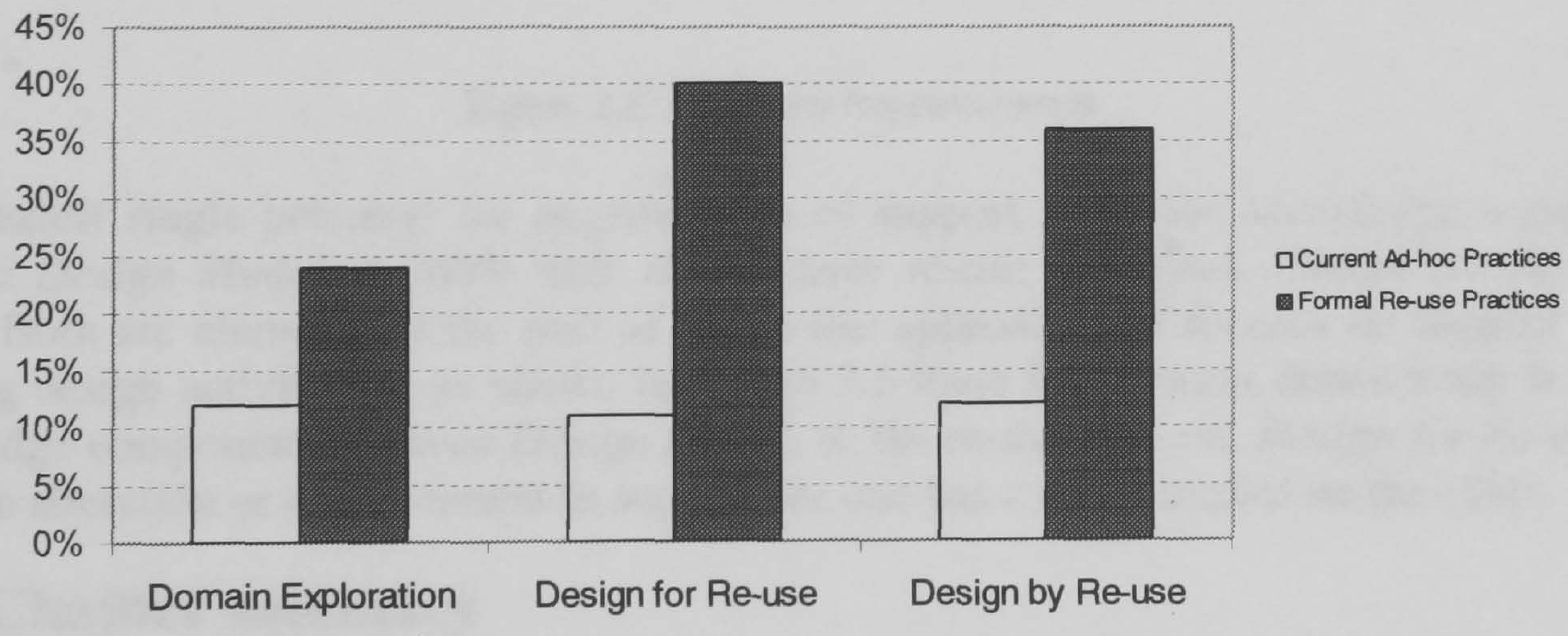


Figure 2.7: Overall Design Re-use process benefits (Duffy and Ferns 1999)

Figure 2.7 illustrates the overall benefits in terms of cost, performance, quality and time of current ad-hoc *re-use* practice (the first column) versus the foreseen overall benefits of formal *re-use* practices (the second column) for each *re-use* process. We can see from the first column in each set that the current benefits from all three *re-use* processes are similar at around 11% - 12%. However, the analysis highlighted that the greatest potential overall benefits of supporting *re-use* were from *Design for Re-use* at 40%.

The analysis also verified the requirement to improve research support for particular *re-use* processes and knowledge components. Figure 2.8 indicates the foreseen potential for improvement i.e. the relation between current and formalised re-use practices for each of the *components* and *processes* of a re-use approach to engineering design. No significant foreseen improvement was found for the component *domain knowledge* and as such it is not included in Figure 2.8.

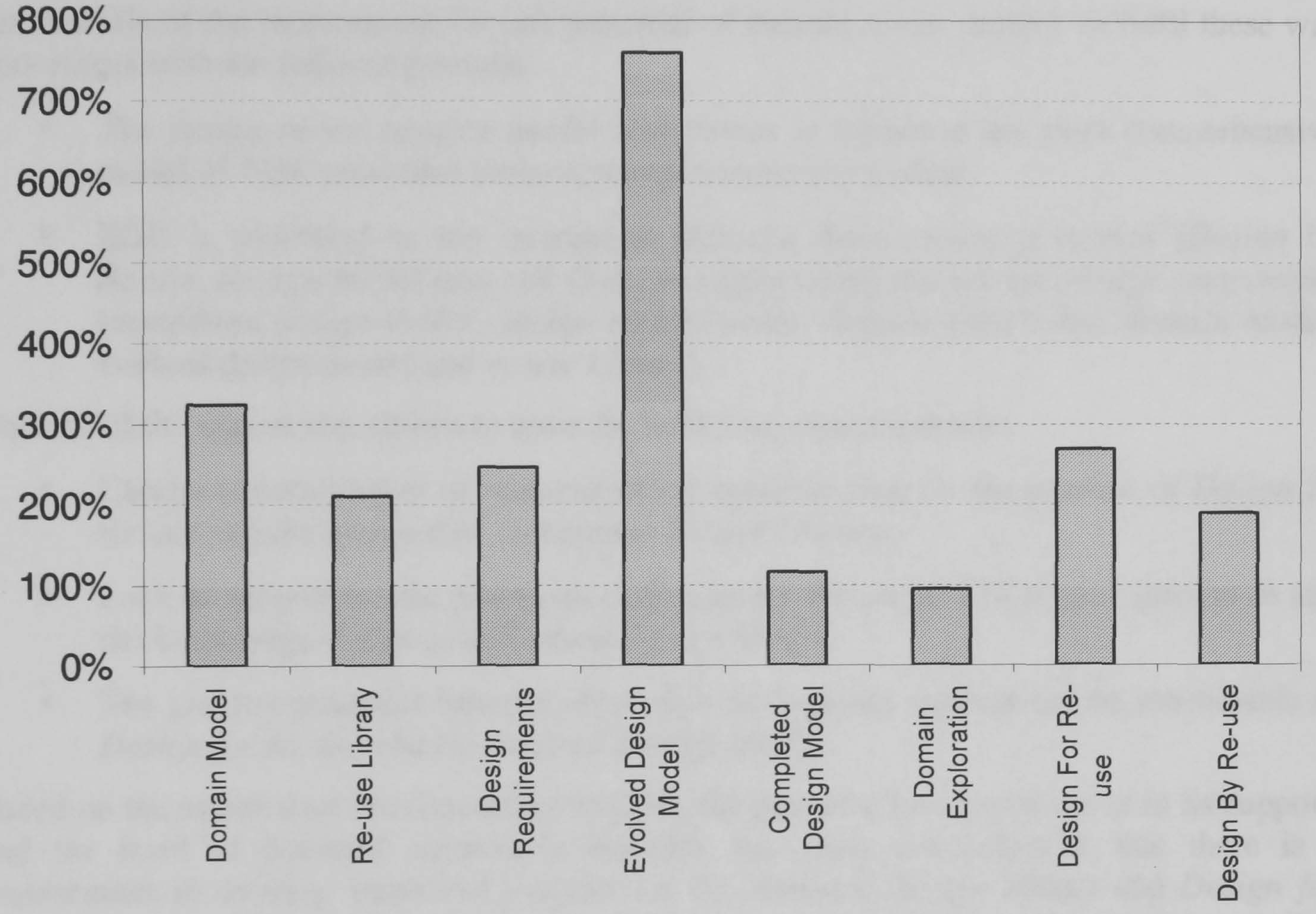


Figure 2.8: Foreseen improvements

The greatest single potential for improvement of support is for the knowledge component *Evolved Design Model* at 760% and, of the three *re-use* processes, *Design for Re-use* at 269%. Both are elements of the part of the *re-use* approach that focuses on support of the ongoing design activity and as shown in Figure 2.6 there is a process dependency from the knowledge component, *Evolved Design Model,* to the re-use process, *Design for Re-use*. As such, an alteration or improvement in support for one has a direct impact on the other.

2.6 Chapter summary

The aim of this chapter was to introduce the concept of EDR and establish its characteristics with respect to the activity of engineering design. In addition, a comprehensive model of EDR was acknowledged. Further, the chapter aimed to highlight the main gaps in EDR support to provide a context for the remainder of the work.

In fulfilment of these aims Engineering Design Re-use (EDR) was:

- Established as an approach with its origins in the realms of software engineering. However, the practices of software re-use were shown to be insufficient for engineering design as the domain deals with more abstract concepts.

- Defined as *the total approach that supports the utilisation of any knowledge gained from a design activity.*
- Shown to have significant potential benefits to an engineering design activity with respect to cost, performance, quality and time.
- Shown to require tools and methods to successfully manage design knowledge acquisition, maintenance and utilisation prior, during and after the design activity.

An analysis of the requirement for and potential of current *re-use* models to fulfil these was undertaken with the following results:

- The *design re-use process model* was shown to represent the most comprehensive model of EDR presented to the research community to date.
- EDR is modelled as the interaction between three *re-use processes* (*Design by Re-use, Design for Re-use,* and *Domain Exploration*) and six *knowledge components* (*completed design model, design requirements, domain knowledge, domain model, evolved design model* and *re-use library*).

Current EDR support was shown to have the following characteristics:

- Compartmentalisation of research effort concentrating on the process of *Design by Re-use* and the knowledge component *Re-use Library.*
- Lack of support for the processes of *Design for Re-use* and *Domain Exploration* and the knowledge component *Evolved Design Model.*
- The greatest potential benefits obtainable from re-use support can be attributable to *Design for Re-use* and *the Evolved Design Model.*

Based on the established characteristics of EDR, the potential for improvement in its support, and the level of potential achievable benefits, the main conclusion is that there is a requirement to develop improved support for the *Evolved Design Model* and *Design for Re-use* process. The *Evolved Design Model* represents knowledge of an evolving design whilst the *Design for Re-use* process is defined as a conscious effort to identify, extract and enhance knowledge elements during design itself. As such, an approach to improve EDR support requires that the knowledge generated during the design activity be captured and supported in some form which supports enhancement of its knowledge content to promote its future re-use.

3 Modular Design and Engineering Design Re-use

Chapter 2 outlined the characteristics of EDR and defined the requirements for improved EDR support through the mapping of current EDR practices to elements of the 'Design Re-use Process Model'. Chapter 3 aims to highlight the correlation between the principles of Modular Design (MD) and improved support for EDR. The field of product structuring ((WDK) 1995; (WDK) 1996; (WDK) 1997; (WDK) 1998; Riitahuhta and Pulkkinen 2001), which underpins MD, and its application in a *re-use* approach, is introduced in section 3.1. Having established the main characteristics of the product structuring ideology section 3.2 defines the features of MD and their relation to the engineering design activity[6]. Section 3.3 outlines the advantages, disadvantages and issues related to the application of MD principles. Section 3.4 defines the support afforded by MD to the EDR processes whilst section. Section 3.5 defines the terminology that will be utilised throughout the remainder of this work to: define requirements (Section 3.6), critically analyse existing approaches (Chapter 4) and define the declarative and procedural knowledge related to the proposed novel MD methodology (Chapter 5).

3.1 Product Structuring

Chapter 2 summarised that to improve EDR support required an approach to support the generated knowledge during the design activity and the enhancement of its knowledge to address the lack of support for the Evolved Design Model and Design for Re-use process. Product Structuring (PS) concerns the activity whereby the structure characteristics of a design or product are defined. Where structure is defined as 'the elements of a system identified by their type and relations between these elements' (Andreasen, Duffy et al. 1995). Not purely limited to physical parts and components, the structuring activity can also be utilised at a far more abstract level where design knowledge is restricted to high level knowledge describing energy transformation, functions, and behaviours. In addition 'structuring' can enhance the content of individual fragments by 'formalising the product information to provide a framework for product information data during its lifecycle' (MacCallum 1995; Yu and MacCallum 1995). Thus, the relationship between this and other knowledge fragments generated during the design lifecycle can be explicitly defined through the PS. As such, the PS can provide a framework to support further knowledge enhancement based on an analysis of say, the similarities and/or dependencies between these knowledge fragments. Due the PS fields' emphasis on support and management of design knowledge, it is investigated as a basis on which actively support improved EDR.

3.1.1 Product Structuring Theories

The basis of product structuring is to define the elements and relations of a product with respect to a chosen *viewpoint*. A number of theories dominate structuring methodology including Andreasen's theory of domains (Andreasen, Duffy et al. 1995; Andreasen, Hansen et al. 1996) and the model-based theory subscribed to by Erens (Erens and Verhulst 1995). Andreasen's structuring principle centres on the synthesis process as a progression from transformation structure (energy, material), through functions (required effects) and organs (function carriers) to the definition of part structures. The basis of structuring principles,

[6] For a history of Modularity the reader is referred to Miller and Elgard Miller, T. and P. Elgard (1998). Defining Modules, Modularity and Modularisation - evolution of the concept in a historical perspective. Design for Integration in Manufacturing. Proceedings of the 13th IPS Research Seminar, Aalborg University, Fugsloe..

here, involve the definition of structures within each domain and understanding the interactions between domains (as shown in Figure 3.1).

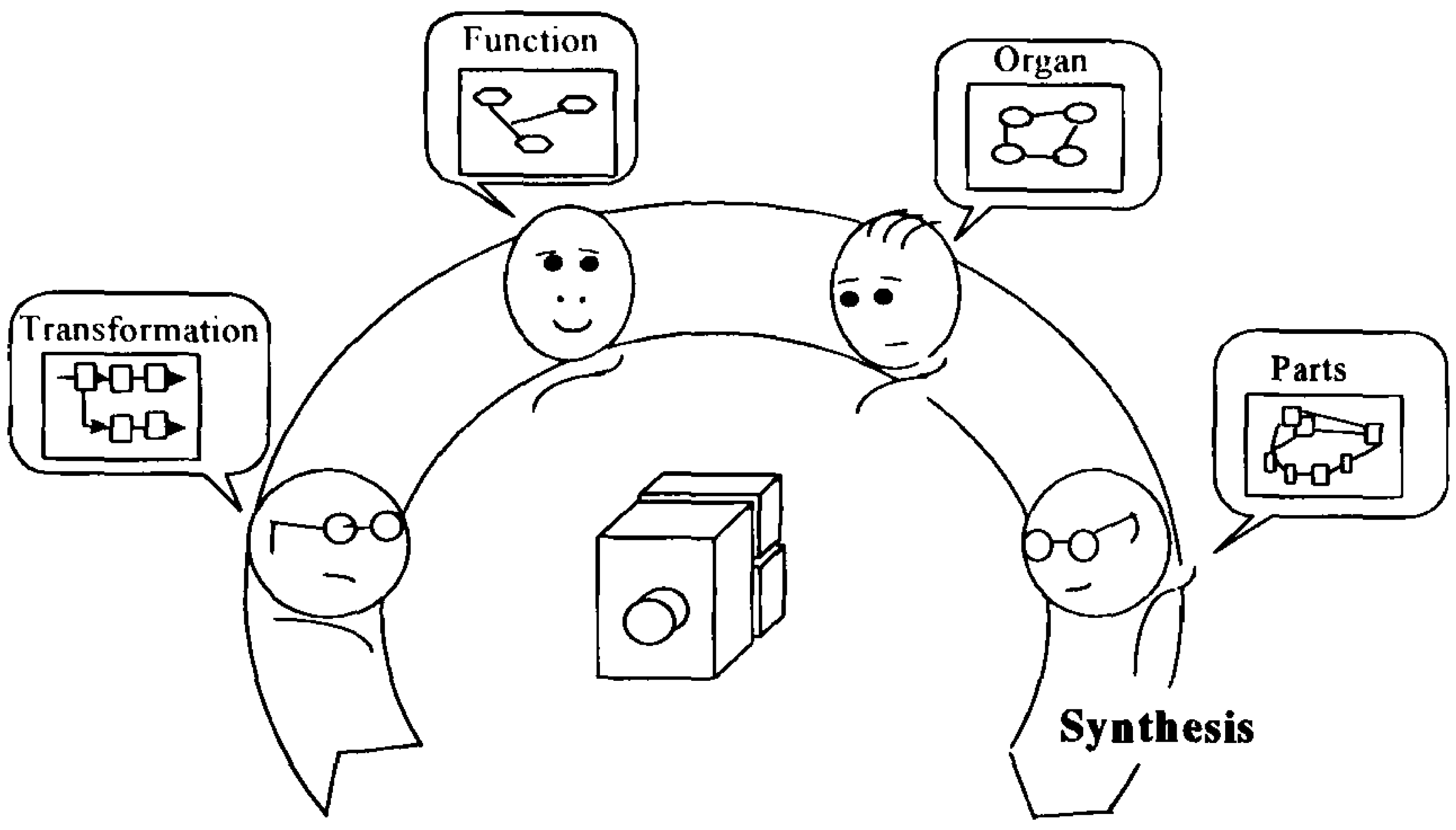


Figure 3.1: Andreasen's Domain Theory (Andreasen 1991)

Erens' (Erens and Verhulst 1995) product model-based theory involves the transformation of a design from a functional model to a technology model towards the definition and construction of a physical model (see Figure 3.2).

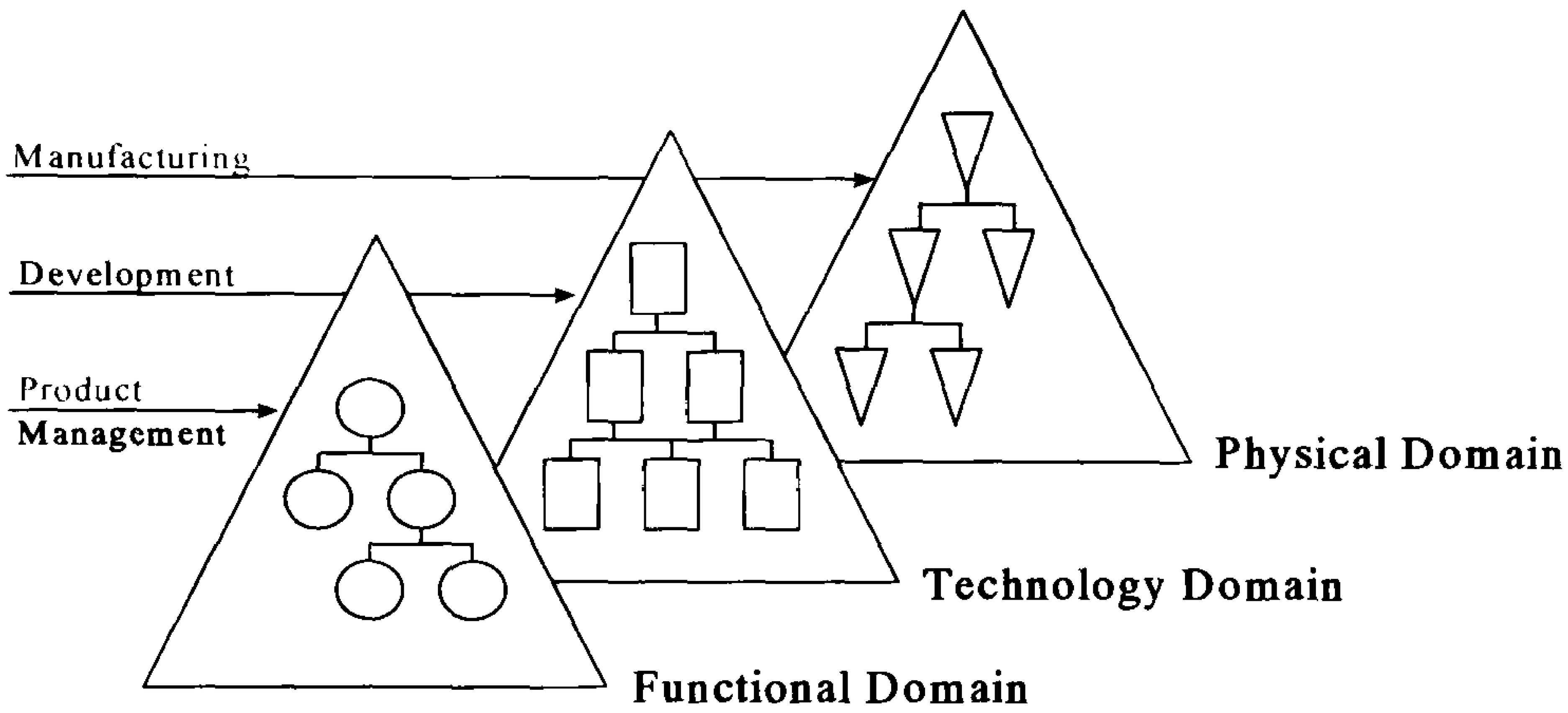


Figure 3.2: Eren's Product Models (Erens and Verhulst 1995)

Both theories structure products in various domains but it is the scope, role and definition of each domain that differ. In general the structuring principles utilised within each theory are the same: decomposition, configuration and rationalisation (see Section 3.1.2).

3.1.2 Categories of Product Structuring

Structuring tends to fall into the categories of *decomposition* (Hansen 1995; Liedholm 1998), *configuration* (Hansen 1995; Yu and MacCallum 1995; Andreasen, Hansen et al. 1996) (occasionally referred to as *composition* (Tichem and Storm 1996), and *rationalisation* (Herbertsson 1995; Erens and Verhulst 1996).

A product's primary effect, primary function, or technology can be decomposed into its constituent elements. This decreases associated design complexity because the integral complexity of each individual element is lower than that of the whole (Hansen 1995). Thus, teams of designers can work in parallel and reduce product development time (Hansen 1995). *Decomposition* is often related with the functional domain (Andreasen, Hansen et al. 1996; Erens and Verhulst 1996) or effects domain (Andreasen, Hansen et al. 1996) where the main function is decomposed into sub functions (Tichem and Storm 1996). Liedholm (Liedholm 1998) states that the *decomposition* activity is utilised to clarify what the product should do and establish the functions of the product.

The *decomposition* activity primarily supports *Design by Re-use* in that it allows designers to breakdown the low-level design requirements into more manageable, less complex constituents. Applied to the re-use library, the *decomposition* activity can breakdown past design solutions into their high level solution concepts providing the potential of utilising both experiential knowledge and design specification goals. Thus, designers can decompose current requirements with a view to map between current and past designs to enhance *Design by Reuse* capabilities.

The elements of a product and the way in which they are built together determine the overall behaviour or function of that product (Andreasen, Hansen et al. 1996). Where a design need is decomposed into low complexity elements, often at an abstracted level, we must proceed to allocate possible solution concepts to each element and *configure* these elements to meet this need. Thus, *configuration* creates an arrangement, from a given set of elements, by defining the relationships between selected elements that satisfy the requirements and constraints of a design (Hansen 1995; Yu and MacCallum 1995; Andreasen and Riitahuhta 1998; Tiihonen, Lehtonen et al. 1998). The process of *configuration* design involves the creation or identification of relations between the elements to ensure that the subsystem realises its function and contributes to the overall purposeful function, in the right manner (Hansen 1995).

Design by Re-use is a form of *configuration* as it re-uses previously defined elements to meet a current design need. *Configuration* as with *Design by Reuse* requires that rationalised (structured) sources of past design knowledge be available.

Rationalisation in product structuring involves the systematic organisation of knowledge related to the products domain to form a rational conception of a model that is free from radical or specific quantities. *Rationalisation* can take a number of forms including the definition of product architectures (Herbertsson 1995; Erens and Verhulst 1996), platforms (Elgard 1998). Generic product architectures are defined to form a 'stable structure and provide a consistent environment for new component development (Erens and Verhulst 1996). Such product architectures arise from *rationalisation* over a number of products, are a more stable model of design than the physical models and can be re-used to create new versions of the product (Erens and Verhulst 1996). Where physical models consist of components that are liable to change, generic product architectures facilitate mapping of a more consistent functional arrangement to physical components and the interactions between these (Herbertsson 1995). *Rationalisation* can also occur when components within a number of products are redefined to produce a platform of products. This platform forms a 'common structure from which a stream of derivative products can be efficiently developed and produced' (Elgard 1998). Platforms form a re-usable 'foundation of product elements, technologies, knowledge as means of supporting product variety and increasing re-use of engineering knowledge' (Elgard 1998).

Rationalisation in structuring promotes the process of *Domain Exploration* by exploring completed design models and their associated domain knowledge. Such exploration results in a deeper understanding of the elements and relationships that combine to facilitate effective design in the domain. Successful *rationalisation* of design knowledge can promote the *Re-use Library*, of parts, concepts and knowledge and a generic product architecture model that can subsequently be utilised through the *Design by Re-use* process.

3.1.3 Organisational Product Structuring

It is argued that successful EDR relies not only on an understanding of the theories and methodologies behind it but also how it fits within the design process, the product development strategy and the overall company strategy (see Figure 3.3). For a strategy geared towards continued enhancement and re-use of a company's knowledge resources, the role of product structuring is that of supporting, maintaining and promoting these resources for *re-use.*

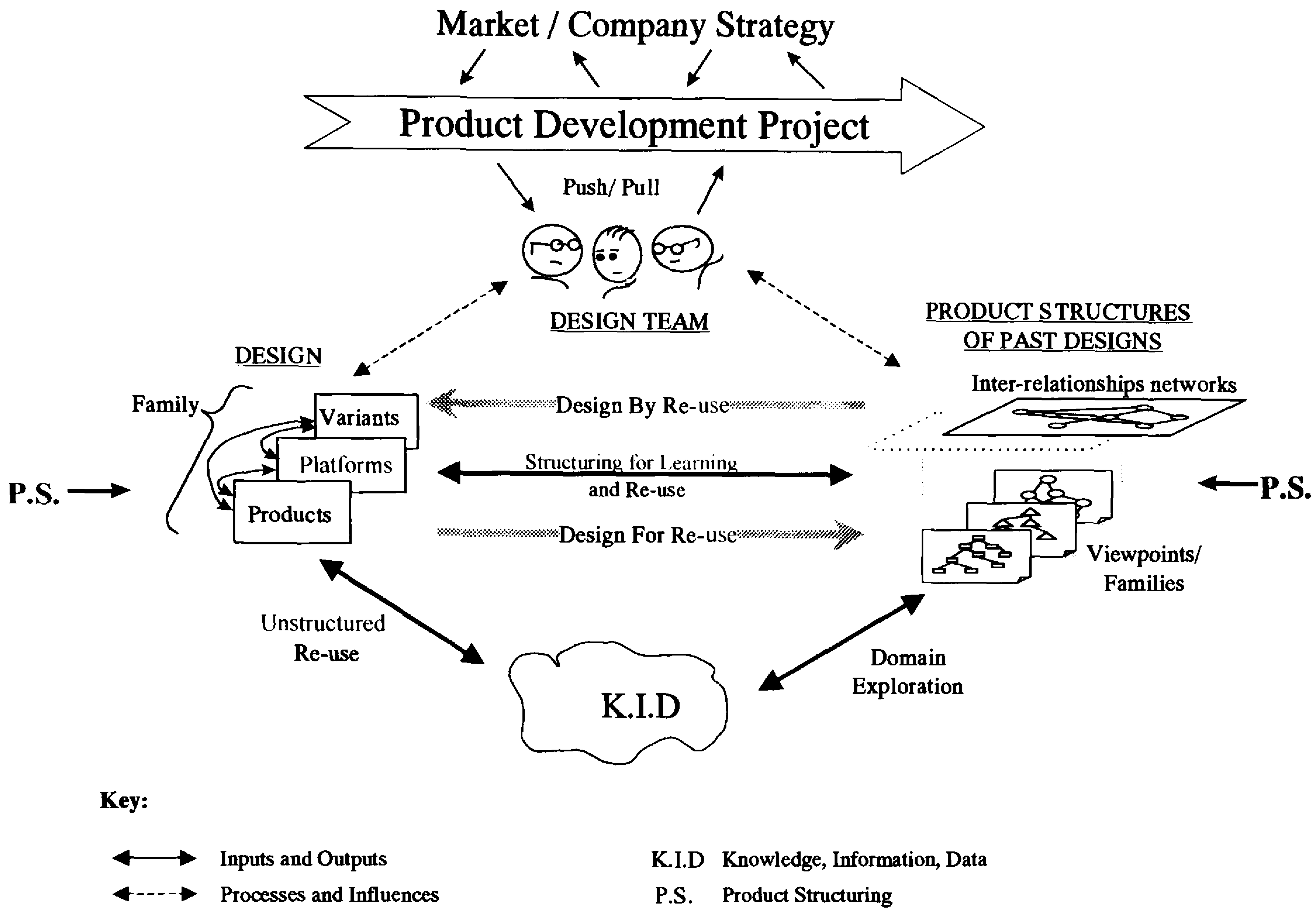


Figure 3.3: Organisational Product Structuring and Re-use (Smith and Duffy 2001)

Figure 3.3 illustrates the roles of product structuring (P.S) and re-use within a design organisation. It shows how the overall organisational and product development strategies constrain the design team throughout any design project. The diagram depicts both structured and unstructured re-use. There is a need to develop tools, techniques and methodologies to support currently unstructured re-use. Such support includes rationalisation from past design cases and structuring of current design knowledge to facilitate a structured approach to *re-use* and *learning.* As discussed in Both the structuring and re-use principles must also develop to satisfy organisational goals whilst meeting the knowledge requirements of the design team in both current and future design projects.

The design team are 'pulled' by the overall company and product development strategy and in turn 'push' the individual design project by carrying out a number of processes and activities. The design team are also subject to a number of influences including their own knowledge and experience. Thus, a design team draws from past design experience and re-uses experiential knowledge to further the current design project within the overall design requirements.

Product Structuring is essential to a *re-use* approach in that over a period design generally produces a considerable amount of knowledge, information and data (K.I.D) [7], which without structure remains a mass of vague elements whose significance and relations are difficult to define and understand. With structure however, this K.I.D can aid designers in finding specific knowledge elements useful in a new design i.e. *Design by Re-use.* Such K.I.D from past designs can be structured using a number of different principles from viewpoints to families and inter-linked with a series of networks or indexes.

If product structuring is then employed during current or new design processes for products, parts, assemblies and families the K.I.D generated during this evolving process can be structured and stored in a formal manner, and appropriately structured to promote its use in future design, i.e. *Design for Re-use.* As shown in the diagram we would thus have a bi-directional flow between past and current design which would result in structuring principles to support *learning* from design knowledge and its subsequent *re-use.*

Figure 3.3 illustrates that product structuring is indeed an essential element in the *re-use* approach as it can facilitate the organisation of K.I.D. However, current *re-use* in design organisations is predominantly unstructured (Duffy and Ferns 1999), with a deficiency of tools, techniques and methodologies to support the overriding product structuring theories (Duffy, Smith et al. 1998; Smith 2002). Hence an increased understanding of the processes of product structuring is required to enable this K.I.D to be structured in design organisations for re-use. For instance, Duffy and Legler's work (Duffy and Legler 1998) addresses this need by proposing a methodology to structure (rationalise) past designs. This rationalisation provides a basis upon which to efficiently retrieve specific cases for re-use and presents a means upon which to generalise, enhance and re-use past experiential knowledge.

3.2 Modular Design characteristics

Modular Design (MD) is a 'natural extension of product structuring principles' (Knox 1984). For example, one of the 'basic requirements for the building of product families based on *product platforms* is a modularised *product architecture'* (Hofer and Gruenenfelder 2001). MD is a methodology for executing the activity of design. It has gained increasing prominence, as a design methodology to meet market demands to 'quickly and globally deliver a high variety of customised products' (O'Grady and Liang 1998). According to Smith and Reinertsen (Smith and Reinertsen 1997), modular product design can facilitate this as 'economies of scope are gained by using the modular components over and over in different products; and customisation is gained by the myriad of products that can be configured'. Hofer and Gruenenfelder (Hofer and Gruenenfelder 2001) noted that this call for greater product variances 'lead often to a disadvantageous cost position due to the efforts for individualised solutions and make it more difficult to profitably put products to the market'. Existing design methodologies that focussed on the creation of individualised products for, in extreme cases, individual customers, are no longer appropriate to maintain competitiveness and profit margins. Thus, methodologies that supported product customisation whilst achieving the cost benefits of mass production were sought, leading to the term *mass customisation.* Modular Design is a design methodology often associated with this drive for *mass customisation* due to its focus on the development of products with distinct detachable modules for rapid product development, possible re-use of long-lasting

[7] Data is taken here to be the most basic symbolic elements such as numerical numbers of words e.g. 5, length, metre, gap, false, true, and safe. Information provides a meaning to data such as length is equal to 5, the gap is 2, and safety can be true or false. Knowledge provides added meaning to information and allows inferences to be deduced or abducted e.g. if the length is greater than the gap then safe is true.

modules, efficient upgrading, reconfiguration and other lifecycle engineering objectives (Gu, Hashemian et al. 1997). As a result research into the theory and application of MD has increased. For example, principles of MD have been applied to many areas including ship systems (Blade, Klinge et al. 1998), process plants (Humphries and Radcliffe 2001), power products (Sosale, Hashemiam et al. 1997) and air-conditioning systems (Chang and Ward 1995). As a premise to support platform design it has also been applied to products such as light rail vehicles (Lashin and Doblies 2001), automobiles (Hofer and Gruenenfelder 2001), industrial products (Berti, Germani et al. 2001), and power tools (Otto 2001).

3.2.1 A Module

A *module* is commonly described as a group of 'functionally' or 'structurally' independent components clustered such that 'interactions are localized within each module and interactions between modules are minimised' (Sosale, Hashemiam et al. 1997). However, according to Miller and Elgard (Miller and Elgard 1998) 'the meaning of the term module has changed from being defined by physical presence into being defined by structure and functionality'. This is illustrated by considering a software module. The module has no physical presence and is thus defined purely by its functional characteristics. Thus, a module can no longer be defined from a purely geometric perspective such as one might view a LEGO™ block. In addition a module may also be defined in terms of the abstract design knowledge for which a physical realisation will evolve through the design activity. The importance of this abstract knowledge from earlier in the design activity to EDR has been discussed throughout Chapter 2. Miller and Elgard (Miller and Elgard 1998) have posited the term knowledge module to broaden the definition of a module to support both the immaterial and physical. This is an essential aspect of MD for its application in EDR as the *knowledge module* 'is the preliminary stage to the physical module' (Miller and Elgard 1998) and provides a mechanism through which to capture re-usable fragments of knowledge. Miller (Miller and Elgard 1998) states that 'intellectual re-use of earlier stages.....blurs the boundaries between knowledge management and conventional modularisation'. Accepting the existence of a *knowledge module,* and moving away from expressing modularity in terms of physical components, a *module* within this work is defined as a group of *concepts* whose interdependencies are;

- maximised internally within a group of *concepts* (module) and
- minimised externally between groups of *concepts* (modules).

Where, a *concept* is defined as an element of the product being designed expressed from a particular design *viewpoint.*

3.2.2 Modularity

Modularity is as a property of the product structure. One of the key attributes of modularity is that it is not an absolute value of the product structure; it exists in more than one form and at more than one level. In the first instance, a modular architecture can take on a number of *types* as illustrated in Figure 3.4 including:

a) Component swapping modularity – different components paired with the same basic product (modules) .

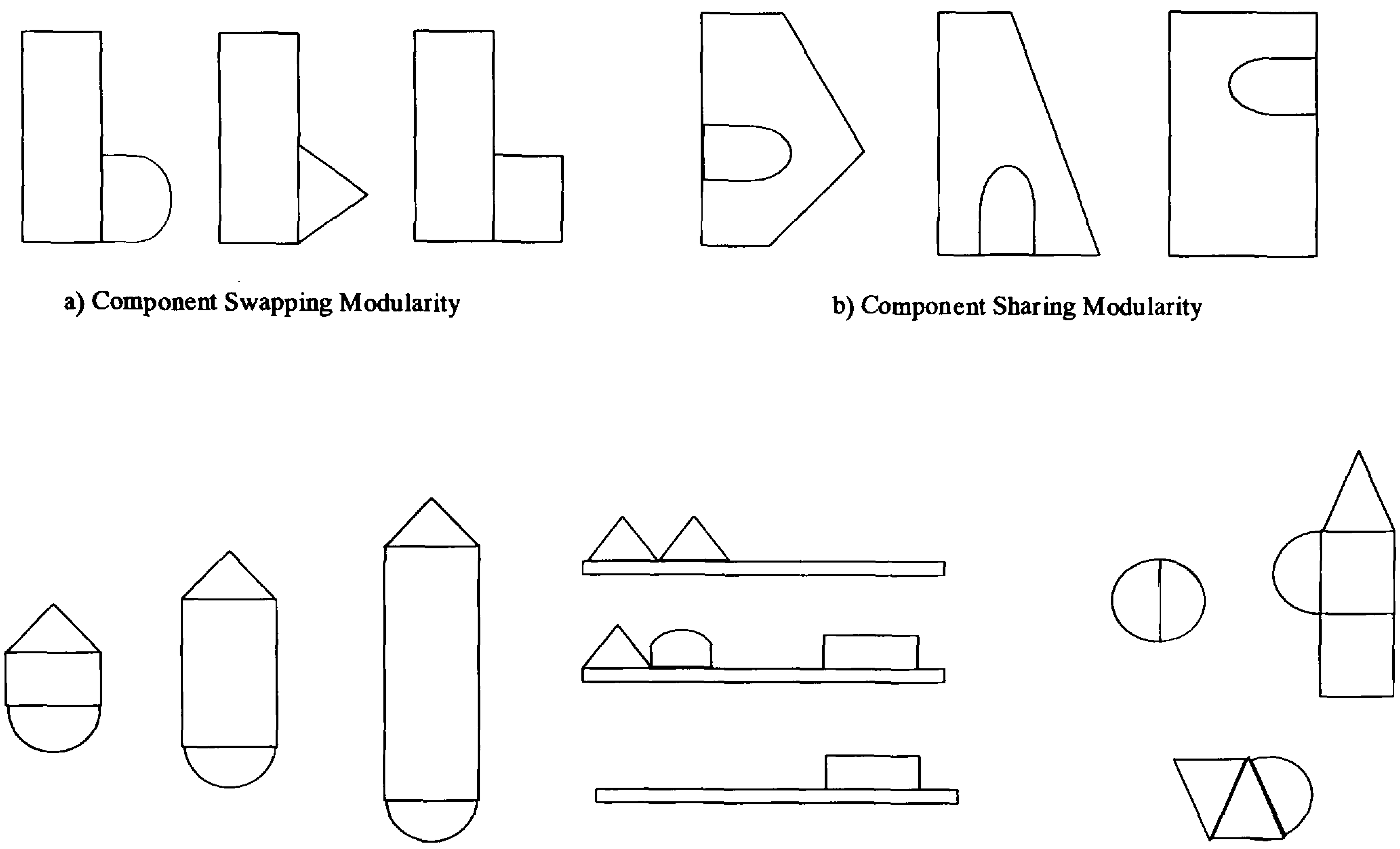
b) Component sharing modularity – a core module (or modules) used across different products to provide economies of scope.

c) Cut-to-fit modularity – one or more components or modules is continually variable within preset or practical limits.

d) Bus modularity – a standard structure or interface which can accept a number of different components or modules.

e) Sectional modularity – allows the configuration of any number of different types of components in an arbitrary way, as long as each component is connected to another at a standard interface.

f) Mix modularity - a combination of the above *types*.



a) Component Swapping Modularity b) Component Sharing Modularity c) Cut-to-Fit Modularity d) Bus Modularity e) Sectional Modularity

Figure 3.4: Types of modularity (Ulrich and Tung 1991)

These *types* of modularity can be used to support product variance over a product family or generations of these.

Secondly, due to the 'complexity of real life systems more sophisticated modularisation techniques are needed' (Philippi 1998) as it is not always possible to define 'neatly packaged' modules with clear boundaries. Thus, the *modularity* of a product can be equated to a relative value on a scale, which runs between integral and modular, where (Smith and Reinertsen 1997):

- an integrated product is one where the product functions are broadly distributed throughout the system, and,
- a modular product is one where the components are grouped into distinct detachable modules that fulfil a specific product function(s).

Thirdly the *modularity* of a product may depend on the *viewpoint* that is taken of it. Accordingly, the classification of a product as 'modular or integral is also not a constant property of a structure, but depends on the point of view [*viewpoint*] we have when we observe it' (Elgard and Miller 1998). Recognising this Jiao and Tseng (Jiao and Tseng 1999) state 'that the main challenge for today's design methodologies is to support these multiple *viewpoints* to accommodate different modelling paradigms within a single coherent framework.' This is an important aspect when considering MD as a support mechanism for *re-use* as design related knowledge has been shown to exist with a variety of *viewpoints* and that the activity of mapping between these evolves the design activity (Section 2.3.1). Thus, an EDR centred MD methodology would need to support design knowledge across *viewpoints* and facilitate *mapping* between these.

Ulrich et al (Ulrich and Tung 1991) state that though the understanding of modularity has become more abstract and more related to functionality than geometry, a module is fundamentally defined as a physical unit. However, based on Miller et al (Miller and Elgard 1998) expression of the need to explore the concept of modules in relation to 'knowledge management and modules seen as knowledge carriers' we develop the concept of *Knowledge Modularity* (KM). The work presented in this thesis explores this KM concept in relation to its applicability to support design knowledge modelling and enhancement. Such exploration is essential to facilitate the development of an approach on which to facilitate MD for improved EDR, termed here, *Knowledge Modularisation* (KMn) [8].

3.2.3 A modular product

Due to the need for that individual *module* functions and/or structures to eventually combine to realise the overall function and/or structure of the product, the modules can never truly be independent and must be defined together with the product to which they belong. Further 'between module' or 'interface' constraints must be considered for modules to be successfully configured to meet overall product requirements. A comprehensive knowledge of the *dependencies* within the product is required to define the boundaries of, and the interfaces between, modules. This requirement in noted by, amongst others, Galvin (Galvin 1999), Gu et al (Gu, Hashemian et al. 1997), Knox (Knox 1984) and O'Grady and Liang (O'Grady and Liang 1998). This *dependency* knowledge between these concepts in *viewpoints* can be a viewed from a number of *perspectives,* for example, energy, information, and material; and/or spatial relations (Knox 1984).

Similar to that of the definition of modules, the definition of a modular product has evolved from its traditional historic perspective. Traditionally, a modular product would have been defined as a product that fulfils various functions through the combination of distinct building blocks (Blackenfelt 2001). However, more recently this has evolved into a more generic definition whereby a product is composed of building blocks chosen for company specific reasons (Blackenfelt 2001). Thus, a modular product is no longer necessarily defined based on its functional *objective*, but on a specific *lifecycle objective(s)* that are aligned with product or company strategy. Typical examples of such *lifecycle objectives* are ease of; assembly, maintenance, re-cycling, and disposal. Thus, the *dependency* knowledge of a product may also be considered from the *perspective* of specific *lifecycle objectives.* Obviously, it would be ideal if one modular product configuration could fulfil all the requirements of each *lifecycle objective*. However, similar to the case with *viewpoints* (Section 3.2.2) a modular product configuration that achieves one *lifecycle objective* may not achieve another. Sosale et al (Sosale, Hashemiam et al. 1997) states that 'it is the designers' responsibility to make trade-off decisions' such as these. However, research undertaken as part of this work (Smith, Robb et al. 2001) identified that the designer required some evidence of the relative modularity of differing product configurations and the impact of lifecycle objectives on this to support such decision-making.

3.2.4 Modular Design research

Modular Design (MD) research can generally be grouped into 3 categories; those associated with; the identification of modules, the design of modules, and designing with modules

[8] *Knowledge Modularisation* (KMn) can be considered the activity of defining the knowledge modularity of design related knowledge

(O'Grady and Liang 1998). Again, taking the case of product platform design and utilisation we can see this distinction in the application of MD research to this area, as shown in Figure 3.5.

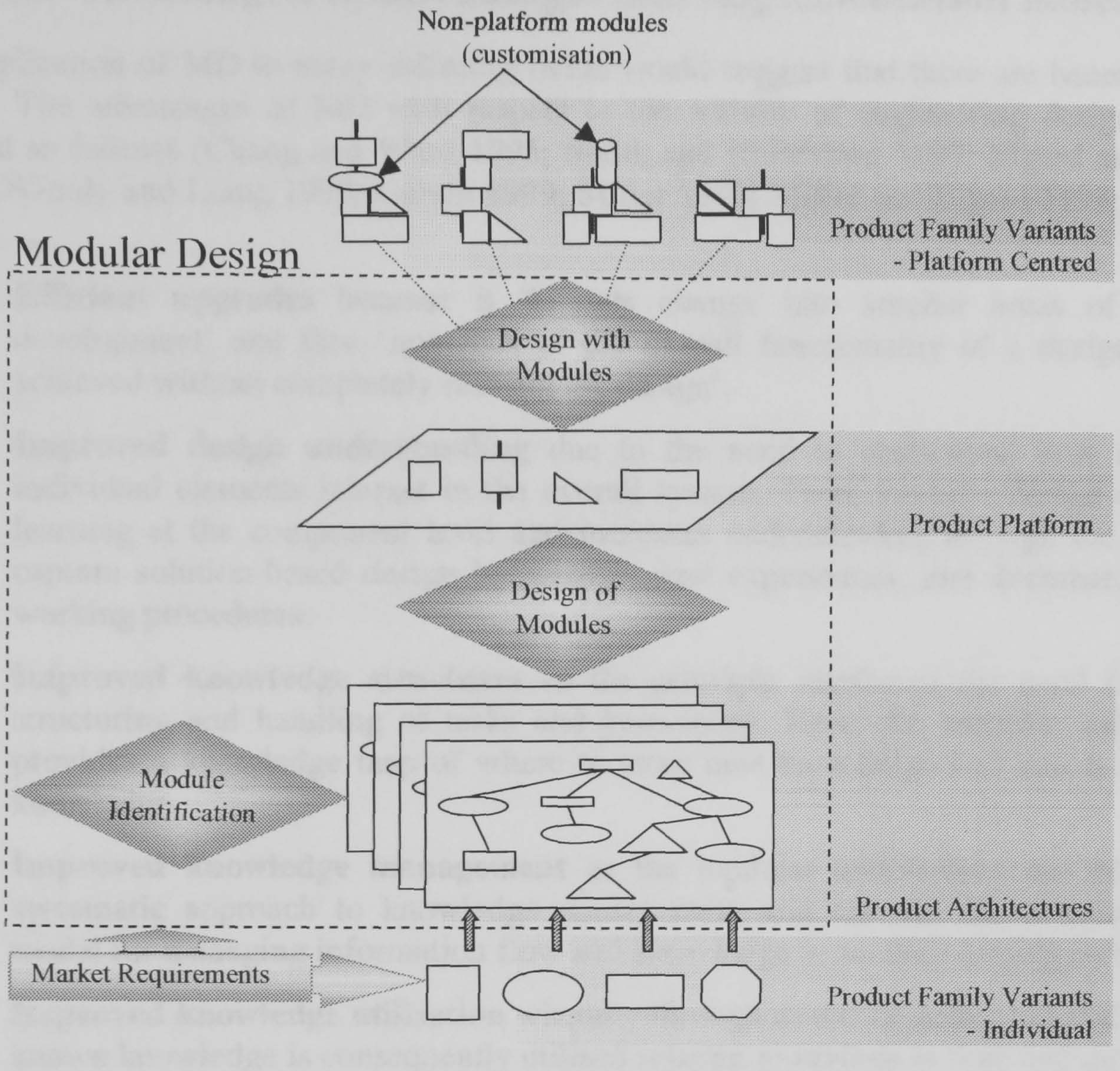


Figure 3.5: Architecting a product family

Figure 3.5 depicts the process of 'architecting' a product family utilising a product platform. We see, at the bottom of Figure 3.5, that we have an initial set of individual product family variants (designed to meet individual customer requirements). The first step in the *platforming* process is to define the individual product architectures for each variant. A *module identification* process is applied to these architectures where *modules* are defined based on their functional and/or specific market characteristics (Dobrescu and Reich 2001; Hofer and Gruenenfelder 2001; Lashin and Doblies 2001; Otto 2001). After the module boundaries have been defined, the *design of modules* process determines their structural and geometric characteristics. Finally, there is the process of *design with modules* whereby product variants are created based on the *configuration* of the previously defined platform *modules* characterised by their functional and physical features. At this point *customisation* can occur, where customer specific functions, not available from the modular platform system, are developed as original assemblies (Lashin and Doblies 2001).

Platform Design is one specific application of MD where in general a pre-existing product family is required as its basis. However, MD is a methodology that has the potential to be utilised as the basis for any design activity not just those based on pre-existing designs. As stated by Galvin (Galvin 1999) a modular product can occur in the case of a single product or a family of products that use certain components across the entire range. The work covered in this thesis focuses on the *module identification* process. Further, the domain of application deals predominantly with individual products to improve the re-use capabilities of the product and its associated knowledge.

3.3 MD advantages, disadvantages and implementation issues

The application of MD to many differing fields would suggest that there are benefits to be gained. The advantages of MD with respect to the activity of engineering design can be outlined as follows (Chang and Ward 1995; Smith and Reinertsen 1997; Elgard and Miller 1998; O'Grady and Liang 1998; Galvin 1999; Miller 1999; Miller and Elgard 1999; Muffato 1999):

- **Efficient upgrades** because it 'bounds change into smaller areas of product development' and thus 'upgrades of the overall functionality of a design can be achieved without completely redoing the design'.
- **Improved design understanding** due to the need to understand how and why individual elements interact in the overall system. Thus, modular design enhances learning at the component level and increases understanding through the drive to capture solution-based design knowledge, past experiences, and documentation of working procedures.
- **Improved knowledge structures** as the principle reinforces the need for better structuring and handling of tasks and knowledge. Here, the modular architecture provides a knowledge map of where to store new knowledge and access previous knowledge.
- **Improved knowledge management** as the modular architecture can promote a systematic approach to knowledge management and can aid the definition of a model for managing information flow and knowledge in product development.
- **Improved knowledge utilisation** whereby through re-use of defined modules, well-known knowledge is consequently utilised relating to savings in time and money.
- **Rapid product development** as it permits the concurrent development of modules. In addition, existing modules can be rapidly reconfigured and new modules introduced.
- **Strategic Flexibility** that allows companies to respond to changing markets and technologies by rapidly and inexpensively creating product variants derived from different combinations of existing or new modules.
- **Reduction in complexity** as it improves or creates design clarity and simplifies part inventories.
- **Reduction in costs** due to the rapid development opportunities afforded by modular designs, where costs and lead times are cut, and business efficiency improves through the re-use of design and manufacturing processes.

Despite the plethora research as to the benefits of modular design there are a number of potential disadvantages of the modular design (Ishii, Lee et al. 1995; Hatton 1996; Gu, Hashemian et al. 1997; Smith and Reinertsen 1997), including:

- **Cost**

  Modularity adds costs, as it requires the exploration and specification of [additional] interfaces. These additional interfaces result in extra design effort and increased costs as robust interfaces are generally more expensive.

- **Performance**

  In integral products the characteristics sharing of functions and geometric nesting of components, eliminates redundancy and minimizes the volume a product occupies. Thus modular products are potentially larger and with inferior product performance. The performance disadvantage of the modular product arises from the fact that interfaces are frequently the weak links of the system.

In addition, there are a number of barriers to the implementation of MD in practice (Chang and Ward 1995; Sosale, Hashemiam et al. 1997; Burke and Miller 1998; Huang and Kusiak 1998; Juengst and Heinrich 1998; Miller and Elgard 1998; O'Grady and Liang 1998; Philippi 1998; Miller 1999), including:

- **Theoretical understanding**

  There is a gap is the research communities understanding of the core phenomenon of MD itself in relation to their understanding of the benefits it may provide. Little work has been on these research issues and modularity has been treated in literature in an abstract from and it has not been satisfactorily explored in industry.

- **Practical implementation**

  The gap in researchers understanding of phenomena of MD itself is mirrored by its lack of application in practice. This can be attributed to a lack of design theories and tools in the mechanical world that serve as articulate procedure for designers to follow in practising modular design.

  In addition, successful implementation depends on strong management support and commitment from the designers. However, many designers are sceptical about following rules of a common architecture as they feel it limits the possibility for fulfilling customer needs, and limits creativity and previous studies have not always concluded that modularisation is a good idea (Hatton 1996). However, it has been suggested that this is due to the focus on re-working conventional designs to include modular design when studies have shown that to be successful, a modularisation strategy must be incorporated at project inception

- **The modularisation activity**

  Designers face a difficult task as the modular structuring task is huge and often very difficult. In the first instance there is no objective scale to define what is essential functionality and it depends on the perception of the system. In addition, it is not sufficient to consider geometry alone, since information, energy and material also create important relations and it is expected that conflicts will arise between these. It is the designers' responsibility to make trade-off decisions and resolve these conflicts. However due to the lack of articulate MD procedures and explicit knowledge related to the nature of the systems, there is little support to aid the designers in making these decisions and defining the MD.

Section 3.2 illustrated the potential for application of MD principles to manage the business implications associated with the increasing market demand for product variance and customisation. This section has outlined the benefits and disadvantages of, and issues related to, MD in research and practice. However, despite evidence as to the continued development of the modular design field, there is a significant lack of approach to support its practical application. Indeed, Philippi (Philippi 1998) concluded that 'to be able to handle the complexity of real-world systems in a comfortable manner more sophisticated modularisation techniques are needed' and Huang and Kusiak (Huang and Kusiak 1998) conceded that 'approaches are needed to determine modules, represent modularity, optimise modular design and assess the impact of modularity on the design process, manufacturing and management'.

3.4 The relation between modular and re-use principles

The benefits of MD as outlined in Section 3.3 centre on a greater capacity for structuring both physical parts and product knowledge to better manage the relation between market requirements and the design product. Hence, they support increased utilisation of experiential knowledge for new product development and thus provide an approach on which to actively support re-use.

MD has been shown to support architecting of a product platform and platforms have been described previously as a *rationalisation* of elements (generally components) across a product family. Thus, it can be argued that modularisation supports this *rationalisation* process and as a consequence *Domain Exploration* as discussed in section 3.1. MD can facilitate rapid new product development through the *configuration* of previously defined modules and thus can be seen to support *Design by Re-use.* However, as discussed through Figure 3.3 the utilisation of product structuring principles during current or new design processes to support the K.I.D generated during the evolving process can facilitate *Design for Re-use*. Thus, MD applied as a *methodology* during a current and evolving design activity can provide active support for *Design for Re-use*. This support is provided through the definition and structuring of knowledge and product modules to facilitate their future *re-use* to support the design activity for product variants and/or future product generations.

3.5 Terminology

The following section provides a clarification of definitions of specific terminology as utilised within the context of this work.

3.5.1 Viewpoint

A *viewpoint* represents a structured view of engineering design required by the designer to evolve the engineering design activity to a suitable conclusion (Bhatta and Goel 1992; Gero 1992; Duffy and Kerr 1993; Andreasen, Duffy et al. 1995; Erens and Verhulst 1995). The notion of a *viewpoint* of design is illustrated in Figure 3.6. For example, designers may require viewpoints based on their current focus: geometrical, numerical, spatial, functional, mechanical, behavioural, and structural.

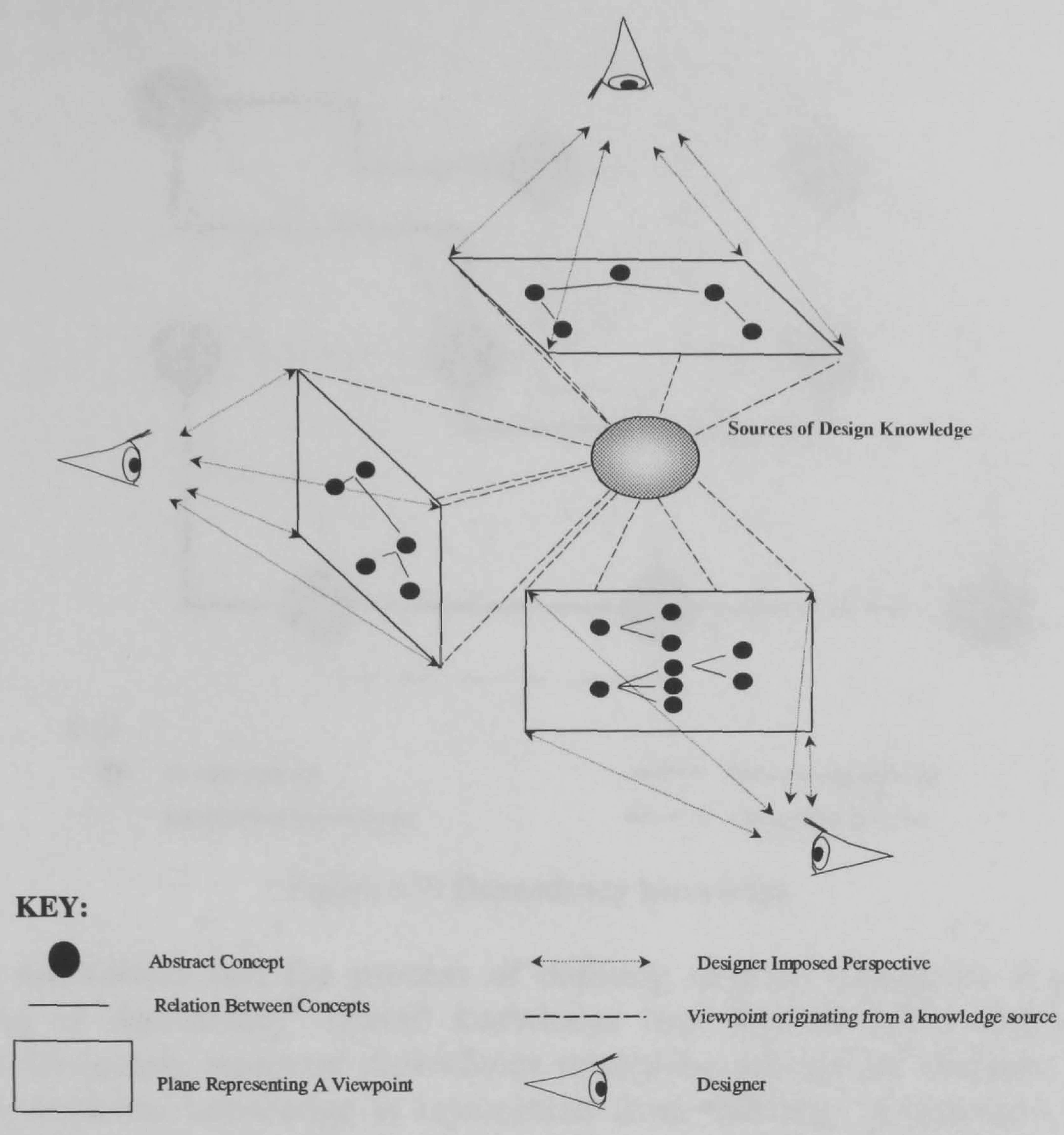


Figure 3.6: Viewpoint approach in design (Duffy and Kerr 1993),

3.5.2 Design lifecycle objectives

A *lifecycle objective* can be considered to be the expression of a required or preferential need with respect to an individual or group of product stakeholders from any stage of the entire product lifecycle (of which the 'design process' can be considered only one phase i.e. customer, designer, manufacturer, assembler, user, maintainer, disposer). The prime concern of this research however is the lifecycle phase of 'design', from 'requirement to definition', and how best to support this 'for re-use'.

3.5.3 Perspective

A *perspective* represents *dependency* knowledge from a particular standpoint as defined by the designer. Figure 3.7 presents an example of a simplified design *viewpoint* showing 9 design *concepts* and their *interdependencies*. The figure illustrates that design *concepts* can have different types of *interdependency*. For example, we see concept A has a material dependency with concept B, an energy dependency with concept D, and both a material and energy dependency with concept E.

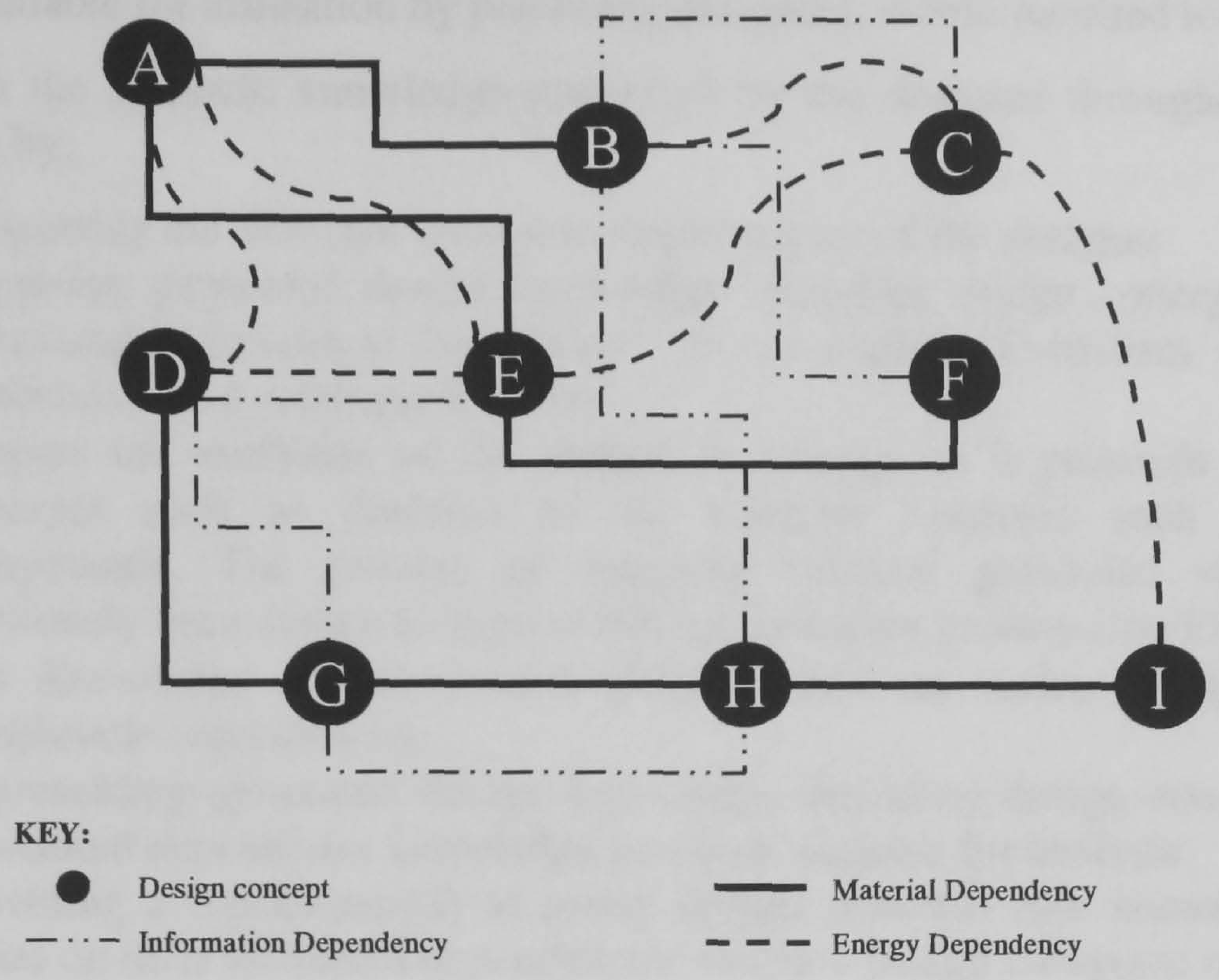


Figure 3.7: Dependency knowledge

It has been established that the process of defining module boundaries is reliant on an understanding of *dependency*[9] related knowledge (see Section 3.2.3) and as such it is important to adequately represent *dependency* related knowledge for analysis. Thus, within this work *dependency* knowledge is represented from differing *perspectives* as shown in Figure 3.8. This allows the designer to explicitly represent and manage *dependency* knowledge based on the particular strategic interests or objectives of the design activity or organisation.

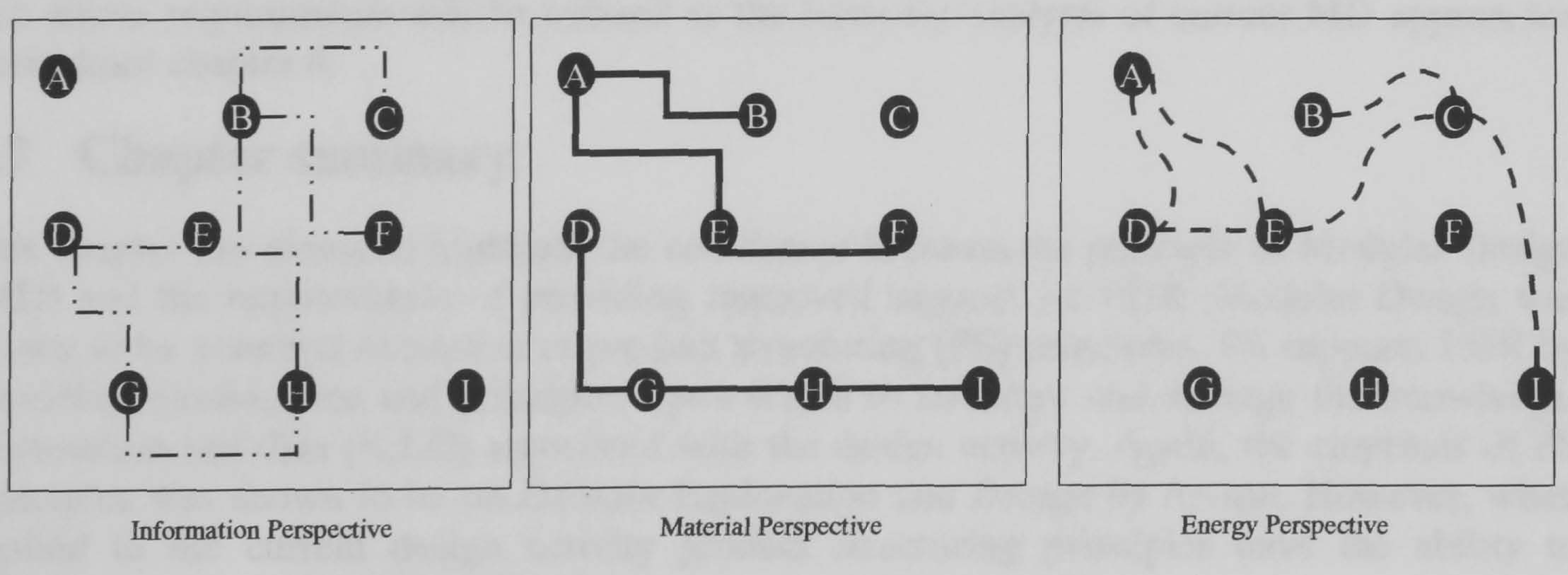


Figure 3.8: Perspectives of dependency knowledge

3.6 Modular design requirements for improved EDR support

The principles of Modular Design (Section 3.2 to 3.4) have been shown to map to the requirements of improved EDR support (Chapter 2). The following list presents a

[9] A *dependency* ≠ *interdependency*. An *interdependency* is a bi-directional dependence between two components whereas a *dependency* illustrates a direction to the dependence.

requirements framework for an MD methodology supporting improved EDR. An MD *methodology,* suitable for utilisation by practising designers, would required to:

- Support the dynamic knowledge generated by the designer throughout the design activity by;
  - Supporting the different *viewpoint* requirements of the designer
  - Capturing generated design knowledge including design *concepts*[10] and their associated *dependency* knowledge[11] in an explicit formalism to support its *exploration* and subsequent *re-use*
  - Support the *evolution* of the design knowledge as it proceeds from abstract *concepts* such as *function* to the concrete *concepts* such as *structural* components. The process of *mapping* between generated *viewpoints* has previously been shown to support this evolutionary process (see Figure 2.4)
- Support *Knowledge Modularisation* (KMn) based on various design *viewpoints* and/or *lifecycle objectives* by:
  - Representing generated design knowledge including design *concept* and their associated *dependency* knowledge in a form suitable for analysis.
  - Providing a mechanism(s) to group design *concepts* into *knowledge modules* based on their identified *dependencies* within a design *viewpoint* or alternatively across design *viewpoints.*
  - Providing a mechanism(s) to optimise the module identification process based on the acknowledged *dependency* knowledge *perspectives.*
  - Providing knowledge of the impact of *trade-off* decisions, when conflict arise based on differing functional, spatial and lifecycle objective *perspectives,* to aid the designer in defining an appropriate modular solution.
- Support the potential *re-use* of generated *knowledge modules* by:
  - Explicitly representing and storing knowledge from a design activity including design *concepts, dependencies* and *module identification* knowledge within and across design *viewpoints.*

The above requirements will be utilised as the basis for analysis of current MD approaches throughout chapter 4.

3.7 Chapter summary

This chapter has aimed to highlight the correlation between the principle of Modular Design (MD) and the requirements of providing improved support for EDR. Modular Design was shown to be a natural extension of product structuring (PS) principles. PS supports EDR by providing mechanisms and principles upon which to structure and manage the knowledge, information and data (K.I.D) associated with the design activity. Again, the emphasis of PS principles was shown to be on *Domain Exploration* and *Design by Re-use.* However, when applied to the current design activity product structuring principles have the ability to provide support to *Design for Re-use.*

MD was defined as a product structuring principle applicable to both single products and product families. The main focus of current MD approaches is on the definition of distinct detachable modules (*component groupings*) within the product to support creation of product

[10] A design *concept* is defined as an element of the design expressed from a particular design *viewpoint. Concepts* are not restricted to representing physical elements (i.e. components) of the design (see Section 3.2.1.)

[11] *Dependency* knowledge is defined as the relational knowledge between *concepts* from different *perspectives* including, but not limited to, energy, information, material and spatial (see Section 3.2.3)

variety and their re-use over product families and product generations. The definition of *component groups,* when based on some form of dependence or similarity, enhances the knowledge related to that component in that the designers has additional knowledge of that component, its relation to others, and its status within the overall design. Thus, MD applied during a current design activity can provide an approach to actively support the process of *Design for Re-use.*

The requirement to support earlier design phases requires that the currently accepted definition of a *module* be extended to include the more abstract *concepts* associated with conceptual design. The term *knowledge module* is used to define the phenomena. Thus throughout this work, modules are defined as *concept groupings* rather than *component groupings*. As such, where *component groupings* are argued to enhance the knowledge related to components, *concept groupings* can enhance the knowledge content of design *concepts.*

Modularity is shown to exist on a scale that is a measure of the internal and external dependencies that are attributable to the modules (*concept groupings*) within the product architecture. Further, the degree of modularity is shown to be dependant on the *viewpoint* taken of the product. The support of design knowledge over such *viewpoints* was shown to be central to improved EDR support.

The process of defining a modular product was shown to be dependant on an understanding of the *dependency* knowledge of the product's components (or in the case of this work *concepts*). *Dependency* knowledge has been shown to be much more complex than the geometric or spatial relations alone, for example, functional *dependencies* such as energy, information and material and *lifecycle objectives* can also have an impact on the modular product. *Lifecycle objectives* represent the embodiment of specific company or product strategies such as ease of maintenance or re-cycling. Thus, *dependencies* between concepts exist with respect to their impact on the fulfilment of these *lifecycle objectives.*

Based on the defined characteristics of both EDR and MD, the requirements for an MD methodology to support improved re-use have been outlined in Section 3.6.

4 Critical Review of Existing Modular Design Methodologies

Chapter 3 established the correlation between the requirement to improve EDR support and capabilities of the product structuring principle Modular Design (MD). Through a discussion of the characteristics of MD the requirements of an MD methodology supporting EDR were outlined (Section 3.6). Chapter 4 presents a critical review of current approaches to MD against the requirements of improved EDR support. Current approaches are reviewed throughout section 4.1 with respect to the requirements identified in Section 3.6. Section 4.2 analyses the main findings of the critical review whilst Section 4.3 summarises the chapter.

4.1 Existing MD methodologies

The following considers current approaches to modular design and demonstrates their failure to fulfil the requirements of MD *methodology* for improved EDR support. Section 4.1.1 outlines the MD methodologies that are to be reviewed. The ability of reviewed methodologies to facilitate design viewpoints is outlined in Section 4.1.2 and 4.1.3. Section 4.1.4 covers the capture and representation of design concepts, and dependency knowledge and perspectives of these including lifecycle objectives. Section 4.1.5 covers module identification and optimisation whilst section 4.1.6 addresses the ability of existing methodologies to support trade-off decision-making.

4.1.1 Reviewed modular design methodologies

There are a number of existing methodologies that support the application of MD principles to varying areas of the engineering domain from the 'designed product' to the 'production system' (see Table 4.1). In addition, MD methodologies can support the exploration and identification of modularity or the selection and/or configuration of pre-defined modules, components and/or products. The work presented in this thesis is concerned with the 'designed product' and more specifically with the supporting knowledge generated throughout the design activities that evolve the product from specification to definition. Thus, MD methodologies that focus on the 'designed product' and the 'exploration and identification' of modularity will be reviewed with particular reference to the requirements outlined in Section 3.6.

Product centred MD methodologies can be defined as that which support modularisation over multiple or single product(s). However, given the context of this thesis, the focus here is predominantly on methodologies that support MD in the context of a single product (see Section 1).[12]

[12] Though not the primary focus, MD methodologies that pertain to the other aspects presented in Table 4.1 are cited during the remainder of this review to describe, illustrate or clarify a point of particular relevance.

MD Methodologies Categorisation			Representative Work
The Designed Product	Exploration and Identification	Single Product	(Ishii, Lee et al. 1995; Kusiak and Huang 1996; Erixon 1997; Gershenson and Prasad 1997; Gershenson and Prasad 1997; Gu, Hashemian et al. 1997; Kamrani 1997; Sosale, Hashemiam et al. 1997; Erixon 1998; Huang and Kusiak 1998; Marshall, Leaney et al. 1998; Nilsson and Erixon 1998; O'Grady and Liang 1998; Gershenson, Prasad et al. 1999; Salheih and Kamrani 1999; Jarventausta and Pulkkinen 2001).
		Multiple	(Chang and Ward 1995; Muffato 1999; Zamirowski and Otto 1999; Gonzalez-Zugasti and Otto 2000; Berti, Germani et al. 2001; Gonzalez-Zugasti, Otto et al. 2001, Gonzalez-Zugasti, 2000; Hofer and Gruenenfelder 2001; Otto 2001; Pedersen, Allen et al. 2001; Schellhammer and Karandikar 2001; Simpson, Maier et al. 2001)
	Selection		(Ouyang, Chenggang et al. 1996)
	Configuration		(Rosen 1996; Tseng and Jiao 1997; Juengst and Heinrich 1998; O'Grady and Liang 1998; Jiao and Tseng 1999; Bracewell and Shea 2001; Dobrescu and Reich 2001; Lashin and Doblies 2001; Siddique and Rosen 2001)
Production System/Units			(He and Kusiak 1998; He and Kusiak 1997); Miller 1999); Rogers and Bottaci 1997); and (Zhou and Irani)

Table 4.1: Reviewed Work

[shaded box] Reviewed category of MD work

(Jiao and Tseng — Additional reviewed methodologies (see footnote 12)

4.1.2 Viewpoint support

The focus of the majority of the MD methodologies, which meet the review characteristics established in section 4.1.1, is on one particular viewpoint or another. Based on the definition of a *viewpoint* given in section 3.5.1, these can generally be grouped into 3 distinct categories: functional, behavioural, and structural.

Function

The focus of Kusiak and Huang (Kusiak and Huang 1996; Huang and Kusiak 1998), and Ouyang et al (Ouyang, Chenggang et al. 1996) is 'functional modularity'. Here, modules are expressed in terms of product (sub)functions. Kusiak and Huang (Kusiak and Huang 1996) state that it is 'desirable to modularise early in the design process' however 'the information to identify the modules might not be available'. Thus, module parameters are 'fuzzily' defined based on product functions and their relation to cost and performance. The realisation of these defined functional modules as physical entities is left to the designer. Their methodology focuses on the functional level to support the designer in a deeper investigation of the 'cost verses performance' trade-offs of potential solutions. As such early representation of modularity supports the designers' subsequent decision-making process while helping to maintain the overall functional integrity of the design.

Ouyang et al (Ouyang, Chenggang et al. 1996) utilises 'functional modularity' to support module selection based on customer requirements. They (Ouyang, Chenggang et al. 1996) functionally decompose pre-existing 'machine tool' modules and the customer requirements that they facilitate. They present a Computer Aided Design (CAD) approach, which incorporates an Expert System (ES), Case-Based Reasoning (CBR) and Artificial Neural Networks (ANN), to select the most suitable pre-existing modules. Thus, the most suitable modules are made available for either their direct re-use, or as the basis for further development by the design team. The approach requires initial acquisition of a substantial amount of domain knowledge. Ouyang et al state that such 'knowledge acquisition is the bottle-neck of intelligent design' and have tried to minimise this problem by combining the differing acquisition capabilities of the three different approaches (ES, CBR, ANN).

The methodologies utilise 'functional modularity' to support objectives of the product lifecycle. For example, the aim of Kusiak and Huangs work (Kusiak and Huang 1996; Huang and Kusiak 1998) is to represent modularity as early in design as possible to support trade-off decisions as the design evolves. Whilst Ouyang et als' (Ouyang, Chenggang et al. 1996) focus is on the initial selection of pre-existing modules based on customer requirements. Thus the focus of these approaches is upon supporting the objectives of those stakeholders concerned with the early product lifecycle, i.e. defining customer requirements and their subsequent incorporation into early conceptual design decisions such as the definition of 'fuzzy modules' for further development (as depicted by Figure 4.1).

Behaviour

'Behavioural modularity' expresses modules in terms of the technical solutions (Erens and Verhulst 1995) or working principles (Zhang 1998) that fulfil the functional requirements of a design. For example, Erixons' Modular Function Deployment (MFD) approach looks at modularity across the wider spectrum of the design process (Erixon 1996; Erixon 1997; Erixon 1998). However, the module identification phase focuses on 'behaviourally expressed modularity' by clustering technical solutions (or working principles) based on their correlation with a number of module drivers. The module drivers represent high-level lifecycle objectives from stakeholders including the designer, salesperson, and assembler. Technical solutions are then grouped into modules based on their ability to satisfy the requirements of such objectives. Thus, as depicted in Figure 4.1, 'behaviourally expressed modularity' broadly supports the objectives of the mid-product lifecycle, i.e. by expressing modules that satisfy the requirements of those stakeholders involved in the product definition phase.

Structure

'Structural modularity' is the focus of, amongst others, Gershensen (Gershenson and Prasad 1997; Gershenson and Prasad 1997; Gershenson, Prasad et al. 1999), Gu et al (Gu, Hashemian et al. 1997), Ishii et al (Ishii, Lee et al. 1995), Kamrani (Kamrani 1997), and Sosale et al (Sosale, Hashemiam et al. 1997). Here, modules are expressed as groups or clusters of components, parts or assemblies. Due to the need to have previously defined the component parts, the application field of 'structural modularity' is often redesign in domains where the product is mature and the parts inventory stable.

Gershenson, defines MD methodologies for service and maintenance (Gershenson and Prasad 1997), and manufacturability (Gershenson and Prasad 1997). Both methodologies decompose an existing product into a component tree diagram, which represents the general partitioning of components in that product. A Service Mode Analysis (SMA) (Gershenson and Prasad 1997) or manufacturing graph (Gershenson and Prasad 1997) is also developed for the product, which represents the general partitioning of service (Gershenson and Prasad 1997) or manufacturing operations (Gershenson and Prasad 1997) for the product. An analysis of the component tree against the SMA or manufacturing graph allows the designer to map service or manufacturing operations to the components and thus identify improvements in component 'groupings' based on the requirements with the operations in question. Gu et al (Gu, Hashemian et al. 1997), Sosale et al (Sosale, Hashemiam et al. 1997) and Ishii et al (Ishii, Lee et al. 1995) focus on modularity for product retirement whereby physical parts are grouped into modules based on their similarity in areas such as life span, material, maintenance level, disposal method, and recycling capabilities. Kamrani (Kamrani 1997) utilises 'structural modularity' to gain potential process improvements. Kamrani's methodology (Kamrani 1997) is applied to well-developed design domains where technology, materials and design are mature and subject to little flux. The aim of the methodology is to maximise kinship in terms of manufacturing requirements and assembly operations.

'Structurally expressed modularity' focuses predominantly on objectives of the later product lifecycle i.e. by expressing modularity that satisfies the requirements of those stakeholders involved in the manufacture, use and disposal of the product (see Figure 4.1).

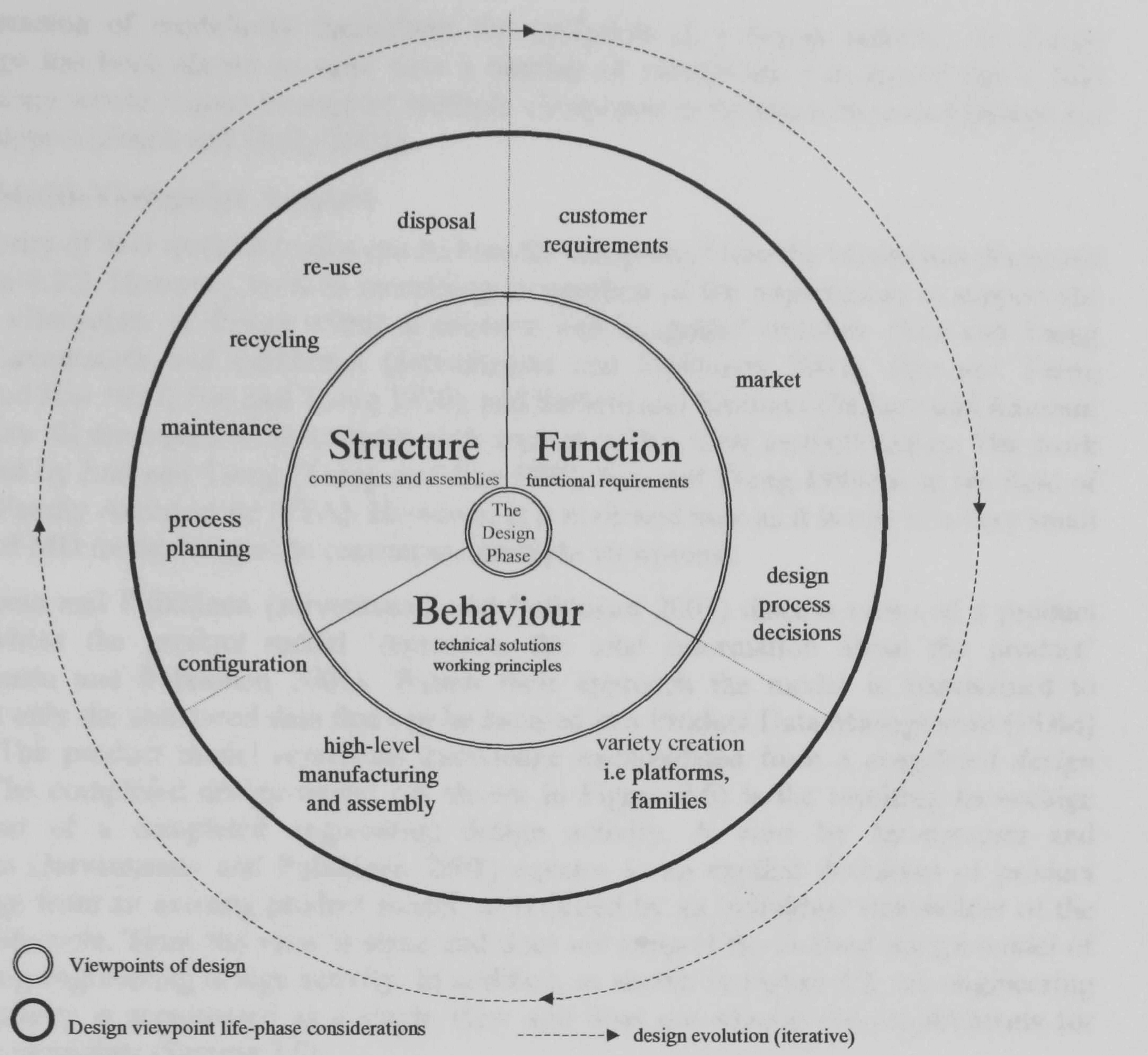


Figure 4.1: The life-phase consideration associated with the viewpoints in MD

Each MD methodology *viewpoint* discussed above has been shown to support objectives relating to particular stakeholders from different *phases* of the product *lifecycle* as depicted by Figure 4.1. The figure depicts the design phase of a product lifecycle illustrating the viewpoints utilised in MD and the design life-phase considerations managed by each. The radial axis progresses from the general (design phase) to the specific (life-phase considerations). Design is characterised by iteration (Smith and Eppinger 1993), and the dashed line represents the iterative evolution of design knowledge.

A general characteristic of each methodology, regardless of the *viewpoint,* is that design knowledge generated out-with the considered viewpoint is generally not explicitly captured, modelled nor utilised as part of the methodology. Kamrani (Kamrani 1997) notes this point when he expresses concern that 'conceptual design modules', those of a functional to behavioural nature, cannot meet the constraints of later detailed stages of design. Further, it can be said that due to the nature of 'structural modules' they fail to capture and/or explicitly represent knowledge from earlier conceptual phases of design.

Based on the requirements of an MD approach, on which to actively support EDR, (see Section 2.6 and Section 3.6) a MD methodology requires to support the dynamic knowledge generated by the designer by facilitating the capture and management of design knowledge and their relations across varying *viewpoints* throughout the *evolution* of the design. Thus, the developed methodology requires to support the systematic representation, examination and expression of modularity throughout the evolution of a design activity. As design knowledge has been shown to exist over a number of viewpoints it is argued that a MD methodology would require to support multiple viewpoints to facilitate improved *Design for Re-use* support (Smith and Duffy 2001).

4.1.3 Multi-Viewpoint support

The majority of MD methodologies can be broadly categorised into the viewpoints discussed in section 4.1.2. However, there is increasing recognition of the requirement to support the multiple viewpoints of design within a coherent and integrated structure (Jiao and Tseng 1999). Jarventausta and Pulkkinen (Jarventausta and Pulkkinen 2001), Jiao and Tseng (Tseng and Jiao 1997; Jiao and Tseng 1999), and Salheih and Kamrani (Salheih and Kamrani 1999) have all attempted to encompass such aspects within their methodologies. The work carried out by Jiao and Tseng (Tseng and Jiao 1997; Jiao and Tseng 1999) is in the field of Product Family Architecture (PFA). However, it is reviewed here as it is one of a very small number of MD methodologies to encompass multiple viewpoints.

Jarventausta and Pulkkinen (Jarventausta and Pulkkinen 2001) discuss *views* of a product model where the product model 'represents the total information about the product' (Jarventausta and Pulkkinen 2001). Within their approach the model is streamlined to represent only the structured data that can be handled in a Product Data Management (PDM) system. The product model represents knowledge encapsulated from a *completed design model*. The completed design model (as shown in Figure 2.6) is the resulting knoweldge component of a completed engineering design activity. A view by Jarventausta and Pulkkinen (Jarventausta and Pulkkinen 2001) equates to an explicit definition of product knowledge from an existing product model, as required by an individual stakeholder of the product lifecycle. Thus, the view is static and does not support the *evolved design model* of an ongoing engineering design activity. In addition, as shown in Figure 4.2, the engineering design activity is represented as a single view and does not support the requirements for multiple viewpoints (Section 3.6).

Figure 4.2 illustrates the distinction between the *views* adopted by Jarventausta and Pulkkinen and the *viewpoints* which are proposed as part of this work. Jarventausta and Pulkkinen propose that the engineering design activity be viewed as an attached view of the product model. The attached view is a static representation of product knoweldge as it relates to a particular stakeholder (i.e. the engineering designer) whereas viewpoints encapsulate the differing knowledge requirements of engineering designer which evolve the product from abstract concepts to a concrete defintion. Thus, Jarventausta and Pulkkinen (Jarventausta and Pulkkinen 2001) utilise modular design principles to support the decomposition and clustering of product knowledge from an existing product model to support improved product data management.

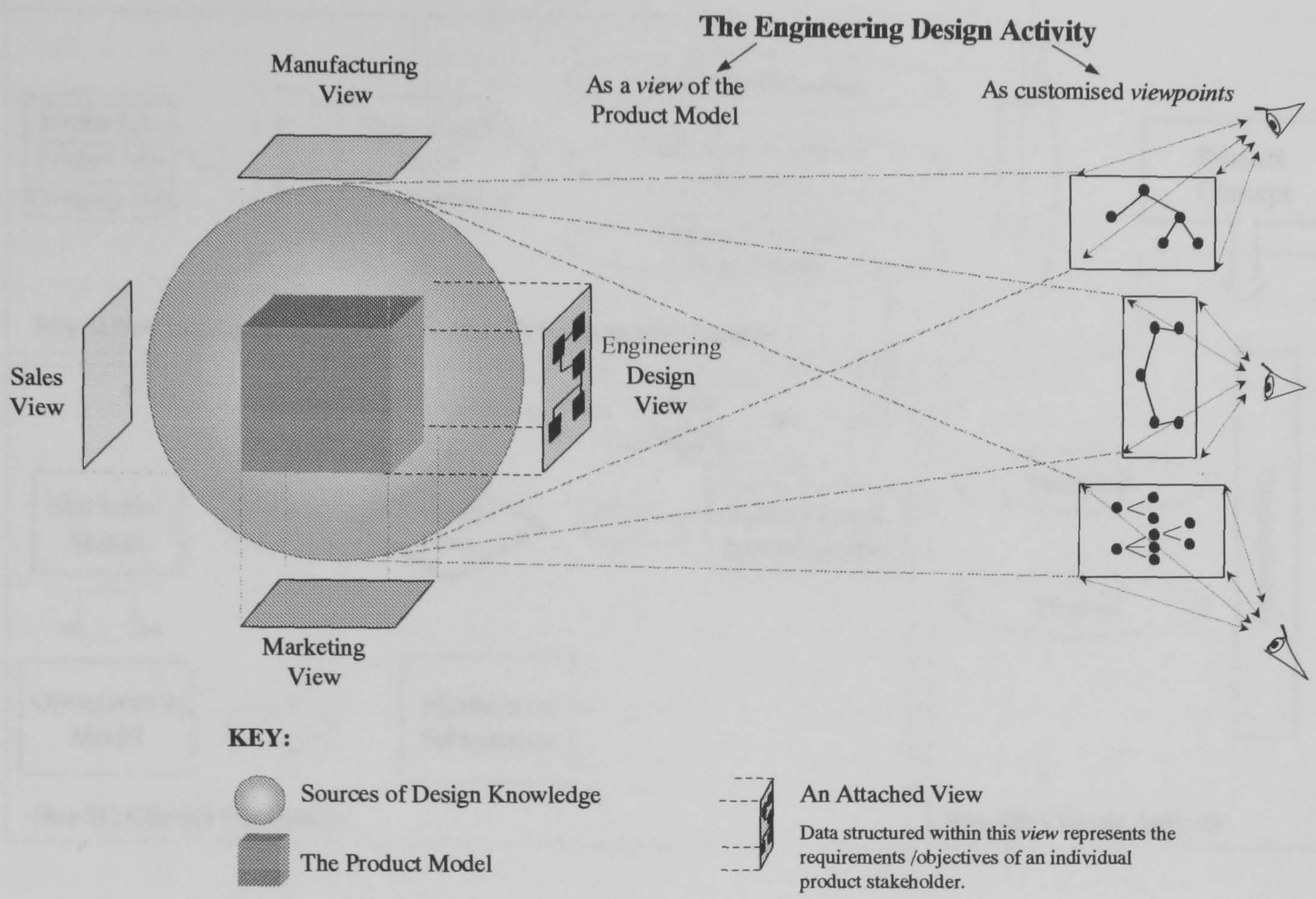


Figure 4.2: The engineering design activity as a *view* (Jarventausta and Pulkkinen 2001) of the product model and as *viewpoints* (Duffy and Kerr 1993)

Acknowledging the need to support different *viewpoints* of the design Salheih and Kamrani (Salheih and Kamrani 1999) note that the principle of modularity 'can be applied in product design, design problems, production systems, or all three'. They present a 4-step modular design methodology, as shown in Figure 4.3, covering the design process from need to concept. The aim of the methodology is to determine the modularity that exists in design concepts and utilising these modules as the basis for the allocation of development teams. The detailed design of such modules is left to the development teams.

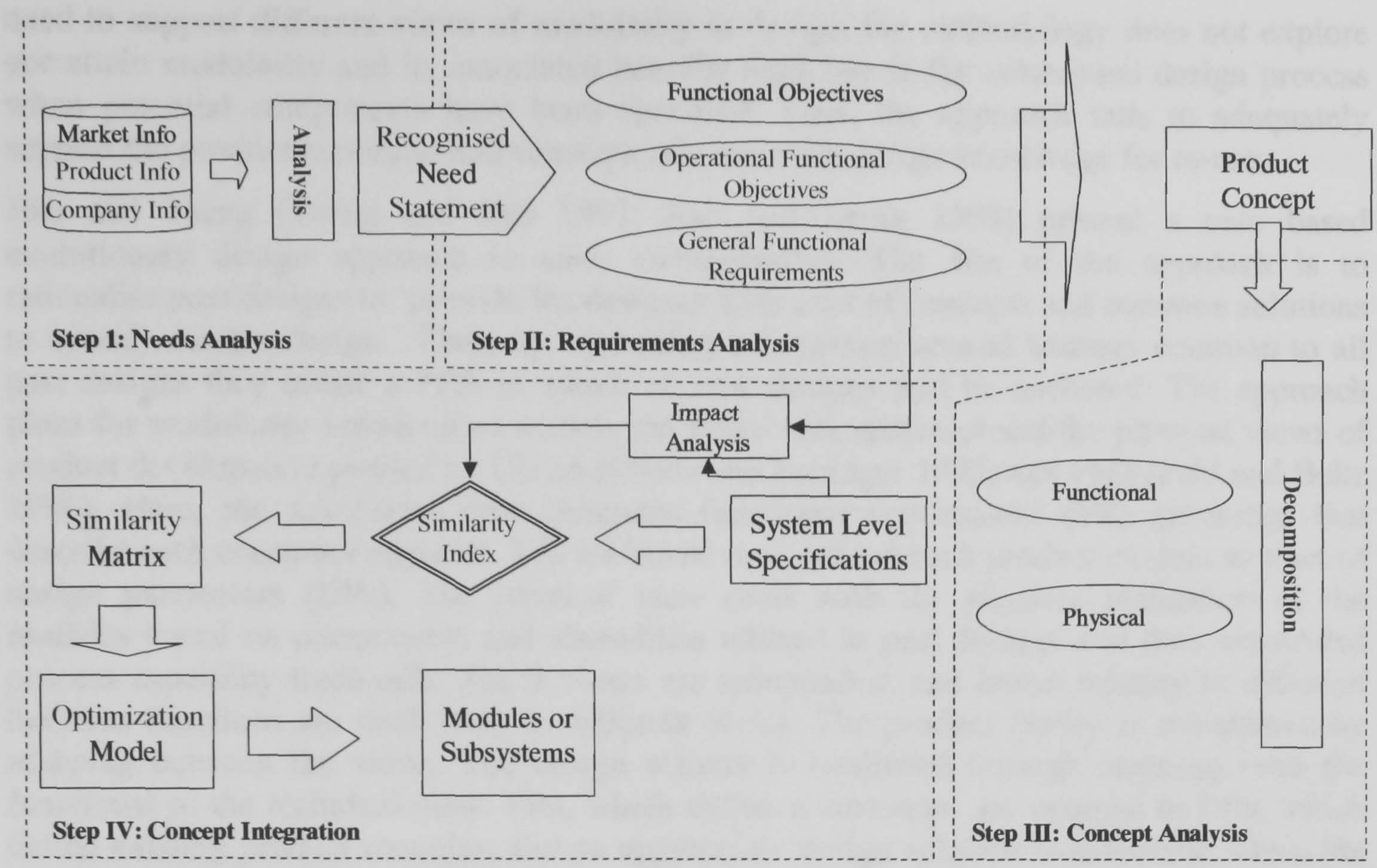


Figure 4.3: Macro level Design for modularity (Salheih and Kamrani 1999)

As illustrated in Figure 4.3, the 4 methodology steps are as follows:

1. Needs Analysis – the outcome of which is a 'needs statement' that fully defines the product in terms of its functional needs and physical limitations. These are arranged into groups and prioritised according to their importance.
2. Requirements Analysis – the outcome of which is list of product requirements. These are broken down into: the functional objectives needed to meet customers' primary needs, the operational functional requirements that are primary requirements of the product (both functional and physical objectives), and secondary objectives (known as general functional requirements) which are desirable but do not effect the main function of the product.
3. Concept Analysis – the outcome of which is a basic functional and physical decomposition of the product conceptual designs.
4. Concept Integration - the outcome of which is the arrangement of basic components into modules and their subsequent integration into a functional system.

Product modularity is only formally explored in the last step (step 4: concept integration) and through one viewpoint (a structurally oriented viewpoint). The previous, three steps are data gathering steps which support the module identification and integration activities of Step 4. Thus, the product is specified to a high level before modular analysis occurs. Further, due to the need to decompose the 'product concept designs' and gather a significant amount of market, customer and product data their methodology pertains predominantly to cases of redesign and/or where the design and problem domain are well developed, documented and understood. Thus, they term their approach as 'macro level' as they observe modularity only at the level where the granularity of design problem and process is at a relatively low level i.e. basic needs, requirements and component types. Despite an acknowledgement of the need to support different views of modularity in design, the methodology does not explore nor attain modularity and its associated benefits until late in the conceptual design process when potential components have been specified. Thus, the approach fails to adequately support the product evolution and consequently maintain design knowledge for re-use.

Jiao and Tseng (Tseng and Jiao 1997; Jiao and Tseng 1999) present a case based evolutionary design approach to *mass customisation*. The aim of the approach is to rationalise *past designs* to 'provide the designer with a set of concepts and common solutions to specify current design'. Thus, by organising information around features common to all past designs they create a PFA in which all new designs will be anchored. The approach plans for modularity across *views* namely the functional, technical and the physical views of product development posited by Ulrich (Ulrich and Eppinger 1995) and Pahl (Pahl and Beitz 1996). Here, the *functional view* generates functional requirement (FR) groupings that describe each consumer segment. The *technical view* defines each product module as a set of design parameters (DPs). The *physical view* deals with the physical realisation of the modules based on components and assemblies utilised in past designs and their associated process capability trade-offs. The 3 views are independent and issues relating to different business functions are dealt with in different views. The product family is maintained by *mapping* between the views. The design activity is facilitated through mapping from the functional to the technical view. FRs, which define a customer, are mapped to DPs, which define existing product modules, and an appropriate design solution is generated. Thus, the approach aims initially at defining the underlying modularity of already existing products and their associated customer base, and subsequently at ensuring all new products are anchored in the existing PFA.

Jarventausta and Pulkkinen (Jarventausta and Pulkkinen 2001) and Jiao and Tseng (Tseng and Jiao 1997; Jiao and Tseng 1999) focus on past designs, representing the modularity within *completed design model(s)* to support improved data management (Jarventausta and Pulkkinen 2001) and the design of new product family variants (Tseng and Jiao 1997; Jiao and Tseng 1999). Thus, their methodologies apply to cases of redesign or cases that do not 'include novel engineering design tasks, but systematic variant design' (Jarventausta and Pulkkinen 2001). A further limitation of their methodologies is that they require access to an abundance of customer and marketing information. For example, Jiao and Tseng apply their methodology to industrial products as particular features of the market, in comparison with the consumer product market, make customer requirements analysis easier including; advanced customer product knowledge, the concrete factors on which purchase decisions are made, and the limitations in the number and type of customers. On the other hand, Salheih and Kamrani (Salheih and Kamrani 1999) support current design and the *evolved product model*. However, the limitations of their approach in terms of the requirements for an abundance of product and market data (similar to that noted in (Tseng and Jiao 1997; Jiao and Tseng 1999; Jarventausta and Pulkkinen 2001)) and the failure to express product modularity until the product components are defined (at least at a conceptual level), negates this as a comprehensive MD methodology for improved EDR support.

All three methodologies acknowledge that one can take multiple *views* of product knowledge. However, the methodologies tended to consider the modularity of an engineering design from a single view. The work presented in this thesis focuses solely on engineering design activity and its associated *viewpoints* with the intention of furthering our understanding of the modularity of design *knowledge* within and across these viewpoints.

4.1.4 Capture and representation of concepts and dependency knowledge

Chapter 2 identified the importance of consistent capture of generated design knowledge, whilst Chapter 3 outlined the importance of knowledge related to design *concepts* and *dependencies* between these, to the exploration, identification and definition of modularity within the design activity. As such, the capabilities of existing MD methodologies with respect to the capture and representation of design concepts and dependency knowledge is the focus of the following section. The section is split into the representation of design concepts, concept dependencies and lifecycle objectives[13].

As discussed in Section 2.3, consistent capture and modelling of design knowledge during the activity of design can facilitate improved EDR support. It is suggested that the application of a consistent and formalised approach to design concept knowledge modelling would facilitate the generation of *knowledge modules* and the subsequent utilisation of modularity as a facilitator of design knowledge management (i.e. *knowledge modularisation*). Thus, an MD methodology that supports consistent and formalised capture of design concepts and the subsequent exploration of the modularity of these concepts would improve EDR support.

Concepts

Gershensen (Gershenson and Prasad 1997; Gershenson and Prasad 1997; Gershenson, Prasad et al. 1999); Gu et al (Gu, Hashemian et al. 1997), Ishii et al (Ishii, Lee et al. 1995), Kamrani (Kamrani 1997), and Sosale et al (Sosale, Hashemiam et al. 1997) represent modularity from a purely structural viewpoint where modules are physical entities. Modules are defined as groups of design components and/or sub-assemblies. Abstract design knowledge from earlier in the design activity is not explicitly expressed as part of the MD activity. These MD methodologies do not facilitate representation of earlier, more abstract, function and behaviour concepts and as such, cannot adequately support the notion of *knowledge modules.*

A number of methodologies represent modularity from a functional (Kusiak and Huang 1996; Ouyang, Chenggang et al. 1996; Huang and Kusiak 1998) or behavioural (Erens and Verhulst 1995) viewpoint. These recognise that a module need not be restricted to being a physical entity. However, these MD methodologies represent only the design concepts from the particular viewpoint that they embody and not those generated during other viewpoints. Knowledge related to other viewpoints is not explicitly captured or represented as part of the methodology. Thus, although these MD methodologies embody the notion of the *knowledge module* they are not adequate facilitators of *knowledge modularisation* as they fail to capture, represent or maintain significant portions of design related knowledge. Consequently, they do not facilitate inter-viewpoint modularisation, i.e. the maintenance of the modular solution as the design activity evolves.

Existing methodologies that encompass a multi-viewpoint approach (Tseng and Jiao 1997; Jiao and Tseng 1999; Salheih and Kamrani 1999; Jarventausta and Pulkkinen 2001), as discussed in Section 4.1.3, represent knowledge from different *viewpoints*. However, as discussed above the viewpoints utilised by both Jarventausta and Pulkkinen (Jarventausta and Pulkkinen 2001) and Jiao and Tseng (Tseng and Jiao 1997; Jiao and Tseng 1999) are not representative of those generated during the design activity phase of a product life but cover the product lifecycle at a relatively low-level of detail. In such methodologies concepts related to the design activity are represented within a single viewpoint (Jarventausta and Pulkkinen 2001) or the mapping from one viewpoint (technical) to the consecutive viewpoint (physical) (Tseng and Jiao 1997; Jiao and Tseng 1999). Thus, the multiple viewpoints, which can be taken of design knowledge, are not explicitly formalised nor is the modularity of these viewpoints explored and expressed. Both methodologies (Tseng and Jiao 1997; Jiao

[13] The fulfilment of *Lifecycle Objectives* has been shown in Section 3.2.3 to impose further *dependencies* between design *concepts.*

and Tseng 1999; Jarventausta and Pulkkinen 2001) are based on the exploration of past designs (*completed design models)* and do not support the evolution of design knowledge, and the respective viewpoints, as generated during the design activity. On the other hand, Salheih and Kamrani (Salheih and Kamrani 1999) support current design activities. However, they (Salheih and Kamrani 1999) do not utilise a consistent modelling formalism throughout the design activity and only explore modularity in the later stages of the design activity. For example, knowledge from earlier in the design process is expressed only as System Level Specifications (SLS). SLS represents one-to-one relationships between components with respect to their functional and physical characteristics. For example, an SLS can be based upon the functional and physical requirements of a customer group expressed during market analysis. The defined SLS are utilised as the basis for a dependency analysis of the product *components*. Thus, the methodology utilises the more abstract knowledge of the design to define the functional and physical dependence between the components (concrete concepts). Thus the resulting methodology expresses modularity late in the design process (only in terms of physical entities) and does not explicitly explore the modularity of more abstract design concepts i.e. knowledge modularity. Thus, due to the requirements for a number of existing *completed design models* (Tseng and Jiao 1997; Jiao and Tseng 1999; Jarventausta and Pulkkinen 2001) and/or the lack of a consistent knowledge modelling mechanism (Salheih and Kamrani 1999), current multi-viewpoint approaches fail to adequately represent the *concepts* (and their respective *viewpoints*) generated during the design activity and consequently cannot facilitate the exploration and definition of *knowledge modularity* in the *evolved design model.*

Dependencies

Relations, both within and between modules, can be considered to involve complex *dependencies* related to functional, behavioural, structural, spatial, information, energy and material *perspectives* of the design (see Section 3.2.3). Section 3.2.4 established that the process of exploring and defining modularity is reliant on an understanding of dependency related knowledge.

In general in the literature dependencies between concepts have been represented as lines in a graph (Ishii, Lee et al. 1995) (Kusiak and Huang 1996) or tree (Gershenson and Prasad 1997; Gershenson and Prasad 1997); as illustrated in Figure 4.4(a), or entries in a matrix (Erixon 1996; Erixon 1997; Gu, Hashemian et al. 1997; Sosale, Hashemiam et al. 1997; Erixon 1998; Huang and Kusiak 1998; Jarventausta and Pulkkinen 2001) as in Figure 4.4(b) or a combination of both (Gershenson, Prasad et al. 1999; Jiao and Tseng 1999; Salheih and Kamrani 1999).

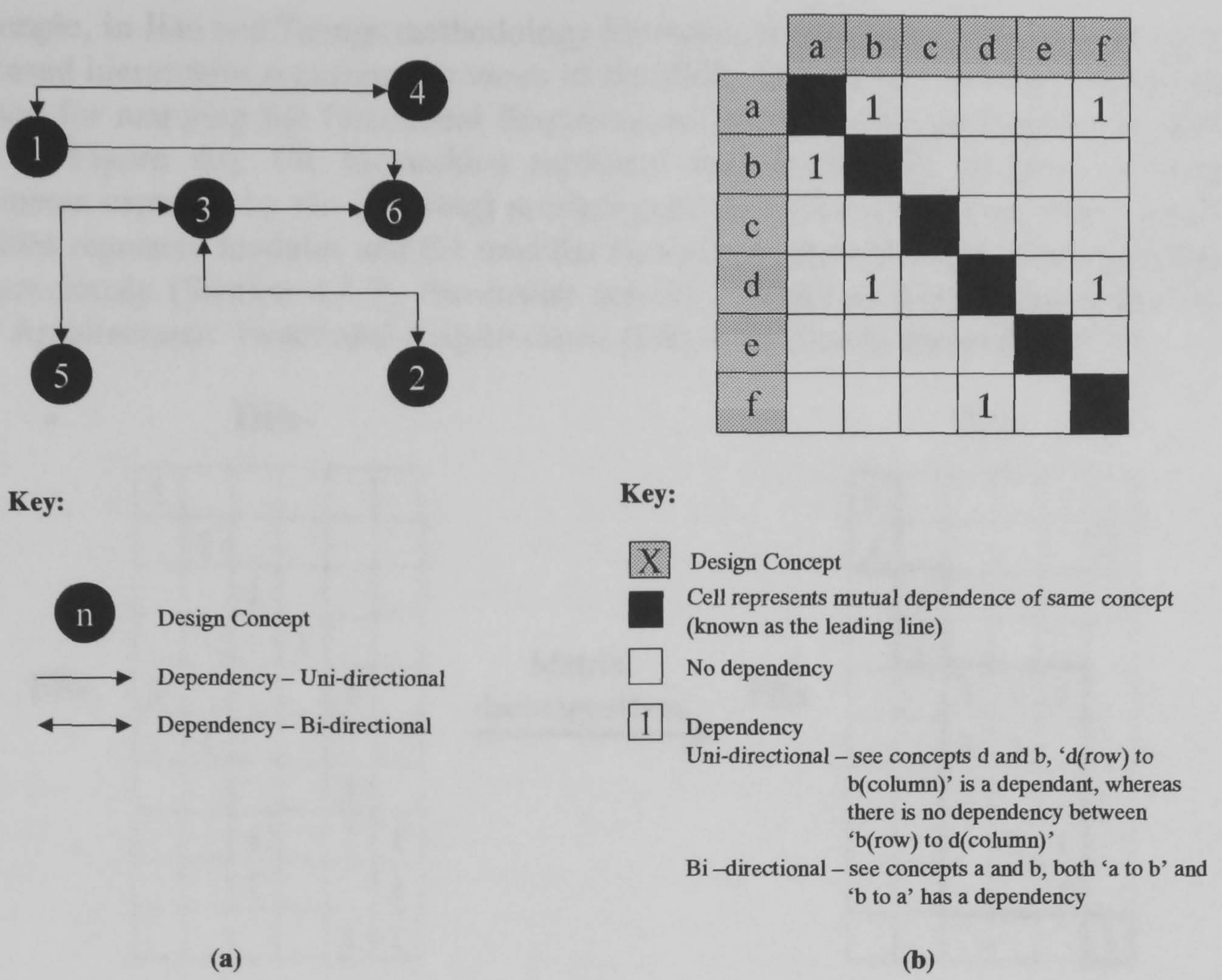


Figure 4.4: Graph and matrix based representations of dependencies

The matrix can be square (same concepts are represented in both row and column) as in (Salheih and Kamrani 1999; Jarventausta and Pulkkinen 2001) or non-square (different concepts are represented by the row than the columns) as in (Jiao and Tseng 1999). Further, the dependencies within a square matrix can be symmetrical (the dependencies are bi-directional and are mirrored by the 'leading line') or asymmetrical (the dependencies are directional and may differ on each side of the 'leading line'). For example, consider the dependency between a power supply unit and a visual display unit (VDU). The VDU is dependant on the power supply unit but the same is not true for the reverse case. In general energy relations such as these are uni-directional. However, the dependency between two components that are physically linked is, by nature, a bi-directional relation.

It is possible to convert between these representations (Pimmler and Eppinger 1994) as illustrated in Figure 4.5.

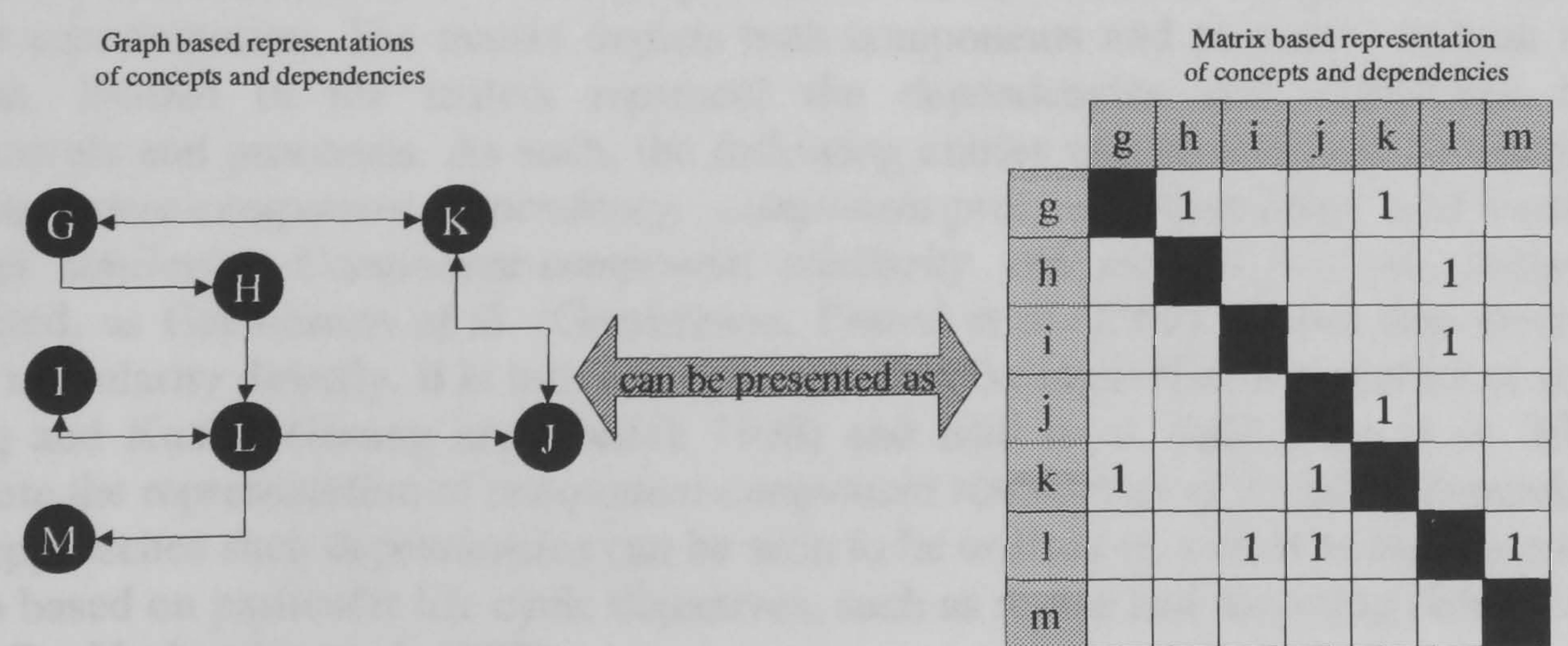


Figure 4.5: Dependency data presented as both graph to matrix representations

For example, in Jiao and Tsengs methodology for *mass customisation* (Jiao and Tseng 1999) graph based hierarchies represent the *views* of the PFA, whilst a matrix based representation is utilised for mapping the Functional Requirements (FRs) to Design Parameters (DPs) as shown in Figure 4.6. FR hierarchies represent the 'underlying patterns of customer requirements captured by the [existing] product portfolio' (Jiao and Tseng 1999) whilst DP hierarchies represent modules and the modular structure of the existing product portfolio. As stated previously (Section 4.1.3), the design activity is facilitated by mapping the Product Family Architectures' Functional Requirements (FR) to its Design Parameters (DP).

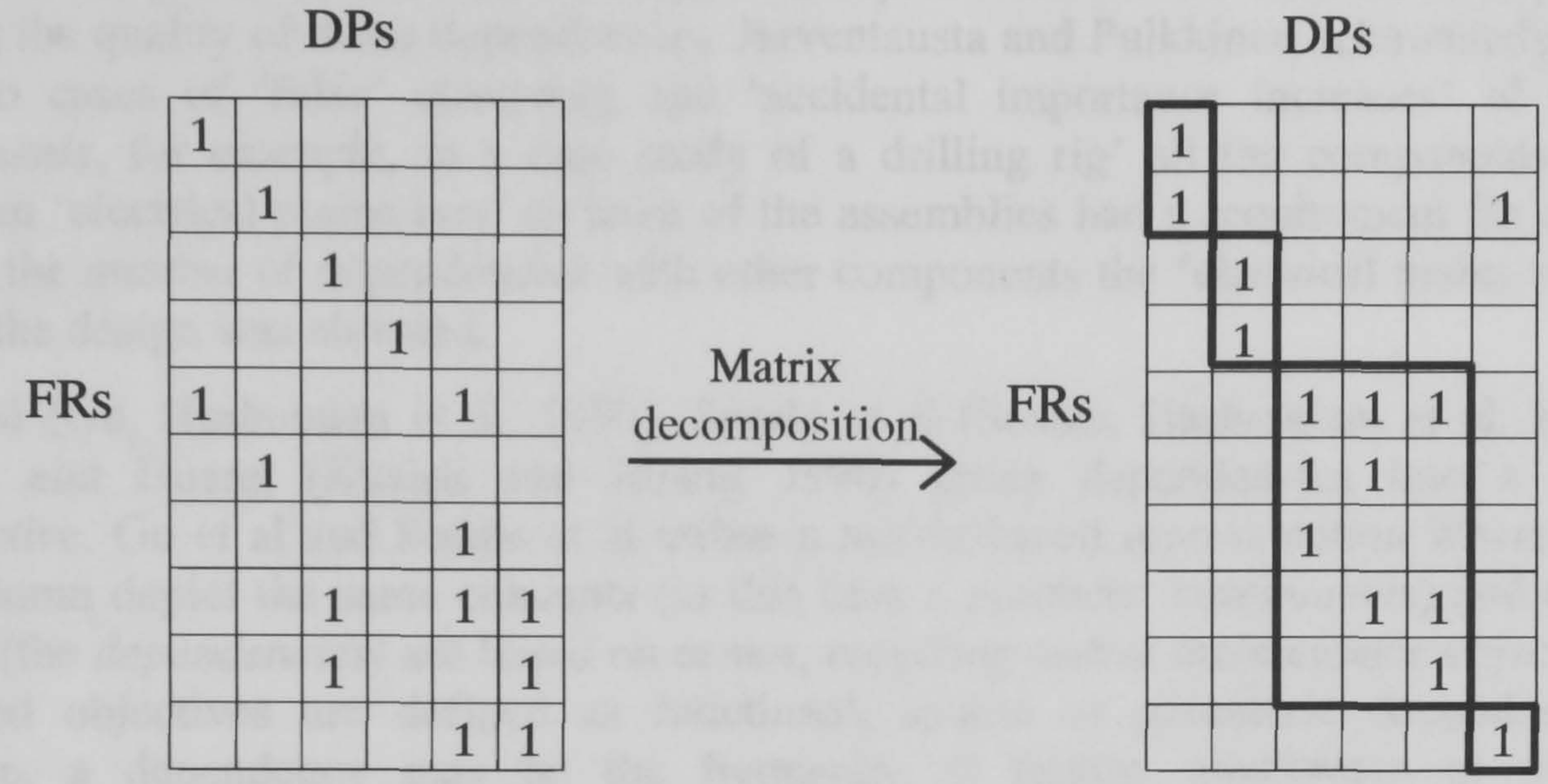


Figure 4.6: The design activity - mapping from FRs to DPs (Jiao and Tseng 1999)

Here, the matrix is non-square, i.e. the concepts in the rows and columns differ. The row represents functional requirements of the PFA. The columns represent the design parameters associated with the PFA. The dependencies represent a 'what-how' relationship (also referred to as a *causal relation* (Zhang 1998)) and dependencies either exist (1) or do not exist (0). Salheih and Kamrani (Salheih and Kamrani 1999) utilise a similar mix of both graph and matrix based representations of design concepts and their dependencies within their methodology. However, the similarity matrix, which is analysed to identify independent component modules, is a square, symmetrical matrix.

A similar mix of graph and matrix based representations is utilised in work by Gershenson et al. An existing product is decomposed into a graph-based representation of its assemblies with 'part-of' relations. The manufacturing processes for the bottom level components are attached to the bottom level of the graph. However, the modular analysis is based upon a matrix representation. The matrix depicts both components and processes in both row and column. Entries in the matrix represent the dependencies and similarities between components and processes. As such, the following entries can be made in the matrix based on *component-component dependency, component-process dependency* and *component-process similarity. Component-component similarity* and *process-process similarity* are neglected, as Gershenson et al. (Gershenson, Prasad et al. 1999) believe that these do not affect modularity directly. It is interesting to note that Gu et al. (Gu, Hashemian et al. 1997), Huang and Kusiak (Huang and Kusiak 1998) and Ishii et al. (Ishii, Lee et al. 1995), all advocate the representation of *component-component similarities* as a type of dependency. In their approaches such dependencies can be seen to be utilised as a basis to facilitate modular design based on particular life cycle objectives, such as re-use and recycling (Ishii, Lee et al. 1995; Gu, Hashemian et al. 1997).

In work by Jarventausta and Pulkkinen (Jarventausta and Pulkkinen 2001), and earlier work by Gershenson and Prasad (Gershenson and Prasad 1997; Gershenson and Prasad 1997), dependencies between concepts are based on a single *perspective* of design. Jarventausta and Pulkkinen (Jarventausta and Pulkkinen 2001) utilise a matrix-based representation of design concepts and their dependencies whilst Gershenson and Prasad (Gershenson and Prasad 1997; Gershenson and Prasad 1997) utilise a tree-based representation. In all cases, the dependencies represent 'part-of' relations between two components and have a value of either '1' or '0', i.e. they either exist or do not exist. No indication of their direction or importance is given. Thus, their subsequent analysis relies on the amount of dependencies and not the quality of those dependencies. Jarventausta and Pulkkinen acknowledge that this leads to cases of 'false' clustering and 'accidental importance increases' of secondary *components*, for example, in a case study of a drilling rig' all the components clustered inside an 'electrical mains box' as most of the assemblies had a requirement for electricity. Due to the number of dependencies with other components the 'electrical mains box' status within the design was elevated.

Gu et al (Gu, Hashemian et al. 1997), Sosale et al (Sosale, Hashemiam et al. 1997), and Kusiak and Huang (Kusiak and Huang 1996) group dependencies into a combined *perspective*. Gu et al and Sosale et al utilise a matrix-based representation where both row and column depict the same concepts (in this case a products' components) and the matrix entries (the *dependencies)* are based on re-use, recycling and/or maintenance objectives. The specified objectives are defined as functional, spatial or geometric dependencies. For example, a dependency may be the frequency of failure, attachment or down time similarities. An interaction analysis is carried out between all components and each defined objective. The resulting interaction matrices are combined into an overall weighted average matrix. This matrix depicts dependencies as the weighted average of all the previously analysed component interactions. The matrix is symmetrical about the leading line. Thus, the direction of the dependency has been neglected.

Kusiak and Huang recognise that different dependencies can exist based on functional similarities such as 'geometric, temporal, force, electrical, thermal and photometric' (Kusiak and Huang 1996). However, all dependencies with their presented work are represented by one connection between the nodes of an interaction graph as shown in Figure 4.7.

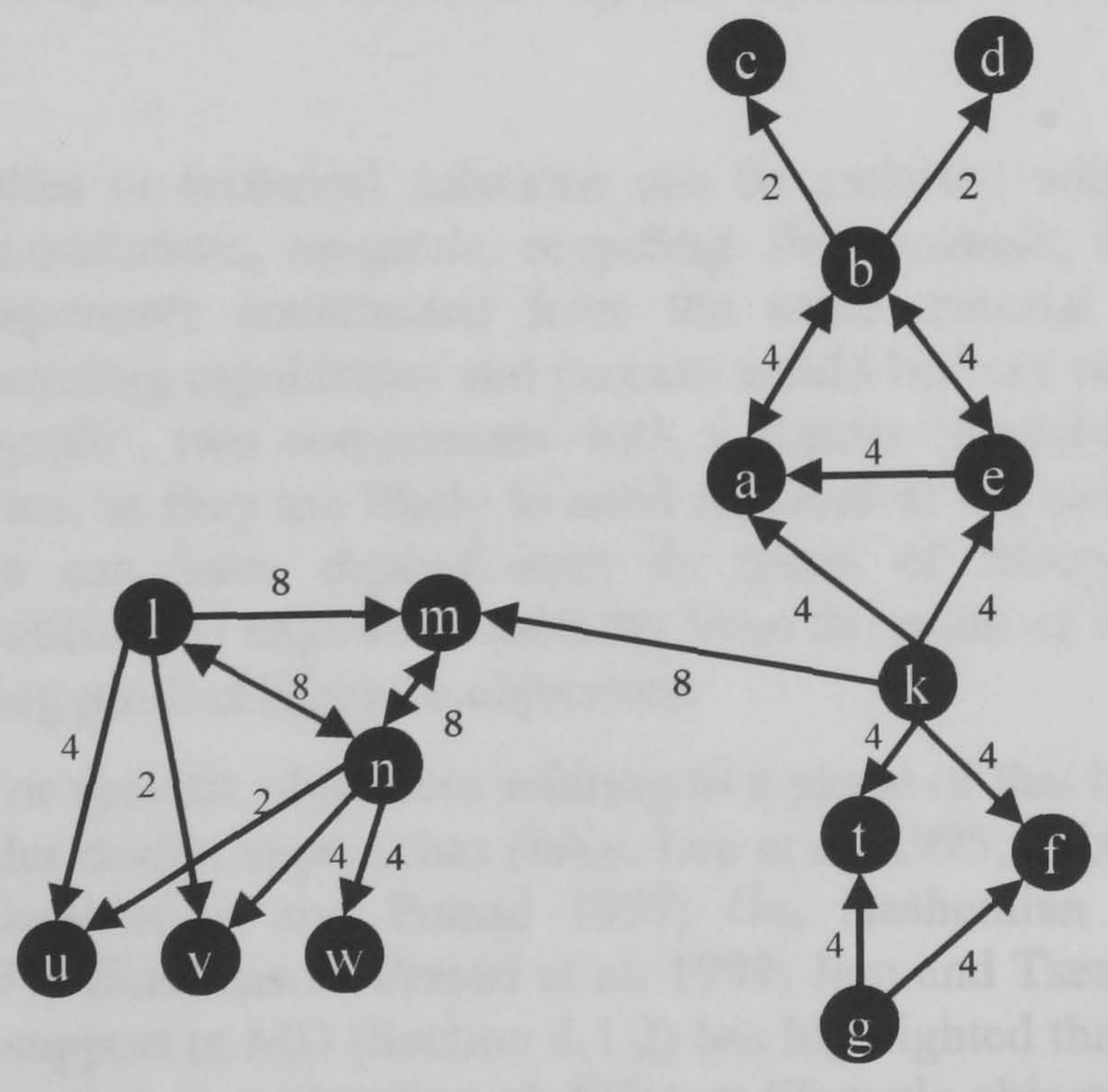


Figure 4.7: An interaction graph (Kusiak and Huang 1996)

Therefore, it is not clear if their approach actually makes such a distinction between these different types of dependencies when identifying product modules. However, it is clear that the approach recognises that dependencies can be directional and have varying degrees of importance. These are depicted by the direction(s) of the interactions between nodes of the graph and the figure attributed to these interactions.

Later work by Huang and Kusiak (Huang and Kusiak 1998) utilises a matrix-based representation of the product components and their interactions (*dependencies*). In addition to an interaction matrix, which represents the functional dependence between components, a suitability matrix is also defined. The suitability matrix is a 'component-component incidence matrix', which 'represents the suitability of components for inclusion into a module' (Huang and Kusiak 1998). Akin to their earlier publication (Kusiak and Huang 1996), the between-component *dependencies* have both direction and importance ratings but no explicit representation of the different *perspectives* that can be taken of such *dependencies*.

It can be seen that in many MD methodologies *dependencies* are predominantly treated as one-dimensional i.e. that a relationship exists. However, it has been shown that *dependencies* can be viewed from different *perspectives,* can have varying degrees of importance, and can be directional. The importance of these *dependencies* to the process of MD has also been discussed (Section 3.2.3).

MD methodologies have made attempts to overcome the inadequacies of *dependency* representation by assigning weighted measures (Kusiak and Huang 1996; Gu, Hashemian et al. 1997), or similarity functions (Huang and Kusiak 1998) to *dependencies.* However, none fully support *dependency* characteristics in terms of type, importance, and direction. In addition, *dependencies* exist between design *viewpoints.* These *dependencies* represent the mapping from one *viewpoint* to another and have been shown to support the evolution of the design activity. No approach has been found which adequately supports design *viewpoints* and, as a consequence, do not explicitly support these *dependency* types. Thus, it is suggested that a more complete representation of *dependency* knowledge which provides more adequate support of 'within' and 'between' module *dependencies,* from different *perspectives* and across *different viewpoints,* would better support the exploration and identification of *knowledge modularity* in the engineering design activity.

Lifecycle objectives.

Components, assemblies or technical solutions can be analysed with respect to lifecycle objectives such as manufacture, up-grade, recycling. For example, from the objective of 'recycling', two components constructed from the same material would have a high dependence, as the recycling capabilities and process would be very similar. Likewise, from the objective of 'upgrade', two components with a similar 'useful-life-in-service' would have a high dependence, as they are likely to need replaced at the same time. Thus, we see that design *concepts* can have *dependencies* in terms of lifecycle objectives. Such *dependencies* can be utilised to explore modularity from the point of view of its capabilities with respect to fulfilling product lifecycle objectives.

The design lifecycle, or specific objectives relating to a phase of this lifecycle, are the focus of a number of modular design approaches (Ishii, Lee et al. 1995; Erixon 1997; Gershenson and Prasad 1997; Gershenson and Prasad 1997; Gu, Hashemian et al. 1997; Sosale, Hashemiam et al. 1997; Gershenson, Prasad et al. 1999; Jiao and Tseng 1999). A review of the design viewpoint support in MD (Section 4.1.2) has highlighted that different viewpoints are favoured for the support or exploration of different lifecycle objectives during the design life phase (Figure 4.1).

MD methodologies by Gershenson et al. (Gershenson and Prasad 1997; Gershenson and Prasad 1997); and Kamrani (Kamrani 1997) explore objectives related to manufacturing and assembly and their impact on MD whilst methodologies by Ishii et al (Ishii, Lee et al. 1995) and Sosale et al (Sosale, Hashemiam et al. 1997) focus on re-use, recycling and retirement objectives. These methodologies focus on the 'structural' viewpoint and require the decomposition of an existing product (Gershenson and Prasad 1997; Gershenson and Prasad 1997; Kamrani 1997; Sosale, Hashemiam et al. 1997) or a product definition which has evolved to the component or assembly level (Ishii, Lee et al. 1995). Thus, out-with the boundary of that particular viewpoint the designer has no formal mechanisms to further explore the implications of the chosen objectives on product knowledge from other viewpoints.

Later work by Gershenson et al (Gershenson, Prasad et al. 1999) and Gu et al (Gu, Hashemian et al. 1997), and work by Erixon (Erixon 1997), embody a broader lifecycle approach. For example, Erixon (Erixon 1997) analyses the technical solutions of existing design with respect to a number of module drivers. Module drivers are 'important reasons' for the implementation of design changes (Erixon 1997), in terms of, amongst others, design, variety, management and quality. Thus, module drivers represent the explications of a series of *lifecycle objectives* for the product such as carry over, styling, service and recycling. The approach utilises MD as a means to plan for design changes i.e. grouping solutions based on their relation to specified objectives to restrict future design changes to a select number of modules. Both Gershensen et al and Gu et al present similar approaches. Again, these methodologies require the functional and/or structural decomposition of a *completed design model* and thus support redesign and incorporation of modularity into existing products (or product families).

Different *viewpoints* have been shown to be capable of supporting analysis of *objectives* related to different product life phases during the design life-phase (see section 4.1.2, Figure 4.1). The reviewed methodologies, were found to support lifecycle objectives, however, all focus on a particular viewpoint with no mapping to additional viewpoints. As such the approaches cannot support an analysis of the trade-offs that exist between design viewpoints with respect to their associated lifecycle objectives.

It is clear that dependencies play a significant role in the exploration and determination of the MD. However, current MD methodologies fail to adequately support these both within individual *viewpoints* and across *multiple viewpoints* of the design activity. The methodology proposed within this work aims to provide a more adequate representation of dependency knowledge to support its utilisation to facilitate *knowledge modularisation* throughout the design activity.

4.1.5 Module identification

The main characteristics that determine modularity have been defined as the degree of dependency (interaction) between components (in the case of this work concepts) both: within a module and of different modules (Ulrich and Tung 1991; Kusiak and Huang 1996).

The criteria for the optimal modular product structure can thus be defined here as the clustering of concepts such that the degree of dependency and interaction is:

- maximised internally within groups (modules), and
- minimised externally between groups (modules).

The challenge for modular design research is to identify this optimal modular structure. Firstly, there is the requirement for the adequate modelling of concept and dependency knowledge to support its analysis. Secondly, the process of Module Identification (MI) requires some form of analysis of the design concepts and their associated dependencies to determine the modules or modularity that exists in the design. The MI process requires that optimal groups (or clusters) of concepts are determined based on some applicable criteria. Finally, the modules must be identified and communicated to the designer by some appropriate means.

Modelling

Concept and *dependency* knowledge modelling is discussed in the previous section (4.1.4). The modelling requirements have been shown to be generally fulfilled using either interaction graphs or matrix techniques.

Analysis

The analysis (grouping or clustering process) can be achieved manually (Erixon 1996; Erixon 1997; Gershenson and Prasad 1997; Gershenson and Prasad 1997; Erixon 1998), through the application of some form of heuristic (Kusiak and Huang 1996) or algorithm (Sosale, Hashemiam et al. 1997; Huang and Kusiak 1998; Jiao and Tseng 1999; Salheih and Kamrani 1999), or through a combination of both (Zamirowski and Otto 1999; Jarventausta and Pulkkinen 2001).

In work by Erixon (Erixon 1997) a Module Indication Matrix (MIM) represents the assessment of 'technical solutions' against a set of 'module drivers'. A 'score' is given for each technical solution, which represents the sum of all their relations to the 'module drivers', indicative of the appropriateness for module inclusion. The 'user' can manually peruse the matrix and can 'pick out' and 'mark' a technical solution for inclusion in the module. Gershenson et al (Gershenson and Prasad 1997; Gershenson and Prasad 1997) also carry out the MI process manually, for example, they peruse a product and process decomposition to identify related partitioning in each. Based on their proposed definition of modules significant partitions represent 'modules'. A manual approach to MI is subject to human error and as such it is not possible to guarantee that an optimal modular solution has been identified. The case studies described in both works have a limited number of concepts and associated dependencies. However, as the capability of a humans' short term memory is limited to seven elements (plus or minus two) (Miller 1965), this limits our capacity for processing information, which may consequently limit the effectiveness of manual MI in more complex cases. As such, alternative methods to support the MI process have been sought.

The use of heuristics and/or algorithms to support the MI process is proposed in a number of works. For example, work by Zamirowski and Otto (Zamirowski and Otto 1999) utilises function and variety heuristics to examine the arrangement of function and flows to identify possible modules. However, this process of partitioning by heuristics is carried out manually and the authors report that they are 'often impractical to implement, since they interact with so many other flows'. Jarventausta et al. (Jarventausta and Pulkkinen 2001) overcome some of the issues associated with manual MI by utilising an algorithm (Cluthill-McGee algorithm) to cluster the concept dependencies in a relational matrix close to the diagonal and then manually apply a 'degree clustering' method to identify modules. They state that 'clustering should be computerised'. Degree clustering group's components inside others, based on the relations that exist between them, such that the outcome is one item with a number of modules clustered inside it. The results are hierarchical so that a 'PDM system can handle it' (Jarventausta and Pulkkinen 2001). The authors report that the results are 'raw' and provided cases of 'false' clustering. In addition the MI process, which uses the applications Microsoft Excel and Matlab, was time consuming, taking 'two to three days' and '45 clustering rounds'. However, the utilisation of an algorithm was shown to ease clustering by arranging relations 'nearby each other' (Jarventausta and Pulkkinen 2001). Huang et al (Huang and Kusiak 1998), Kusiak et al (Kusiak and Huang 1996), Salheih et al (Salheih and Kamrani 1999) utilise heuristics and algorithms though it is not clear from their published work whether the process is computationally supported.

Sosale et al (Gu, Hashemian et al. 1997; Sosale, Hashemiam et al. 1997) and Jiao and Tseng (Tseng and Jiao 1997; Jiao and Tseng 1999) both utilise computational support during the MI process. For example, Sosale et al utilise an objective function, which is modelled as the 'maximum interaction score within the modules', to determine a value for the interaction between components in the design. The simulated annealing algorithm is applied to generate varying module configurations. For each newly generated module configuration, the objective function is determined and if its value is smaller than that of the previous configuration it is selected as the new configuration.

Jiao and Tseng (Tseng and Jiao 1997; Jiao and Tseng 1999) also claim to computationally support the MI process. However, their case-based approach the MI process is concerned with the identification of pre-existing modules that map to current customer needs and not the identification of the potential concept groupings that constitute these modules as defined here.

As can be seen from the previous examples, there are a number of heuristics and algorithms that exist which may be applied to the problem of module identification. Whilst many of the heuristics and algorithms are based on the manipulation of the matrix by a set of predefined rules, simulated annealing (Kirkpatrick, Gelatt et al. 1983) as utilised the approach by Sosale et al (Gu, Hashemian et al. 1997; Sosale, Hashemiam et al. 1997) represents a form of optimisation. Other forms of optimisation techniques include, amongst others, Genetic Algorithms (GA) (Holland 1962) and Tabu search (Glover and Laguna 1993). The algorithms tend to have a number of parameters that affect their performance, for example, the annealing schedule for simulated annealing and research by Whitfield et al. (Whitfield, Duffy et al. 2001) has demonstrated that these parameters are intrinsically linked to the domain.

Identification and communication

Despite the availability of varying methods to both model and optimise concept and dependency knowledge, there are disparities with respect to the methods for the *identification* and *communication* of modules within these optimised models. For example, in Jarventausta and Pulkkinen (Jarventausta and Pulkkinen 2001), and Salhieh and Kamrani (Salheih and Kamrani 1999) modules are identified manually by perusing an optimised matrix. In Sosale et al. (Sosale, Hashemiam et al. 1997) the results of the optimisation are presented in a list of components each with a value indicating the module number to which the component is assigned. However, it is not clear from their published work (Gu, Hashemian et al. 1997; Sosale, Hashemiam et al. 1997) how these modules are extracted from the optimised matrix. The rationale given for their module groupings would suggest that it is subject to manual interpretation of the matrix.

Due to the complex nature of the module identification problem it may not always be appropriate, or possible, to manually identify modular concept groupings within graphs or matrices, as in Jarventausta and Pulkkinen, (Jarventausta and Pulkkinen 2001). This is especially pertinent as the design space becomes more highly constrained i.e. when there are a large number of inter-dependencies between components. In such cases, the clustered matrix/graph may be densely populated and on first perusal yield no significant modules. Therefore, an alternative analysis of the optimised model would be required to facilitate module identification. Such analysis would conceivably require the utilisation of further resources such as domain specific knowledge. This would result in additional time and effort on the part of the human designer and/or the development of computational support systems. Further, when modules are readily identifiable, whether manually or automatically, it may not always be appropriate to return a list of definitive modules as suggested by Sosale et al. (Sosale, Hashemiam et al. 1997). For example, what is modular from one perspective (e.g. assembly) may not be modular from another (e.g. maintenance) (Miller and Elgard 1998). Thus, we see that the modular design problem is further complicated by the fact that differing modular configurations support different perspectives of the problem. In addition, the modular design may also exist over different hierarchical levels of the product structure. Thus, the inherent hierarchical modularity is not exposed when the outcome of the identification phase is presented as a list (Sosale, Hashemiam et al. 1997) or as definitive module boundaries (Salheih and Kamrani 1999).

Given the above, the following must be determined in order to develop a means to facilitate improved module *identification* and *communication*:

- Inherent modularity.
- Potentially differing modular configurations.
- Differing hierarchical levels of modularity in the product structure.

The novel methodology proposed in this work aims to address the above MI requirements. The proposed solution aims to combine the strengths of both the designer, in terms of their domain and problem specific knowledge, and of computer support, in terms of its advanced capabilities for rapid and precise analysis and calculation. It is intended to support the knowledge generated by the designer throughout the design activity in terms of design concepts from differing viewpoints and represent these for analysis. The analysis will utilise the strengths of computational support to analyse the modularity of this generated knowledge to both structure it to enhance its knowledge content and maintain a modular solution over design viewpoints.

4.1.6 Inherent modularity and supporting trade –off decisions

There is a requirement (see Section 4.1.5) to identify and communicate to the designer the inherent modularity of a design with respect to the varying strategic *life-cycle objectives* required of the design, i.e. design, manufacture, use, upgrade, disposal. An understanding of the capabilities of differing module configurations to support these objectives, either individually or collectively, supports the designer in making trade-off decisions between these. It is suggested that a requirement of an MD methodology that supports 'a life-cycle view' of design is the identification and communication of optimal module configurations for the differing *perspectives* that can be taken during the product *life-cycle* and across any combination of these.

Current MD methodologies which present the results of the modular analysis as a list (Sosale, Hashemiam et al. 1997) or as definitive module boundaries (Salheih and Kamrani 1999) has been shown to be insufficient at supporting this requirement. Work by Gershenson et al (Gershenson, Prasad et al. 1999) constituted the only reviewed case where the inherent (termed relative) modularity of a product was addressed. However, the work by Gershenson et al identifies the relative modularity of existing products and their assemblies to facilitate redesign. The methodology considers the modular product to be a well-designed product. The product is structurally decomposed into its constituent assemblies, termed 'modules' and the analysis is based these. A low modularity value, either at the assembly or product level, indicates a candidate for design improvement. A methodology based on elimination and reconfiguration is then applied to the assemblies and the 'redesigned product' is re-evaluated to determine its new relative modularity value. The methodology is applied at the assembly, as opposed to the component level, and as such is applied at a relatively low level of granularity. Thus, the products modularity represents a retrospective indicator of the 'goodness' of the *completed design models* assemblies and the work presented in this thesis focuses on the application of MD principles to design concepts generated in the *evolving design model* to support design knowledge for *re-use.*

4.2 Summary of existing approaches

The need for research in the above areas of MD (Sections 4.1.4 to 4.1.6) has been noted by a number of practitioners including Miller (Miller and Elgard 1998), Gu (Gu, Hashemian et al. 1997), Jiao (Jiao and Tseng 1999) and Ishii (Ishii, Lee et al. 1995). The benefits of modular design suggest a potential to support the process of *Design for Re-use*, However problems lie in our current understanding of how to utilise the principle of MD to support design knowledge, to meet the needs of the designer in identifying and representing the inherent *knowledge modularity* to support *re-use.*

Table 4.2 provides an overview of the support afforded by existing MD methodologies with respect to the requirements of an MD methodology to provide improved EDR support. It can be seen from that existing methodologies already provide some support. However, the critical review has clarified that, of the existing MD methodologies, none has satisfied all the requirements, due to the following limitations.

- Insufficient support of the design knowledge

  None of the methodologies were able to adequately capture, model and represent design knowledge. In particular, no current methodology:

  - Adequately supports the evolution of design knowledge. *Viewpoints* represent a structured view of engineering design and are generated by the designer to evolve the engineering design activity. Mapping between these *viewpoints* has been shown to evolve design knowledge from the abstract to the concrete (Figure 2.4). Section 4.1.3 discussed the inadequacies of current methodologies with respect to supporting the multiple *viewpoints* of the design activity.
  - Supports the consistent modelling and representation of design *concepts* from the varying *viewpoints* as they evolved from the abstract to the concrete.
  - Allows the designer to represent and consequently utilise knowledge related to the differing *dependencies,* which the design *concepts* are subject to both within and across *viewpoints,* throughout the evolution of the design.

- Insufficient support of the *Knowledge Modularity*

  No methodology was shown to support the exploration of modularity of the knowledge throughout the evolution of the design activity. In particular, there is:

  - Lack of support for the exploration and generation *knowledge modules* with particular insufficiencies with respect to the abstract *concepts* from early design.
  - Significant insufficiencies in both the modelling and representation of the evolution of *knowledge modules,* from the abstract to the concrete *concepts,* throughout the design activity.

- Significant limitations in the MI process

  The MI process, required to analyse *concept* and *dependency* knowledge to facilitate the modularisation of the design and its associated knowledge, has significant limitations. In particular:

  - The methodologies as a whole exhibit limited computational support for the analysis (grouping or clustering) and optimisation phases of the MI process.
  - The identification of modules in optimised models is limited to predominantly manual interpretation. The process requires a more formalised approach.
  - Inherent or relative modularity, which exists with respect to the differing *dependency* types, is not explicitly identified or represented in existing approaches.

Evaluation Criteria			Reviewed Methodologies: (Erixon 1998)	(Gershenson and Prasad 1997) (Gershenson and Prasad 1997)	(Gershenson, Prasad et al. 1999)	(Gu, Hashemian et al. 1997) (Sosale, Hashemiam et al. 1997)	(Huang and Kusiak 1998) (Kusiak and Huang 1996)
Viewpoint Support	Single	Function	No	No	No	No	Yes
		Behaviour	Yes	No	No	No	No
		Structure	No	Yes	Yes	Yes	No
	Multiple		No	No	No	No	No
Design Knowledge Representation	Representation		Matrix	Tree	Tree represents decomposition of product into assemblies **Matrix** support modular analysis	Matrix	Matrix
	Concepts		Technical Solutions – columns in matrices	Components	Components	Components	Product Functions
	Dependencies	Functional, Spatial etc.	N/A	Represents only the 'part–of' relation – no weight or direction characteristics are represented	Tree structure represents only the 'part–of' relation – no weight or direction characteristics are represented. Matrix represents component and process dependencies and similarities	Function, spatial and geometrical –dependencies have weighting but no direction	Recognise that geometric, temporal, force, electrical, thermal and photometric dependencies exist – no explicit representation of these provided – dependencies have direction and weight characteristics
		Lifecycle Objectives	Module Drivers – rows in matrices. Entries in matrix represent relation of 'technical solution' to 'drivers'	Manufacturing, Service and Maintenance objectives – no explicit definition of related dependency knowledge	Declares the method as a 'Lifecycle View' but only considers later lifecycle objectives from manufacture onwards.	Re-use, recycling and maintenance – dependencies related to these factors are represented in a single perspective of the dependencies.	N/A
Design Knowledge Evolution			No	No	No	No	No
Modular Analysis			Manual Selection	Manual Selection	Manual Selection	Objective Function Simulated Annealing	Decomposition Algorithm
Module Identification / Communication			Manual Perusal	Manual Perusal	Manual Perusal	List Structure – possibly based on manual interpretation of optimised matrix	Manual Perusal
Inherent / Relative Modularity			No	No	Yes, but as a retrospective indicator of the 'goodness' of the design	No	No

Table 4.2: Overview of existing MD methodologies

Evaluation Criteria			Reviewed Methodologies			
			(Ishii, Lee et al. 1995)	(Jarventausta and Pulkkinen 2001)	(Jiao and Tseng 1999) (Tseng and Jiao 1997)	(Salheih and Kamrani 1999)
Viewpoint Support	**Single**	**Function**	No	N/A	Yes	N/A
		Behaviour	No	N/A	Yes	N/A
		Structure	Yes	N/A	Yes	N/A
	Multiple		No	Yes – engineering design activity represented as a single view – *views* equate more readily to the definition of a *perspective* posited in this work	Yes – however the focus is different – the approach plans for modularity based on the PFA of past designs. Applies to redesign and evolutionary design only.	Yes – modularity is only explored and defined late in the process - during the structure viewpoint – others viewpoints treated as data gathering for this step
Design Knowledge Representation	**Representation**		Graph	Matrix	Graph represent 'views' Matrices map between these 'views'	Graph represent decomposition of design knowledge Matrix supports modular analysis
	Design Activity Concepts		Components – nodes	Dependant on the 'view' chosen	FRs and DPs – mapping between these constitutes the design activity	Components and SLS
	Dependencies	**Functional, spatial etc.**	N/A	'part–of' relation – no importance or direction characteristics	'part-of' and 'kind-of' relations represented by graph structure	
		Lifecycle Objectives	Represents dependencies between components based on comparisons of their technology life-cycle	N/A	'what–how' relations (known as causal relations) represented in matrix structure	
Design Knowledge Evolution			No	No	No	No
Modular Analysis			Suggests the use of algorithms to 'clump' components	Cluthill McGee Algorithm Manual 'degree' clustering	Kusiak and Chow Algorithm (Kusiak and Chow 1987)	P Median Optimisation Model
Module Identification / Communication			N/A	Manual perusal of clustered matrix	Tabular Form	Manual Perusal of an optimised matrix
Inherent Modularity			No	No	No	No

Table 4.2: Overview of existing MD methodologies cont...

The deficiencies of existing approaches provide some validation for the need to determine both novel declarative knowledge, which defines the elements that constitute a holistic MD methodology for re-use, and novel procedural knowledge, which determines its application to the design activity. In particular, the identification of weaknesses in existing methodologies has highlighted the following aspects required for further research:

- Enhancement of design knowledge support

  To provide a more comprehensive model of design knowledge to facilitate the subsequent exploration of the existing *knowledge modularity* through:

  - The modelling and representation of a more comprehensive set of design *viewpoints*.
  - The application of a consistent approach to the modelling and representation of design *concepts* from the abstract to the concrete.
  - The application of a more comprehensive and consistent approach to the modelling and representation of *dependency* knowledge, both within and across the generated design *viewpoints*.

- Improvement in support for *Knowledge Modularisation (KMn)*

  To provide the appropriate mechanisms required to support *KMn* based on the provision of an enhanced model of design knowledge, including:

  - The development of mechanisms to explore and generate *knowledge modules* from the early stages of the design activity.
  - The development of mechanisms to model and represent the evolution of *knowledge modules* throughout the activity of design.

- Improvement in the MI process

  To provide computational support for the analysis and optimisation phase of the MI process, including:

  - The development of a formal mechanism to identify and communicate the modules that exist in optimised models.
  - The development of a novel means to explore, identify and communicate the inherent modularity. For example,
    - Identifying and communicating the differing potential modular configurations that exist in the optimised model.
    - Identifying and communicating modularity with respect to a model optimised for, any individual or required set of, identified lifecycle objectives.

To address on the deficiencies identified in current MD methodologies and based on the aspects of further research highlighted above, the author has derived three hypotheses, as follows:

1. *Evolutionary design support with modular clustering can support design knowledge re-use*
2. *The application of a genetic algorithm will support the exploration and optimisation of* knowledge modules *in the product structure'*
3. *Formalising the definition of a* module *can support the identification of inherent modularity in the product structure'*

The hypothesis are embodied within a novel MD methodology, presented in Part II, and tested and evaluated through their implementation in a number of case studies in Part III of this work.

4.3 Chapter summary

MD methodologies have been critically reviewed based on the requirements for a MD methodology to support improved EDR. The resulting findings are summarised in Table 4.2 and Section 4.2. The review has highlighted the inadequacies of existing approaches and identified areas for further research and enhancement based upon these.

The work presented in Part II of this thesis focuses on the development of a novel MD methodology through the definition of the required declarative and procedural knowledge.

Part 2 – Solution Definition and Embodiment

5 A Multi-Viewpoint Modular Design Methodology

A novel 'Multi-Viewpoint Modular Design methodology' has been developed in the work presented in this thesis that aims to satisfy the requirements of an MD methodology for improved EDR and address the previously outlined inadequacies of existing MD methodologies. An overview of this new MD methodology is given in Section 5.1. The methodology has four main elements, i.e. the shaded components of Figure 5.1. These four elements are outlined in the following sections: Section 5.2 presents the *Modelling Formalism*, Section 5.3 presents the *Optimisation Mechanism*, Section 5.4 the *Modular Identification Mechanism* and Section 5.5 the methodologies *Mapping Mechanism.* A proposed application process is outlined in Section 5.6. Details of the techniques employed to embody each component are provided in Chapter 6. Section 5.7 summarises the chapter.

5.1 Overview of the Methodology

The methodology has been developed to support the exploration and maintenance of a modular design by modelling *viewpoints* and *perspectives* of design knowledge, optimising these models, identifying their inherent hierarchical modular structure and then *mapping*[14] between these viewpoints. The Multi-Viewpoint MD methodology has four main elements; a *Modelling Formalism,* an *Optimisation Mechanism,* a *Module Identification Mechanism* and a *Mapping Mechanism.* Figure 5.1 presents an overview of the Multi-Viewpoint Modular Design Methodology.

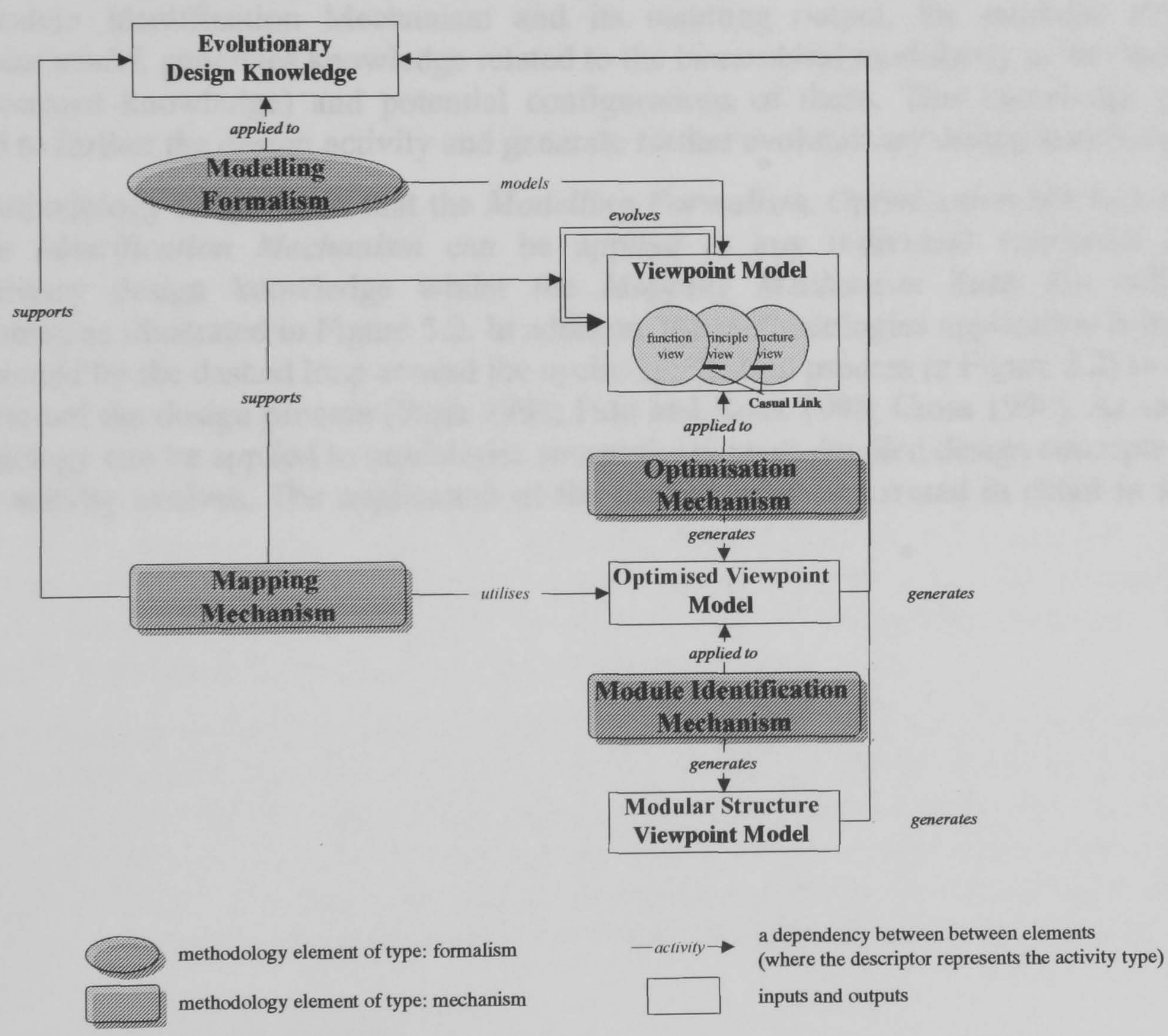


Figure 5.1: An overview of the Multi-Viewpoint Modular Design Methodology

[14] Mapping represents the activity whereby a more abstract concept (for example, a *function concept*) is defined as being realised by a more concrete design *concept* (for example, a *working principle concept*).

As can be seen from Figure 5.1, a *Modelling Formalism* is applied to the evolutionary design knowledge generated during the design activity. The output from the formalism is a *viewpoint model*. The Modelling Formalism can be utilised to model different viewpoints of the knowledge generated and the relations between these viewpoints (depicted as shaded circles within the viewpoint model[15] output in Figure 5.1).

The *Optimisation Mechanism* can be applied to each individual generated *viewpoint model* to create an *optimised viewpoint model*. The application of the Optimisation Mechanism and the resulting optimised viewpoint model generate further design knowledge[16]. This design knowledge can be utilised to further the design activity and consequently evolves its associated knowledge. In addition, the optimised viewpoint model is utilised as an input to the *Mapping Mechanism*.

The *Mapping Mechanism* supports the evolution of the viewpoint models by mapping between individual viewpoints, i.e. by defining the causal relations between design concepts in different viewpoints (see footnote 15). Thus, the mechanism is applied across and not within individual viewpoints as with the other elements of the proposed methodology. This mapping process maintains the modular solution as the viewpoints become successively more concrete, i.e. as they move from representing abstract concepts such functions to more concrete concepts such as parts. As such, the Mapping Mechanism supports the generation of evolutionary design knowledge.

The *Module Identification Mechanism* is applied to the optimised viewpoint model to support the interpretation of inherent modularity in the product structure. The application of the Module Identification Mechanism and its resulting output, the *modular structure viewpoint model,* generates knowledge related to the hierarchical modularity in the design (or its associated knowledge) and potential configurations of these. This knowledge can be utilised to further the design activity and generate further evolutionary design knowledge.

The methodology is cyclic, in that the *Modelling Formalism*, *Optimisation Mechanism* and *Module Identification Mechanism* can be applied to any individual viewpoint of the evolutionary design knowledge whilst the *Mapping Mechanism* links the individual viewpoints, as illustrated in Figure 5.2. In addition, the methodologies application is iterative (represented by the dashed loop around the cyclic application process in Figure 5.2) to reflect the nature of the design process (Pugh 1991; Pahl and Beitz 1996; Cross 1998). As such the methodology can be applied to modularise successively more detailed design concepts as the design activity evolves. The application of the methodology is covered in detail in section 5.6.

[15] The relations between *viewpoints* are depicted by the dark line (here, termed the *causal link*) between the different *viewpoints*. The causal-link relation is defined within Chapter 6.
[16] The design knowledge is related to optimum clusters of *concepts* and is discussed in greater depth in 5.3 and in Chapter 6.

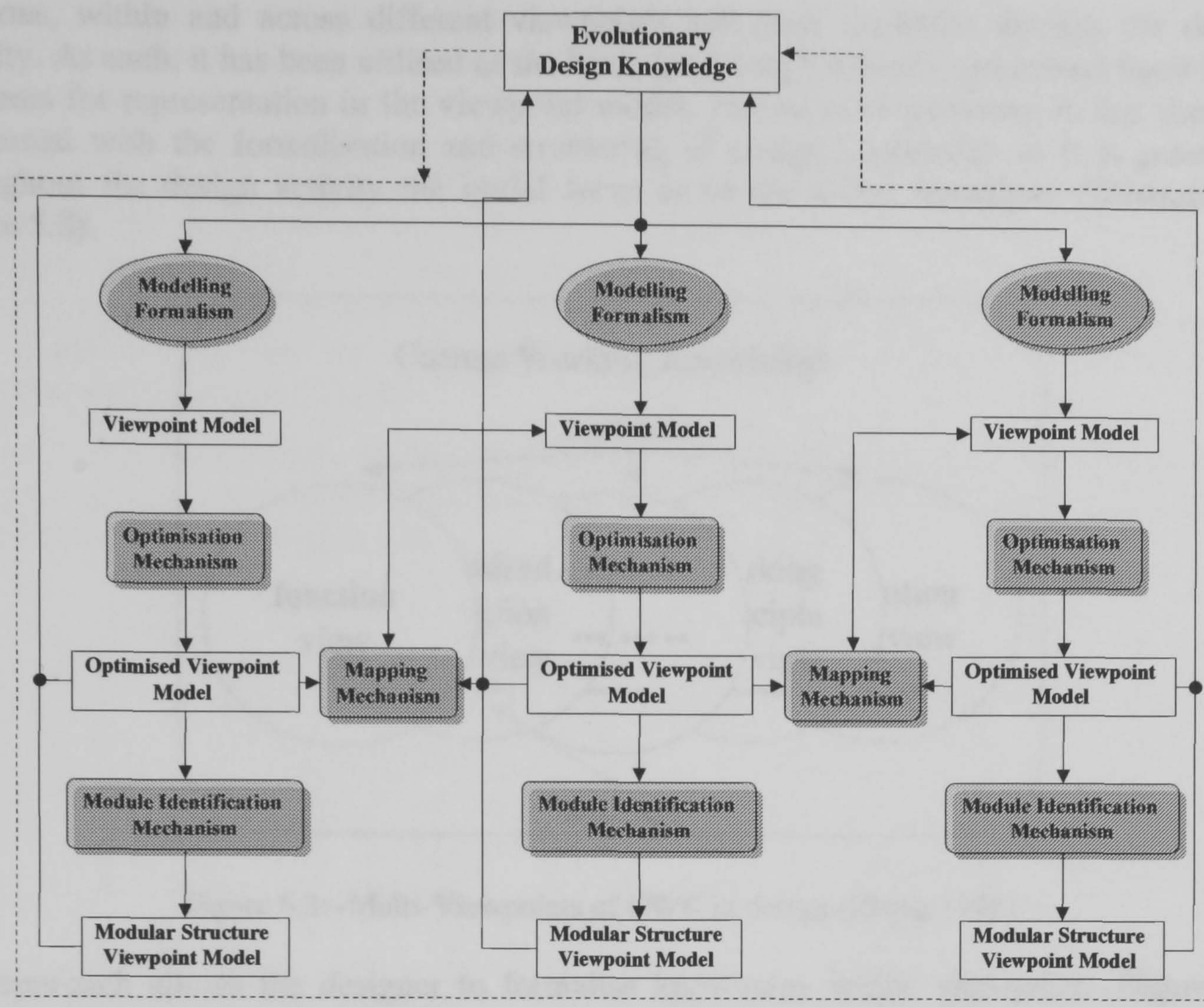


Figure 5.2: The cyclic and iterative nature of the Multi-Viewpoint MD Methodology

5.2 Modelling Formalism

To be able to computationally support the concept of *Knowledge Modularisation (KMn)* evolutionary design knowledge types and the relations, which can exist between them, require to be defined. In addition, the elements of evolutionary design knowledge that can be utilised to support *KMn* must be identified and extracted from those defined. Thus, evolutionary design knowledge must initially be formalised and then the knowledge elements appropriate for *KMn* extracted from this formalism. The extracted elements require to be represented in some form that supports their future modular analysis (by the Optimisation Mechanism). As discussed previously in Section 4.1.4, the representation generally takes the form of a graph or matrix-based representation. However, in this approach a matrix-based representation is utilised (see Section 5.2.2).

The *knowledge formalism* provides the knowledge upon which to base the *matrix formalism* to define individual *viewpoint models*. The viewpoint models are the basis for a clustering analysis (by the *Optimisation Mechanism*) and a modular analysis (by the *Module Identification Mechanism*). Thus based on the requirements to both define and represent evolutionary design knowledge, to support its modular analysis, the Modelling Formalism consists of two parts: an *evolutionary knowledge formalism* and a *dependency matrix formalism.*

5.2.1 Knowledge Formalism

The formalism of evolutionary design knowledge was the focus of previous work by Zhang (Zhang 1998). Zhang developed a knowledge formalism that takes a Multi-Viewpoint Evolutionary Approach (MVEA) to formalising both current working knowledge (CWK) and domain knowledge (DK). The formalism defines design knowledge elements, and their relations, within and across different viewpoints and their evolution through the design activity. As such, it has been utilised as the basis from which identify and extract knowledge elements for representation in the viewpoint model. As the work presented in this thesis is concerned with the formalisation and structuring of design knowledge as it is generated throughout the design activity the initial focus is on the CWK formalism (illustrated in Figure 5.3).

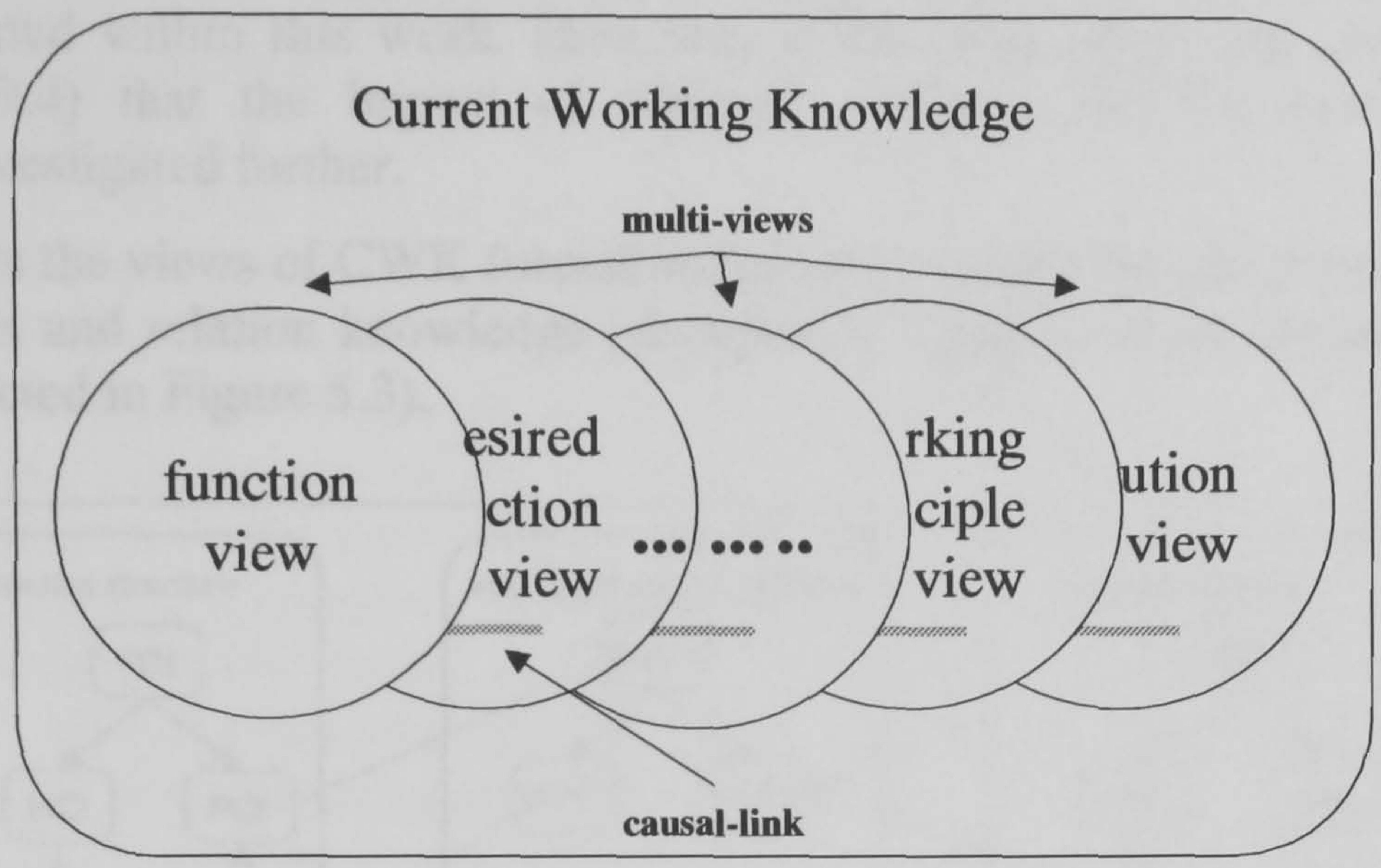


Figure 5.3: -Multi-Viewpoints of CWK in design (Zhang 1998)

The approach allows the designer to formalise knowledge within viewpoints. Figure 5.3 illustrates the multiple views that can be taken of CWK[17]. The figure illustrates that there are number of views with causal-link dependencies between them (see footnote 15).

The main elements of the formalism are illustrated in Figure 5.4. The approach allows the designer to encapsulate knowledge of the 'ideas' of the design (formalised as design *concepts*). Design *concepts*, their *attributes* (input matters, output matters, behaviour properties, principle properties, parts, etc.) and *constraints* (both on the concept and attributes are formalised (as depicted Figure 5.4a) within each individual viewpoint. In addition, the relations that exist between concepts both within and between viewpoints are formalised (Figure 5.4b).

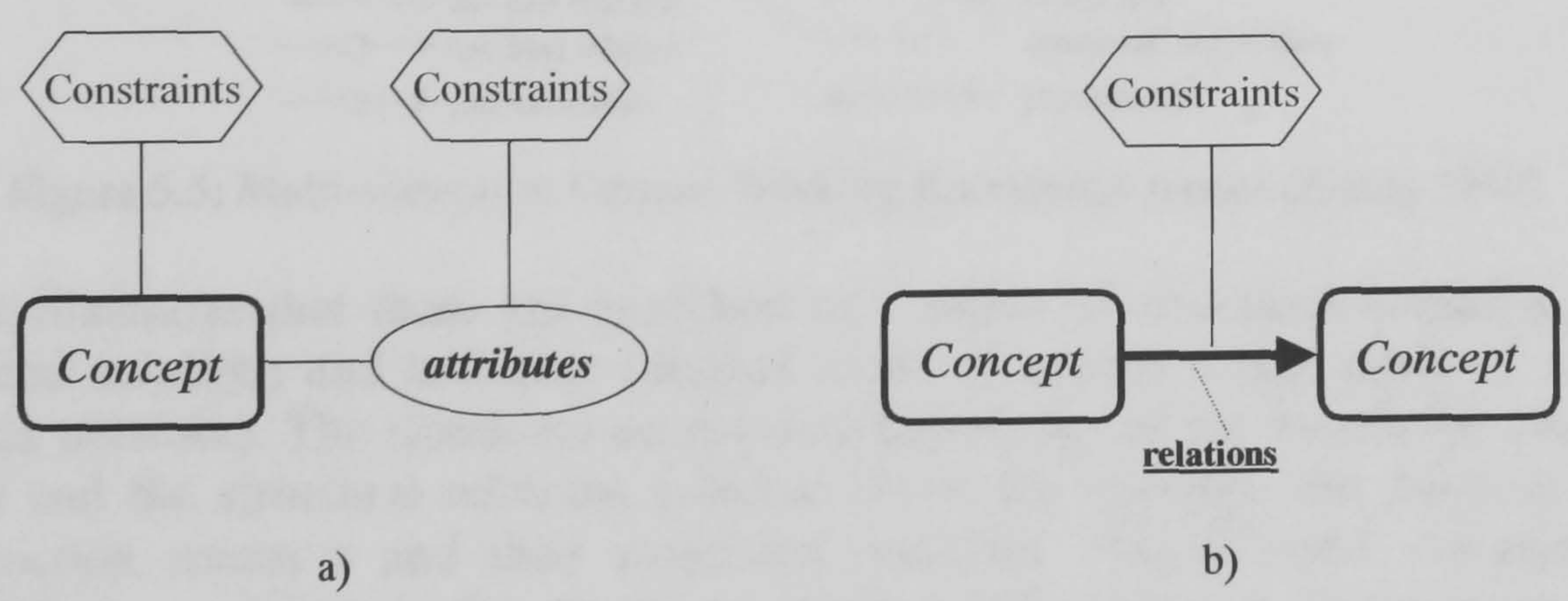


Figure 5.4: Main elements of CWK (Zhang 1998)

Concept constraints indicate application conditions whilst *attribute constraints* represent dependencies between individual attributes. Relations between concepts can be of a structural nature (*has-kind, a-kind-of, a-part-of, has-part*) or associative nature (*functional-*

[17] the views are detailed later in Figure 5.5 and their utilisation in the proposed methodology is defined in Chapter 6.

dependency, physical-link) and can have a direction (see Figure 5.4b). Relations between viewpoints are formalised as a *causal-link* (as illustrated in Figure 5.3 and Figure 5.5). Relations may also have associated constraints. However, the focus of the work presented in this thesis is on defining the declarative and procedural knowledge required to provide a foundation on which to facilitate *KMn*. As such, the primary interest of this work is in the formalism of the design knowledge *concepts* and their *relational/dependency* knowledge and how this can be utilised to support the *KMn* theory. Thus, constraint knowledge is not explicitly supported within this work. However, it has been suggested, as part of further work (Section 9.4) that the impact of concept, attribute and relation constraints on modularity be investigated further.

Figure 5.5 depicts the views of CWK formalised as basic structures and networks illustrating how the concepts and relation knowledge (depicted in Figure 5.4) interrelate within multi-viewpoints (depicted in Figure 5.3).

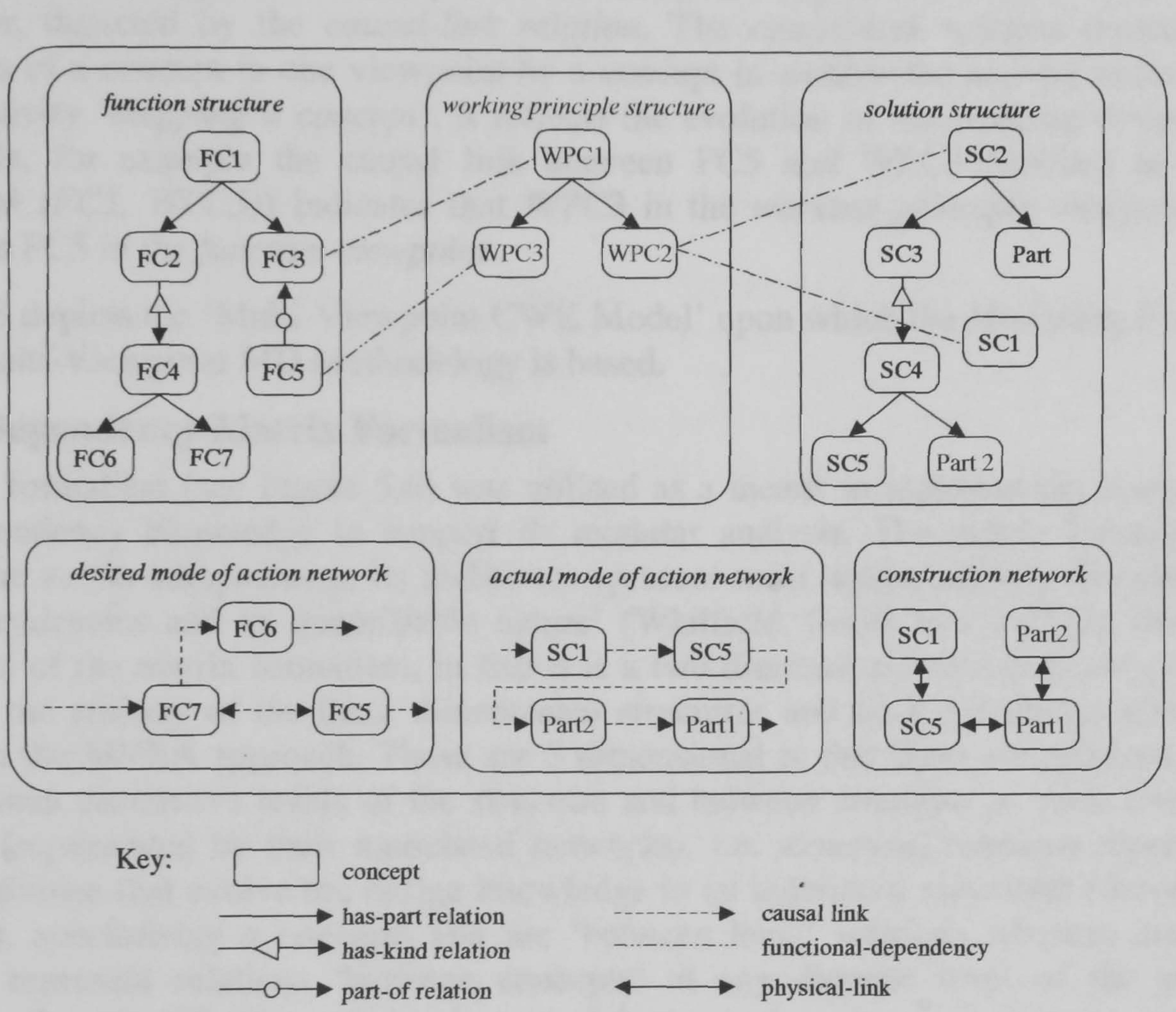


Figure 5.5: Multi-viewpoint Current Working Knowledge model (Zhang 1998)

The figure illustrates that these are modelled as a series of *structures* (function, working principle and solution) and *networks* (desired mode of action, actual mode of action and construction network). The structures encapsulate knowledge of the viewpoint concepts[18] in the design and the structural relations between these; for example, the *function structure* defines *function concepts* and their *structural relations*. The different notations on the relations between concepts in the structure represent different types of structural relations. Structural relations represent the design activities that have been carried out to evolve the CWK, for example, the has-part relation between concepts FC4 and FC6 (defined as relation *has-part(FC4, FC6)*) indicates that FC6 has been generated by decomposing or dividing

[18] The *concept* denotations FC, WPC, SC and Part are defined in Chapter 6. However, for the purposes of explanation at this stage it is sufficient to know that they represent *concept* types formalised for each different *viewpoint*. A *concept* in the *function viewpoint* is denoted as FC, the *working principle viewpoint* as WPC and the *solution viewpoint* either SC or Part.

FC4. Thus, the structures are dynamic, in that they reflect the current status of the designers' knowledge of the design, therefore as the designer(s) understand more about the design, more concrete concept knowledge is generated and the structures evolve.

The *networks* depict viewpoint concepts and their associative relations and represent a specific part of the *structures*, i.e. the concepts from the bottom most level of the structure (at any discrete point in time) and the associative relations between these. *Associative relations* represent the functioning sequence or physical attachment between concepts. Thus networks depict the most detailed and concrete viewpoint concepts and a type of associative relation between them, for example, the *actual mode of action network* in Figure 5.5 represents the most concrete *solution concepts* (SC1, SC5, PART1 and PART2 from the solution structure) and the *functional dependency relations* between these whilst the *construction network* represents the same concepts and the *physical-link relation* between them. As shown in Figure 5.5 concepts in one viewpoint may have a relation with concepts in another, depicted by the *causal-link relation*. The *causal-link relation* represents the realisation of a concept in one viewpoint by a concept in another the activity accomplished by the activity '*mapping a concept*'. It reflects the evolution of the working design across viewpoints, for example the causal link between FC5 and WPC3 (defined as relation *causal-link (FC5, WPC3)*) indicates that WPC3 in the *working principle viewpoint* is the realisation FC5 in the *function viewpoint*.

Figure 5.5 depicts the 'Multi-Viewpoint CWK Model' upon which the *Modelling Formalism* for the Multi-Viewpoint MD Methodology is based.

5.2.2 Dependency Matrix Formalism

A matrix formalism (see Figure 5.6) was utilised as a means to represent the concepts and their dependency knowledge to support its modular analysis. The matrix formalism was chosen due to 'its compactness, its ability to represent most design activity knowledge and their dependencies and its quantifiable nature' (Whitfield, Smith et al. 2002). Due to the limitations of the matrix formalism, in that it is a two dimensional representation, it cannot represent the entirety of the three dimensional structures and their associated networks as defined in the MVEA approach. These are 3 dimensional in that there are relations defined both between successive levels of the structure and between concepts on each level of the structure (represented by their associated networks), i.e. *structural relations* represent the design activities that evolve the design knowledge in an individual viewpoint (*decomposing a concept, specialising a concept*) and are 'between level' relations whereas *associative relations* represent relations 'between concepts' at any discrete level of the structure. However, the identification of *product modularity* is based on *components* and their functional and physical dependency with one another. Thus, it is suggested that a similar utilisation of *design knowledge concepts* and their functional and physical dependency relations can facilitate the identification of *knowledge modularity*. The bottom level of each MVEA structure is representative of such knowledge (as shown in Figure 5.5). In the MVEA approach this is formalised as the *'desired mode of action network'* for the function structure, and both the *'actual mode of action network'* and *'construction network'* for the solution structure (discussed in greater detail in Section 6.1.1).

The formalised knowledge is represented within a dependency matrix formalism as illustrated in Figure 5.6. The representation of the formalised design knowledge concepts and their dependencies within a matrix is termed the *Viewpoint Model.* A viewpoint model is created for the differing viewpoints of interest throughout the design phase i.e. *function, working principle, structure*. Within the viewpoint model the matrix rows and columns represent the same concepts (in the same order) and dependencies as entries in the matrix body (see Figure 4.8).

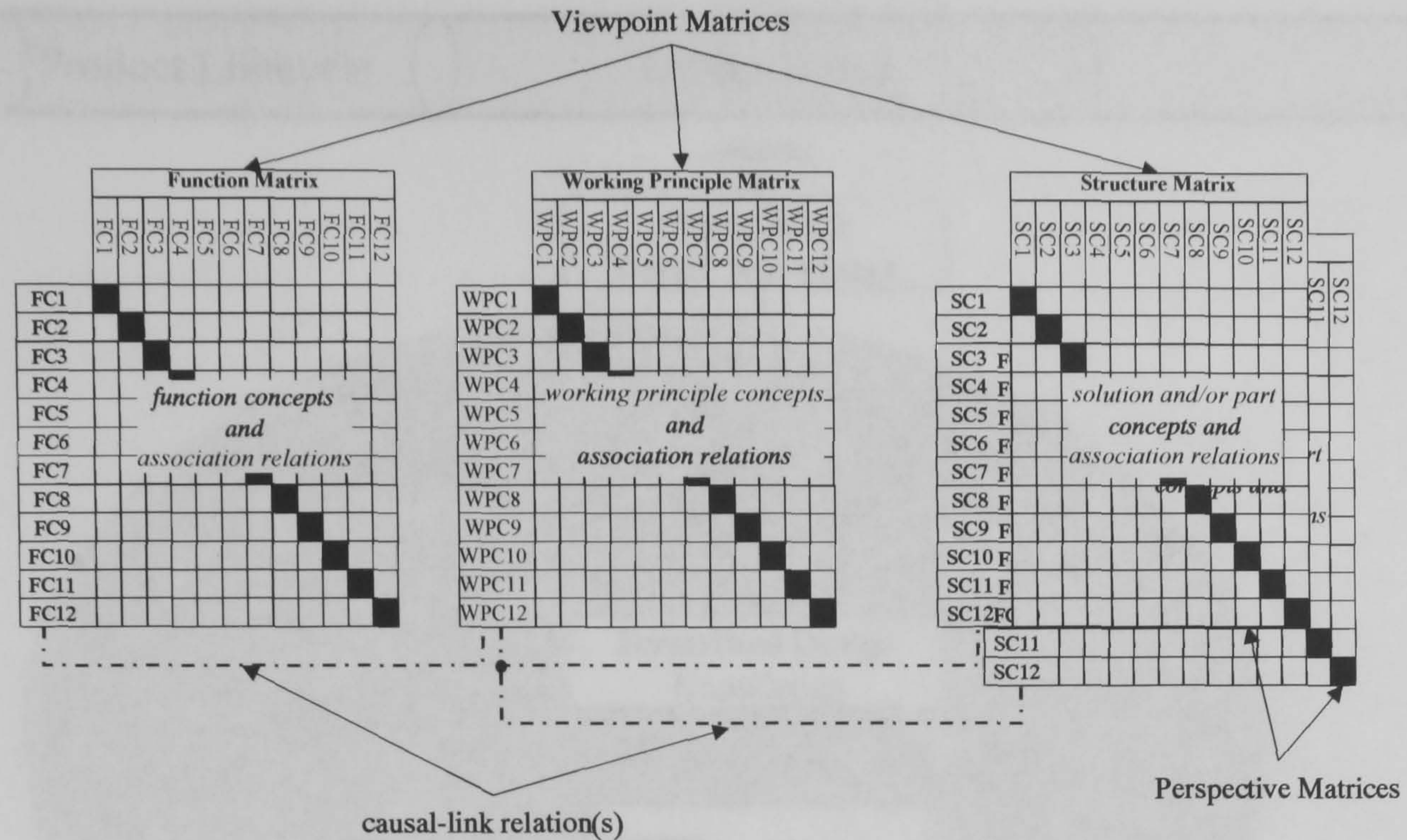


Figure 5.6: The Dependency Matrix Formalism

Figure 5.6 illustrates that the viewpoint 'structure' has two networks associated with it, i.e. the *actual mode of action network* and the *construction network*. These networks are embodied within the structure viewpoint in the proposed Multi-Viewpoint MD Methodology. The networks represent the most detailed and concrete solution concepts (MVEA definition) and the functional dependencies (*actual mode of action network*) and physical-link relations (*construction network*) between these. Thus, each network depicts the concepts and the dependency between them from different *perspectives*, i.e. *functional* and *physical*. A viewpoint model represents design concepts and the associative relations between these from a particular viewpoint, i.e. *function, working principle,* and *structure*. However, as illustrated above (the *actual mode of action network* and *construction network* example) there can be different types of associative relations within each individual viewpoint. Thus, *perspectives* of a *viewpoint model* (as illustrated in Figure 5.6) can be defined to allow the designer to represent the different types of dependency between concepts in a viewpoint. Thus the designer can choose to optimise the modularity based on a particular dependency type (*perspective*) within a viewpoint.

Figure 5.7 provides an overview of the *Modelling Formalism* application within the Multi-Viewpoint MD Methodology proposed.

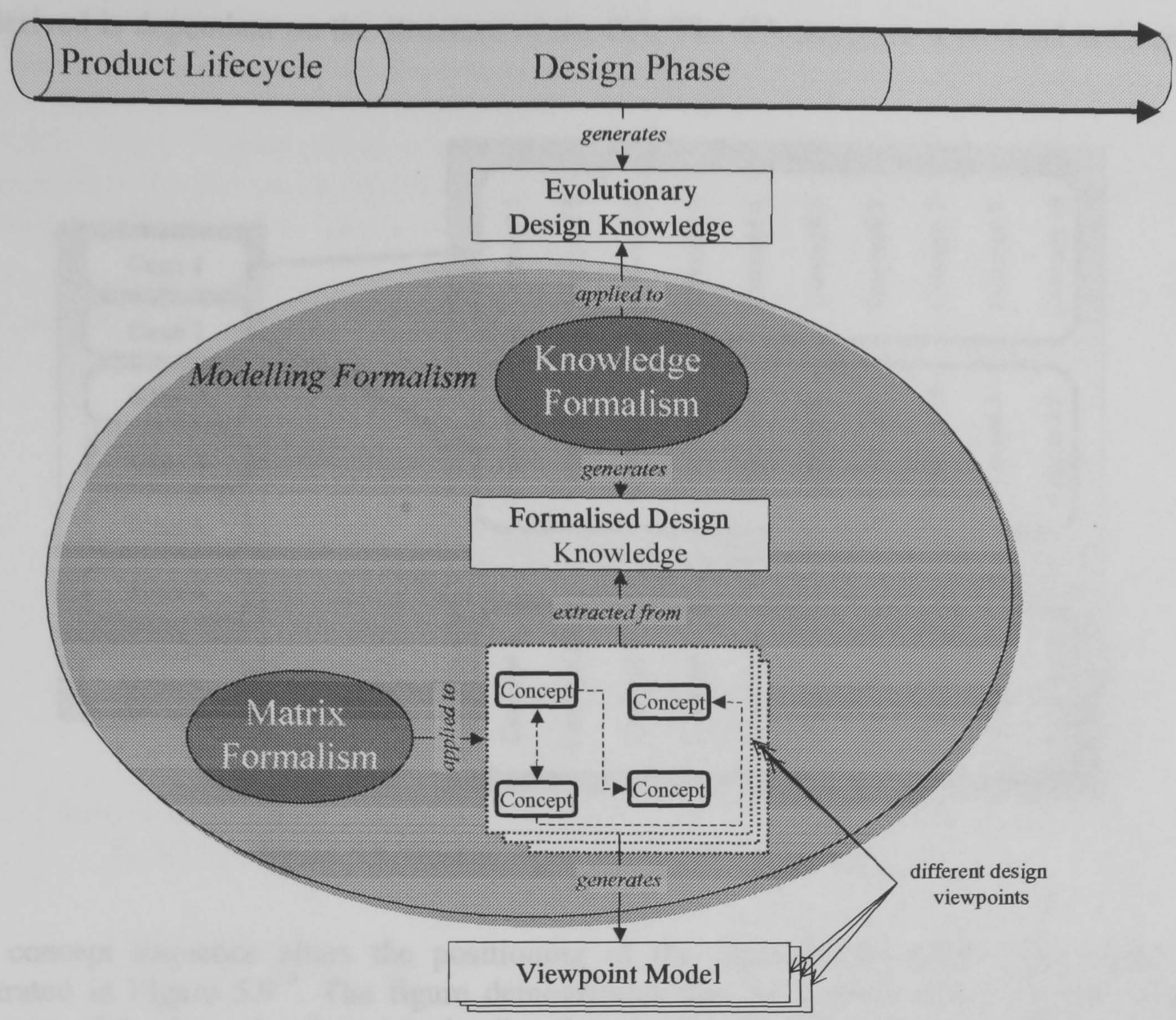


Figure 5.7: Overview of Modelling Formalism

Figure 5.7 depicts the *evolutionary design knowledge* generated during the design phase of the product lifecycle. The MVEA *knowledge formalism* is applied to this evolutionary design knowledge to generated *formalised design knowledge.* The most concrete and detailed concepts and their associative dependency relations of each individual viewpoint are extracted from the formalised design knowledge. A matrix formalism is applied to this extracted knowledge to generate individual viewpoint models. Thus, the output of the Modelling Formalism element of the methodology is a *viewpoint model.*

5.3 The Optimisation Mechanism

The viewpoint model, the output from the methodologies *Modelling Formalism* element, is required to support a modular analysis of the design concepts. The analysis clusters design concepts such that highly dependant concepts are grouped together with the aim of identifying the optimum concept groupings. The application of the *Optimisation Mechanism* aims to facilitate the identification of an optimal or near-optimal modular structure. As such the Optimisation Mechanism requires to facilitate the generation of clustered viewpoint models with potential concept groups (modules), based on the knowledge represented in the viewpoint model, and evaluate these against some criteria which is representative of modularity. Thus, the optimisation model has two parts: a *Genetic Algorithm* (GA) and a *Clustering Criteria* (CC).

The GA generates populations of 'genes' (potential solutions) each representing a variation of the viewpoint model based on the order of the concepts within it (as illustrated in Figure 5.8). The concept order(s) generated by the GA are applied to both column and row abstractions for each gene. The number of generation and the method by which these 'genes' are derived is dependant on the structure of the GA. The GA structure is covered in Chapter 6.

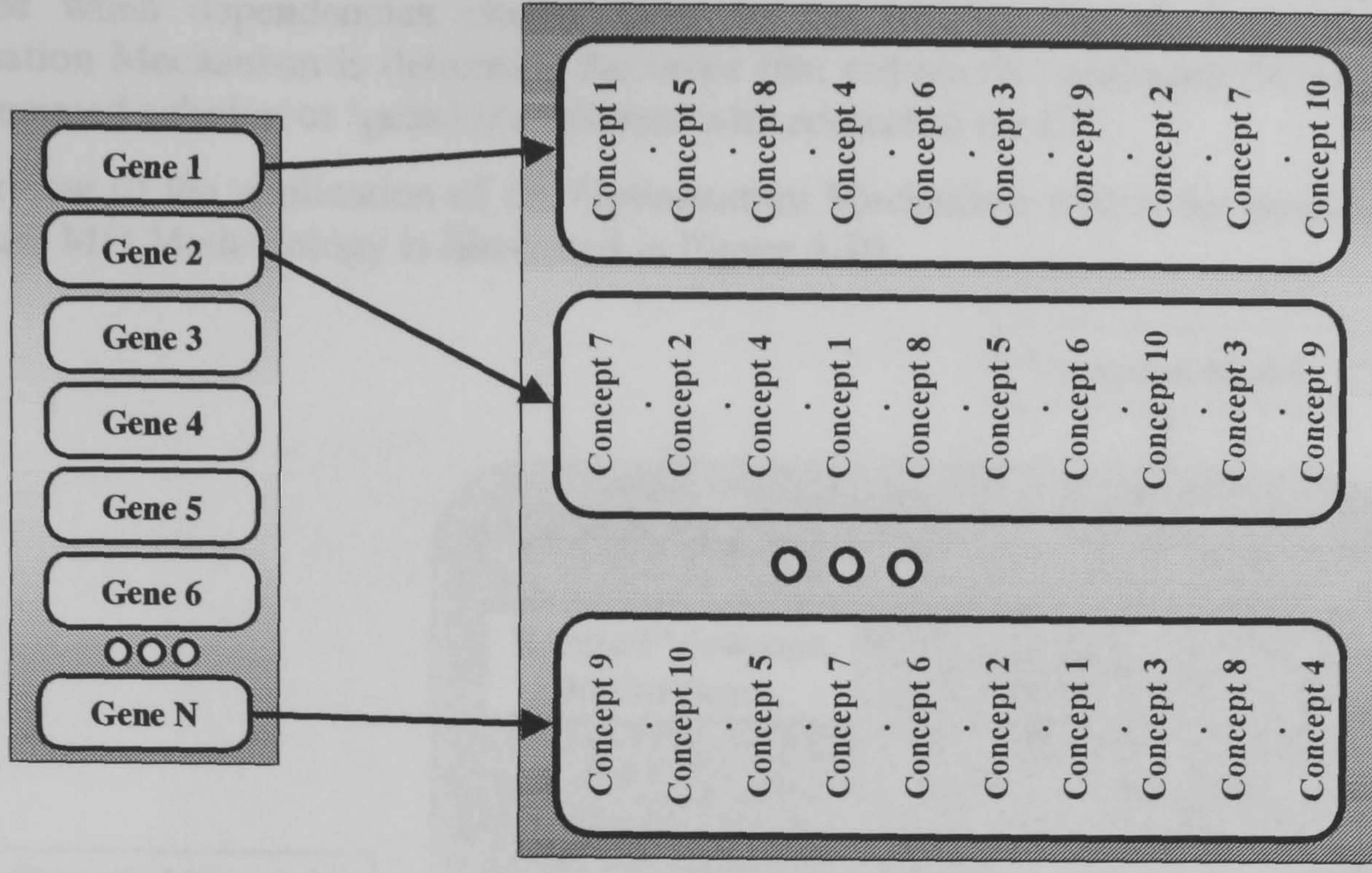


Figure 5.8: Genetic representation of concept order

The concept sequence alters the positioning of the dependencies within the matrix as illustrated in Figure 5.9[19]. The figure demonstrates that the concept order impacts on the closeness of the dependencies to the leading line. As dependencies move towards the leading line it is possible to identify groups of closely related dependencies, as can be seen by the right hand matrix in Figure 5.9. The objective of the Optimisation Mechanism is to identify the concept order that optimises these dependency groups i.e. determine the order that clusters the dependencies most closely around the leading line.

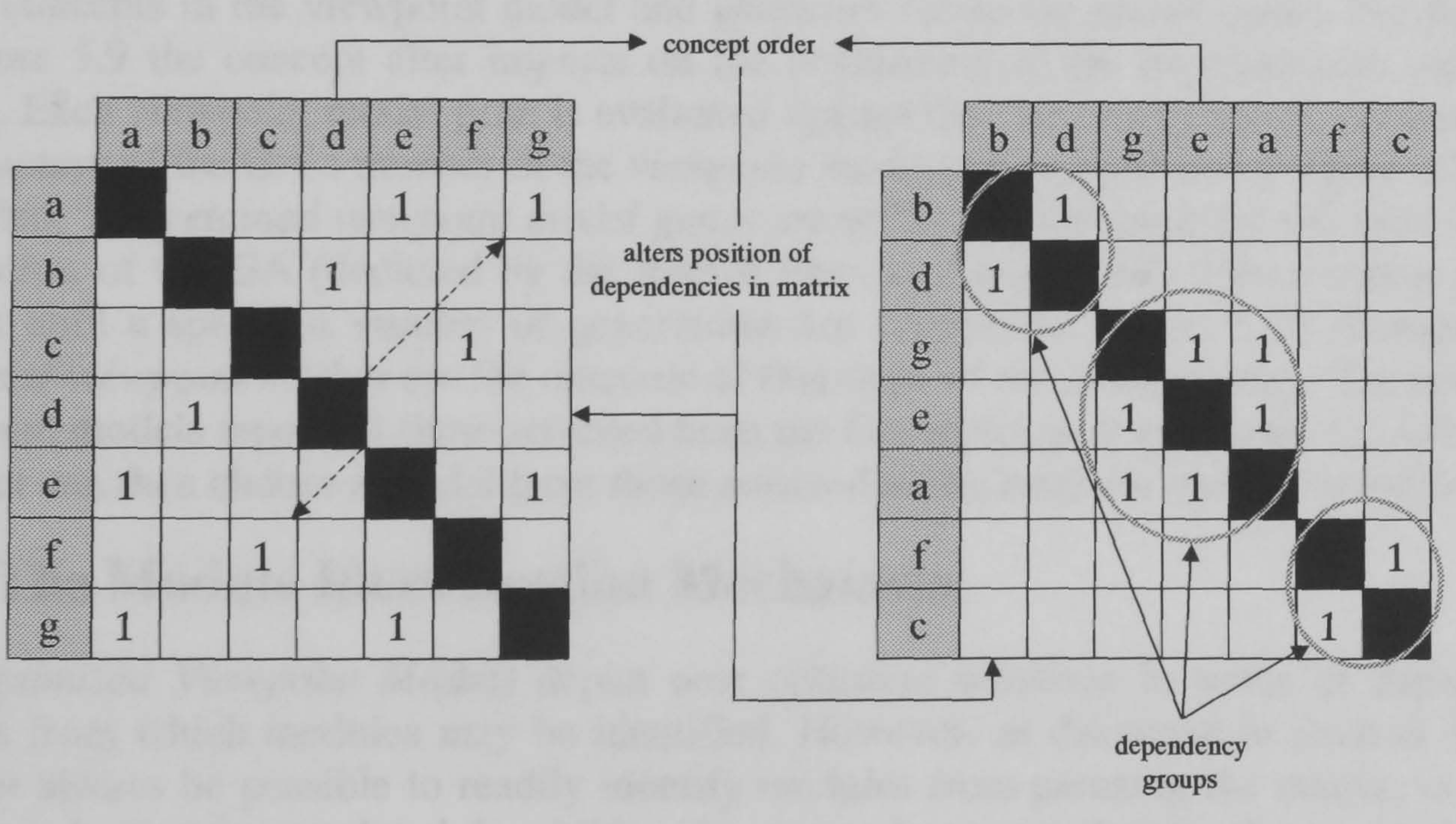


Figure 5.9: The effect of the concept order on the dependencies positioning in the viewpoint model

[19] The examples utilised in this chapter do not represent knowledge from a practical system. They author fabricated the examples for explanatory purposes only.

The CC (defined in Chapter 6) is a measure of the dependency knowledge's distance from the leading line (as depicted by the dashed lines in Figure 5.9). Thus, the clustering criterion nears a minimum the closer the dependencies get to the leading line. As concept groups are identified when dependencies cluster close to the leading line the objective of the Optimisation Mechanism is determine the order that returns the minimum CC value. Thus, each generated solution or 'gene' is evaluated with respect to the CC.

An overview of the application of the Optimisation Mechanism within the proposed Multi-Viewpoint MD Methodology is illustrated in Figure 5.10.

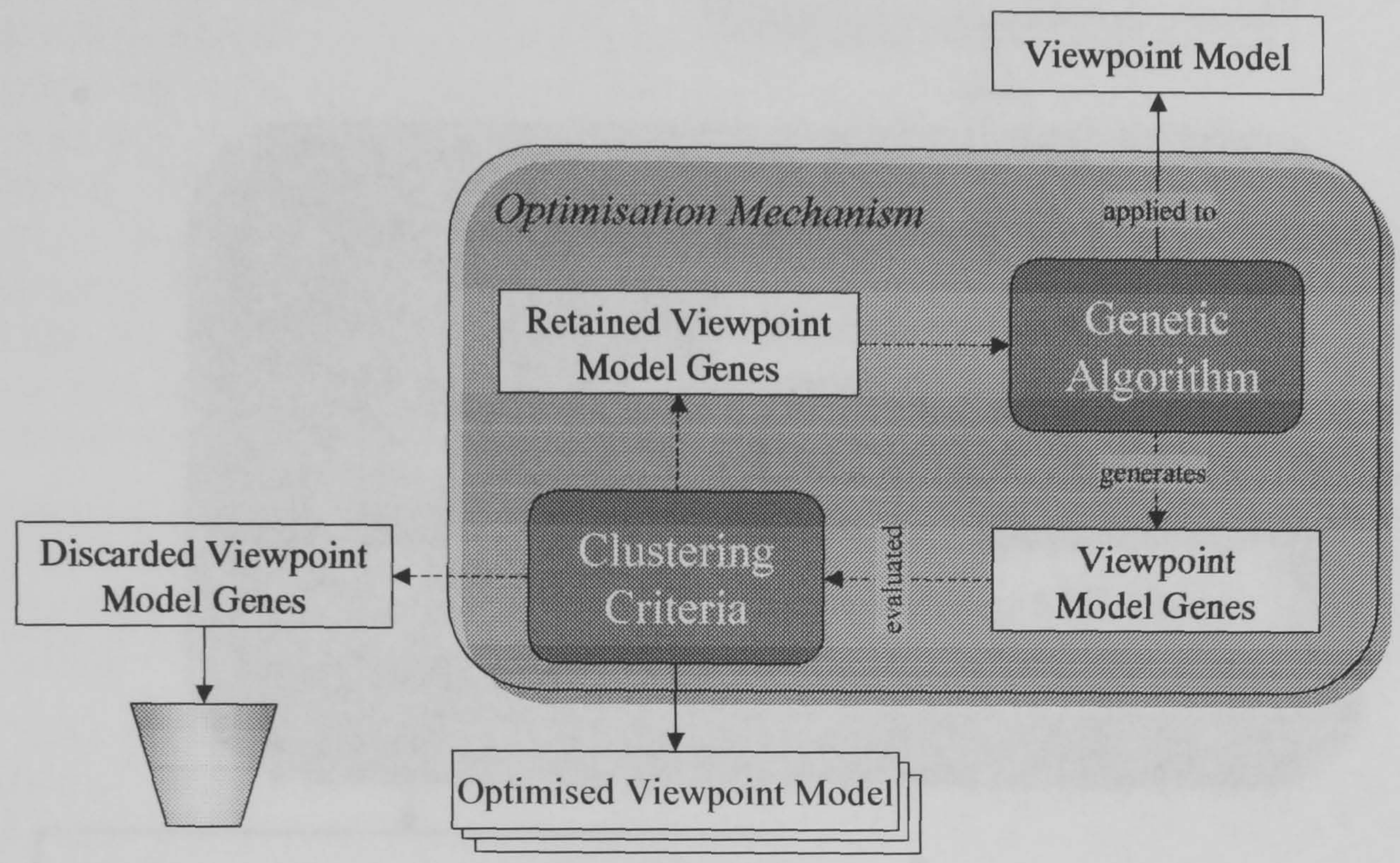


Figure 5.10: The Optimisation Model

The figure illustrates that the GA is applied to the viewpoint model. The GA alters the order of the concepts in the viewpoint model and generates *viewpoint model genes*. As illustrated in Figure 5.9 the concept alter impacts on the positioning of the dependencies within the matrix. Each viewpoint model gene is evaluated against the clustering criteria. Depending on the structure of the GA a number of the *viewpoint model genes* are retained whilst others are discarded. The *retained viewpoint model genes* are utilised as the basis for the next iterative application of the GA (depicted by the dashed lines in Figure 5.10). This iterative process repeats until a specified number of generations are completed. Figure 5.10 illustrates that *optimised viewpoint models* are the outcome of this stage of the methodology. The optimised viewpoint models represent those returned from the GA with a near minimum CC value. The designer can then choose a model from those returned as the basis for *module identification*.

5.4 The Module Identification Mechanism

The *Optimised Viewpoint Models* depict near optimum solutions in terms of dependency clusters from which modules may be identified. However, as discussed in Section 4.1.5, it may not always be possible to readily identify modules from perusing the matrix, i.e. when the matrix is densely populated. In addition, it may not be appropriate for the result of the MI process to be a list structure or definitive module boundaries, as modularity is a relative value and may exist over hierarchical levels of the product structure (see Section 4.1.6). Thus, a further *Module Identification Mechanism* has been defined to address these issues. The Module Identification Mechanism consists of two parts: a *Module Strength Indicator* (MSI) *Function* and a *Modular Structure Matrix* (MSM) *Representation.*

The *MSI* function (detailed in Chapter 6) represents a measure of the modularity of a concept grouping based on the definition of an optimal modular structure (see section 3.2.1). The *MSI* function is applied to all possible concept groupings within the optimised viewpoint model. The *MSM* is a representation of the resulting *MSI* values in a matrix form. The output from the *Module Identification Mechanism* is the *Modular Structure Viewpoint Model* as illustrated by the overview in Figure 5.11.

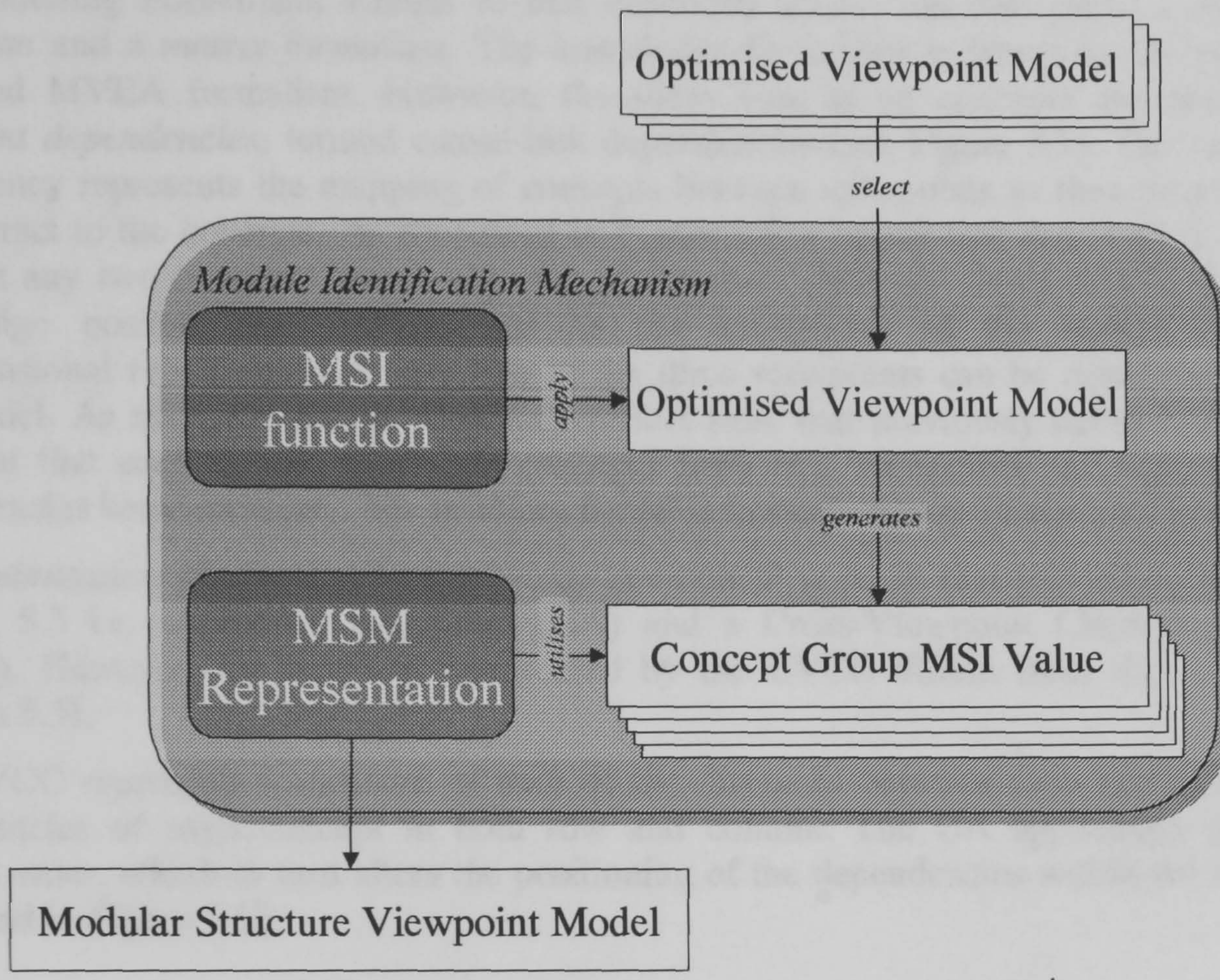


Figure 5.11: Overview of Module Identification Mechanism

The figure illustrates that an optimised viewpoint model is selected and the *MSI function* is applied to that model. The *MSI* function application generates a value for each possible concept grouping within the model. The set of MSI values, returned for all the potential concept groupings in the matrix, are utilised to determine the MSM representation. The MSM representation depicts the inherent modularity of the product structure.

5.5 The Mapping Mechanism

The three methodology elements described through Sections 5.2 to 5.4 are applicable to each of the design viewpoints separately. This results in the definition of a modular structure for individual viewpoints but not the identification of the overall modularity of the product structure or maintenance of the modularity across viewpoints of the product structure. However to support the evolution of the modular design there is a requirement to map between the individual viewpoints. Thus, a *Mapping Mechanism* is defined to support *cross-viewpoint modular development* and *analysis*.

The *Mapping Mechanism* is a key element in providing a coherent and integrated framework required to support the capture and exploration of knowledge from multiple viewpoints and to evolve the modular solution from the abstract to the concrete. The mechanism also supports analysis of the effect of a change in 'modularity' focus i.e. change in design concepts or dependencies including those associated with product *lifecycle objectives*. Further, when utilised in a *Design by Re-use* scenario the mechanism could facilitate the analysis of the impact of design changes and support partial re-use of the design solutions (modules) and their associated knowledge. The Mapping Mechanism consists of two parts, a *Modelling Formalism* and an *Optimisation Mechanism.* Though based on the same principles as the methodology elements defined in Sections 5.2 (*Modelling Formalism*) and 5.3 *(Optimisation Mechanism)*, there are a number of key aspects that differ in each. The following defines the two parts of the Mapping Mechanism and illustrates the key differences between these and the methodology elements defined previously.

The Modelling Formalism, similar to that described above, has two parts: a *knowledge formalism* and a *matrix formalism.* The knowledge formalism is based on the previously described MVEA formalism. However, the focus here is on *concepts* and their *cross-viewpoint dependencies,* termed causal-link dependencies (see Figure 5.5). The causal-link dependency represents the mapping of concepts between viewpoints as they progress from the abstract to the concrete. As illustrated in Figure 5.5, a causal-link dependence can exist between any two concepts across any two of the three different viewpoints of the design knowledge posited here. Again, due to the limitations of the matrix formalism (2Dimensional representation) only two of the three viewpoints can be represented in any one model. As such, the *matrix formalism* differs from that previously defined (in Section 5.2.2) in that each matrix represents concepts from two viewpoints and the causal-link dependencies between these. The resulting model is termed a *Cross-Viewpoint Model.*

The *Optimisation Mechanism* has the same component parts as that previously defined in Section 5.3 i.e. a Genetic Algorithm (GA) and a Cross-Viewpoint Clustering Criteria (CVCC). However, the measure represented by the CVCC differs from that of the CC (Section 5.3).

The CVCC represents a measure of sum of the distances between each two consecutive dependencies of each concept in both row and column. The GA application alters the concept order, which in turn alters the positioning of the dependencies within the matrix as illustrated in Figure 5.12.

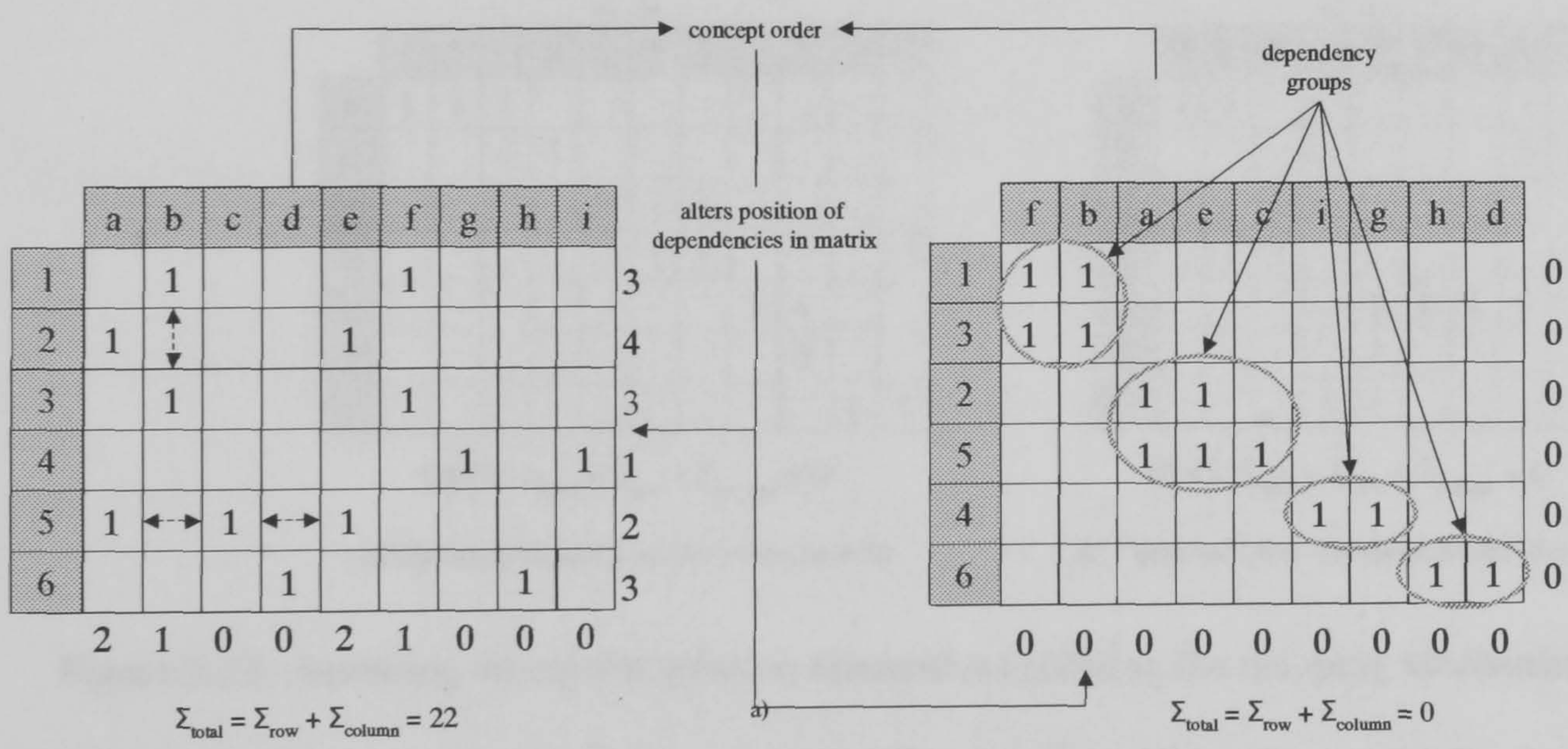


Figure 5.12: The effect of the concept order on the dependencies positioning in the cross viewpoint model

The figure demonstrates that the concept order impacts on the closeness of the dependencies to each other in any row or column. As dependencies move closer to each other it is possible to identify groups of dependent concepts from different viewpoints of design, as can be seen by the right hand matrix in Figure 5.12. The objective of the *Cross-Viewpoint Optimisation Mechanism* is to identify the concept order that optimises these dependent concept groups, i.e. determine the concept order that clusters the dependencies most closely to each other in row and column to minimise the CVCC. The minimum value that the CVCC can have is zero as illustrated in Figure 5.12.

The *Cross-Viewpoint Optimisation Mechanism* allows the designer to map the modular solution across viewpoints. A causal-link dependence between two concepts (C_i and C_j) of different types (i.e. from different viewpoints, *A* and *B* respectively), depicted as causal-link(C_i and C_j), denotes that concept C_i (in viewpoint *A*) is realised (or partially realised) by concept C_j (viewpoint *B*). Thus, the designer may utilise these dependencies to maintain the optimum modular solution returned from one viewpoint (say, viewpoint *A*) in another (say, viewpoint *B*). This 'modular maintenance' process is discussed in greater detail in Section 6.4.2.

In addition, the designer may generate a number of different potential solutions to realise viewpoint *A*. The *Mapping Mechanism* allows the designer to evaluate these different solutions with respect to their ability to maintain the modular solution. Figure 5.13 depicts two different solutions to viewpoint *B* i.e. solution *B* and *B'*. These solutions have been defined as potential realisations of viewpoint *A* (from Figure 5.9).

Formalised Design Knowledge concepts (viewpoint B)

optimum order from viewpoint A

	1	2	3	4	5	6	7	8	9	10	11
b	1						1				1
d								1			1
g			1	1	1						
e				1	1						
a		1				1					
f		1									
c									1	1	

CVCC $\Sigma_{total} = \Sigma_{row} + \Sigma_{column} = 13$

	1	7	11	8	3	4	5	6	2	9	10
b	1	1	1								
d			1	1							
g					1	1	1				
e						1	1				
a								1	1		
f									1		
c										1	1

CVCC $\Sigma_{total} = \Sigma_{row} + \Sigma_{column} = 0$

(a) Optimised Cross-Viewpoint Model (A to B)

Formalised Design Knowledge concepts (viewpoint B')

	1'	2'	3'	4'	5'	6'	7'	8'	9'
b				1					1
d			1			1			
g		1		1			1		
e				1			1		
a	1		1		1			1	
f	1								
c	1	1							

CVCC $\Sigma_{total} = \Sigma_{row} + \Sigma_{column} = 21$

	9'	4'	7'	2'	1'	8'	5'	3'	6'
b	1	1							
d								1	1
g		1	1	1					
e		1	1						
a					1	1	1	1	
f					1				
c				1	1				

CVCC $\Sigma_{total} = \Sigma_{row} + \Sigma_{column} = 6$

(b) Optimised Cross-Viewpoint Model (A to B')

Figure 5.13: Assessing viewpoint solution alternatives utilising the mapping mechanism

Figure 5.13a depicts the *Cross-Viewpoint Model(A* to *B)* whilst Figure 5.13b depicts the *Cross-Viewpoint Model(A* to *B')*. The *Cross-Viewpoint Model (A* to *B)* depicts a possible solution to realising viewpoint A and has an optimum CVCC value of 0 (Figure 5.13a). However, *Cross-Viewpoint Model (A* to *B')* depicts an alternative solution to the realisation of *viewpoint A*. The optimum CVCC value returned for this solution is 6 as depicted by the *Optimised Cross-Viewpoint Model(A* to *B')* in Figure 5.13b. Thus, if we consider *concepts* of the *Optimised Cross-Viewpoint Model(A* to *B')* and their realisation of the three main clusters defined in the *Optimised Viewpoint Model A* (clusters{b,d},{g,e,a} and {f,c} as shown in Figure 5.9) we see that these are realised by viewpoint *B'* concepts {3',4',6',9'}, {1',2',3',4',5',7',8'} and {1',2'} respectively. Thus, in the case of the viewpoint *B'* solution there are a number of viewpoint B' concepts ($1', 2', 3', 4'$) which realise concepts from two or more of the groupings defined in the *Optimised Viewpoint Model A* (Figure 5.9). Thus, there are significantly more concepts in the viewpoint B' solution, than that of viewpoint B solution, which would require to be reconsidered (further decomposed and/or redesigned) to maintain the modular solution. Thus the designer can assess alternative solutions, based on the objective of maintaining the modular solution, utilising the *Mapping Mechanism*. In the case of Figure 5.13, viewpoint B represents the more optimal solution.

Figure 5.14 illustrates the inputs, outputs and interaction of the constituent parts of the methodology's *Mapping Mechanism*. The Mapping Mechanism is utilised to maintain the modular solution. The Mapping Mechanism consists of two parts: a modelling formalism and an optimisation mechanism. The modelling formalism (similar to that defined in Section 5.2) utilises the MVEA knowledge formalism and a matrix formalism. The modelling formalism generates *Cross-Viewpoint Models*. The figure illustrates that a *Cross-Viewpoint Model* can be generated between the concepts of any two viewpoints. The model depicts the causal-link relations between concepts of any two viewpoints. The Cross-Viewpoint Model extracts knowledge of the optimal solution for one viewpoint (say, the function viewpoint) and knowledge of the concepts that realise this solution in another viewpoint (say, the working principle viewpoint). Thus, in a *Cross-Viewpoint Model*$_{(function\ to\ working\ principle)}$ the row represents the modular solution from the *function viewpoint* (both *concepts* and their near optimal order) and columns represent the *working principle concepts* formalised from the evolutionary design knowledge generated as the design activity evolves. Entries in the matrix body represent causal relations from the function concepts to working principle concepts.

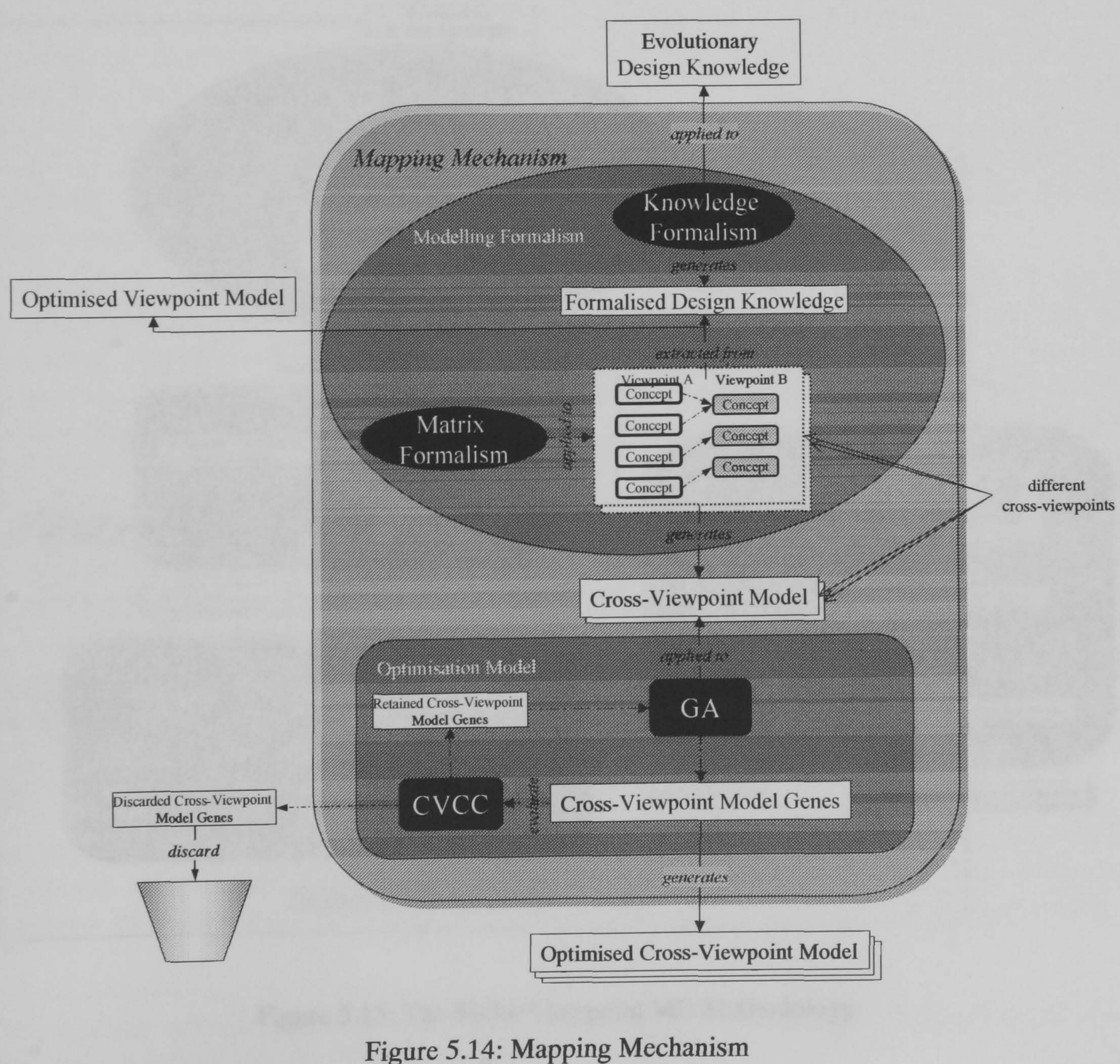


Figure 5.14: Mapping Mechanism

The optimisation mechanism consists of a Genetic Algorithm (GA) and a Cross Viewpoint Clustering Criteria (CVCC). The GA is applied to optimise the working principle concept sequence based on the optimum function concept order, i.e. the function concept order is maintained whilst the working principle concept order is re-sequenced by the GA. The GA produces a number of *Cross-Viewpoint Model Genes*, which are evaluated against the CVCC. The objective of the optimisation is to minimise the CVCC value. Based on the structure defined for the GA, the genes are either discarded or retained, the retained genes are utilised as the basis for the next GA application iteration. The GA iteration continues until a pre-defined number of generations are completed. The *Optimised Cross-Viewpoint Models* represent those returned by the GA with a near minimum CVCC value. The returned optimised models can be used to analyse the capabilities of a viewpoint solutions to maintain the modular design. As an analysis of the Optimised Cross-Viewpoint Models can highlight concept candidates for further consideration, for example decomposition into sub-concepts and/or redesign, they can be utilised as the basis to support the evolution the CWK.

Having defined the elements of the Multi-Viewpoint MD Methodology Figure 5.15 illustrates the entire methodology, its elements, their constituent parts and the interaction between these. The methodology is applied to viewpoints of the design activity phase of the product lifecycle at discrete intervals in a cyclic and iterative nature as illustrated in Figure 5.2.

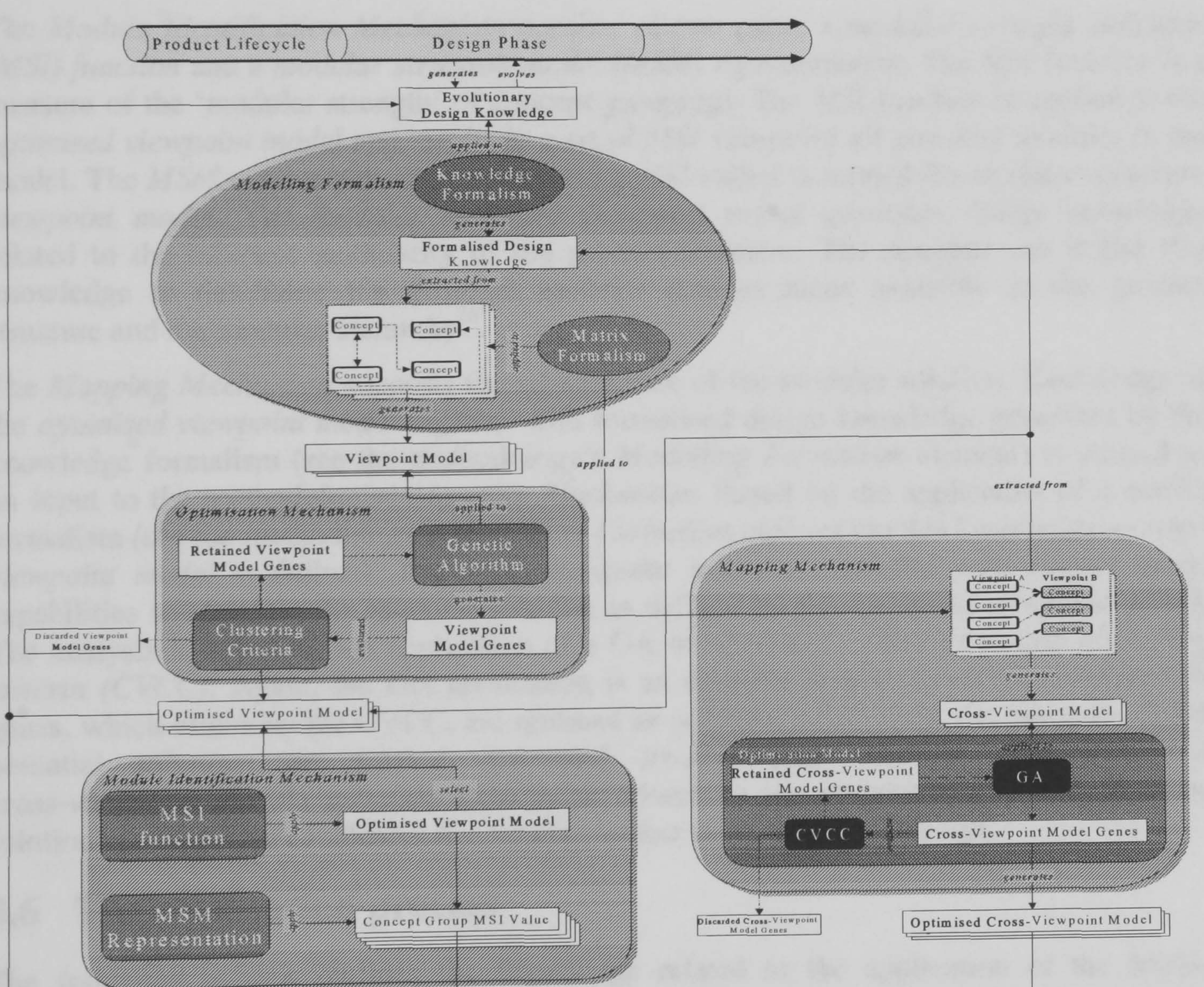


Figure 5.15: The Multi-Viewpoint MD Methodology

Figure 5.15 depicts the entire methodology as applied to an iteration of the design activity. The figure illustrates that evolutionary design knowledge is generated during the design phase of the product lifecycle. As part of the *Modelling Formalism* element of the methodology a *knowledge formalism* is applied to the generated design knowledge which is formalised within a number of design viewpoints, i.e. *function, working principle and structure*. The evolutionary design knowledge is formalised as concepts and their dependencies. A *matrix formalism* is applied to this formalised design knowledge to generate both *viewpoint models* and *cross-viewpoint models*. The *viewpoint model* becomes the input to the *Optimisation Mechanism*.

A Genetic Algorithm (GA), part of the *Optimisation Mechanism*, is applied to the *viewpoint model*. The GA application results in a number of *viewpoint model genes* that are evaluated with respect to a defined clustering criterion (CC). The objective of the *Optimisation Mechanism* is to minimise the CC. The GA application is an iterative process in which a number of genes, which minimise the CC, are returned as potential solutions to the designer. These solutions are termed *optimised viewpoint models*. The designer can select an *optimised viewpoint model* as the input to both the *Module Identification Mechanism* and the *Mapping Mechanism*. In addition, the *optimised viewpoint model* generates design knowledge with respect to the optimum grouping of *concepts*. This design knowledge can in turn be utilised to support the evolution of the design phase of the product lifecycle and its associated knowledge.

The *Module Identification Mechanism* consists of two parts: a *modular strength indicator (MSI) function* and a *modular structure model (MSM) representation*. The *MSI* function is a measure of the 'modular strength' of concept groupings. The *MSI* function is applied to the *optimised viewpoint model* and results in a set of *MSI* values for all potential modules in the model. The *MSM* representation of the resulting *MSI* values is termed the *modular structure viewpoint model*. The *modular structure viewpoint model* generates design knowledge related to the inherent modularity of the product structure. The designer can utilise this knowledge to determine the different modular configurations available in the product structure and the modular hierarchy[20].

The *Mapping Mechanism* supports the maintenance of the modular solution. Knowledge of the *optimised viewpoint model* together with formalised design knowledge generated by the knowledge formalism (see the methodology's *Modelling Formalism* element) is utilised as an input to the methodologies *Mapping Mechanism*. Based on the application of a matrix formalism (see the methodology's *Modelling Formalism* element) to this knowledge a *cross-viewpoint model* is defined. The *cross-viewpoint model* is analysed with respect to its capabilities to maintain the modular solution as defined by the *optimised viewpoint model*. The analysis is based on the application of a GA to minimise a *cross-viewpoint clustering criteria (CVCC)*. Again, the GA application is an iterative process in which a number of genes, which minimise the CVCC, are returned as potential solutions to the designer. These potential solutions are termed *optimised cross-viewpoint models*. The *optimised cross-viewpoint models* generate knowledge related to the maintenance of the modular solution and highlight candidates that require further consideration in design.

5.6 The application process

The following section outlines the knowledge related to the application of the Multi-Viewpoint MD Methodology to design. As the methodology was developed to support improved EDR the procedural knowledge it is defined based on its application within a

[20] The modular configuration and hierarchy based knowledge generated by the *Module Identification Mechanism* is defined in greater detail in Chapter 6

current design activity (i.e. *for re-use*). Figure 5.16 denotes the procedural steps for applying the Multi-viewpoint MD Methodology as illustrated in Figure 5.15 to the design process.

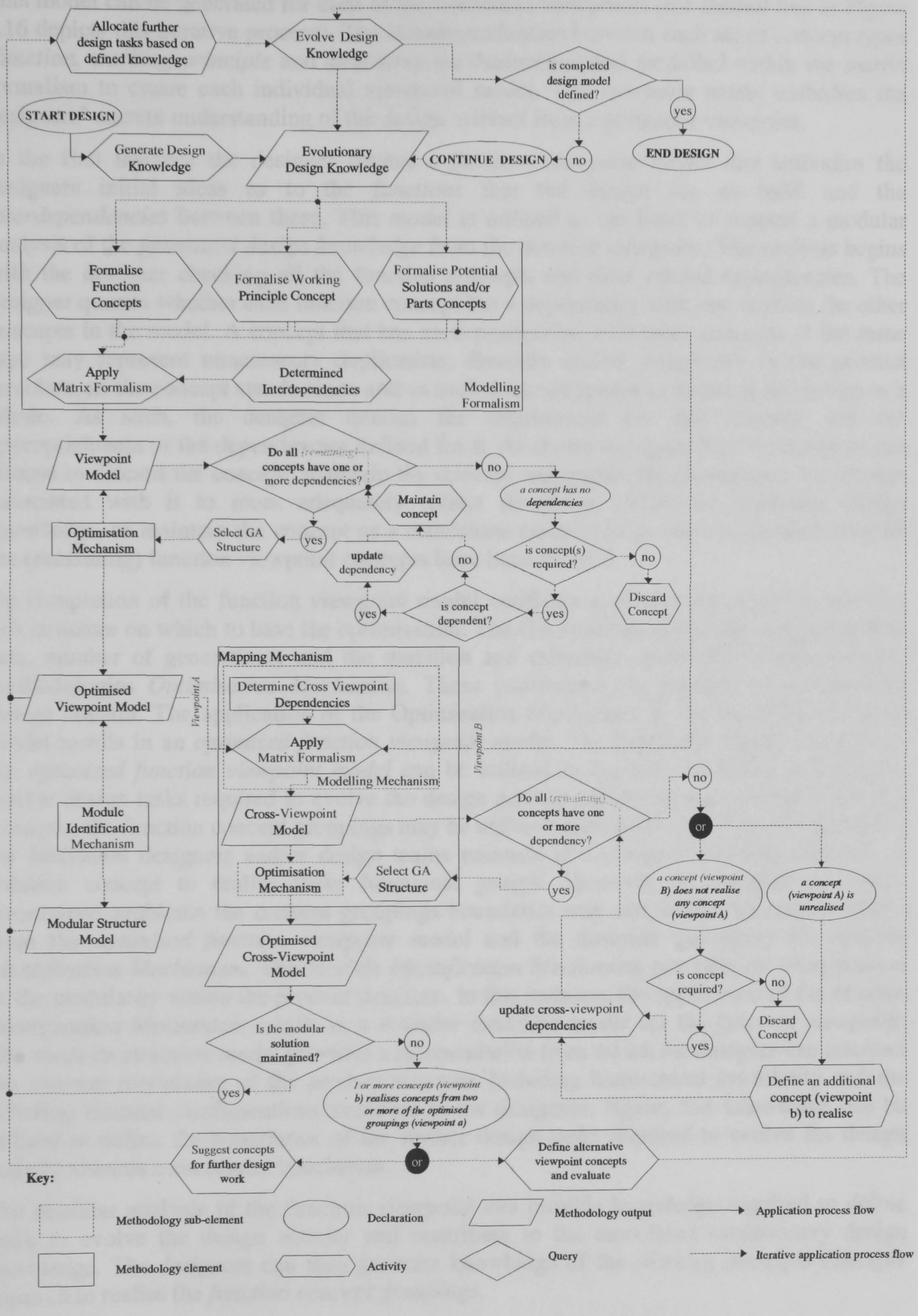


Figure 5.16: Application procedure during design

As illustrated in Figure 5.16 the design activity starts and knowledge is generated and utilised as the basis for the methodology's *Modelling Mechanism*. The generated design knowledge is formalised as initial function, working principle and structure concepts in design. A matrix formalism is applied to define a model on which to support further analysis. This model can be generated for each of the individual viewpoints (the dashed line in Figure 5.16 depicts this iterative process). The interdependencies between each set of *concept types* (*function, working principle* and *structure*) are determined and modelled within the matrix formalism to create each individual viewpoint model. The *viewpoint model* embodies the designers' current understanding of the design artefact from a particular viewpoint.

In the first instance the designer creates a *function viewpoint model* that embodies the designers initial ideas as to the *functions* that the design has to fulfil and the *interdependencies* between these. This *model* is utilised as the basis to support a modular analysis of the *generated design knowledge* from the *function viewpoint*. The analysis begins with the designer checking all the function concepts and their related dependencies. The designer queries whether each function concept has a dependency with one or more the other concepts in the model. A concept that has no dependencies with other concepts of the same type may represent unnecessary duplication, diversity and/or complexity in the product structure, as the concept stands alone and as such does not appear to augment the design as a whole. As such, the designer queries the requirement for that concept and the appropriateness of the dependencies defined for it. As shown in Figure 5.16 the designer can choose to discard the concept, maintain the concept and update the dependency knowledge associated with it to more adequately reflect its status within the generated design knowledge, or maintain the concept as a standalone entity. The process is repeated until all the (remaining) function viewpoint concepts have been verified.

On completion of the function viewpoint model verification process the designer selects a GA structure on which to base the optimisation. The GA structure determines the population size, number of generations, and the mutation and crossover operators of input into the methodologies *Optimisation Mechanism.* These parameters are intrinsic to the modular design domain. The application of the Optimisation Mechanism to the function viewpoint model results in an *optimised function viewpoint model.* The *concept groupings* defined in the *optimised function viewpoint model* can be utilised as the basis to define and allocate further design tasks required to evolve the design activity and its associated knowledge, for example, the function concept groupings may be utilised as the basis to define the boundaries for individual designers and/or design teams research into potential working principle or solution concept to realise these functional groups. However, in complex or highly constrained problems the concept groupings boundaries may not always be clearly visible from the *optimised function viewpoint model* and the designer can apply the *Module Identification Mechanism.* The *Module Identification Mechanism* provides an interpretation of the modularity within the product structure. In this instance, the application of the *Module Identification Mechanism* results in a *modular structure model* for the *function viewpoint.* The *modular structure model* provides a representation from which the designer can interpret the inherent modularity of the product structure including hierarchical modularity and the differing modular configurations available to the designers. Again, this knowledge can be utilised to define the boundaries of the further design tasks required to evolve the design activity towards a successful conclusion.

The modular analysis of the function viewpoint can provide knowledge required to define tasks to evolve the design activity and contribute to the associated evolutionary design knowledge. The designers can then generate knowledge of the *working principle concepts* required to realise the *function concept groupings.*

The working principle concepts can be analysed both as an individual model of a viewpoint (a *working principle viewpoint model*) and with respect to its capabilities to maintain the modular solution. The analysis of the working principle viewpoint model is carried out through the same process as that described above for the function viewpoint model. The analysis results in knowledge of optimum *working principle groupings* on which to base further design tasks and research required to support the realisation these groupings as structural entities, i.e. *solution concepts* or *parts*. The analysis of the capabilities of *working principle concepts* to maintain the modular solution defined in the *function viewpoint* is achieved through the application of the methodology's *Mapping Mechanism.*

The *Mapping Mechanism* analyses the working principle concepts that have been defined by the designers to realise the function concepts. As such, the results (the *concept order*) defined in the *optimised function viewpoint model* is utilised as the basis for the analysis. A matrix formalism is applied to both the function and working principle concepts. The *cross-viewpoint dependencies* between these are defined and represented in the matrix body. The analysis begins with the designer checking all the dependencies between the function and working principle concepts. The designer queries if the concepts in each viewpoint have at least one more dependencies with the concepts from the alternative viewpoint, i.e. whether each *working principle concept* realises (or partially realises) a *function concept* and whether each *function concept* is realised by at least one *working principle concept*. Each concept in a preceding, more abstract, viewpoint (termed viewpoint *A* in Figure 5.16) should be realised by one or more concepts in the following, more concrete, viewpoint (termed viewpoint *B* in Figure 5.16) to maintain the integrity of the design, for example, if a function concept is not realised by a concept in the working principle viewpoint the design will evolve without embodying aspects of the defined functionality. The designer checks the dependencies and identifies a scenario where either: (i) a concept in viewpoint *B* which does not realise any concepts from viewpoint *A*, or (ii) alternatively a concept from viewpoint *A* that is not realised by any concepts in viewpoint *B*. In the event of the first case (scenario (i) above) case the designer can assess the viewpoint *B* concept and consider whether it is required. If the viewpoint *B* concept is deemed to be required then the designer is obligated to update the defined dependencies to better reflect the concept's dependency to viewpoint *A*. On the other hand if the concept is deemed to be unnecessary the designer can choose to discard it. In the second case (scenario (ii) above), where a viewpoint *A* concept is not realised by any concepts in viewpoint *B* the designer is required to define an additional viewpoint *B* concept to realise the neglected viewpoint *A* concept. In the case where a new concept is defined and added to viewpoint *B* the dependencies within the *cross-viewpoint model* required to be updated to reflect the alteration to the model. This process is repeated until all remaining concepts in the cross-viewpoint model have been verified.

On completion of the *cross-viewpoint model verification*, the designer requires to define a GA structure for *Mapping Mechanism's* sub element, the *optimisation mechanism.* The *optimisation mechanism* is applied to the *working principle viewpoint* whilst the *function viewpoint* is maintained. This allows the designer to optimise the working principle viewpoint with respect to the objective of maintaining the modular solution (as defined in the *function* viewpoint). The resulting *optimised cross-viewpoint model* can be utilised to assess the performance of the working principle concepts with respect to their capabilities to maintain the modular solution. If the modular solution is maintained the designer can utilise the findings to define further design tasks necessary to evolve the design activity and consequently its associated knowledge. The modular solution is not maintained the designer can identify design concepts from the *optimised cross-viewpoint model* that require further design activities to be carried out on them, i.e. design concepts from viewpoint *B* which fulfil concepts from more than one concept grouping in viewpoint *A*. In addition, the designer may conclude that the defined viewpoint concepts (viewpoint *B*) have such a poor performance with respect to maintaining the modular solution (defined in viewpoint *A*) and take the decision to define an alternative set of viewpoint concepts to evaluate.

The process is repeated for the individual structure viewpoint (the *structure viewpoint model)* and a mapping from the working principle viewpoint to the structure viewpoint (the *cross-viewpoint model working principle to structure)*. This application to the function, working principle and structure viewpoints represents one cycle of the methodology itself. However, the design activity is an iterative process and designers' knowledge of a design from a particular viewpoint evolves as the design progresses. Thus, the designer generates knowledge of more concrete and detailed viewpoint concepts as the design evolves. As such the entire methodology can be applied at various stages of the iterative process of design as illustrated by the dashed lines in Figure 5.16 and depicted in Figure 5.17.

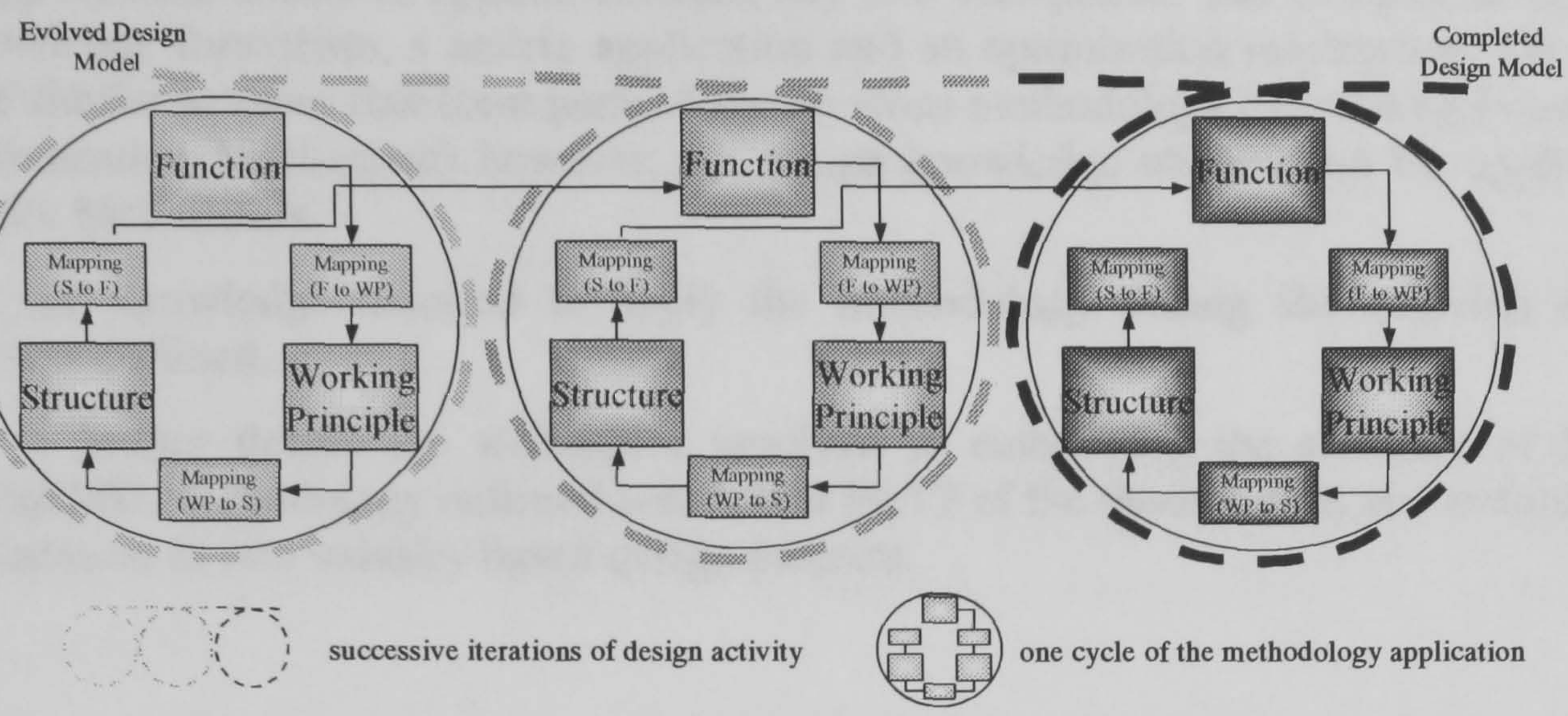


Figure 5.17: An overview of the methodology's cyclic and iterative application in the design activity

Figure 5.17 depicts the methodologies application in the iterative process of design. The figure illustrates that the design activity evolves over a number of iterations from the abstract to the concrete (depicted by the successively darker dashed loops in Figure 5.17). Within each iterative loop the design model evolves (design knowledge) from abstract concepts (function) to more concrete concepts (structure). The methodology is applied to structure the knowledge of the evolved design model in each loop and maintain the modular solution throughout this iteration. However, as the design evolves the designers' knowledge of the evolved design model and its associated design concepts within each viewpoint becomes increasingly more detailed and concrete (depicted by the successively darker depiction of the methodology in Figure 5.17). As such, the methodology is successively applied until a completed design model is defined.

5.7 Chapter summary

This chapter introduced the declarative and procedural knowledge that defines a novel MD methodology to support improved EDR developed through the work presented in this thesis i.e. the Multi-Viewpoint Modular Design Methodology. The novel methodology has been developed to support the concept of *knowledge modularisation* to aid in the evolution of a modular design solution while structuring its associated knowledge. The Multi-Viewpoint MD Methodology consists of four main elements: a *Modelling Formalism*, an *Optimisation Mechanism*, a *Module Identification Mechanism* and a *Mapping Mechanism*. The first three elements are applicable within each *viewpoint* whilst the later is utilised to support knowledge related to the maintenance of the modular design, i.e. across *viewpoints*.

The *Modelling Formalism* models design knowledge within a chosen *viewpoint* using elements of a previously developed knowledge formalism and a matrix application. The resulting model is known as the *Viewpoint Model*.

The *Optimisation Mechanism,* consisting of a genetic algorithm (GA) and a clustering criterion (CC), is applied to the *Viewpoint Model.* The different generations of the *Viewpoint*

Model generated by the GA are evaluated against the CC. Those that result in minimisation of the CC are presented to the designer as *optimised Viewpoint Models.*

The *Module Identification Mechanism* consists of a module strength indictor (MSI) function and a modular structure matrix application and is applied to the designers chosen *optimised Viewpoint Model.* The modular structure matrix is applied to the returned MSI values and the resulting representation is termed a *modular structure model.*

The *Mapping Mechanism* is applied between any two viewpoints. The mechanism consists of a knowledge formalism, a matrix application and an optimisation mechanism. The three parts are similar to those that form parts of the previous methodology (Modelling Formalism and Optimisation Mechanism) however, the design knowledge utilised and the application process for each differs.

Finally, the knowledge required to apply the methodology during the evolving design activity was outlined.

Chapter 6 further details the techniques involved in embodying the elements of Multi-Viewpoint MD Methodology outlined here whilst Part 3 of the thesis details and evaluates its implementation in two industry based design projects.

6 Approach Formalisms and Mechanisms

The following chapter details the formalisms and mechanisms that embody the four main elements of the Multi-Viewpoint MD Methodology outlined in Chapter 5. Section 6.1 details the *Modelling Formalism,* the *Optimisation Mechanism* is described in Section 6.2, the *Module Identification Mechanism* in Section 6.3 and the proposed *Mapping Mechanism* in Section 6.4. Section 6.5 summarises the chapter.

6.1 Modelling Formalism

The *Modelling Formalism* (outlined in Section 5.2) consists of two parts: a *knowledge formalism* (Section 6.1.1) and a *dependency matrix formalism* (Section 6.1.2). The output is a *Viewpoint Model.*

6.1.1 Knowledge Formalism

The formalism is based on the Multi-Viewpoint Evolutionary Approach (MVEA) (Zhang 1998). The following summarises the elements of relevance to the work presented in this thesis. Table 6.1 indicates the elements of the formalism that are elaborated upon in the following sections.

This section outlines the formalisms within MVEA of *function, working principle, part* and *solution concepts* and the *functional* and *physical link dependencies* between these. The knowledge is formalised into three *viewpoint models: function, working principle* and *structure* (Section 5.2.1).

CONCEPT	
Concept	Indicates the existence of an idea
View	Specific viewpoint that is taken of a design
CONCEPT TYPES	
Function	Verb-noun pair that indicates the purpose of the function e.g. reduce speed
Working Principle	Working principle in the working design e.g. lever-principle
Part	A part in the working design e.g. gear
Solution	A solution in the working design e.g. gear-pair
RELATION TYPES	
Functional- dependency	Matter flow between concepts (functioning sequences)
Physical-link	A physical connection between two concepts
Causal-link	A mapping relation denoting the realisation (or partial realisation) of a concept of one type (from one viewpoint) by a concept of different type (from another viewpoint)

Table 6.1: MVEA elements of interest

Concepts

There are four types of concepts required to model Current Working Knowledge (CWK): *Function, Working Principle, Solution* and *Part* (Zhang 1998) as defined in Table 6.2.

Concept	Definition
Function	The purpose of a design, for example *'Reduce-Speed'*
Working Principle	The techniques employed to implement the functions of a design, for example *'Lever-Principle'*
Part	The physical components which realise functions and working principles of a design for example *'Gear'*
Solution	The substantial realisation of the functions and working principles of a design, for example *'Gear-Pair'*

Table 6.2: Concept descriptions

The *concepts* can have input matters (IM), output matters (OM) [21], and behaviour properties (BP)[22]. The IM and OM have three different types of matter associated with them: *Energy, Information* and *Material.* The *part concept* (*PART*) can also have characteristics (Char.) of the type: *Shape (SP), Dimensions (D), Surface Qualities* (SQ) and *Material (M).*

Dependencies

Dependencies between *concepts* have been shown to be important to the determination of *modularity* of a designed product (see Section 3.2.1). The notations for the MVEA relation formalisms are given in Table 6.3.

The formalism of the *functional-dependency, physical link* and *causal-link* relations are covered in detail within this section. For the formalism of relations *has-kind, a-kind-of, has-part,* and *a-part-of* the reader is referred (Zhang 1998). The *causal-link* relation supports the activity of '*mapping*' between different types of *concepts.* As such, the *causal-link* relation supports the formalism of Cross-Viewpoint Models.

Due to limitations of the matrix formalism (see section 5.2.2), the *has-kind, a-kind-of, a-part-of* and *has-part* relations are not explicitly represented in the viewpoint models. However, their impact and potential utilisation to support the concept of *knowledge modularisation* is discussed in future work (Section 9.4).

[21] The IM represents the initial states of the operand before it is transformed by the concept and the OM represents the final state of the operand after it is transformed by the concept, for example, consider the function concept reduce speed, here, the IM would represent the 'input speed', whilst the OM would represent the 'output speed'.

[22] The BP represents the external working states and/or how the IM are transformed to the OM, for example, again consider the function concept reduce speed, here, the BP would represent the 'speed ratio'.

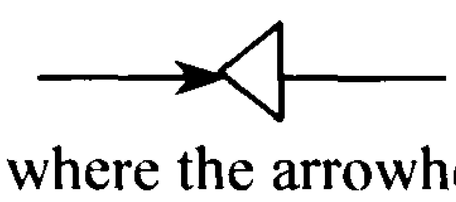
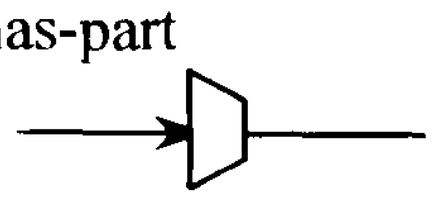
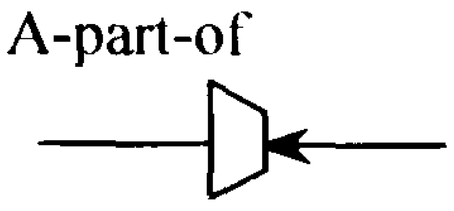
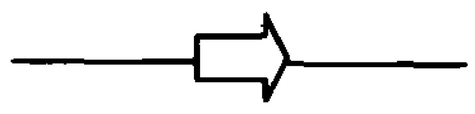
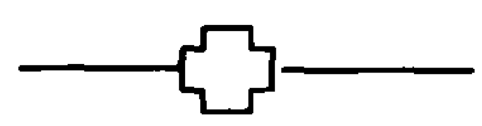
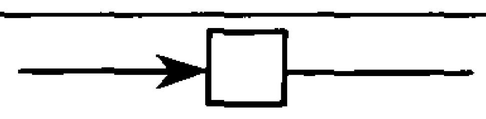
Relation	Notation	Description
has-kind[23]	 where the arrowhead denotes the direction of the specialisation	Organise *concepts* of the same type at different abstraction levels. *Has-kind* or *a-kind-of* relations represent knowledge that supports the design activities '*specialising a concept*' and '*selecting an alternative concept*'.
has-part, a part-of	has-part  A-part-of 	Organise *concepts* of the same type at different levels of detail i.e. represents the design activity *decomposing a concept*.
functional-dependency	 where the block arrow notation represents the direction of matter flow	Represents the matter flow between concepts of the decompositions of concepts.
physical-link	 where the non-arrowhead notation reflects the nature of the dependence i.e. implies reverse relation.	Represents a physical dependence between the decompositions of concepts.
causal-link	 where the arrowhead denotes the direction of the mapping	Represents the causal relationship between concepts of different types. Reflects the evolution of the working design across views that is accomplished by the design activity '*mapping a concept*'.

Table 6.3: MVEA relations (Zhang 1998)

Functional dependency

A *functional dependency* relation can exist between two *concepts* of the same type at the same level of decomposition. The *functional-dependencies* determine the functioning sequences of the *concepts*. The *functional-dependency* (C_i, C_j) is illustrated in Figure 6.1.

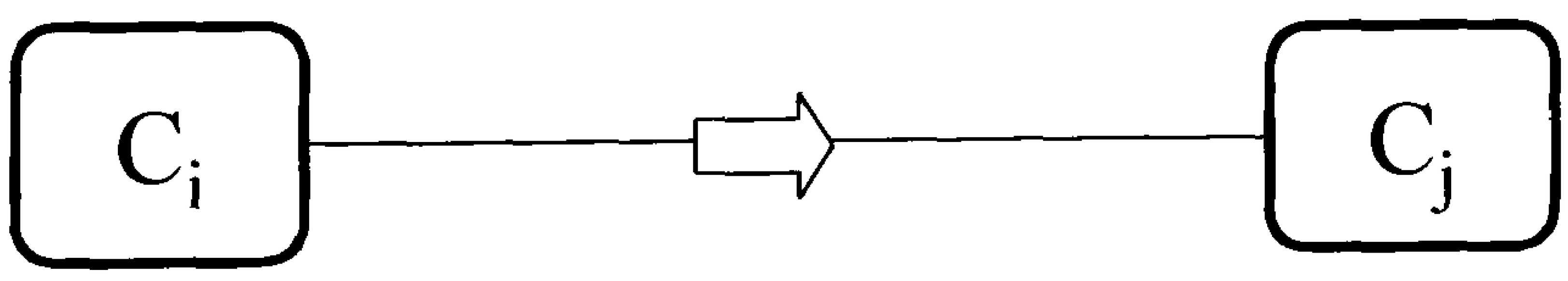


Figure 6.1: Functional-dependency (C_i, C_j), (Zhang 1998)

A *functional dependency* represents matter flow between two *concepts*. Matter can be of the types: Information, Material and Energy. Thus, depending on the matter of interest to the designer, the knowledge of the design can be viewed from the *perspectives* of information,

[23] An '*a-kind-of*' relation reflects the activity of *generalisation. Generalisation* is not used in evolving CWK.

material, and energy flow between the *concepts*. Matter flow is directional and as such *functional dependencies* are directional i.e. functional-dependency (Ci, Cj) ≠ functional-dependency (Cj, Ci). The block arrow notation in Figure 6.1 indicates the direction of the functional dependency.

Physical-link dependencies

A *physical-link* relation can exist between two *concepts* of the same type and at the same level of decomposition. The *physical-link* represents the physical connection between two *concepts*. Naturally, a *physical-link* implies that the reverse relation is true i.e. *physical-link*(C_i, C_j) → *physical-link*(C_j, C_i) (where → denotes 'implies'). The *physical-link*(C_i, C_j) is illustrated in Figure 6.2.

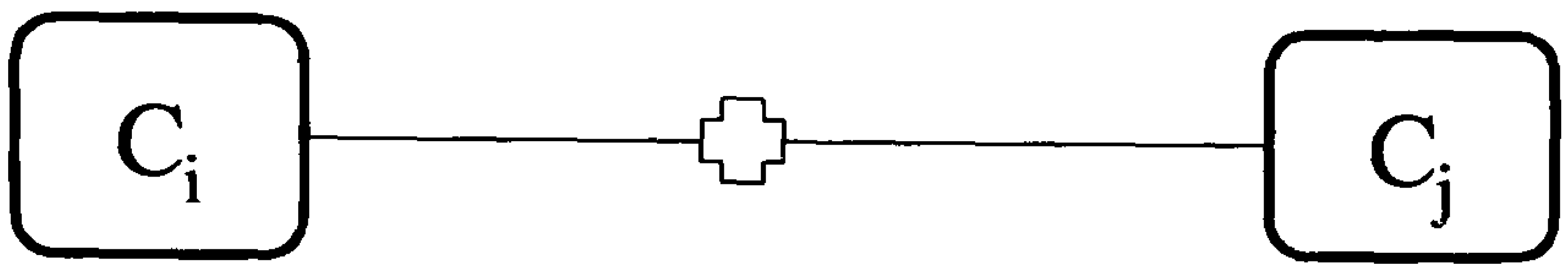


Figure 6.2: Physical-link(C_i, C_j), (Zhang 1998)

Thus, we can also view design knowledge from the *perspective* of *physical-link dependencies* between *concepts*.

Dependencies related to lifecycle objectives

As discussed in Section 3.2.4 and Section 4.1.4, *modularisation* is also seen as a potential principle to develop product definitions that can enhance the fulfilment of *lifecycle objectives* such as ease of manufacture, maintenance, re-cycling, and disposal.

It is posited here that the *function*, *working principle* and *solution concepts* can also have *characteristics* associated within them, similar to that formalised for the *part concept*. The *characteristics* represent the *concepts'* attributes with respect to particular lifecycle objectives. For example, a *function concept* may have the characteristic *consumer-group*, a *working-principle* may have a characteristic *technology-lifespan*, and a solution concept may have the characteristic *life-in-service*. Thus, two *concepts* of a particular type can have a *similarity-dependence* in terms of their associated lifecycle *characteristic* (as illustrated in Figure 6.3). A *similarity-dependence* relation can exist between two *concepts* of the same type and at the same level of decomposition. The nature of the relation *similarity-dependence* implies that the reverse relation is true i.e. *similarity-dependence* (C_i, C_j) → *similarity-dependence* (C_j, C_i). Thus, in such cases the aim of the modular design process is to determination the optimum product modularity that fulfils the particular life-cycle objective e.g. the determination of the optimum clustering of *concepts* based on their *similarity-dependence* with respect to these *characteristics*.

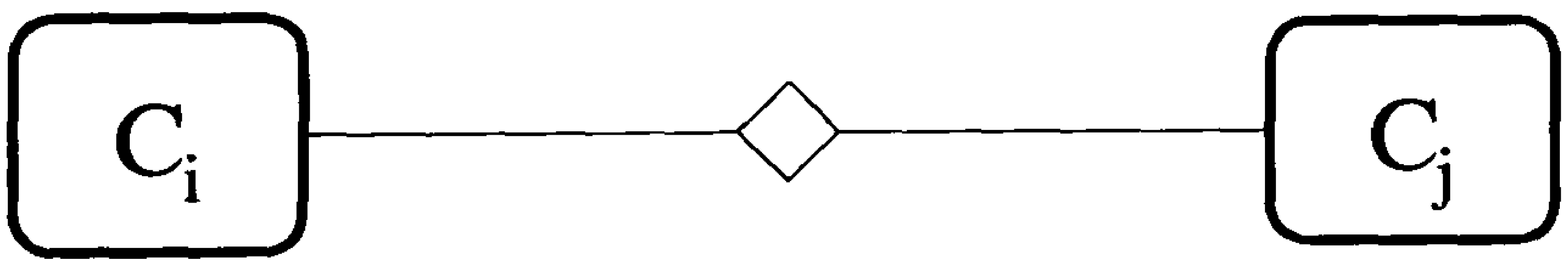


Figure 6.3: Similarity-Dependence (C_i, C_j)

The *characteristics*, which can be defined for a *concept*, reflect the particulars of the required design lifecycle objective(s). Table 6.4 provides some potential examples of these characteristics and the lifecycle objectives which they can embody (the *concept* type to which they may be applicable is based on the findings of Section 4.1.2, illustrated in Figure 4.1).

Life-cycle objective	Concept	Aim	Characteristic
mass-customisation	function	Creation of 'custom modules' to customise product for individual user whilst maintaining mass-production efficiency	consumer-requirements
team design and distribution	function	Create design tasks groupings based on distribution of expertise i.e. reduce travel, communication requirements	expertise-requirement
reduce maintenance	working-principle	Reduction of the number of alterations and/or duration between during routine maintenance	technology-life-span
improve ease of manufacture	solution and/or part	Reduce the number and/or type of operations required	manufacturing operation-type
ease of re-use, recycling and disposal	solution and/or part	Reduction in the number of or distribution of materials in product.	material-type recycle-operation

Table 6.4: Lifecycle objectives and potential concept characteristics

Concepts with similar *lifecycle characteristics* would thus be dependent on each other from the *perspective* of that lifecycle objective. For example, two *working principle concepts* with a *technology-life-span* of 10 years would have a *similarity-dependence.*

It can be seen that design knowledge can be viewed from a number of different *perspectives* that represent design *concepts* and the differing types of *dependencies* that can exist between them, for example, *functional dependencies* (material, information, and energy perspectives), *physical link perspective,* and *similarity dependence* (material type, technology life-span, and manufacturing-operation perspectives).

Causal-link relation

The previously defined dependency formalisms were concerned with the dependency between concepts of the same type i.e. within the same *viewpoint.* However, the relation 'causal-link' has been defined to reflect the 'mapping' of a concept in one viewpoint to one or more concepts in another viewpoint. As such, the *causal-link* relation is utilised for the formalism of a Cross-Viewpoint Model (see Section 6.4.1) and not the formalism of individual Viewpoint Models.

Figure 6.4 denotes a *causal-link* relation from a concept (C_i) to another concept of a different type (C_j), represented as *causal-link*(C_i, C_j). The arrowhead notation in Figure 6.4 indicates the direction of the relation.

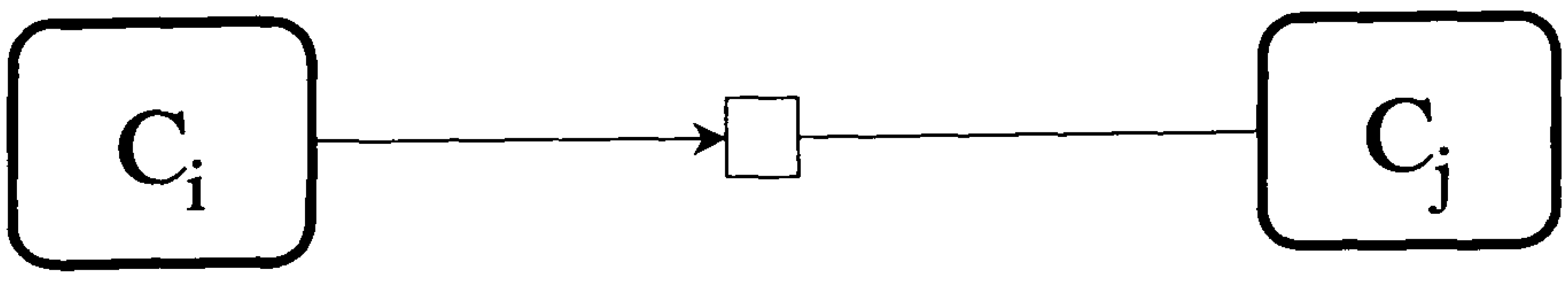


Figure 6.4: Causal-link(Ci, Cj)(Zhang 1998)

Knowledge represented by Viewpoint Models

A number of different viewpoints of design knowledge can exist, the viewpoints utilised in the work presented in this thesis are based on those identified in MD research namely, *function, working principle* and *structure* (see Section 4.1.2).

Table 6.5 clarifies the boundaries of the knowledge represented within each Viewpoint Model and the including the inclusion of the *similarity-dependence* relation within the views formalised as part of the MVEA (Zhang 1998)

Viewpoint Models in the MD Methodology	Description	Elements encompassed from the MVEA formalism	Variations from MVEA formalism
Function Viewpoint Model	The most concrete and detailed *function concepts* and the *associative* relations that exist between them.	The bottom level *concepts* of the *function structure* and the *associations* that can exist between these i.e. the *desired mode of action network*	The inclusion of *similarity-dependence* relations between *function concepts.*
Working Principle Viewpoint Model	The most concrete and detailed *working principle concepts* and the *associative* relations between these.	The bottom level *concepts* and of the *working principle structure* and the *associations* that can exist between these.	The inclusion of *similarity-dependence* relations between *working-principle concepts.*
Structure Viewpoint Model	The most concrete and detailed *solution concepts* and *parts* and the *associative* relations between these.	The bottom level *concepts* of the *solution structure* and the *associations* that can exist between these i.e. the *actual mode of action network* and the *construction network.*	The inclusion of *similarity-dependence* relations between *solution concepts* and parts.

Table 6.5: Viewpoint descriptions

The evolution of the design activity generates successive new levels of the structures (see Section 5.2.2). Therefore the *Viewpoint Model,* the outcome of the modelling formalism element, represents the most concrete and detailed concepts and their *functional, physical* and *similarity dependence* at a discrete time during the design activity.

The conditions for determining the most concrete and detailed *concepts* within a *function structure* at any time are defined in work by Zhang (Zhang 1998). The conditions for determining the most concrete and detailed *working principle concepts* in *working principle structure* of CWK, can be defined as:

$$\text{iff} \neg \exists WPC_j \, ((R_{ij}(WPC_i, WPC_j) \equiv (\text{has-part} \vee \text{has-kind})) \vee (R_{ji}(WPC_j, WPC_i) \equiv \text{a-part-of}))$$

Equation 6.1

Where: iff – if and only if

¬ ~ not – connective (negation)

∃ ~ existential quantifier

≡ ~ equivalent quantifier ('there exists' or 'some' quantifier)

∨ ~ and – connective (conjunction)

WPC_i and WPC_j are two different working principle concepts of CWK;

$R_{ij}(WPC_i, WPC_j)$ is the relation from WPC_i to WPC_j and $R_{ji}(WPC_j, WPC_i)$ is the relation from WPC_j to WPC_i, $R_{ij}(WPC_i, WPC_j)$ and $R_{ji}(WPC_j, WPC_i) \in$ {*has-kind has-part a-part-of functional-dependency similarity-dependence null*}.

The conditions for determining the most concrete and detailed solution concepts and parts within a solution structure at any given time are detailed in work by Zhang (Zhang 1998).

Function Viewpoint

A *function concept* (FC) of a design may have *has-kind, has-part, a-part-of* and *functional-dependency* relations with other function concepts of the design as defined in work by Zhang (Zhang 1998). In addition, for the purposes of analysing *lifecycle modularity* a *function concept* can also have a *similarity-dependence* with other *function concepts* based on their *lifecycle characteristics.*

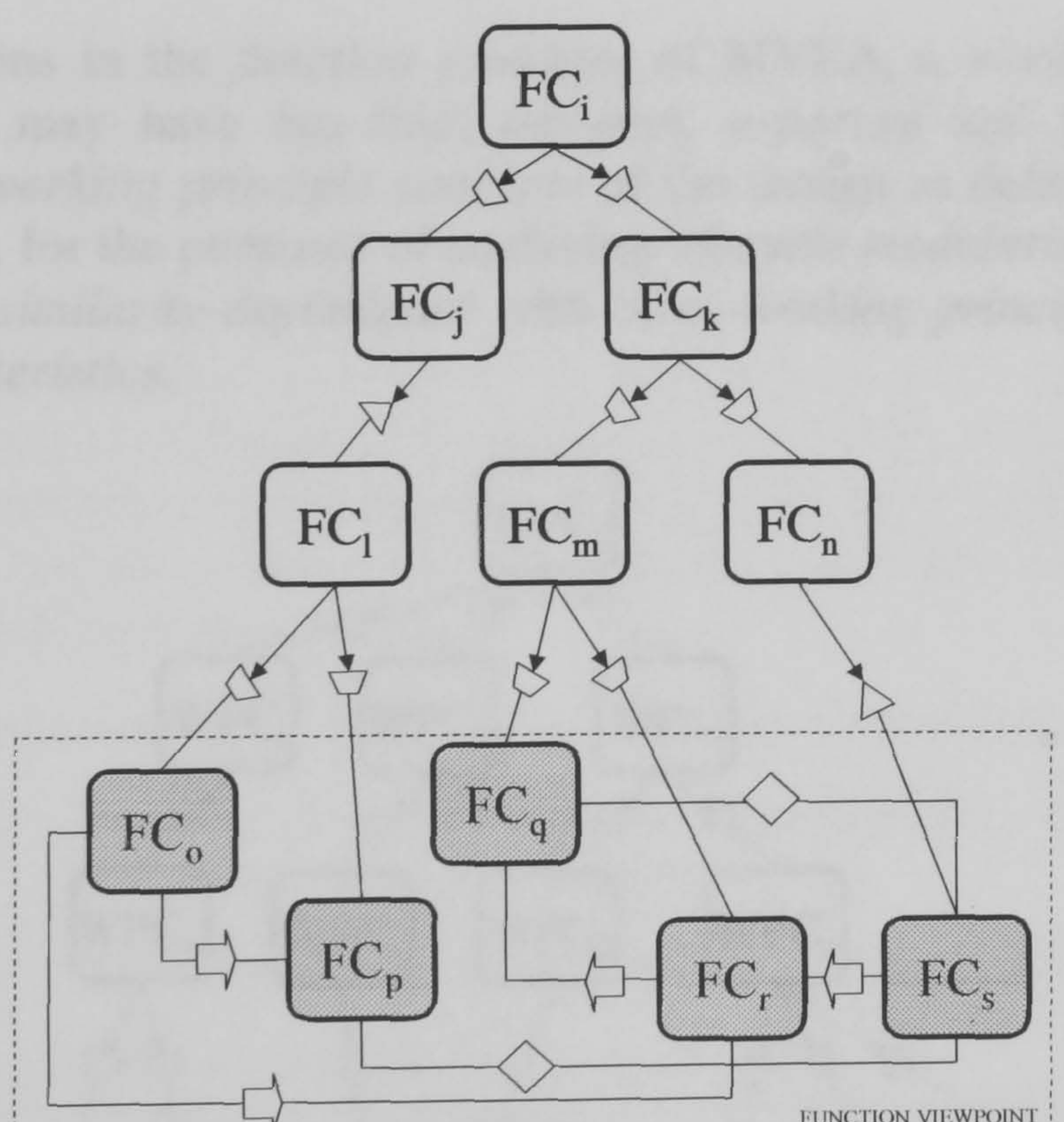


Figure 6.5: A *function viewpoint* of the extended MVEA *function structure*

Figure 6.5 depicts a *function structure* of CWK based on the MVEA (extended to include the *similarity-dependence* relation). The figure illustrates the relations between differing *function concepts* at differing levels of detail. The current most concrete and detailed *function concepts* of the design activity at a specific moment in time are represented as grey rounded rectangles. The *function viewpoint* of the Multi-Viewpoint MD Methodology consists of the bottom level *function concepts* of CWK at a discrete time in the design activity and the association *relations* between them as represented by the dashed box in Figure 6.5.

Notations:
FC_i and FC_j: two different most concrete and detailed function concepts of CWK.
nf: the number of most concrete and detailed function concepts in CWK.
and the following represents the multiplicity of relations is denoted as:
exactly one
optional (zero or one)
many (zero or more)

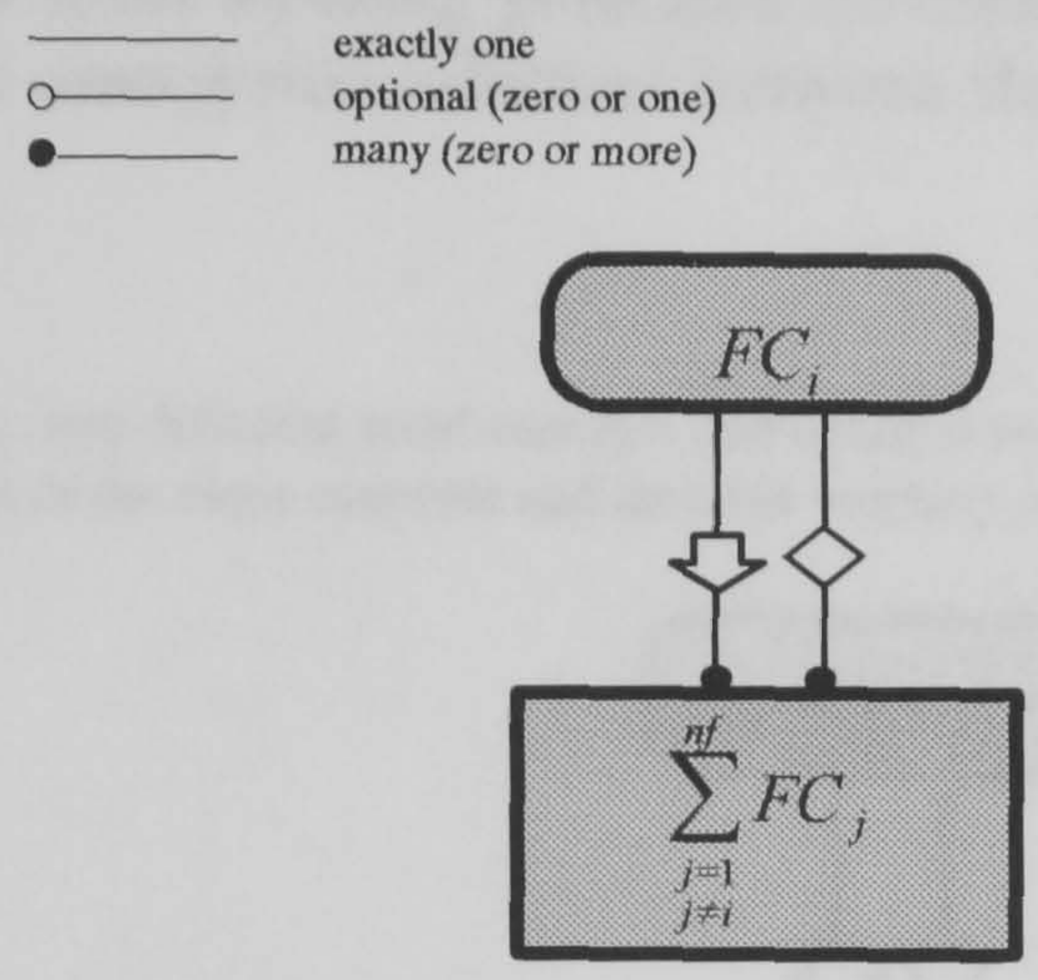


Figure 6.6: Function Viewpoint

As illustrated in Figure 6.6 a *function concept* in the *function viewpoint* can have *functional-dependency* and *similarity-dependence* relations with none or many of the other *function concepts* at this level of the *function structure.*

Working principle

Similar to the relations in the *function structure* of MVEA, a *working principle concept* (WPC) of a design may have *has-kind, has-part, a-part-of* and *functional-dependency* relations with other *working principle concepts* of the design as defined in work by Zhang (Zhang 1998). Again, for the purposes of analysing *lifecycle modularity,* a *working principle concept* may have a *similarity-dependence* with other *working principle concepts* based on their *lifecycle characteristics.*

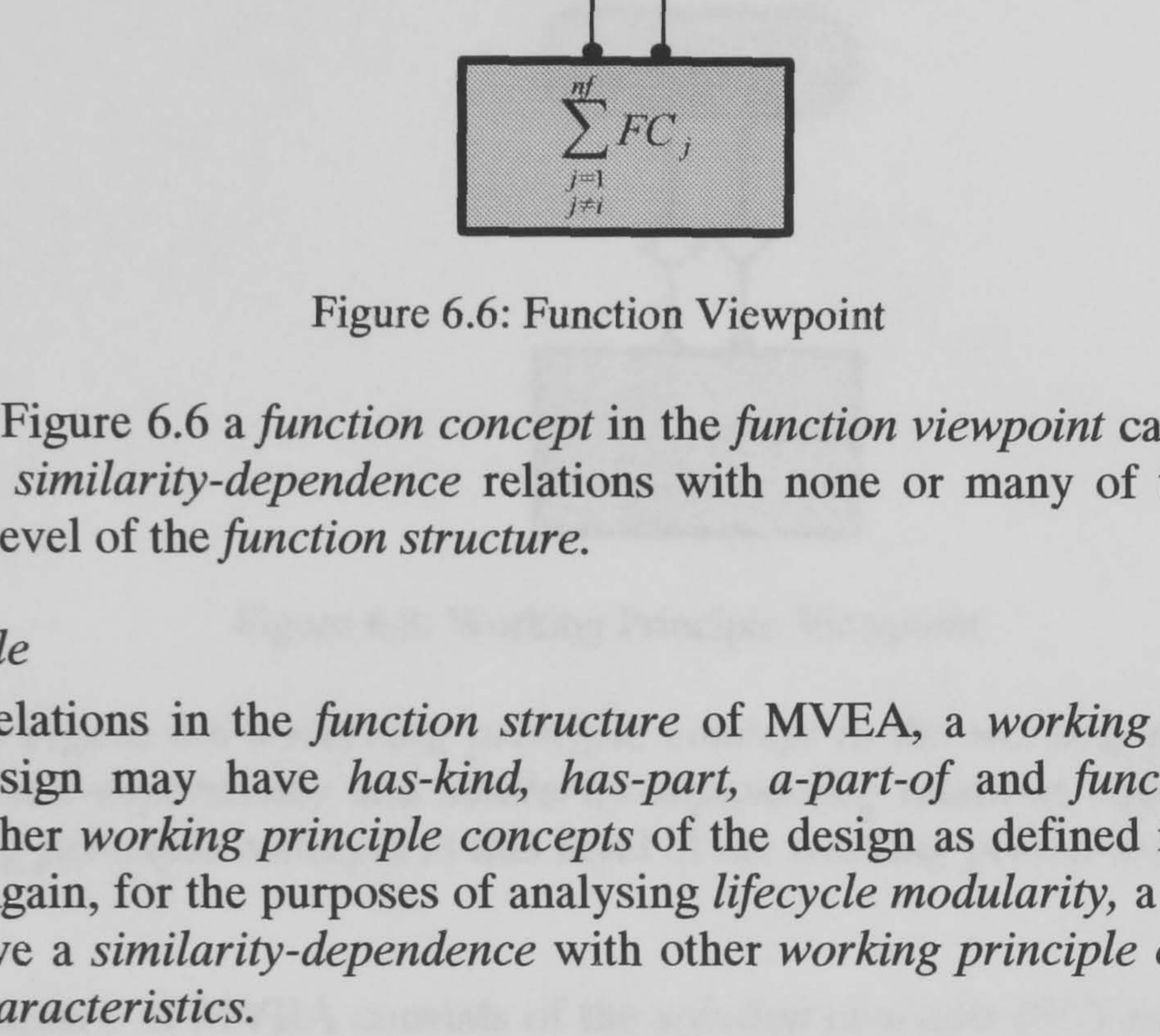


Figure 6.7: A *working principle viewpoint* of the extended MVEA *working principle structure*

Figure 6.7 depicts a *working principle structure* of CWK based on the MVEA (extended to include the *similarity-dependence* relation). The figure depicts *working principle concepts* at ever increasing levels of details. The most concrete and detailed *working principle concepts* of the design activity at a specific moment in time are depicted within grey rounded rectangles. Thus, the *working principle viewpoint* of the Multi-Viewpoint MD Methodology consists of the bottom level *working principle concepts* of CWK at a discrete time in the design activity and the association *relations* between them as represented by the dashed box in Figure 6.7.

Notations:
WPC_i and WPC_j: two different most concrete and detailed working principle concepts of CWK.
nwp: the number of the most concrete and detailed working principle concepts in CWK.

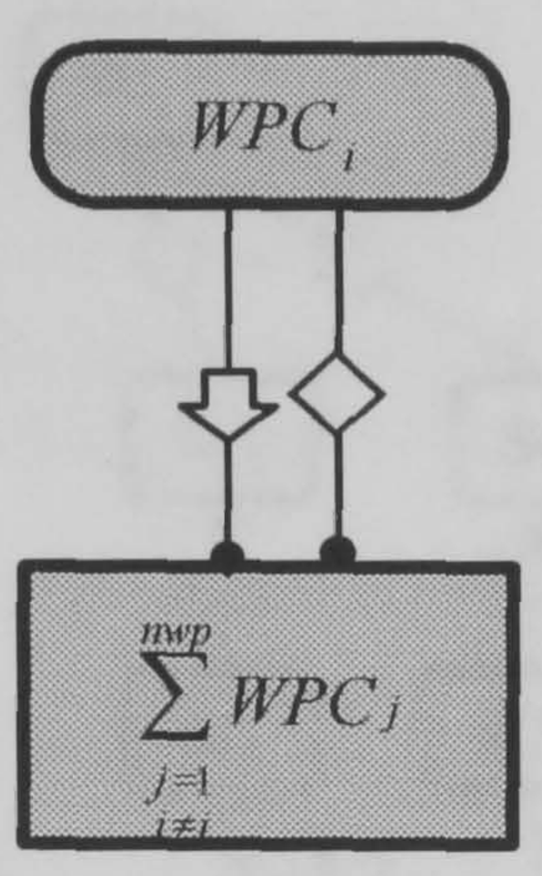


Figure 6.8: Working Principle Viewpoint

As illustrated in Figure 6.8 a *working principle concept* in the *working principle viewpoint* can have *functional-dependency* and *similarity-dependence* relations with none or many of the other *working principle concepts* at this level of the *working principle structure.*

Structure

The *solution structure* of MVEA consists of the *solution concepts* (SC) and *parts* (PART) of a design and the structural and association relations between these see Zhang (Zhang 1998). Within the *solution structure* relationships can exist between: a *solution concept* and other *solution concepts*, a *solution concept* and other *parts*, a *part* and other *solutions concepts*, and between a *part* and other *parts* where:

- *Solution concepts* of the *solution structure* may have;
  - *has-kind, has-part, a-part-of, functional- dependency, physical-link* with other *solution concepts,* and
  - *has-part, functional-dependency, physical-link* with *parts*
- *Parts* of the *solution structure* may have;
  - *a-part-of, functional-dependency, physical-link* relations with *solution concepts* of the *solution structure,* and
  - *has-kind, functional-dependency, physical-link* relations with other *parts.*

Again, for the purposes of analysing *lifecycle modularity,* a *solution concept* may have a *similarity-dependence* with other *solution concepts* and/or *parts,* and similarly a *part* may have a *similarity-dependence* with other *solution concepts* and/or *parts,* based on their *lifecycle characteristics.*

Figure 6.9 depicts a *solution structure* of CWK based on the MVEA (extended to include the *similarity-dependence* relation). The figure depicts *solution concepts* and *parts* at ever increasing levels of details. The most concrete and detailed *solution concepts* and *parts* of the design activity at a specific moment in time are depicted within grey rounded rectangles. Thus, the *structure viewpoint* of the Multi-Viewpoint MD Methodology consists of the bottom level *solution concepts* and *parts* of CWK at any discrete time in the design activity and the association *relations* between them as represented by the dashed box in Figure 6.9.

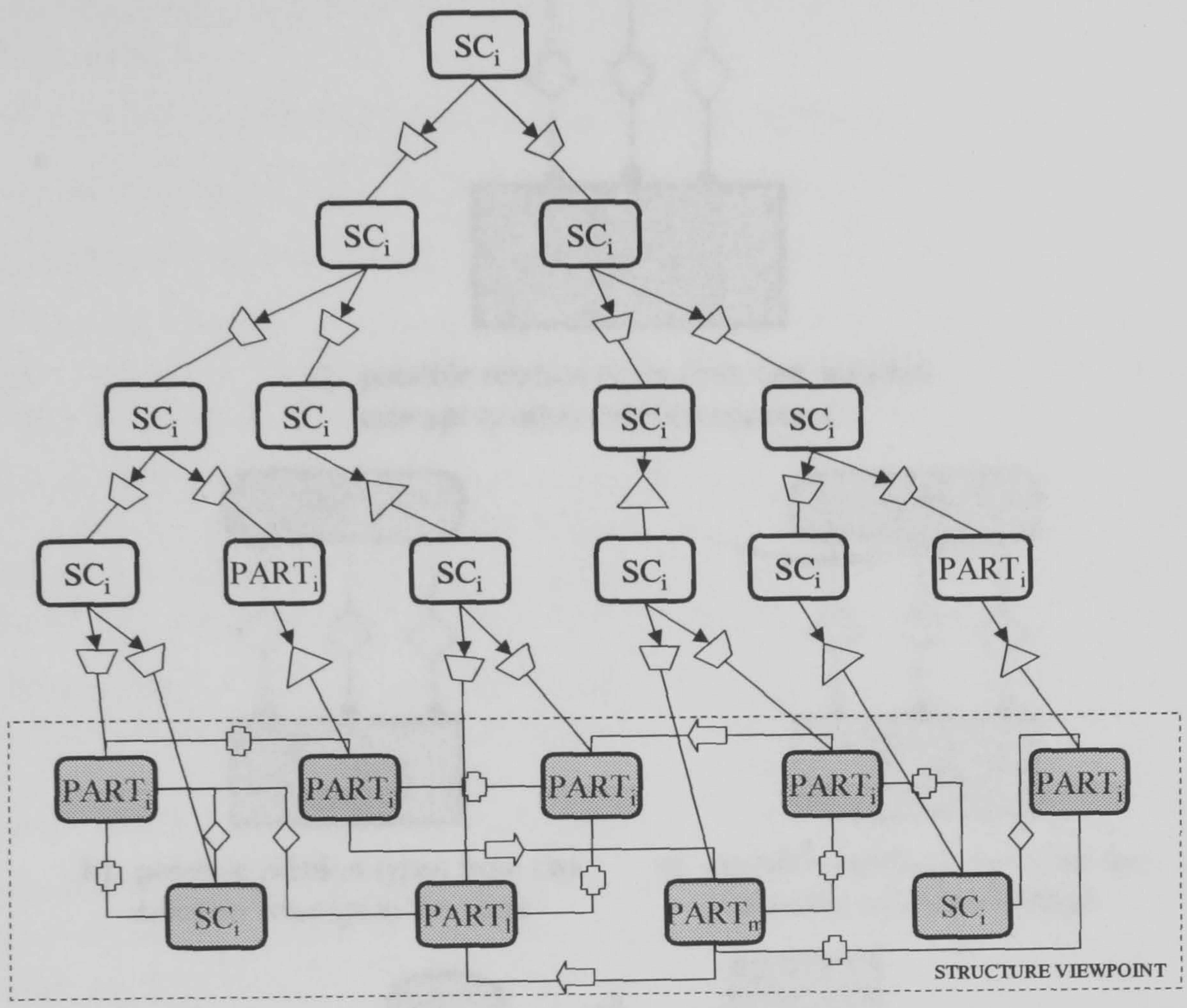
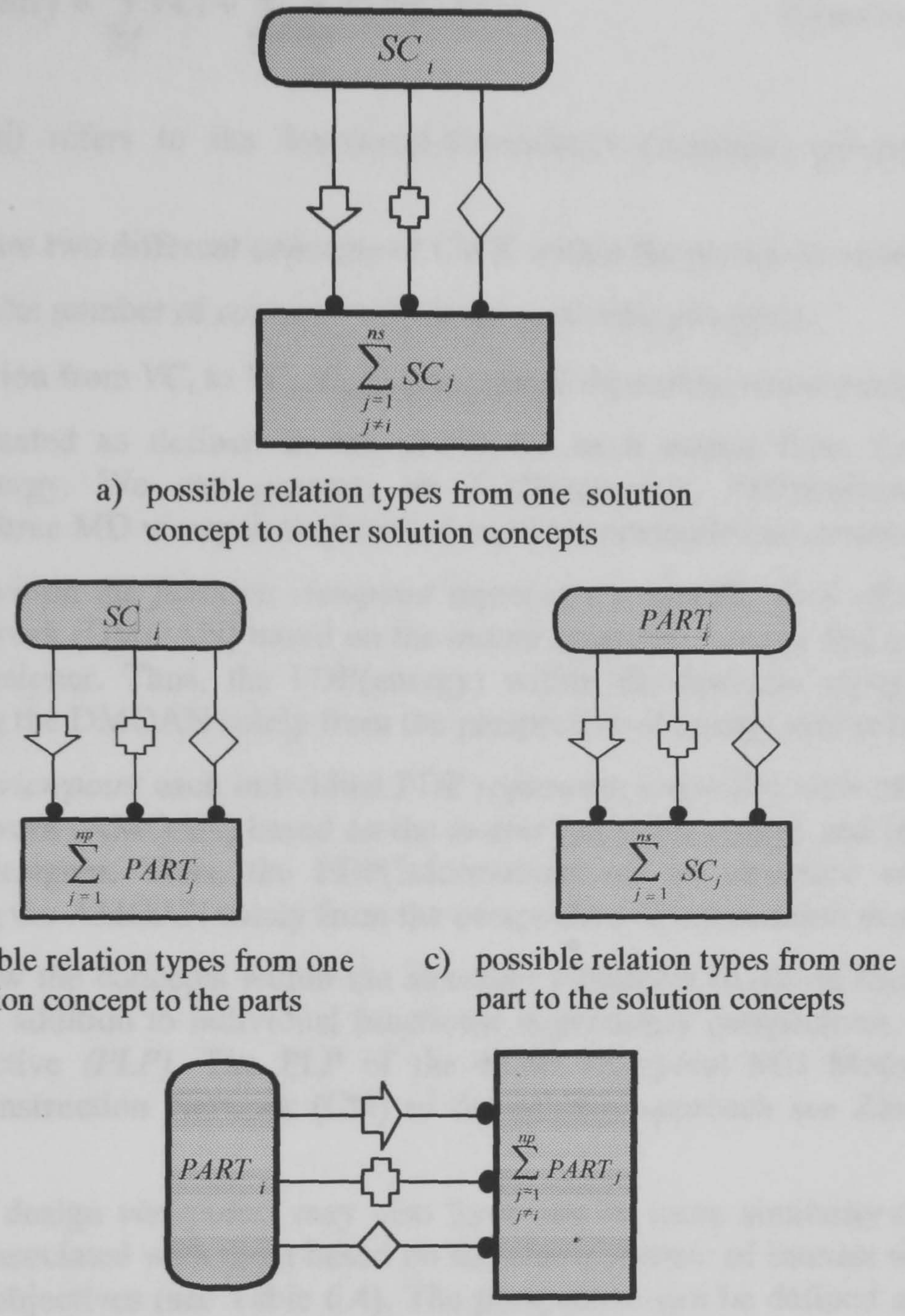


Figure 6.9: A *structure viewpoint* of the extended MVEA *solution structure*

Notations:
SC_i and SC_j: two different most concrete and detailed solution concepts of CWK.
$PART_i$ and $PART_j$: two different most concrete and detailed parts of CWK.
ns: the number of the most concrete and detailed solution concepts in CWK.
np: the number of the most concrete and detailed parts in CWK



a) possible relation types from one solution concept to other solution concepts

b) possible relation types from one solution concept to the parts

c) possible relation types from one part to the solution concepts

d) possible relation types from one part to other parts

Figure 6.10: Possible relation types between *solution concepts* and *parts* in the *structure viewpoint*

As illustrated in Figure 6.10 a *solution concept* or *part* can have *functional-dependency, physical link* and *similarity-dependence* relations with none or many of the other *solution concepts* or *parts* at this level of the *solution structure.*

Viewpoint perspectives

Each viewpoint has a number of different types of association relations that can exist between the viewpoint concepts i.e. *functional-dependency, physical-link* and *similarity-dependence*. Further, the relation types *functional-dependency* and *similarity-dependence* may be considered from different standpoints depending on the matter flow (in the case of *functional-dependency* relations) or characteristics (in the case of *similarity-dependence* relations). Thus, the viewpoint concepts and their association relations may be considered from a number of different standpoints, here termed *viewpoint perspectives*.

For example, a viewpoint from the perspective of material depicts the most concrete and detailed structure concepts and the functional-dependency (material) relations between them as denoted by Equation 6.2.

$$FDP(material) = \sum_{i=1}^{nv1} VC_i + \sum_{i=1}^{nv1} \sum_{\substack{j=1 \\ j \neq i}}^{nv1} R_{ij}(VC_i, VC_j) \qquad \textit{Equation 6.2}$$

where: *FDP(material)* refers to the functional-dependency (material) perspective of a particular viewpoint;

VC_i and VC_j are two different *concepts* of CWK within the particular *viewpoint.*

nv1 refers to the number of *concepts* within the particular *viewpoint*;

R_{ij} is the relation from VC_i to VC_j, $R_{ij} \in$ {*functional-dependency(material)null*}.

An *FDP* can be created as defined in the above for each matter flow, i.e. material, information and energy. We can generate an *FDP(material)*, *FDP(information)* and *FDP(energy)* for all three MD viewpoints (*function, working principle* and *structure*).

An individual FDP within the *function viewpoint* represents a specific view of the Desired Mode of Action Network (DMOAN) based on the *matter* (material, energy and information) of interest to the designer. Thus, the FDP(energy) within the *function viewpoint* is the equivalent of viewing the DMOAN solely from the perspective of energy matter flow.

Within the *structure viewpoint* each individual FDP represents a specific view of the Actual Mode of Action Network (AMOAN) based on the *matter* (material, energy and information) of interest to the designer. Thus, the FDP(information) of the *structure viewpoint* is equivalent to viewing the AMOAN solely from the perspective of information matter flow.

The designer can view the concepts within the *structure viewpoint* based on their *physical-link dependencies,* in addition to individual functional dependency perspectives, *i.e.* from a physical-link perspective *(PLP)*. The PLP of the Multi-Viewpoint MD Methodology is equivalent to the Construction Network (CN) of the MVEA approach see Zhang (Zhang 1998)

All three formalised design *viewpoints* may also have one or more similarity-dependence perspectives (SDP) associated with them based on the *characteristic* of interest with respect to desired life-cycle objectives (see Table 6.4). The perspective can be defined as shown in Equation 6.3.

$$SDP(characteristic) = \sum_{i=1}^{nv1} VC_i + \sum_{i=1}^{nv1} \sum_{\substack{j=1 \\ j \neq i}}^{nv1} R_{ij}(VC_i, VC_j) \qquad \textit{Equation 6.3}$$

where: *SDP(characteristic)* refers to the similarity dependence (characteristic) perspective of the particular viewpoint;

VC_i and VC_j are two different *concepts* within the particular viewpoint.

nv1 refers to the number of *concepts* within the particular viewpoint;

R_{ij} is the relation from VC_i to VC_j, $R_{ij} \in$ {*similarity dependence(characteristic) null*}.

In addition, a combined *perspective* can be created which combines the knowledge from any two or more *perspectives* of the *Viewpoint Model.* The *combined perspective* represents the most concrete and detailed viewpoint concepts and the average value these concepts dependencies across the *perspectives* considered.

6.1.2 Dependency Matrix Formalism

The formalised knowledge requires to be represented by some means to support its subsequent modular analysis. A *Dependency Matrix Formalism* is applied to the generated knowledge to create a matrix-based representation termed, the *Viewpoint Model.*

The *Dependency Matrix Formalism* utilises a Dependency Structure Matrix (DSM), also known as the Design Structure Matrix (Steward 1981). The DSM has been extensively used to represent concepts such as: tasks, resources and parameters, as well as the inter-concept dependencies. The DSM has the ability to represent most design activity relationships and it has seen considerable use in the analysis and management of the product development process (Steward 1981; Kusiak and Park 1990 ; Eppinger, Whitney et al. 1994 ; Coates, Duffy et al. 2000). More recently, the matrix concept has been applied to model various design concepts and their dependencies (Salheih and Kamrani 1999; Jarventausta and Pulkkinen 2001).

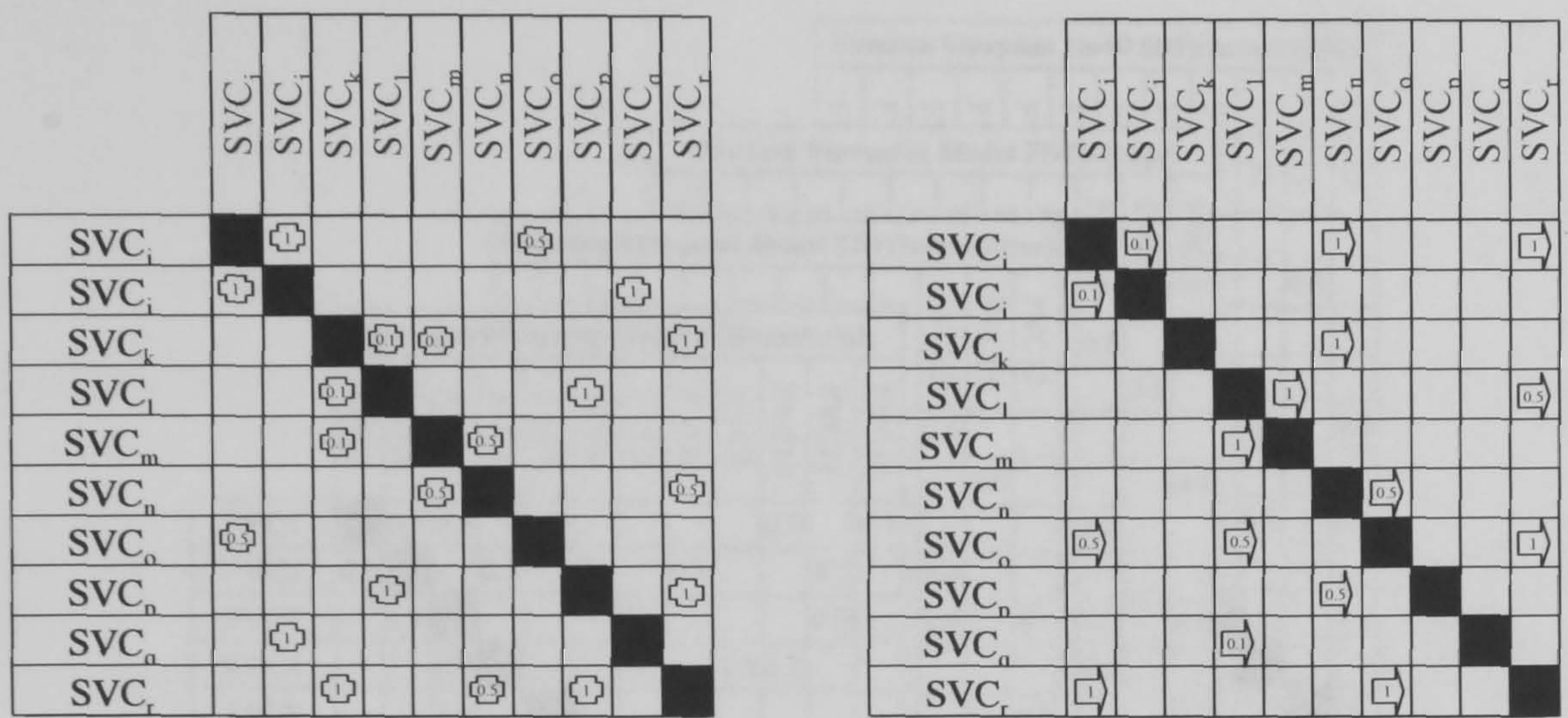
The DSM illustrated in Figure 6.11 consists of *concepts* and their *dependencies*[24]. The figure depicts viewpoint concepts i to r (VC_i, VC_j, ... VC_r) in the row and columns and their inter-dependencies. The *viewpoint concepts* are represented in both the row and column of the matrix in the same order. Within the work presented in this thesis, these *concepts* can be of the type: function or working principle or structure (solution concepts and/or part). The matrix body represents the *dependencies* between these *concepts*. Steward (Steward 1981) originally represented the dependencies in a binary form: 0 to indicate no dependency, and, 1 to indicate a dependency. However, the modelling technique has evolved to reflect a measure of the degree of dependency, termed its weight (Eppinger, Whitney et al. 1994). An entry in the matrix body in Figure 6.11 depicts both the existence of a *dependency* between the two *concepts* (in the row and column) and the weight of that *dependency* (where 0.1 denotes a low degree of dependency and 1 the highest degree of dependency). In this case no entry in a row column indicates that there is no dependency. The black squares in Figure 6.11 are termed the 'leading line' and depict the fact that a *dependency* relation (including no dependency = 0) cannot exist at the row and column intersection of the same *concept*. i.e. $VC_{i(row)}$ cannot have a dependency with $VC_{i(column)}$.

	VC_i	VC_j	VC_k	VC_l	VC_m	VC_n	VC_o	VC_p	VC_q	VC_r
VC_i	■	1					0.5			
VC_j	1	■							1	
VC_k			■	0.1	0.1					1
VC_l			0.1	■				1		
VC_m			0.1		■	0.5				
VC_n					0.5	■				0.5
VC_o	0.5						■			
VC_p				1				■		1
VC_q		1							■	
VC_r			1			0.5		1		■

Figure 6.11: The Dependency Structure Matrix concept

[24] The concepts and dependencies utilised to generate matrices throughout chapter 6 are fabricated examples that have been developed to illustrate details of the methodology. They do not represent real-life examples of design knowledge.

The dependencies shown in the example in Figure 6.11 are symmetrical about the leading line, for example *dependency* (VC_i, VC_j)[25] = *dependency* (VC_j, VC_i). Thus, in the example shown in Figure 6.11 a *dependency* (VC_n, VC_{n+1}) = *dependency* (VC_n, VC_{n+1}). However this is not always the case, for example in the association dependency formalisms provided above it has been illustrated (Section 6.1.1) that a *functional-dependency* (VC_n, VC_{n+1}) ≠ *functional-dependency* (VC_{n+1}, VC_n) as the *functional-dependency* relation is directional. The examples provided in Figure 6.12 illustrate the matrix representation of symmetrical and non-symmetrical dependencies.



a) The dependency structure matrix of concepts SVC_i to SVC_r and their *physical-link* relations.

b) The dependency structure matrix of concepts SVC_i to SVC_r and their *functional-dependency* relations.

Figure 6.12: Perspectives of the Dependency Structure Matrix

Figure 6.12a depicts structural viewpoint concepts i to r (SVC_i, SVC_j, ... SVC_r) and their *physical-link relations* (depicted by the cross notation) whilst Figure 6.12b depicts the same viewpoint concepts and their *functional-dependencies* (depicted by the block arrow notation). By nature *physical-link* relations are symmetrical and as shown in Figure 6.12a in the matrix representation the dependencies are mirrored by the leading line. Thus, a *physical-link* relation will have the same entry in row and column (VC_n, VC_{n+1}) and (VC_{n+1}, VC_n) respectively. However, in Figure 6.12b the direction of the *functional-dependency* is dependant on its position in the matrix. For example, take the *concepts* SVCi and SVC_n, which are subject to the *functional-dependency* (SVC_i, SVC_n), denoted by the entry of a 1 in the row SVC_i and column SVC_n intersection, but not the *functional-dependency* (SVC_n, SVC_i) denoted by the lack of entry in the row SVC_n and column SVC_i intersection.

It can be seen in Figure 6.12b that a *functional-dependency* may also be bi-directional, as in the case of *functional-dependency* (SVC_i, SVC_j) = *functional-dependency* (SVC_j, SVC_i), this is depicted by the same entry in row and column (SVC_i, SVC_j) and (SVC_j, SVC_i) respectively, as in the case of *physical-link* relations.

Figure 6.12 a and b depict the same *viewpoint concepts* from different *perspectives* of that viewpoint i.e. from the perspectives of physical connection (Figure 6.12a) and functional dependence (Figure 6.12b). As discussed in Section 3.2.2, what is modular from one *viewpoint* may not be modular for another and similarly what is modular from one *perspective* of that *viewpoint* may not be modular for another. The key for a modular design practitioner is to determine the optimum modularity with respect to one or a number of

[25] The notation for a dependency between two matrix concepts is: *dependency (row, column)*. The *concept* in the row always preceding that of the column with the direction of the dependency from row to column.

perspectives across different viewpoints of design. Thus, the dependency matrix formalism has been developed to support the representation of design *viewpoints* and *perspectives* of these as *Viewpoint Models*. The following discusses the generation of *perspectives* of these *Viewpoint Models.*

Perspectives of Viewpoint Models

Figure 6.13 illustrates the dependency matrix formalism for *viewpoint perspectives*. The figure depicts the function viewpoint matrix from the *perspective* of *functional-dependency(material)*.

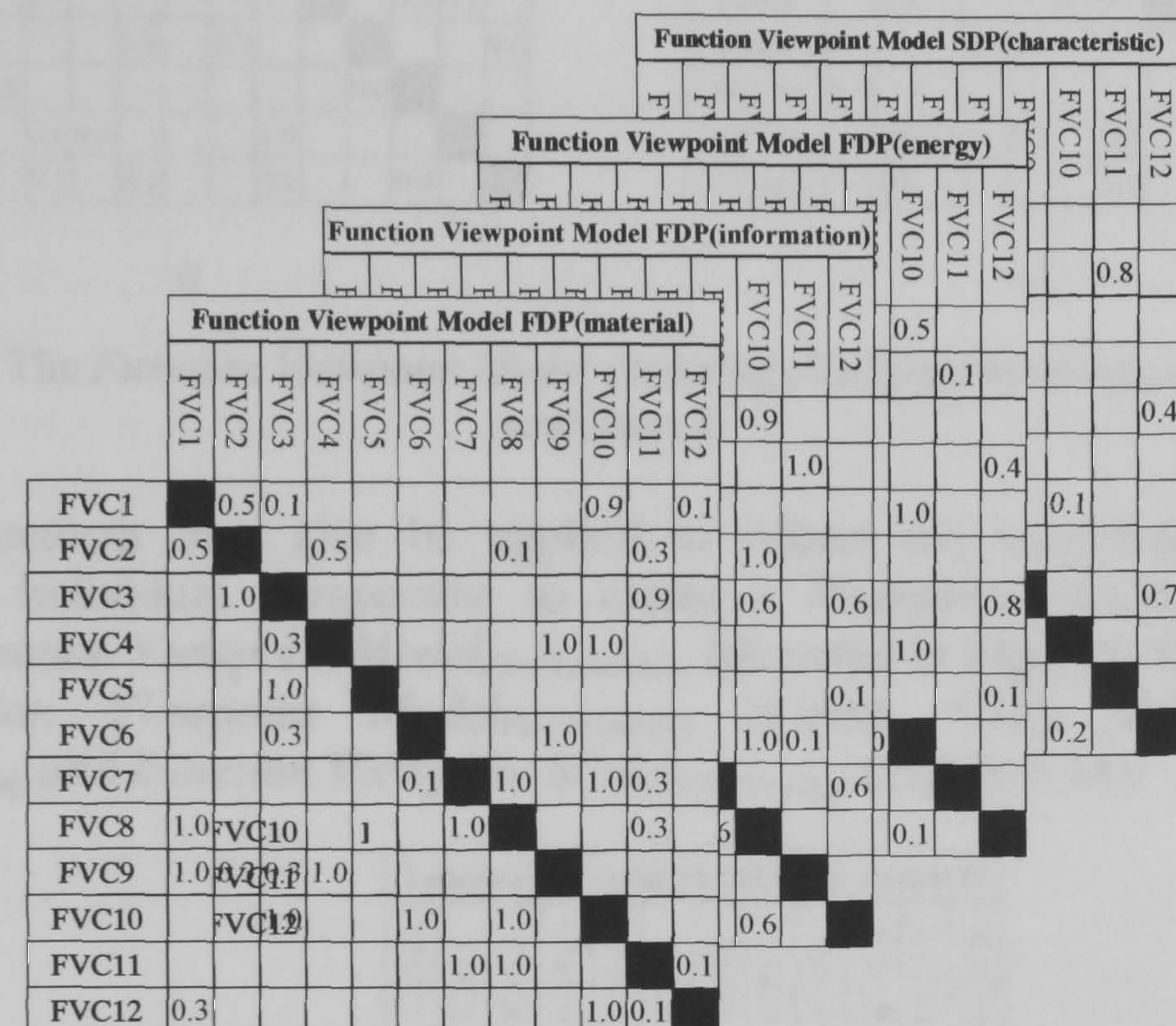


Figure 6.13: The *FDP(material)* perspective of the *Function Viewpoint Model*

The matrices in the background depict potential alternative perspectives that may be generated for the *function viewpoint*: *functional-dependency(information), functional dependency(energy,), similarity-dependence (characteristic)*. The entries in the matrix body depict the weight of the dependencies between *concepts*.

The matrix formalism can be applied to any individual *perspective* of a *viewpoint* to create a *Viewpoint Model*$_{perspective}$. Figure 6.14 depicts two alternative *perspectives* (*FDP(information)* and *FDP(energy)*) of the *Function Viewpoint Model* to that shown in Figure 6.13. *FDP(information)* is shown in Figure 6.14a and *FDP(energy)* in Figure 6.14b. Both *perspectives* depict the same viewpoint concepts (in the same order) but each takes an alternative 'standpoint' of the *dependencies* that exist between them.

Function Viewpoint Model FDP(information)	FVC1	FVC2	FVC3	FVC4	FVC5	FVC6	FVC7	FVC8	FVC9	FVC10	FVC11	FVC12
FVC1		0.6				1.0				0.9		
FVC2	0.6										1.0	
FVC3	1.0							0.5				
FVC4								0.5		1.0		
FVC5						1.0		0.6		0.6		0.6
FVC6	1.0		0.6		1.0		1.0					
FVC7						1.0						0.1
FVC8		0.1								1.0	0.1	
FVC9				0.6		1.0						0.6
FVC10	0.6								0.6			
FVC11		0.6	0.6				1.0					
FVC12		0.1		0.1			0.1			0.6		

a)

Function Viewpoint Model FDP(energy)	FVC1	FVC2	FVC3	FVC4	FVC5	FVC6	FVC7	FVC8	FVC9	FVC10	FVC11	FVC12
FVC1		0.7								0.5		
FVC2	0.7										0.1	
FVC3	0.1	0.9						0.9				
FVC4								0.2				0.4
FVC5		0.1				1.0		0.1		1.0		
FVC6	0.1		0.5		1.0		1.0					
FVC7						0.1						0.8
FVC8		0.1				0.1				1.0		
FVC9				0.1								0.1
FVC10	0.5								1.0			
FVC11		1.0	0.1		0.1		0.4					
FVC12		1.0					0.1			0.1		

b)

Figure 6.14: The *Function Viewpoint Model* depicting *FDP(information)*and *FDP(energy)* respectively

The matrix formalism may also be applied to reflect the knowledge related to any combination of individual *perspective* to create a *Viewpoint Model*$_{combined\text{-}perspective}$, for example, the *Function Viewpoint Model*$_{FDP(combined)}$ (depicted in Figure 6.15) is a combination of the *Function Viewpoint Model*$_{FDP(material)}$ (Figure 6.13), *Function Viewpoint Model*$_{FDP(information)}$ *and Function Viewpoint Model*$_{FDP(energy)}$ (Figure 6.14).

Function Viewpoint Model FDP(combined)	FVC1	FVC2	FVC3	FVC4	FVC5	FVC6	FVC7	FVC8	FVC9	FVC10	FVC11	FVC12
FVC1		0.6	0.03			0.33				0.77		0.03
FVC2	0.6			0.17				0.03			0.47	
FVC3	0.37	0.63						0.47			0.3	
FVC4			0.1					0.23	0.33	0.67		0.13
FVC5		0.03	0.33			0.67		0.23		0.53		0.2
FVC6	0.37		0.47		0.67		0.67		0.33			
FVC7						0.4		0.33		0.33	0.1	0.3
FVC8	0.33	0.07				0.03	0.33			0.67	0.13	
FVC9	0.33	0.1	0.1	0.57		0.33						0.23
FVC10	0.37		0.33			0.33		0.33	0.53			
FVC11		0.53	0.23		0.03		0.8	0.33				0.03
FVC12	0.1	0.37		0.03			0.07			0.57	0.03	

Figure 6.15: The *Function Viewpoint Model* depicting the combined perspective *FDP*$_{combined}$ *(material, information and energy)*

Figure 6.15 *Function Viewpoint Model*$_{FDP(combined)}$ depicts the most detailed and concrete *function concepts* and the average of the *functional-dependency* relations' *material, energy* and *information*. For example, if we take the *functional-dependency* (FVC_3, FVC_2) in the perspective *FDP(material)* it is dependent with a weight of 1 (see Figure 6.13), in *FDP(information)* it is independent (see Figure 6.14a), and in *FDP(energy)* it is dependent with a weight of 0.9 (see Figure 6.14b), thus the average dependency weight is 0.63, which is the value of the dependency shown *FDP(combined)*(FVC_3, FVC_2) as shown in Figure 6.15.

The *concepts* are viewpoint and not perspective specific, i.e. the definition of *concepts* and the sequence in which they are depicted requires to be maintained across all *perspectives* of the individual *Viewpoint Model*. On the other hand, dependencies are *perspective* specific and an alteration of a *dependency* (either its status or weight) in any *individual perspective* does not affect any other *individual perspective.* However, a change to the *dependencies* in an individual *perspective* requires to be propagated to any *combined perspective* of which the altered *perspective* is part.

6.2 Optimisation Mechanism

The *Optimisation Mechanism* consists of two parts: a Genetic Algorithm (GA) and A Clustering Criterion (CC), with the output being Optimised Viewpoint Models. The GA is described in Section 6.2.1 whilst Section 6.2.2 defines the CC.

6.2.1 Genetic Algorithm

Theoretically it is possible for the designer to manually re-order the matrix concepts and thus potentially determine an optimum concept sequence. However, for a relatively small matrix, consisting of 20 concepts, there are over 24 x 10^{17} possible sequences (i.e. 20!). That is the number of possible sequences of concepts is the factorial of the number of concepts, i.e. n! (where n is the number of concepts).

As discussed previously (Section 4.1.5) manual optimisation may not always be an efficient, effective or practical option. Thus, the application of a GA is proposed to enable the exploration of optimal modular structures. The proposed application procedure for the genetic algorithm application to optimising concept structures is illustrated within Figure 6.16.

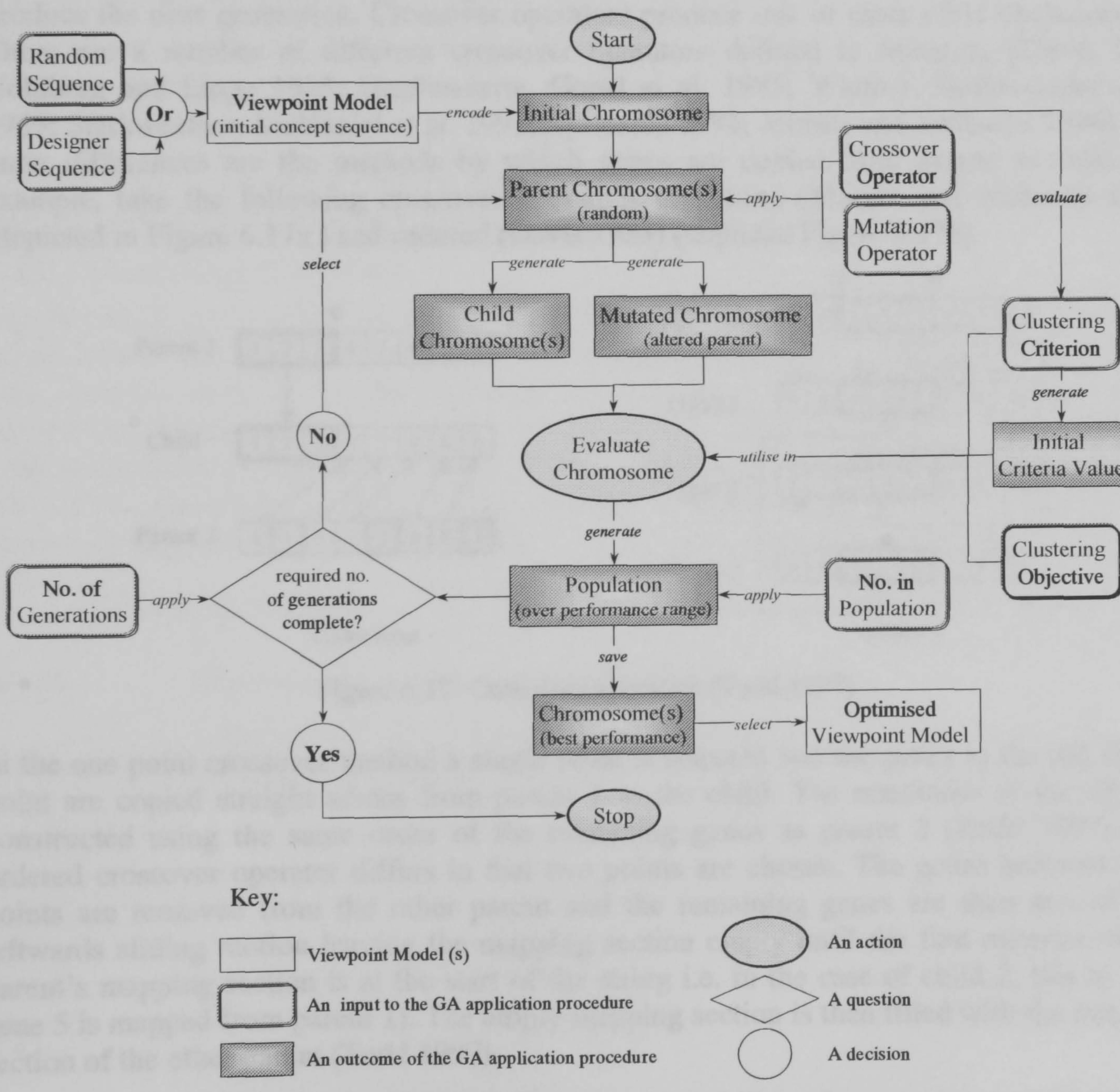
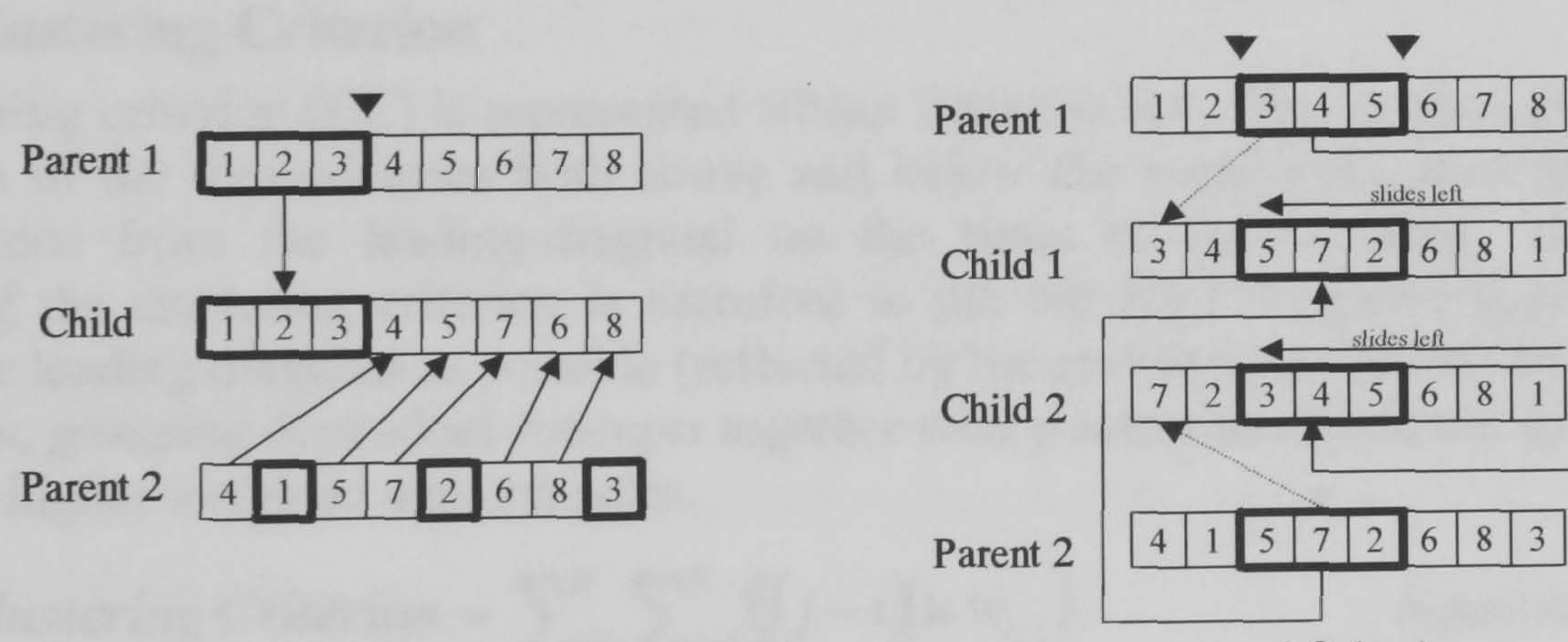


Figure 6.16: The proposed application of a GA for optimisation of concept groups.

This procedure is based on the general structure for a genetic algorithm developed by Goldberg (Goldberg 1989) (see also Chapter 7). As illustrated in Figure 6.16 the start point for the genetic algorithm application is an initial chromosome. In the case of the MD problem, the chromosome represents the order (sequence) of *concepts* in a Viewpoint Model. Thus, the initial chromosome is encoded based on the order of *concepts* in the Viewpoint Model of interest. Depending on the specific objective of the application the initial order may be random or represent a previously defined sequence developed by the designer. Randomisation attempts to ensure that the chromosome represents a unique point in the solution space, such that the group of chromosomes are randomly distributed throughout. The designer defined sequence on the other hand may represent an existing sequence and be utilised to assess the potential for improvement of that design. The initial chromosome is evaluated with respect to the clustering criteria to gain an initial value.

The population size can be set and a roulette-wheel type selection procedure is used to select chromosome candidates for crossover and mutation to produce the next generation. A portion of roulette wheel is given to each chromosome that is proportional to its performance (Goldberg 1989). Thus, chromosomes with higher performance characteristics have a greater chance of surviving the selection procedure, although it is possible for lower performance chromosomes to be passed through to the next generation.

Two parent chromosomes are selected at random and removed from the population. Crossover and mutation operations are then performed upon the selected chromosomes to produce the next generation. Crossover operators produce one or more child chromosomes. There are a number of different crossover operators defined in literature (Davis 1985; Goldberg and Linge 1985; Greffenstette, Gopal et al. 1985; Whitley, Starkweather et al. 1989; Starkweather, McDaniel et al. 1991; Syswerda 1991; Murata and Ishibuchi 1994). The main differences are the methods by which genes are copied from parent to child. For example, take the following crossover operators, one point (Murata and Ishibuchi 1994) (depicted in Figure 6.17a) and ordered (Davis 1985) (depicted Figure 6.17b).



a) One Point a) Ordered

Figure 6.17: Crossover operators (Todd 1997)

In the one point crossover method a single point is selected and the genes to the left of this point are copied straight across from parent 1 to the child. The remainder of the child is constructed using the same order of the remaining genes as parent 2 (Todd 1997). The ordered crossover operator differs in that two points are chosen. The genes between these points are removed from the other parent and the remaining genes are then moved in a leftwards sliding motion leaving the mapping section empty until the first member of this parent's mapping section is at the start of the string i.e. in the case of child 2, this is 7 (as gene 5 is mapped from parent 1). The empty mapping section is then filled with the mapping section of the other parent (Todd 1997).

The two parents are then crossed based upon a probability of crossover to produce two children containing genetic information from both parents. Mutation operators work in a similar manner on a single chromosome to produce a small change in the parent. A number of Mutation operators have been previously defined in literature (Murata and Ishibuchi 1994). The main difference between these is the method by which the parent genes mutate to produce a child, as can be seen from the mutation operators 'two-point adjacent swap' and 'two-point random swap' depicted in Figure 6.18.

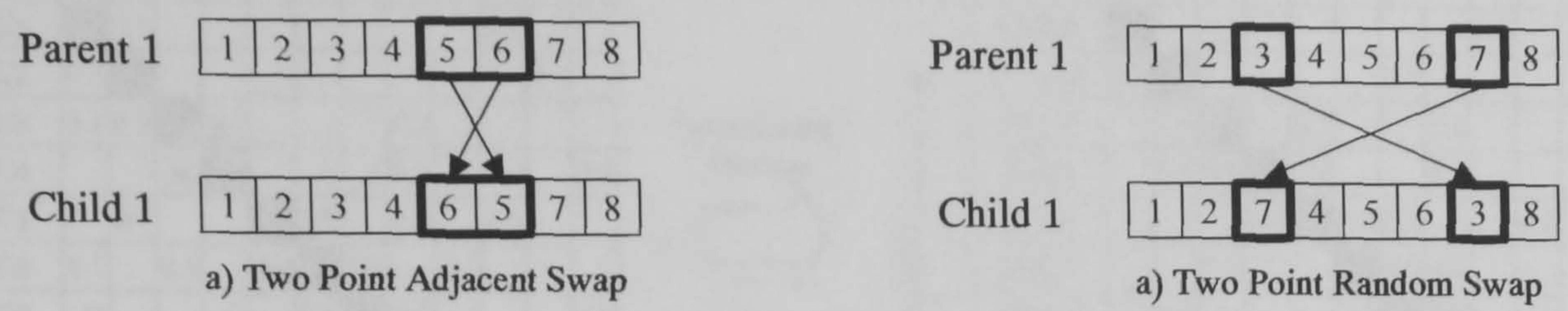


Figure 6.18: Mutation operators (Todd 1997)

The new population is then re-evaluated with respect to the performance criteria (clustering criteria). A check is made to determine whether the GA has completed a pre-determined number of generations, finishing if it has, and otherwise repeating the selection, 'crossover and mutation' and evaluation processes. The best performing chromosomes are saved as potential optimum candidates, i.e. those with the minimum value for the clustering criteria are saved. On completion of the predefined number of generations the designer can select one of the optimised chromosomes. The order of the *concepts* in this chromosome provides the *concept* sequence for the Optimised Viewpoint Model.

GA parameters have been shown to be intrinsically link to the domain (see Section 4.1.5). Thus for the DSM application domain the optimum parameters for the GA structure require to be determined.

The optimisation procedure may be applied to optimise any of the defined *perspectives* of the *Viewpoint Model* including *combined perspectives.*

6.2.2 Clustering Criterion

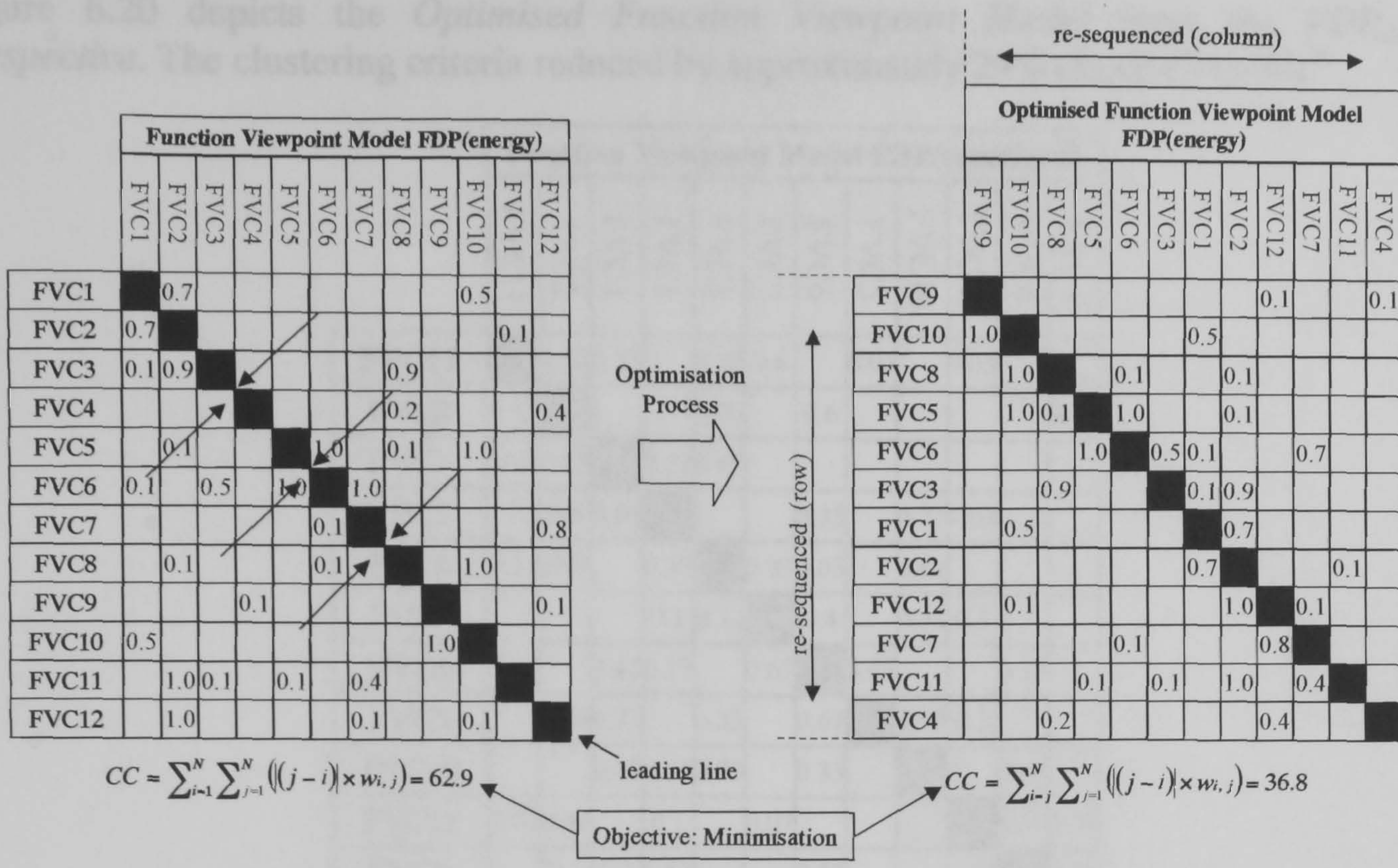
The clustering criterion (CC) is represented within Equation 6.4. The equation represents the summation of the dependencies both above and below the leading-diagonal multiplied by their distance from the leading-diagonal on the basis of their weight. The focus of minimising the clustering criterion is therefore to get the most weighted dependencies as close to the leading diagonal as possible (reflected by the arrows in the matrix body in Figure 6.19). Thus, grouping dependent *concepts* together with priority automatically given towards those with higher weighted dependencies.

$$Clustering\,Criterion = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(|(j-i)| \times w_{i,j}\right) \qquad \textit{Equation 6.4}$$

Where: *N* is the number of concepts in the DSM,

i and *j* are the row and column indices, and

$w_{i,j}$ are the dependency weights.



a) pre-optimisation b) post-optimisation

Figure 6.19: Minimising the Clustering Criterion

Figure 6.19 depicts the *Function Viewpoint Model FDP$_{energy}$* (see Figure 6.14 b) both pre and post optimisation. The figure illustrates that the GA re-sequences the *concept* order with the objective of minimising the clustering criteria. That is, the order from FVC_1, FVC_2, ..., FVC_{12} is changed to FVC_9, FVC_{10}, FVC_5, ... and so forth. A dependency between two *concepts* cannot be violated and must be maintained regardless of the alterations in concept sequence. For example, if we take *functional-dependency$_{energy}$* (FCV_1, FVC_2) we can see that the dependency is maintained in the optimised model despite the alteration of the *concept* sequence. As the *concept* order remains the same in both row and order the leading line also remains intact. The re-sequenced concept order shown in *Optimised Function Viewpoint Model FDP$_{energy}$* (Figure 6.19b) has had the effect of reducing the clustering criteria by 42% (from 62.9 for Figure 6.19a to 36.8 for Figure 6.19b) and it can be seen that the dependencies are clustered closer to the leading line.

The CC value requires to be re-calculated when new *concepts* and/or *concept dependencies* are added to a matrix.

Figure 6.20 depicts the *Optimised Function Viewpoint Model* from the *$FDP_{combined}$ perspective*. The clustering criteria reduced by approximately 29% (from 83 to 59)[26].

Function Viewpoint Model FDP(combined)	FVC11	FVC2	FVC3	FVC1	FVC8	FVC7	FVC6	FVC5	FVC10	FVC12	FVC9	FVC4
FVC11		0.53	0.23		0.33	0.8		0.03		0.03		
FVC2	0.47				0.03		0.6					0.17
FVC3	0.3	0.63		0.37	0.47							
FVC1		0.6	0.03				0.33		0.77	0.03		
FVC8	0.13	0.07		0.33		0.33	0.03		0.67			
FVC7				0.1	0.33		0.4		0.33	0.3		
FVC6			0.47	0.37		0.67		0.67			0.33	
FVC5		0.03	0.33		0.23		0.67		0.53	0.2		
FVC10			0.33	0.37	0.33		0.33				0.53	
FVC12	0.03	0.37		0.1		0.07					0.57	0.03
FVC9		0.1	0.1	0.33			0.33			0.23		0.57
FVC4			0.1		0.23				0.67	0.13	0.33	

Figure 6.20: Optimised Function Viewpoint Model $FDP_{combined}$.

As the *concept* order is *viewpoint* specific, the new *concept* order depicted in Figure 6.20 is propagated to alternative *perspectives* of the *Function Viewpoint Model.* Thus, the impact of the optimisation of one *perspective* (or a *combined perspective*) on others can be ascertained. For example, Figure 6.21 depicts the effect of the *concept order* returned from optimisation of the *Function Viewpoint Model $FDP_{combined}$* on the *Function Viewpoint Model FDP_{energy}*. As can be seen from Figure 6.21 the *FDP_{energy} perspective* has the same concept sequence as that of the optimised *$FDP_{combined}$ perspective.*

The effect of the new sequence has been to increase the clustering criteria *FDP_{energy} perspective* from when it was optimised individually (see Figure 6.19b) by approximately 16% to 43. However the criteria still shows a reduction of 33% from the original Viewpoint Model (see Figure 6.19a)

26 This reduction is based on the application of the clustering criterion (Equation 6.4) to *Function Viewpoint Model $FDP_{combined}$* (shown in Figure 6.15) and *Optimised Function Viewpoint Model $FDP_{combined}$* (shown in Figure 6.20) respectively.

	Function Viewpoint Model FDP(energy)											
	FVC11	FVC2	FVC3	FVC1	FVC8	FVC7	FVC6	FVC5	FVC10	FVC12	FVC9	FVC4
FVC11		1.0	0.1			0.4		0.1				
FVC2	0.1			0.7								
FVC3		0.9		0.1	0.9							
FVC1		0.7							0.5			
FVC8		0.1					0.1		1.0			
FVC7							0.1			0.8		
FVC6			0.5	0.1		1.0		1.0				
FVC5		0.1			0.1		1.0		1.0			
FVC10				0.5							1.0	
FVC12		1.0				0.1			0.1			
FVC9										0.1		0.1
FVC4					0.2					0.4		

Figure 6.21: Optimised Function Viewpoint Model $FDP_{combined}$ effect on the Function Viewpoint Model FDP_{energy}

6.3 Module Identification Mechanism

The *Module Identification Mechanism* was developed to address the difficulties associated with identifying modules as discussed in Section 4.1.5. The *Modular Identification Mechanism* consists of two main parts: the *Module Strength Indicator (MSI)* function and a *Modular Structure Matrix (MSM).*

6.3.1 Module Strength Indicator function

The MSI function was derived to support the determination of the inherent modularity within the product structure. The MSI function is applied to the Optimised Viewpoint Model.

MSI function

The MSI function consists of two parts, Equation 6.5 and Equation 6.6 below. Equation 6.5 provides the designer with an indication of the mean value of the dependencies within a concept grouping for the current sequence. This value depicts the strength of the internal dependencies of a concept grouping. The strength of the internal dependencies is defined as the actual weight of the dependencies (dividend, top part of Equation 6.5) divided by the maximum number of dependencies that may exist within a concept grouping (the divisor, bottom part of Equation 6.5). The maximum value of dependencies that may exist within a *concept* grouping, can be defined as the number of potential dependencies between the *concepts* in the grouping multiplied by the maximum weight value of the dependencies. The maximum number of dependencies within any *concept* grouping is the sum of the available row and column intersections, i.e. squares in the matrix excluding the leading line. Thus, the maximum value of dependencies, of any potential grouping presented in this work, is the sum of the available row and column intersections within the concept grouping multiplied by 1.0 (the maximum value of a dependency).

Equation 6.6 determines a mean value for the external dependencies out with the *concept* grouping. The value represents the actual value of the dependencies out with the *concept* grouping (the dividend(s) in Equation 6.6) divided by the potential maximum value of dependencies out with the grouping (the divisor(s) in Equation 6.6).

The focus is towards identifying concept clusters that have a maximum number of internal dependencies and a minimum number of external dependencies, i.e. concept groupings of high modular value. Therefore subtracting Equation 6.6 from Equation 6.5 can derive the relative modularity of the clustered sequence, with respect to its concepts' internal and external dependencies. The MSI function, Equation 6.7, therefore provides a modularity metric directly related to the overall modularity characteristics of the design artefact.

$$MSI_i = \frac{\sum_{i=n_1}^{n_2}\sum_{j=n_1}^{n_2} w_{i,j}}{(n_2 - n_1)^2 - (n_2 - n_1)} \qquad \text{Equation 6.5}$$

$$MSI_e = \left(\frac{\sum_{i=0}^{n_1}\sum_{j=n_1}^{n_2} w_{i,j} + w_{j,i}}{2\times(n_1\times(n_2 - n_1))}\right)^a + \left(\frac{\sum_{i=n_2}^{N}\sum_{j=n_1}^{n_2} w_{i,j} + w_{j,i}}{2\times((N - n_2)\times(n_2 - n_1))}\right)^b \qquad \text{Equation 6.6}$$

$$MSI = MSI_i - MSI_e \qquad \text{Equation 6.7}$$

Where: n_1 = index of start of the *concept* grouping, and

n_2 = index of end of the *concept* grouping, where $MSI_e(\,)^b = 0$, when $n_2 = N$

N = no. of concepts in the matrix

Given that the maximum dependency weight that can be assigned within the DSM is 1.0, the maximum MSI value that can be returned from Equation 6.7 is 1.0. This represents the strongest possible module solution, i.e. a module consisting entirely of maximum weight internal dependencies (1.0 from Equation 6.5), with no related external dependencies (0.0 from Equation 6.6)

MSI function application

The *MSI* function is applied to every possible concept grouping within the DSM matrix. The resulting values are represented in the Modular Structure Matrix (see Section 6.3.2).

MSI application to the Viewpoint Model

In general, an Optimised Viewpoint Model is the expected basis for *MSI* application. The *MSI* is applied to all possible *concept* groupings within a Viewpoint Model to determine the inherent modularity in the overall structure. Within this work, a *concept* grouping is defined as two or more *concepts* clustered together around the leading line. The application of the *MSI* function provides the designer with a value for the modularity within each grouping. Applied across all the groupings the designer can gain an overview of the inherent modularity and potential module candidates within the model. This information provides a basis on which to facilitate modular design.

Based on the hypothesis that any grouping of two or more *concepts* may be a potential module the *MSI* application process traverses the matrix and calculates a value for all possible *concept* groupings as depicted in Figure 6.22. The figure depicts the process for *MSI* function application to an entire *Viewpoint Model*. For clarity and simplicity of explanation the model is shown without *dependencies*.

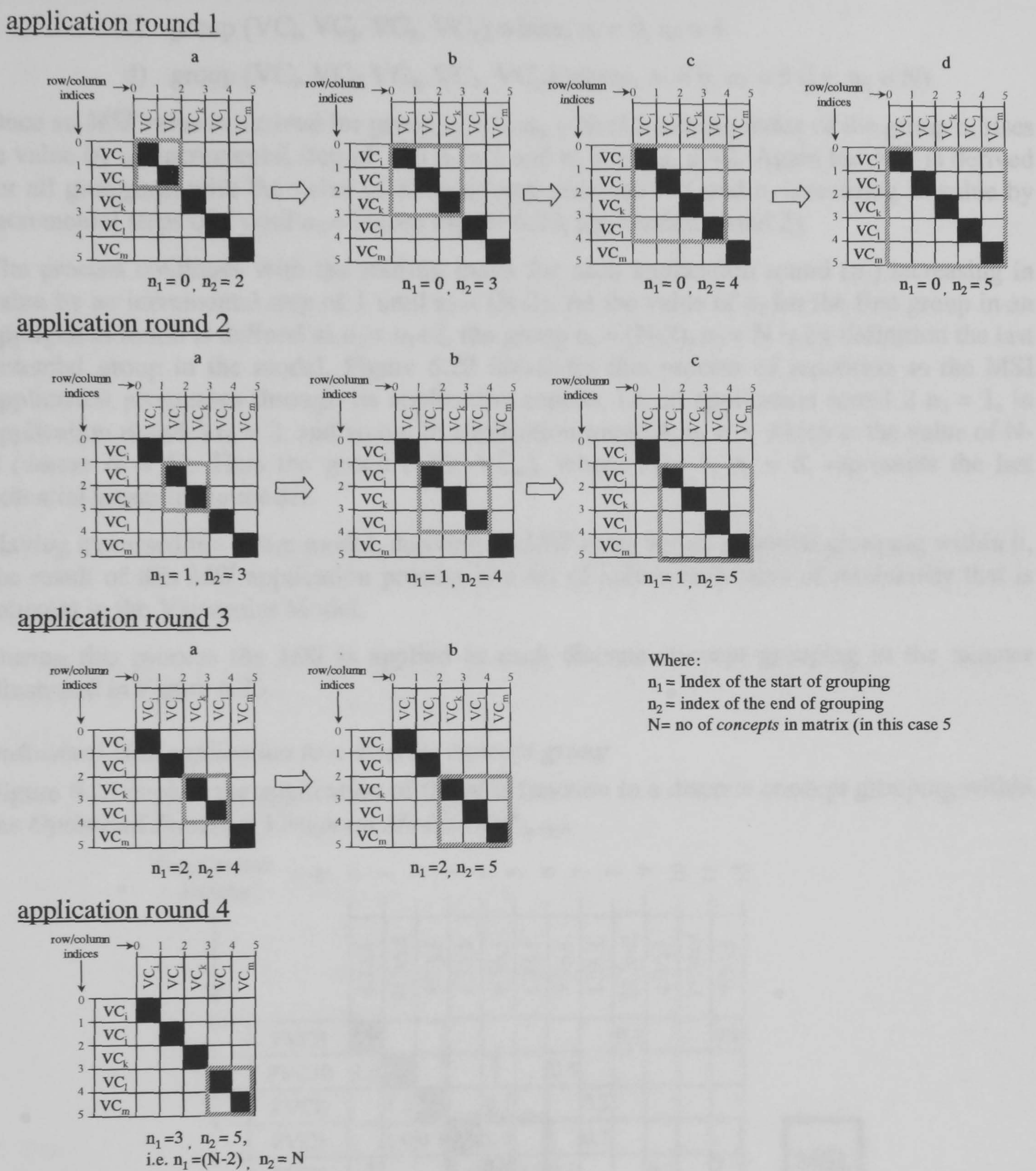


Figure 6.22: MSI model application process

As illustrated in Figure 6.22 (application round 1a), the starting index (n_1) is set to 0 (to represent the start of the model) and the end index (n_2) set to 2 (as this represents the end index of the first potential grouping within a model where the minimum number of *concepts* that a grouping may have is two). Thus, the start and end index for the first grouping in any iteration of MSI application can be defined as: n_1, $n_2 = (n_1+2)$. The *MSI* function is applied to this grouping (see Figure 6.23 and its explanation) and its *MSI* value is derived. This *MSI* application process is repeated for each grouping with $n_1 = 0$ and n_2 increasing in value by incremental steps of 1 until $n_2 = N$ (the number of *concepts* in the model).

Figure 6.22 – application round 1 illustrates this application procedure with the MSI being applied to groupings:

a) group (VC_i, VC_j) where, $n_1 = 0$, $n_2 = 2$

b) group (VC_i, VC_j, VC_k) where, $n_1 = 0$, $n_2 = 3$

c) group (VC_i, VC_j, VC_k, VC_l) where, $n_1 = 0$, $n_2 = 4$

d) group (VC_i, VC_j, VC_k, VC_l, VC_m) where, $n_1 = 0$, $n_2 = 5$ (i.e. $n_2 = N$)

Once an *MSI* value is derived for group $n_1 = 0$, $n_2 = N$, the starting index of the group n_1 rises in value by an incremental step of 1 to $n_1 = 1$ and $n_2 = 3$, i.e. n_1+2. Again the *MSI* is derived for all groupings with the value of n_1 remaining constant at 1 and n_2 increasing in value by incremental steps of 1 until $n_2 = N$ (see Figure 6.22, application round 2).

The process continues with the starting index for each application round (n_1) increasing in value by an incremental step of 1 until $n_1 = (N-2)$. As the value of n_2 for the first group in an application round is defined as $n_2 = n_1+2$, the group $n_1 = (N-2)$, $n_2 = N$ is by definition the last potential group in the model. Figure 6.22 illustrates this process of repetition as the MSI application progresses through its application rounds, i.e. in application round 2 $n_1 = 1$, in application round 3 $n_1 = 2$, and so on. In application round 4, $n_1 = 3$, which is the value of N-2 (where N = 5). Thus the group (VC_l, VC_m), where $n_1 = 3$, $n_2 = 5$, represents the last potential group in the model.

Having traversed the entire model, deriving an *MSI* value for all potential grouping within it, the result of this *MSI* application process is a set of values indicative of modularity that is inherent in the Viewpoint Model.

During this process the *MSI* is applied to each discrete concept grouping in the manner illustrated in Figure 6.23.

Individual MSI application to a discrete concept group

Figure 6.23 depicts the application of the *MSI* function to a discrete concept grouping within the *Optimised Function Viewpoint Model FDP_{energy}*.

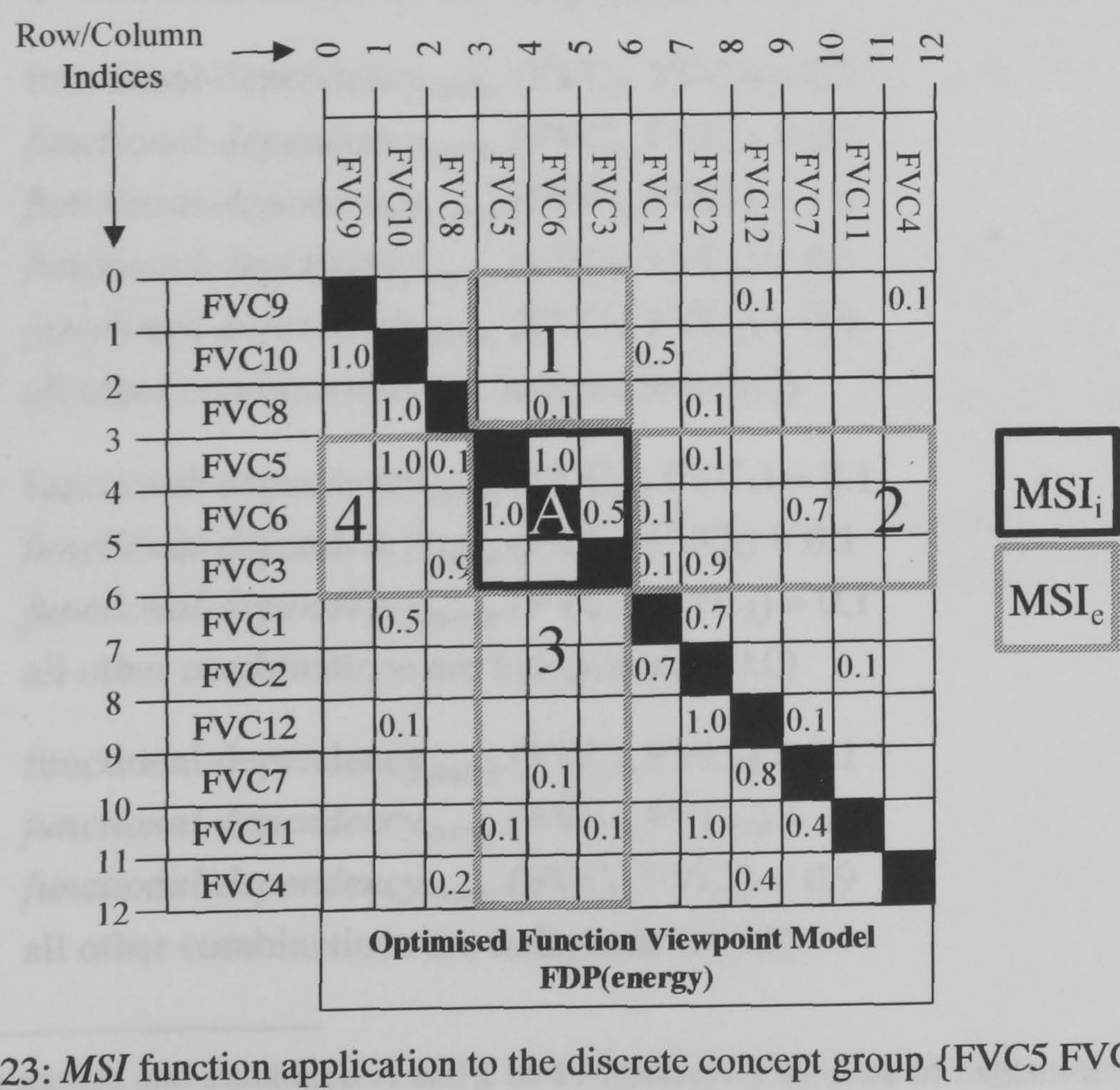


Figure 6.23: *MSI* function application to the discrete concept group {FVC5 FVC6 FVC3}.

The group has been chosen at random from all the groupings that may exist within the model (see Figure 6.22 and its explanation). The *MSI* function (Equation 6.7) is being applied to the concept group {FVC_5 FVC_6 FVC_3}. The black box A depicts the MSI_i part (see Equation 6.5) and the grey boxes (1,2,3,4)[27] the MSI_e (see Equation 6.6).

The dependencies that exist between the *concepts* within the proposed group, i.e. within the black box (A) in Figure 6.23, can be defined as follows:

Box A

functional-dependency$_{energy}$ (FVC_5, FVC_6) = 1.0

functional-dependency$_{energy}$ (FVC_6, FVC_5) = 1.0

functional-dependency$_{energy}$ (FVC_6, FVC_3) = 0.5

functional-dependency$_{energy}$ (FVC_5, FVC_3) (FVC_3, FVC_5) (FVC_3, FVC_6) are all independent. (0.0)

It can be seen within Figure 6.23, that the index of the start and end of *concept grouping* (n_1 and n_2) are 3 and 6 respectively. Thus the resulting calculation to derive the MSI_i value is as follows:

$$MSI_i = \frac{(1.0+1.0+0.5)^{A}}{(6-3)^2-(6-3)} = \frac{2.5}{6} = 0.417 \qquad \textit{Equation 6.8}$$

Equation 6.8 represents the summation of the actual internal dependencies between the *concepts* FVC_5, FVC_6, and FVC_3 (the dividend in Equation 6.8, which is 2.5) divided by the maximum value of the *dependencies* that could exist between these *concepts* (the divisor in Equation 6.8, which is 6).

The dependencies that exist between the *concepts* within the proposed grouping and those *concepts* out with it, i.e. the grey boxes (1, 2, 3, 4) within Figure 6.23, can be defined as follows:

Box 1

functional-dependency$_{energy}$ (FVC_8, FVC_6) = 0.1
all other combinations are independent (0.0)

Box 2

functional-dependency$_{energy}$ (FVC_5, FVC_2) = 0.1
functional-dependency$_{energy}$ (FVC_6, FVC_1) = 0.1
functional-dependency$_{energy}$ (FVC_6, FVC_7) = 1.0
functional-dependency$_{energy}$ (FVC_3, FVC_1) = 0.1
functional-dependency$_{energy}$ (FVC_3, FVC_2) = 0.9
all other combinations are independent (0.0)

Box 3

functional-dependency$_{energy}$ (FVC_{11}, FVC_5) = 0.1
functional-dependency$_{energy}$ (FVC_7, FVC_6) = 0.1
functional-dependency$_{energy}$ (FVC_{11}, FVC_3) = 0.1
all other combinations are independent (0.0)

Box 4

functional-dependency$_{energy}$ (FVC_5, FVC_8) = 0.1
functional-dependency$_{energy}$ (FVC_5, FVC_{10}) = 1.0
functional-dependency$_{energy}$ (FVC_3, FVC_8) = 0.9
all other combinations are independent (0.0)

[27] The boxes are named and numbered A and 1 to 4 respectively for ease of explanation only.

Again, the indexes of the start and end of the *concept* grouping (n_1 and n_2) are 3 and 6 respectively and the number of concepts in the matrix (N) is 12. Thus the resulting calculation to derive the MSI_e value (see Equation 6.6) is as follows (subtext 1, 2, 3 and 4 represents each white dashed boxes above):

$$MSI_e = \left(\frac{((0.1)_1 + (0.1+1.0+0.9)_4)}{2\times(3\times(6-3))}\right)^a + \left(\frac{((0.1+0.1+1.0+0.1+0.9)_2 + (0.1+0.1+0.1)_3)}{2\times((12-6)\times(6-3))}\right)^b$$

$$= \left(\frac{2.1}{18}\right)^a + \left(\frac{2.5}{36}\right)^b = 0.186$$

Equation 6.9

Part a of Equation 6.9 depicts the summation of the actual dependencies within the grey boxes 1 and 4 (dividend of Equation 6.9 part a) and the maximum number of possible dependencies within the grey boxes 1 and 4 (the divisor of Equation 6.9 part a). Part b of Equation 6.9 depicts the summation of the actual dependencies within the grey boxes 2 and 3 (dividend of Equation 6.9 part b) and the maximum number of possible dependencies within the grey boxes 2 and 3 (the divisor of Equation 6.9 part b). Thus Equation 6.9, taken in its entirety, represents the actual value of the dependencies between the *concepts* in the grouping and those *concepts* out with the grouping divided by the maximum number of dependencies that can exist between the grouped *concepts* and those concepts out with the group.

The MSI value (see Equation 6.7) for the grouping would thus be derived as follows:

$$MSI = MSI_i - MSI_e$$

$$\text{MSI} = 0.417 - 0.186 = 0.231$$

The concept group {FVC_5 FVC_6 FVC_3} of the *Optimised Function Viewpoint Model* FDP_{energy} can be seen to have a relatively low modular value (with 1.0 being the highest possible module value). The concept group {FVC_5 FVC_6 FVC_3} has low value of internal dependencies and a number of external dependencies and as such returned a relatively low *MSI* value. Thus, the *MSI* value provides the designer with a measure of the relative modularity of concept groupings and an indication as to the potential of the concept grouping as a module candidate.

6.3.2 Modular Structure Matrix application

The *MSI* application technique results in a series of MSI values for all possible concept groupings in a matrix. An interpretation of these values results in an alternative representation of the DSM termed the *'Module Structure Matrix' (MSM)*. The *MSM* depicts, as different coloured cells the relative modularity of all available *concept* groupings within it.

The returned *MSI* values for the *Optimised Function Viewpoint Model* FDP_{energy} are shown in Figure 6.24 (only the groupings that are not later enveloped by a grouping with a higher *MSI* value are shown).

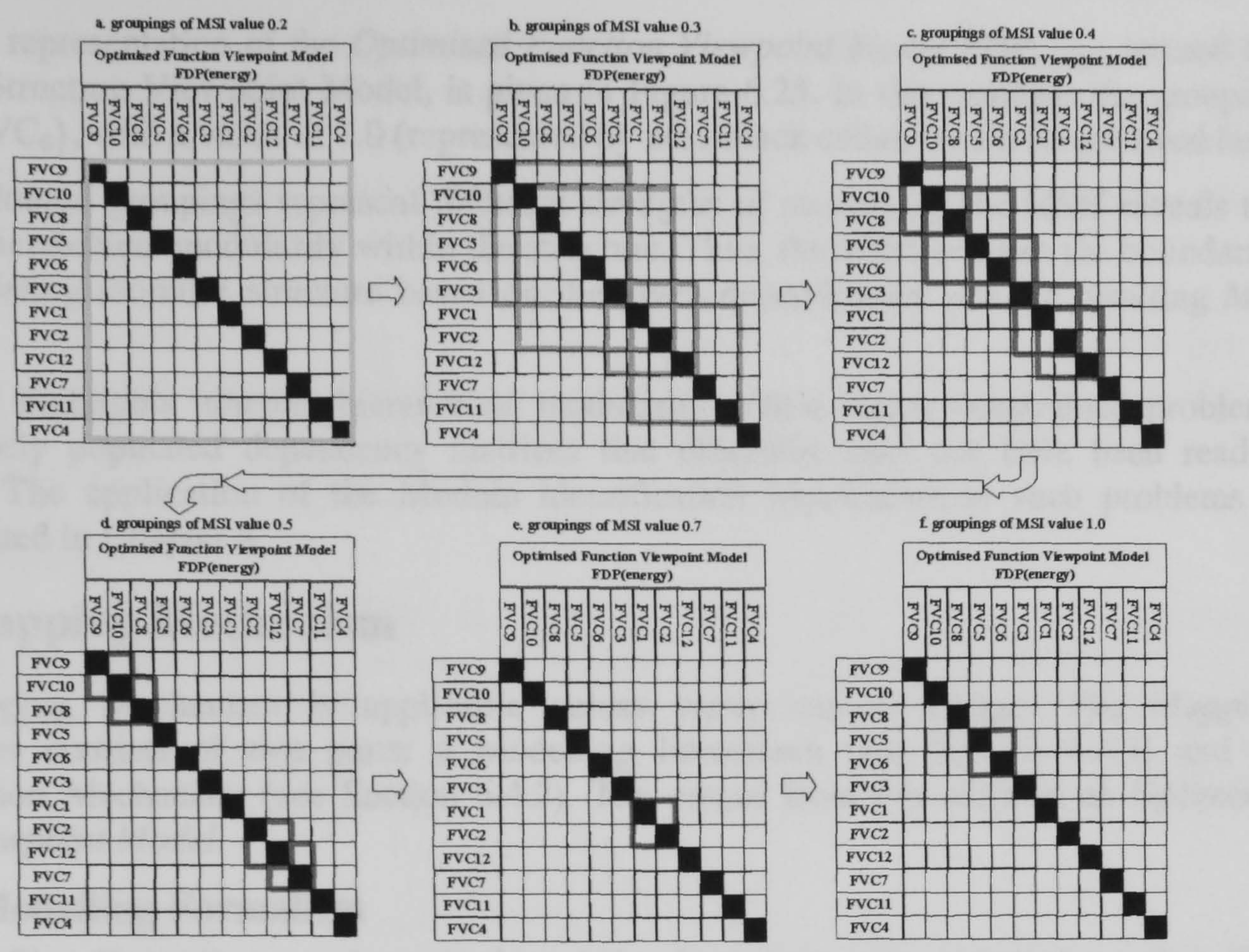


Figure 6.24: MSI Values for the *Optimised Function Viewpoint Model* FDP_{energy}

Figure 6.24, a to f, depict concept groupings with differing values of *MSI*. Though the *MSI* values are of a continuous nature, the cells are coloured utilising an incremental range denoting the values 0.1 to 1.0 (in incremental steps of 0.1). Figure 6.24a illustrates that as a single grouping all the *concepts* in the model have an *MSI* value of 0.2. Increasingly higher values of *MSI* are also recorded in b to f. With the *concept* grouping (FVC5, FVC6) returning a maximum *MSI* value of 1.0. The arrow notations in Figure 6.24 represent the flow with which groupings are coloured in the MSM interpretation of this model and its *MSI* values to ensure that a higher value grouping is not enveloped within a lower value grouping (see arrow notation in Figure 6.24). For example, in Figure 6.25 the grouping {FVC_9, FVC_{10}, FVC_8, FVC_5, FVC_6, FVC_3, FVC_1, FVC_2, FVC_{12}, FVC_7, FVC_{11}, FVC_4} which has a value of 0.2 (depicted as very light grey cells) would be coloured before any grouping with a higher value.

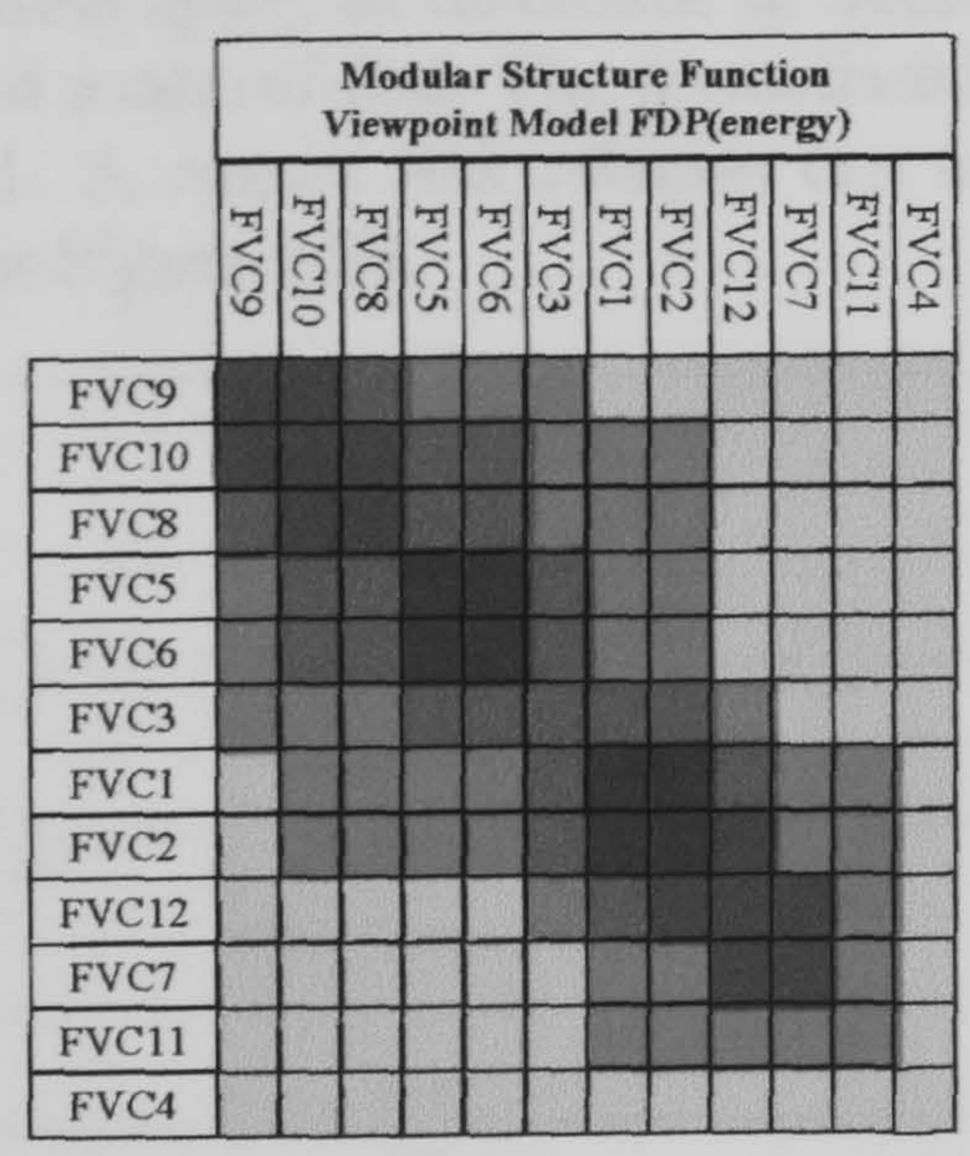


Figure 6.25: The Modular Structure FunctionViewpoint Model FDP_{energy} - an MSM interpretation of the Optimised Function Viewpoint Model FDP_{energy}

The *MSM* representation of the *Optimised Function Viewpoint Model FDP_{energy}*, termed the Modular Structure Viewpoint Model, is given in Figure 6.25. In this example the grouping {FVC_5, FVC_6}, with a value of 1.0 (represented by near-black cells), would be coloured last.

As the coloured groupings represent different strengths of modularity the *MSM* reveals the inherent hierarchical modularity within the structure. Thus, the *MSM* exposes the boundaries of any existing modular structure based on the given dependencies and the resulting *MSI* values.

The *MSM* highlights inherent hierarchical modularity within highly constrained problems with densely populated dependency matrices that otherwise may not have been readily apparent. The application of the Module Identification Mechanism to such problems is demonstrated in Chapter 8.

6.4 Mapping Mechanism

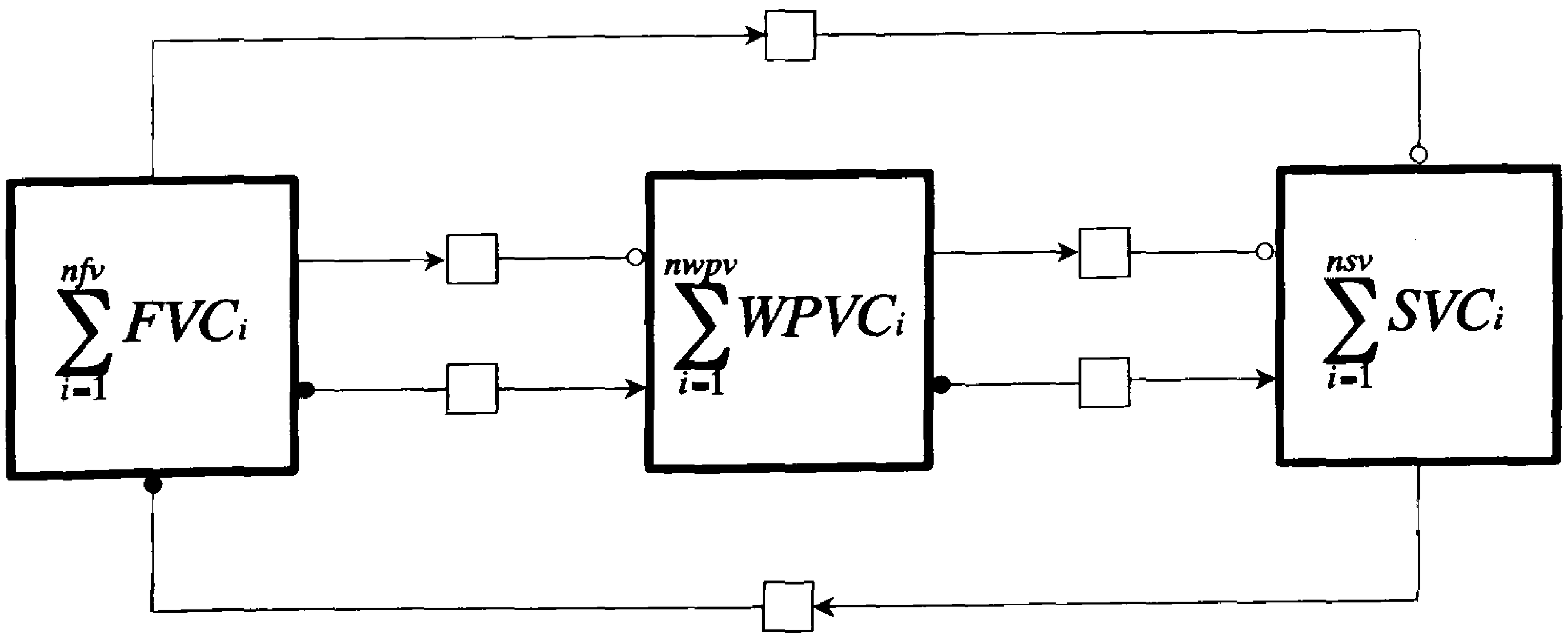
The Mapping Mechanism is applicable across viewpoints of design. The *Mapping Mechanism* consists of two parts: a Modelling Formalism (see Section 6.4.1) and an Optimisation Mechanism (see Section 6.4.2). The output from this stage is an *Optimised Cross-Viewpoint Model.*

6.4.1 Modelling Formalism

The Modelling Formalism consists of a *knowledge formalism* and a *matrix formalism.* The knowledge formalism is based on elements of the MVEA formalism (Zhang 1998) covered previously (Section 6.1.1). The matrix application differs from the DSM application, in that the matrix is non-square and the concepts depicted in the rows and columns are representative of different viewpoints of design knowledge. The outcome of the Modelling Formalism part is a *Cross-Viewpoint Model.*

Knowledge Formalism

The Cross-Viewpoint Model consists of *concepts* from differing viewpoints and their *cross-viewpoint dependencies* (see Section 6.1.1). Having formalised the design knowledge within individual views the *causal-link* relation reflects the evolution of the working design from abstract *concepts* such as function to concrete *concepts* such as parts. Thus, the *viewpoint concepts* are the same as those depicted in individual *Viewpoint Models* and as such the same *concept* formalisms apply as discussed in Section 6.1.1. The *concepts* have a 'what-how' relationship, termed a *causal-link.* The formalism for the *causal-link* relation can also be found in Section 6.1.1. A *causal link* relation can exist between two *concepts* of different types as represented in Figure 6.26.



Notations:

FVC_i:	a concept within the function viewpoint	nfv_i:	the number of concepts within the function viewpoint
$WPVC_i$:	a concept within the working principle viewpoint	$nwpv_i$:	the number of concepts within the working principle viewpoint
SVC_i:	a concept within the structure viewpoint	nsv:	the number of concepts within the structure viewpoint

Figure 6.26: Possible Causal-link relations between concepts of CWK Viewpoints (Zhang 1998)

The figure illustrates that causal-link relations may exist between concepts from the function (FV), working principle (WPV) and structural viewpoint (SV). The multiplicity notation specifies the number of concepts with which another concept can have causal-link relations(Zhang 1998). Thus, it can be seen that there is the possibility of *one-to-one*, *one-to-many* and *many-to one* mappings between concepts of different types.

A minimum of two *Cross-Viewpoint Models* are required to support the maintenance of the modular solution across the *viewpoints* of design (function through working principle to structure), i.e. *Cross-Viewpoint Model$_{(function,\ working\ principle)}$*, mapping from the function to working principle viewpoint, and *Cross-Viewpoint Model$_{(working\ principle,\ structure)}$* mapping from the working principle to structure viewpoint.

Matrix Formalism

A *Matrix Formalism* is applied to the generated knowledge to create a matrix-based representation termed, the *Cross-Viewpoint Model.* A cross-viewpoint matrix can be generated for any two combinations of *viewpoints* of interest to the designer resulting in an asymmetric matrix as illustrated in Figure 6.27.

	WPVC1	WPVC2	WPVC3	WPVC4	WPVC5	WPVC6	WPVC7	WPVC8	WPVC9
FVC1	□								
FVC2			□			□			
FVC3	□	□							
FVC4					□				
FVC5							□	□	
FVC6				□					□
FVC7				□					
FVC8				□					
FVC9					□				
FVC10							□		
FVC11			□						
FVC12	□						□	□	

Notations:
FVC: a concept from viewpoint function
WPVC: a concept from viewpoint working principle
: a causal link relation between two concepts of different types

Figure 6.27: The Cross–Viewpoint Matrix Formalism

Figure 6.27 illustrates the causal-link relations from function viewpoint to the working principle viewpoint. The matrix body depicts the *causal-link* dependence from the function viewpoint to working principle viewpoint denoted by the symbol in the row and column intersection. No denotation in a row column intersection represents the independence of the concepts.

6.4.2 Optimisation Mechanism

The Optimisation Mechanism is similar to that defined in Section 6.2 in that it consists of a *genetic algorithm* and a *clustering criteria* (termed a *cross-viewpoint clustering criteria CVCC).* However the application of the GA and the measure defined by the *clustering* *criteria* differ from that defined in Section 6.2. The following section defines the GA application and cross-viewpoint clustering criteria.

Genetic Algorithm

Due to the limitations of humans short term memory (see Section 4.1.5) the application of a GA is proposed to enable the exploration of the maintenance of near-optimal modular structures across *viewpoints* of design knowledge. The proposed application procedure for the genetic algorithm is similar to that illustrated within Figure 6.16 (Section 6.2.1) and again is based on the general structure for a genetic algorithm developed by Goldberg (Goldberg 1989) (see also Chapter 7). The main difference in the application of the GA is that, unlike in the individual *viewpoint model*, the GA could be applied re-sequence either row or column or both simultaneously. However, as the aim of the work presented in this thesis is to facilitate *knowledge modularisation*, to support *Design for Re-use*, the focus of the following section in on the application of the GA to maintain the modular solution across *viewpoints* of design.

Figure 6.28 depicts the *cross-viewpoint model (function to working principle)* where the row represents concepts from a more abstract viewpoint (function) whilst the columns represent concepts from a more concrete viewpoint (working principle) and the dependencies in the matrix body represent how the concepts in the rows are realised by the concepts in the columns, i.e. how the *function concepts* are realised by the *working principle concepts*.

optimised *function concept* order

	WPVC1	WPVC2	WPVC3	WPVC4	WPVC5	WPVC6	WPVC7	WPVC8	WPVC9
FVC11			□						
FVC2			□			□			
FVC3	□	□							
FVC1	□								
FVC8				□					
FVC7				□					
FVC6				□					□
FVC5							□	□	
FVC10							□		
FVC12	□						□	□	
FVC9					□				
FVC4					□				

Figure 6.28: GA application to maintain the modular solution

Thus, by utilising the near-optimum *concept* order returned for the more abstract view and optimising the concept order in the more concrete viewpoint in relation to these the design can maintain the modular solution across *viewpoints* of the design. As such, the GA is applied to re-sequence the *concepts* represented in the columns of a *cross-viewpoint model*. Thus, taking the example shown in Figure 6.28 (the *cross-viewpoint model (function to working principle)*) it can be seen that the order of the *function concepts* is based on the order defined in the *optimised function viewpoint model*$_{FDP(combined)}$ (extracted from Figure 6.20 above). The aim of the optimisation mechanism in this case is to optimise the order of the *working principle concepts* with the objective of maintaining the modular solution, i.e. determine the *working principle concept* groupings that map to the previously optimised *function concept* groupings.

As illustrated in section 6.2.1 (Figure 6.16 and explanation) the start point for the genetic algorithm application is an initial chromosome (gene). In the case of the MD maintenance problem, the chromosome represents the order (sequence) of *concepts* in the column of the *cross-viewpoint model.* The initial chromosome is evaluated with respect to the *cross-viewpoint clustering criteria* to gain an initial value. As with the GA application defined in section 6.2.1 the population size, number of generation, crossover and mutations operators and the probability of crossover and mutation occurring can be defined by the designer. The GA application process continues as in Section 6.2.1 completing a pre-determined number of generations. The best performing chromosomes are saved as potential optimum candidates, i.e. those with the minimum value for the *cross-viewpoint clustering criteria* are saved.

On completion the designer can select one of the optimised chromosomes. The order of the *concepts* in this chromosome provides the *concept* sequence for the column of the *optimised cross-viewpoint model.*

Cross-Viewpoint Clustering Criterion

The cross-viewpoint clustering criterion (CVCC) is represented within Equation 6.10. The equation represents the summation of the horizontal and vertical distance between consecutive dependencies in both row and column of the matrix as depicted in Figure 6.29.

$$CVCC = \sum_{i}^{N}\sum_{j}^{N}(n_2 - n_1 - 1) + (n_3 - n_2 - 1) + ... + (n_n - n_{(n-1)} - 1) \quad \text{... ...Equation 6.10}$$

Where: N is the number of concept in either row or column

i and j are the row and column indices respectively, and

n is the starting index of the a dependency in either row or column (*where, n_1 is the index of the first dependency in either row or column and n_n is the last dependency in the row or column*)

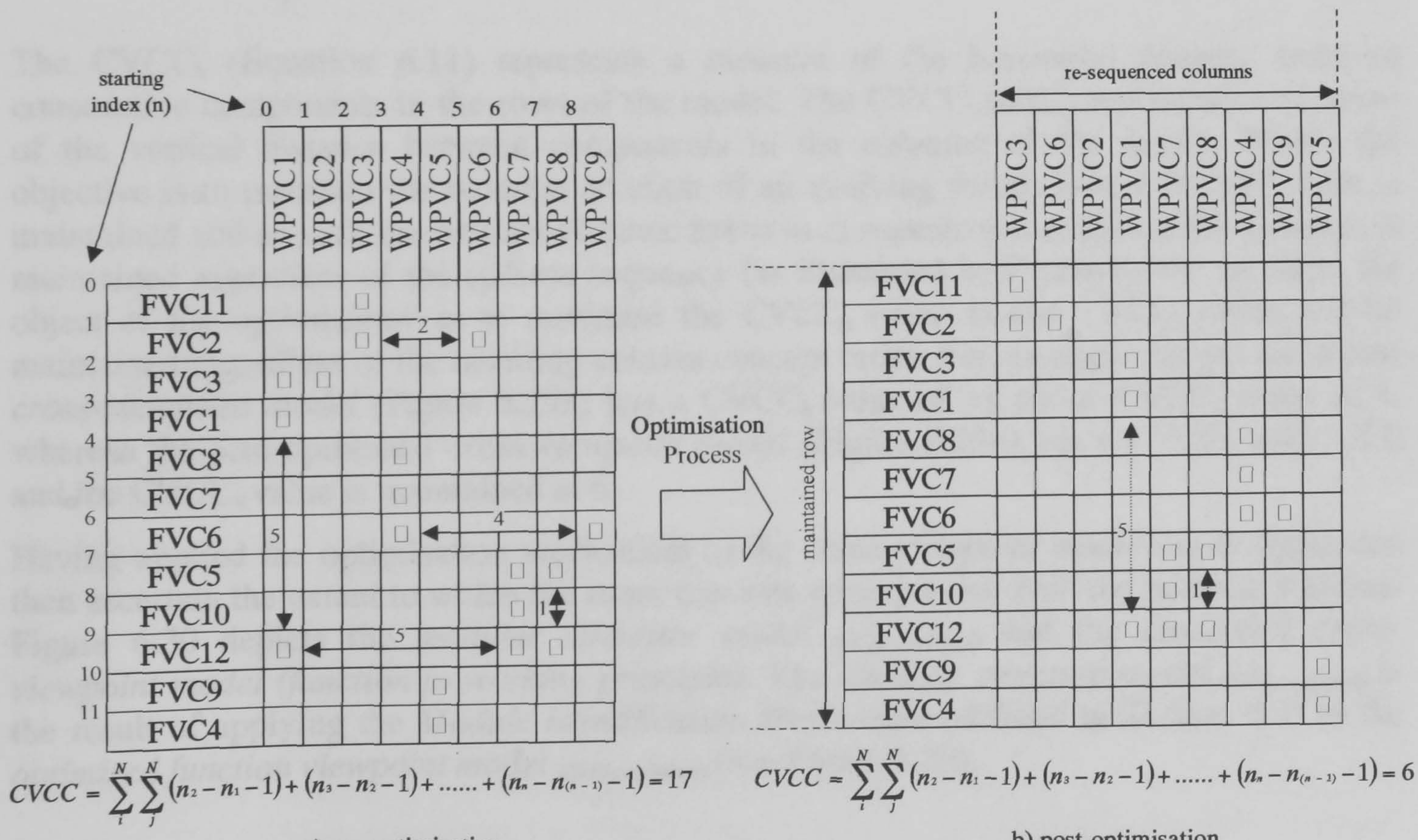


Figure 6.29: Minimising the cross-viewpoint clustering criteria.

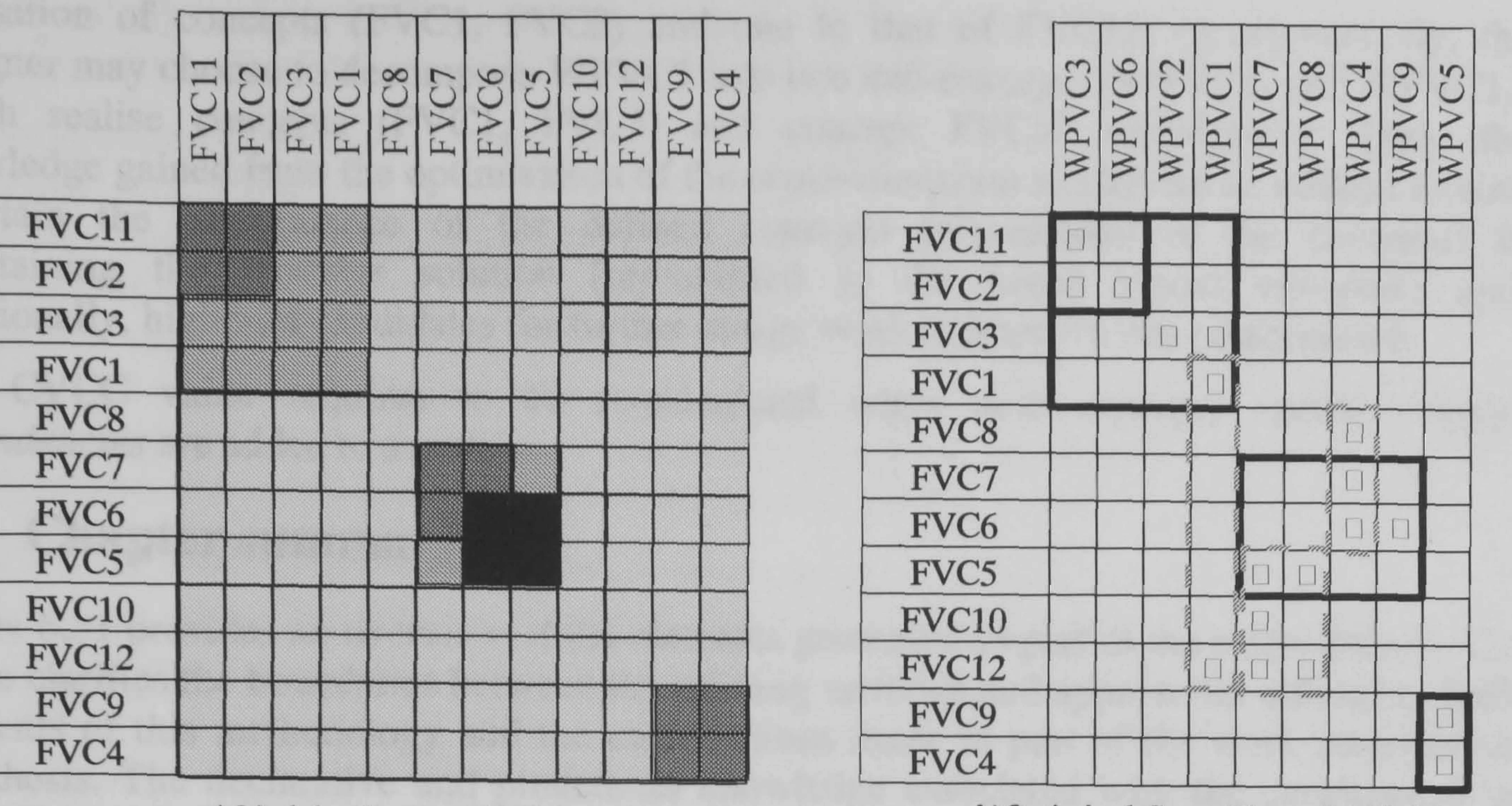
Figure 6.29 depicts the cross-viewpoint model (function to working principle) both pre and post optimisation. The figure illustrate that during the optimisation process the order of the *concepts* represented in the rows are maintained whilst the GA is applied to re-sequence the order of the *concepts* represented by the columns. That is, the concept order WPVC1, WPVC2, WPVC3, WPVC4, ..., WPVC12 is changed to WPVC3,WPVC6, WPVC2, ... and so forth. Again, a dependency between two concepts cannot be violated. The clustering criteria nears a minimum as consecutive dependencies in both row and column move towards each other. For example, consider consecutive causal-link dependencies (in row FVC2), *causal link(FVC2, WPVC3)* and *causal-link(FVC2, WPVC6),* it can be seen in the pre-optimised *cross-viewpoint model* (Figure 6.29a) that there is a distance equal to two matrix squares between these, whereas, in the post-optimised *cross-viewpoint model* (Figure 6.29) these dependencies have moved adjacent to each other and there are no matrix squares between these. For the optimisation mechanism employed to re-sequence both row and column it would be theoretically possible to reduce the CVCC to a minimum of value of zero. However, in the case of maintaining a modular solution, the row sequence is maintained during the optimisation process and as such the vertical distance between concepts in the columns is maintained. For example, consider the consecutive causal-link dependencies (in column WPVC1), *causal link(FVC1, WPVC1)* and *causal-link(FVC12, WPVC1),* it can be seen from the pre-optimised *cross-viewpoint model* that there is a distance of 5 matrix squares between these, a distance which is maintained in the post-optimised *cross-viewpoint model* despite the alteration in the working principle viewpoint concept sequence. Thus the CVCC can be further decomposed into a measure of horizontal (Equation 6.11) and vertical (Equation 6.12) distance between consecutive dependencies in the row or column.

$$CVCC_h = \sum_{i}^{N} (n_2 - n_1 - 1) + (n_3 - n_2 - 1) + \ldots\ldots + (n_n - n_{(n-1)} - 1) \qquad \textit{Equation 6.11}$$

$$CVCC_v = \sum_{i}^{N} (n_2 - n_1 - 1) + (n_3 - n_2 - 1) + \ldots\ldots + (n_n - n_{(n-1)} - 1) \qquad \textit{Equation 6.12}$$

The $CVCC_h$ (Equation 6.11) represents a measure of the horizontal distance between consecutive components in the rows of the model. The $CVCC_v$ (6.12) represents a measure of the vertical distance between components in the columns of the model. Where the objective is to maintain the modular solution of an evolving design model the row order is maintained and as such the vertical distance between consecutive *concepts* in the columns is maintained regardless of the *column* sequence (as illustrated in Figure 6.29). As such, the object of the optimisation is to minimise the $CVCC_h$ value as the $CVCC_v$ value will be maintained regardless of the resulting column concept order. For, example the pre-optimised *cross-viewpoint model* (Figure 6.29a) has a $CVCC_h$ value of 11 and a $CVCC_v$ value of 6, whereas the post-optimised *cross-viewpoint model* (Figure 6.29b) has a $CVCC_h$ value of 0 and the $CVCC_v$ value is maintained at 6.

Having applied the optimisation mechanism to the *cross-viewpoint model* the designer can then ascertain the extent to which the more concrete concepts maintain the modular solution. Figure 6.30 depicts the *modular structure model* $_{FDP(combined)}$ and the associated *cross-viewpoint model (function to working principle)*. The *modular structure model* $_{FDP(combined)}$ is the result of applying the *Module Identification Mechanism* (defined in Section 6.3) to the *optimised function viewpoint model* $_{FDP(combined)}$ (see Figure 6.20).



a) Modular Structure Model $_{FDP(combined)}$ b)Optimised Cross-Viewpoint Model (function to working principle)

Figure 6.30: Maintaining the modular solution

The *cross-viewpoint model* depicts the *concepts* from the *function viewpoint* in the same order as determined in the *optimised function viewpoint model* $_{FDP(combined)}$ and the *working principle concepts* defined to realise these. A number of concept groupings have been defined in the *modular structure model* $_{FDP(combined)}$. The *cross-viewpoint model (function to working principle)* has been optimised with the objective of reducing the $CVCC_h$ to support the maintenance of the modular solution with respect to these *function concept groupings*. It is evident from the *cross-viewpoint model (function to working principle)* that a number of clear concept grouping can be derived. For example, we see that the *function concept grouping* (FVC11, FVC2) has been wholly realised by the *working principle concept grouping* (WPVC3, WPVC6) (depicted by the black boxes overlaid on the cross-viewpoint model in Figure 6.30). Similarly, the *function concept grouping* (FVC9, FVC4) has been wholly realised by the *working principle concept* WPVC5. On the other hand, the *function concept grouping* (FVC7, FVC6, FVC5) is wholly realised by the *working principle concepts* WPVC7, WPVC8, WPVC4 and WPVC9, however, concepts WPVC7, WPVC8, WPVC4 also realise one or more alternative *function concepts* out-with the group (FVC7, FVC6, FVC5) (depicted by the grey dashed line boxes overlaid on the cross-viewpoint model in Figure 6.30). A similar situation arises with the *function concept grouping* (FVC11, FVC2, FVC3, FVC1) which is wholly realised by the *working principle concepts* WPVC3, WPVC6, WPVC2 and WPVC1, however, concept WPVC1 also realise one *function concept* out-with the grouping, i.e. FVC12. In this case the designer may not be too concerned as the *function concept grouping*, which shares these working principle concepts (WPVC7, WPVC8, WPVC4, WPVC1), is representative of the entire product structure (all *concepts* in the *function viewpoint*) of which the *modules*, which they already wholly realise, are a constituent part. Thus there are no conflicts between *working principle concepts* and any strong module candidates in the *function viewpoint*, i.e. no strong modular *function concept groupings* share any two *working principle concepts*. However, were the designer to be concerned, or alternatively had a *working principle concept* realised two *function concepts* in two different modular groupings, the designer may wish to reconsider the *working principle concepts* WPVC7, WPVC8, WPVC4, WPVC1. The designer may wish to maintain the modularity of *function concept grouping* (FVC11, FVC2, FVC3, FVC1) and as such the *working principle concept* WPVC1, which realises FVC1, FVC3 within the grouping and FVC12 out-with the grouping, becomes a candidate for further design work. For example, the designer may choose to further duplicate WPVC1 and assign one *concept* to the realisation of concepts (FVC1, FVC3) and one to that of FVC12, or alternatively, the designer may choose to decompose WPVC1 into two *sub-concepts* ($WPVC1_x$ and $WPVC1_y$) which realise concepts (FVC1, FVC3) and concept FVC12 respectively. Thus, the knowledge gained from the optimisation of the *cross-viewpoint model* can be utilised to both ascertain the performance of the defined *concepts* (represented in the columns) in maintaining the modular solution (represented in the rows) across viewpoint and, additionally, highlight candidates for further design work to improve this performance.

The CVCC value requires to be re-calculated when new *concepts* and/or *concept dependencies* are added to a matrix.

6.5 Chapter summary

Figure 6.31 provides an overview of the elements presented as part of the methodology. The figure clarifies the boundaries between the existing methods and approaches utilised to fulfil elements of this methodology and the contributions made as part of the work presented in this thesis. The declarative and procedural knowledge associated with the construction of these elements (both existing and new) in to a comprehensive methodology framework and their practical application to support the activity of MD is also considered a contribution of the work presented in this thesis.

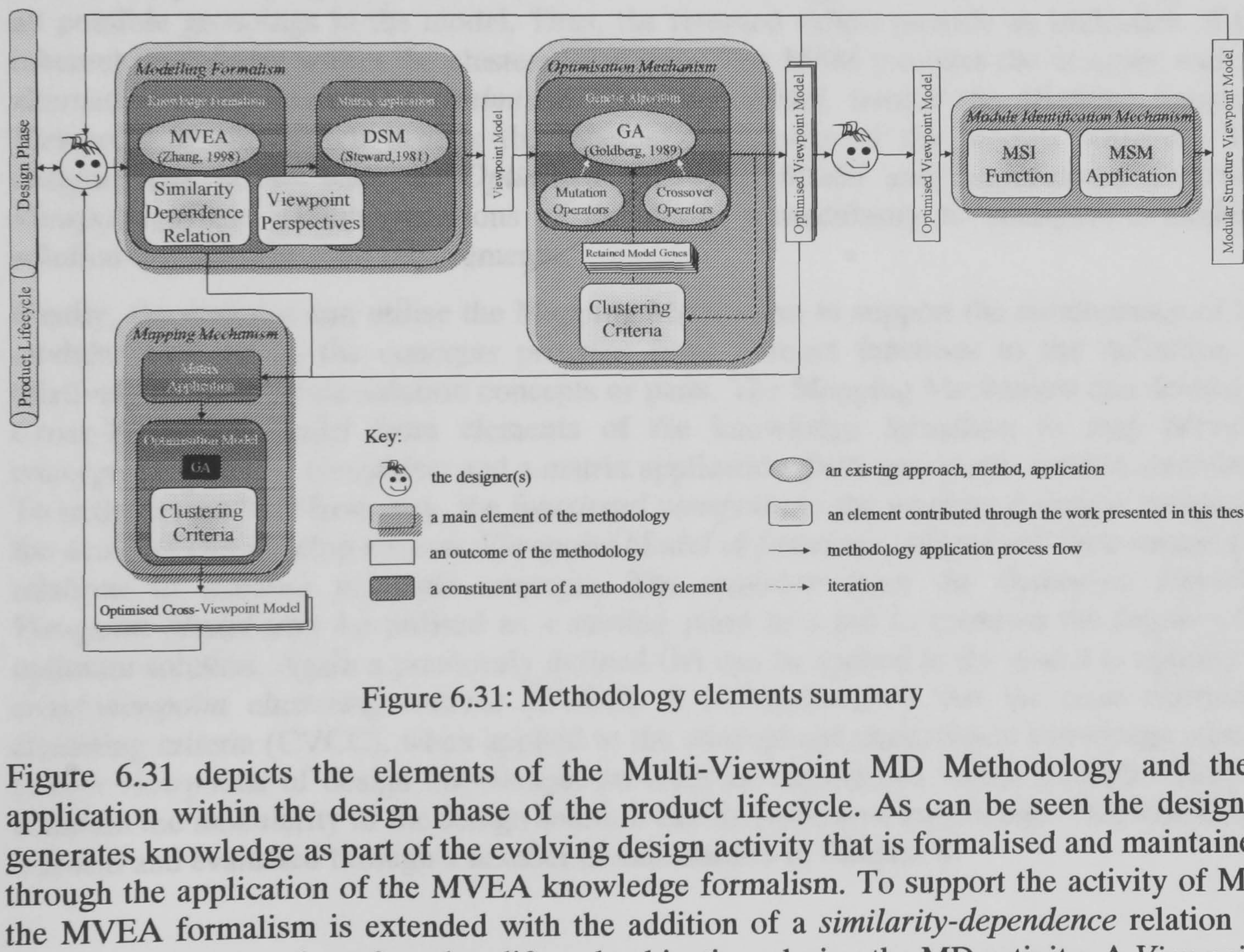


Figure 6.31: Methodology elements summary

Figure 6.31 depicts the elements of the Multi-Viewpoint MD Methodology and their application within the design phase of the product lifecycle. As can be seen the designer generates knowledge as part of the evolving design activity that is formalised and maintained through the application of the MVEA knowledge formalism. To support the activity of MD the MVEA formalism is extended with the addition of a *similarity-dependence* relation to facilitate the exploration of product lifecycle objectives during the MD activity. A *Viewpoint Model* is created through the application of a Dependency Structure Matrix (DSM). The inclusion of *viewpoint perspectives* extends both the current MVEA knowledge formalism and the DSM application. *Viewpoint perspectives* are created to allow the designer to represent the *Viewpoint Model* based on the type of dependency (*functional-dependency, physical-link, similarity-dependence)* and matter *(material, energy, information)* or characteristics (*material-type, useful-technology-lifetime*) of interest at that stage in the design activity and to facilitate the exploration of the impact of these on the MD. The knowledge formalism and matrix application result in a *Viewpoint Model.*

An optimisation mechanism is applied to the *Viewpoint Model* to generate an *Optimised Viewpoint Model.* As can be seen in Figure 6.31 the optimisation mechanism utilises a previously developed GA procedure, utilising crossover and mutation operators (extracted from various literature sources) and a clustering criteria defined as part of the work presented within this thesis. It is hypothesised that the clustering criteria (CC), when applied to the concept and dependency knowledge related to a design activity, provides an appropriate means through which to cluster design concepts to support the identification of modularity. This hypothesis is tested and evaluated through a number of case studies in Chapter 8.

The *Optimised Viewpoint Model* presents an optimum sequence of concepts with respect to the proposed clustering criteria. However, the author proposes that these clusters require further analysis to ascertain their suitability as module candidates. Thus, the *Module Identification Mechanism* has been developed as part of this work to support this analysis. The *Module Identification Mechanism* consists of a Module Strength Indictor *(MSI)* and a *Module Structure Matrix (MSM)*. The *MSI* provides a measure of the modularity of a concept grouping based on its associated internal and external dependencies. The application of the MSI to an *Optimised Viewpoint Model* is novel in that it assumes that any grouping of two or more concepts may represent a module candidate and is applied based on this assumption to all possible groupings in the model. Thus, the returned values provide an indication of the inherent modularity within the clustered concepts. The MSM provides the designer with an alternative representation of *Optimised Viewpoint Model,* termed the *Modular Structure Viewpoint Model,* which depicts the inherent modularity of the product structure. The designer can utilise both the *Optimised Viewpoint Model* and the *Modular Structure Viewpoint Model* as interpretations of the design's modularity to configure a modular solution that satisfies their requirements.

Finally, the designer can utilise the Mapping Mechanism to support the maintenance of the modular solution as the *concepts* progress from abstract functions to the definition of relatively more concrete solution concepts or parts. The Mapping Mechanism can develop a *Cross-Viewpoint Model* from elements of the knowledge formalism to map between concepts in different *viewpoints* and a matrix application (both previously defined elements). To maintain the MD from, say, the *functional viewpoint* to the *working principle viewpoint,* the designer can develop a *Cross-Viewpoint Model* of *function concepts* and their *causal link* relations to *working principle* concepts. The sequence from the *Optimised Function Viewpoint Model* may be utilised as a starting point in a bid to maintain the functionally optimum solution. Again a previously defined GA can be applied to the model to optimise a *cross-viewpoint clustering criteria* (CVCC). It is hypothesised that the cross-viewpoint clustering criteria (CVCC), when applied to the concept and dependency knowledge related to two viewpoints of design knowledge, provides an appropriate means through which to maintain the modularity of the design solution and its associated knowledge. This hypothesis is tested and evaluated through a number of case studies in Chapter 8.

Part 3 –Evaluation, Discussion, Conclusion

7 Computational Methodology Implementation

Chapter 7 presents a computational support system[28]. The methodology has been tested and evaluated through its partial implementation in a computational system and its application to a number of studies. The studies evaluate the methodology's capabilities to fulfil the requirements of an MD methodology for improved EDR support and test the hypothesis posited in Chapter 4. The evaluation process covers four studies, two extracted from published literature and two industrial implementations. Chapter 8 discusses these studies and the resulting findings.

The following covers the current computational implementation for elements of the Multi-Viewpoint MD Methodology. The author was not personally responsible for the programming covered in the following section (see footnote[1] below). However, the implementation of parts of the methodology as a computational tool was undertaken as a means to support its evaluation. The author worked closely with the programmer to ensure the methodology's effective transfer into a computational environment.

7.1 Computational support for the knowledge formalism

It is envisaged that the knowledge formalism (Zhang 1998) would be applied continually throughout the evolving design activity as detailed within the MVEA methodology and supported by DENOTE (Zhang 1998). The dependency matrix formalism would be applied to the required formalised knowledge (maintained by DENOTE) to create a *Viewpoint Model.* DENOTE and the computational tool developed as part of the work presented in this thesis are currently separate tools. However the MVEA knowledge formalism was utilised to provide a formal basis for the work and allow for the integration of these tools at a later date (see Section 9.4, Future Work).

A DSM modelling and analysis system was adapted to reflect elements of the methodology presented in Chapters 5 and 6. The system was previously designed to support process optimisation (Whitfield, Duffy et al. 2001). The system can support multi-criteria optimisation of combinatorial problems. Its adaptation to support the Multi-Viewpoint MD Methodology included the addition of the project type 'modularisation' which includes the additional programming of:

- Modularisation projects
- *Viewpoint* and *perspectives* capabilities.
- Clustering Criteria.
- The *Module Identification Mechanism*, including the:
  - MSI function.
  - MSI application process.
  - MSM representation.
- *Viewpoint mapping* capabilities.

The following section discusses each of these elements in the context of the computational support afforded by the DSM system.

[28] Note that the system programming was undertaken by Dr Ian Whitfield. Dr Whitfield adapted a previously developed process-modelling system to reflect the methodology presented in the previous part of this thesis. The author worked closely with Dr Whitfield during this period to ensure effective computational implementation of the methodology.

7.1.1 Generating modularisation projects

Figure 7.1 shows the interface of the DSM modelling tool with a dialogue box indicating that a project 'new name' of the type 'modularisation' is to be created.

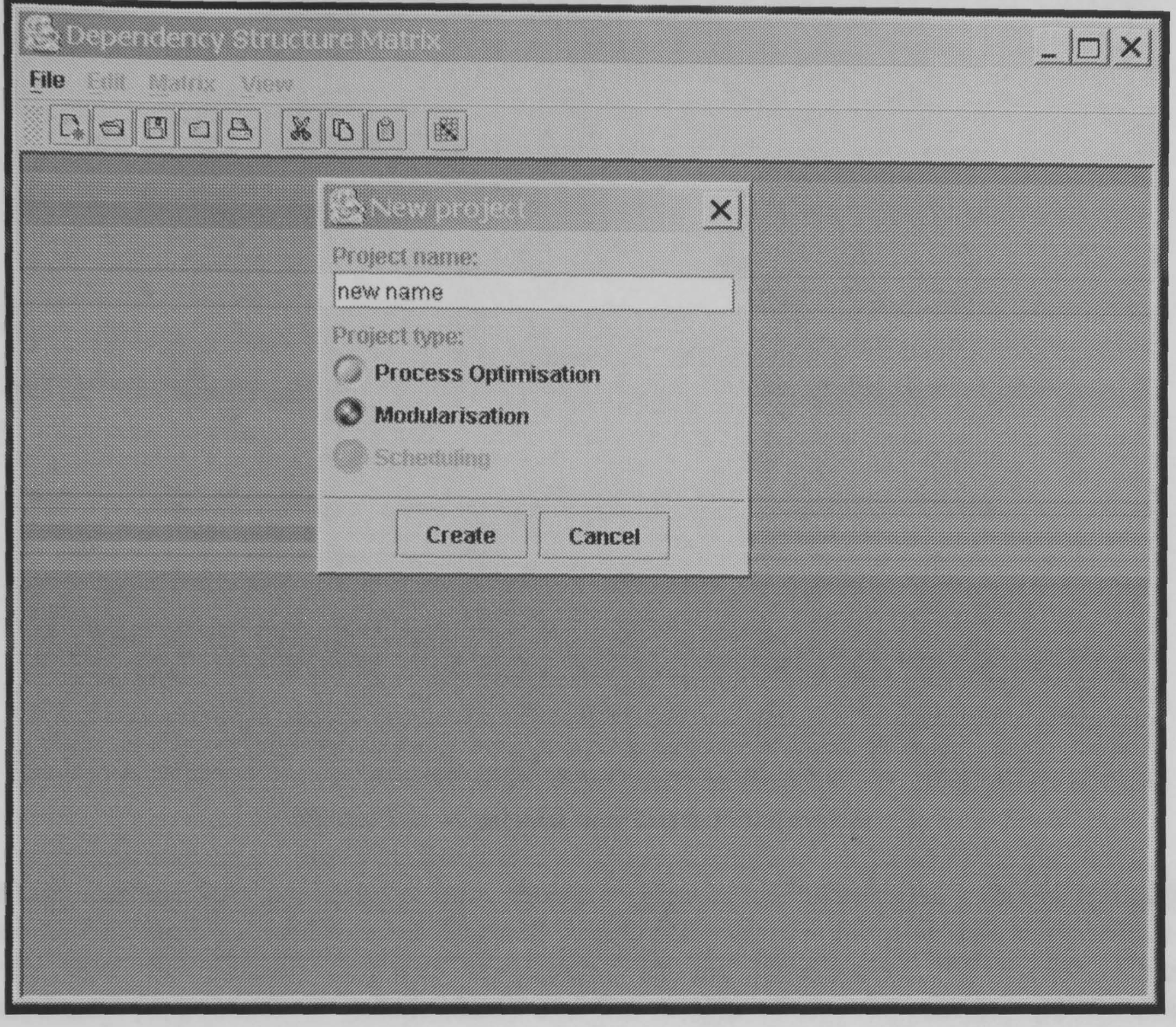


Figure 7.1: DSM modelling and analysis tool interface

Once a new project is created as illustrated in Figure 7.1 the system allows the creation of a number of matrices within the project.

7.1.2 Viewpoint and perspective creation

Viewpoint matrices

A matrix is generated for each viewpoint of interest. Figure 7.2 shows the project 'demonstration' and the three viewpoint matrices: function, working principle and structure[29]. Each matrix may contain any number of *concepts* with the matrix changing size automatically as concepts are added or removed. This is illustrated in Figure 7.2 where the function viewpoint matrix has 12 concepts, the working principle viewpoint matrix 9 concepts and the structure viewpoint matrix 20 concepts.

29 The project 'demonstration' and its contents do not represent knowledge from a practical design. The author fabricated the project for explanatory purposes only. The project is based on the examples utilised in Chapter 6 to provide continuity.

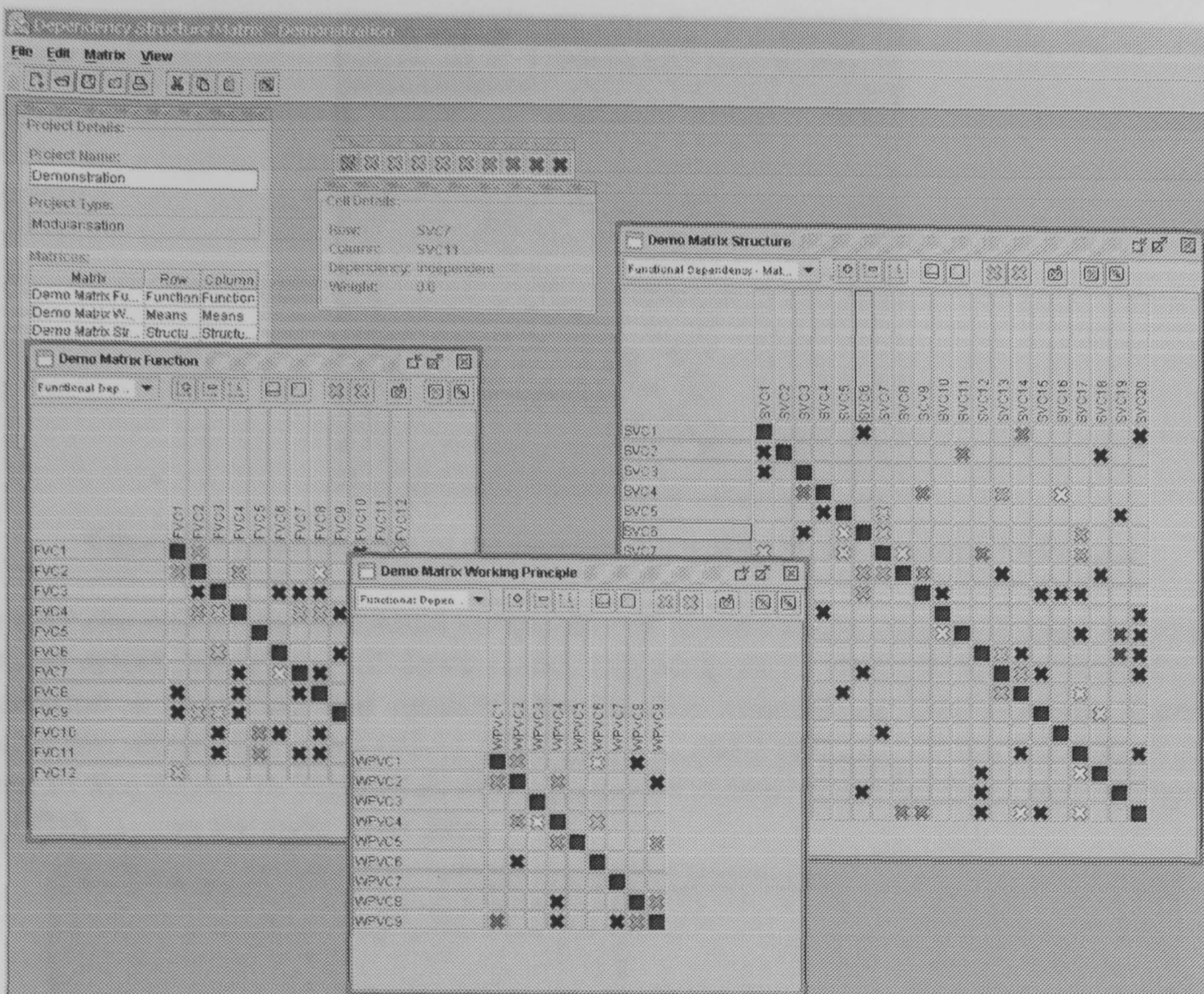


Figure 7.2: Viewpoint matrices in DSM system

The user can use the 'cell details' box, shown in Figure 7.2, to navigate a matrix to aid the process of assigning dependencies. The cell details box provides the user with information as to the row and column *concepts* represented by the intersection, the nature of their dependency (dependent or independent) and the weight of the dependency (independent concepts have a zero weighting). The data within the 'cell details' box changes accordingly as the user scrolls the mouse across a selected matrix. Dependencies may be entered into the matrix body as illustrated in Figure 7.2. Selecting a cell within the matrix will change the state of the dependency from 'independent' to 'dependent' or vice versa. The user may also change the weight of the dependency, which is reflected by its colour (the user can alternate between a colour and greyscale gradient).

Viewpoint perspective creation

Figure 7.3 illustrates the dependency matrix formalism for viewpoint perspectives within the system. The figure depicts the function matrix from the *perspective* of *functional-dependency(material)*. The system displays the CC value within the criteria area (bottom left hand corner of Figure 7.3) and automatically recalculates and updates this when new *concepts* and/or *concept dependencies* are added to the matrix. This allows the user to gain instant feedback as to the effect of a change on the clustering criteria. The pull-down menu illustrates the alternative perspectives available as: *functional-dependency(information), functional dependency(energy,) functional-dependency(combined)*, *similarity-dependence(characteristic)* and *combined all.*

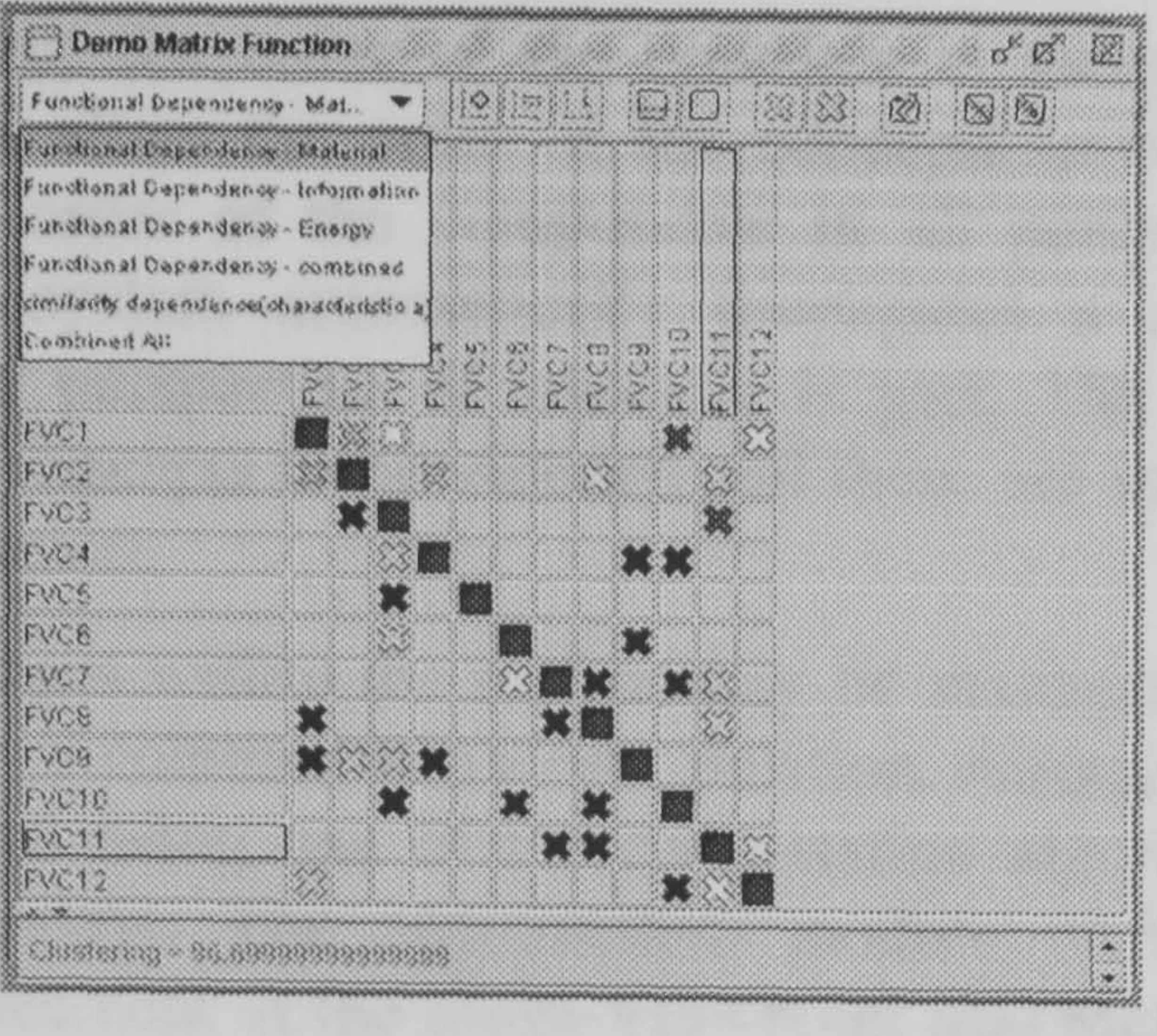


Figure 7.3: A perspective of the Function Viewpoint Model depicting $FDP_{(material)}$.

The user can create any individual *perspective* of interest to the modular design activity. The user can select, using the pull-down menu, any perspective of the *Viewpoint Model* to view or perform a clustering and module identification analysis on. For example, Figure 7.4 depicts two alternative perspectives ($FDP_{(information)}$ and $FDP_{(energy)}$) of the function viewpoint model to that shown in Figure 7.3. $FDP_{(information)}$ is shown on the left-hand side of Figure 7.4 and $FDP_{(energy)}$ on the right hand side.

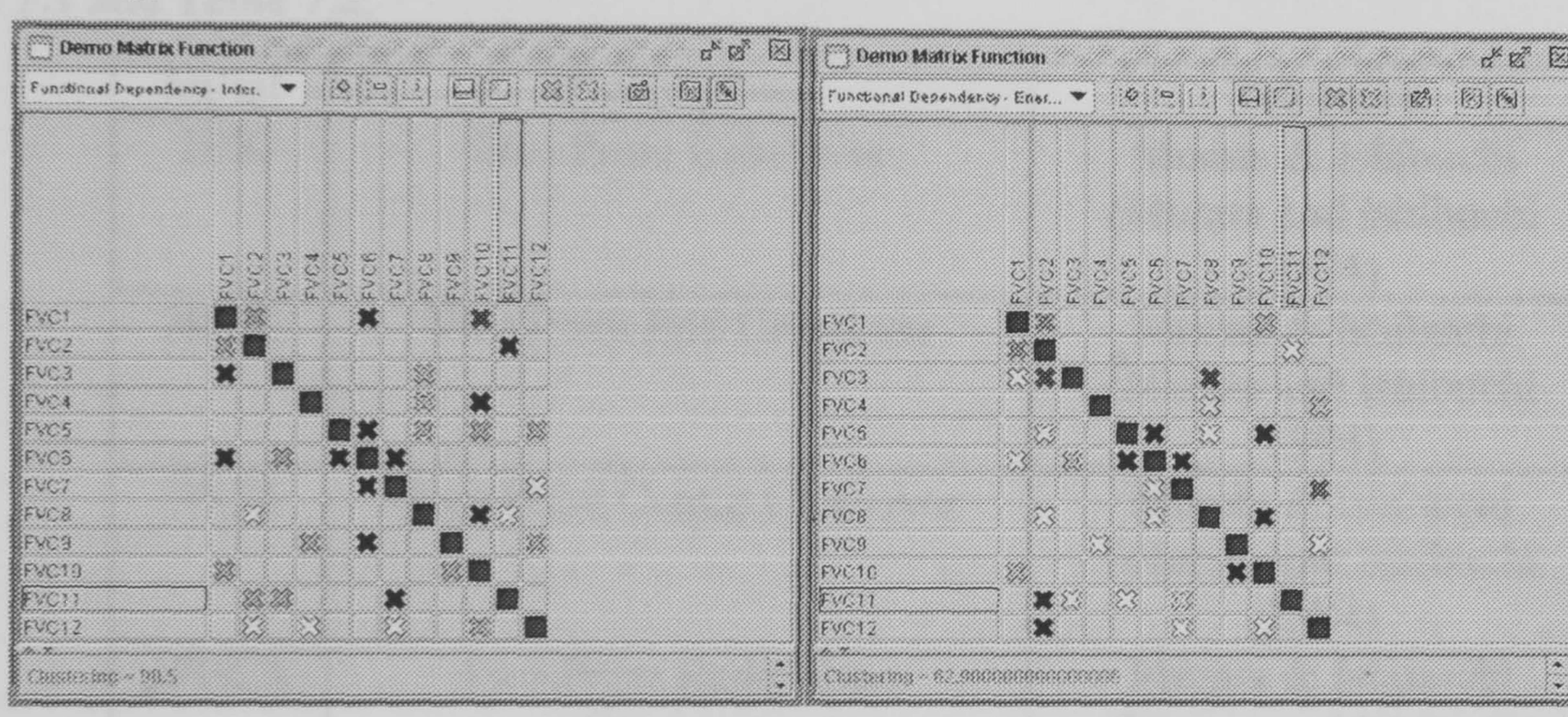


Figure 7.4: The Function Viewpoint Model depicting $FDP_{(information)}$ and $FDP_{(energy)}$ respectively

Figure 7.5 illustrates the combined perspective $FDP_{combined}$ that combines $FDP_{(material)}$ (see Figure 7.3), $FDP_{(information)}$ and $FDP_{(energy)}$ (see Figure 7.4) to depict function concepts and the average of the functional-dependency relations' *material, energy* and *information.*

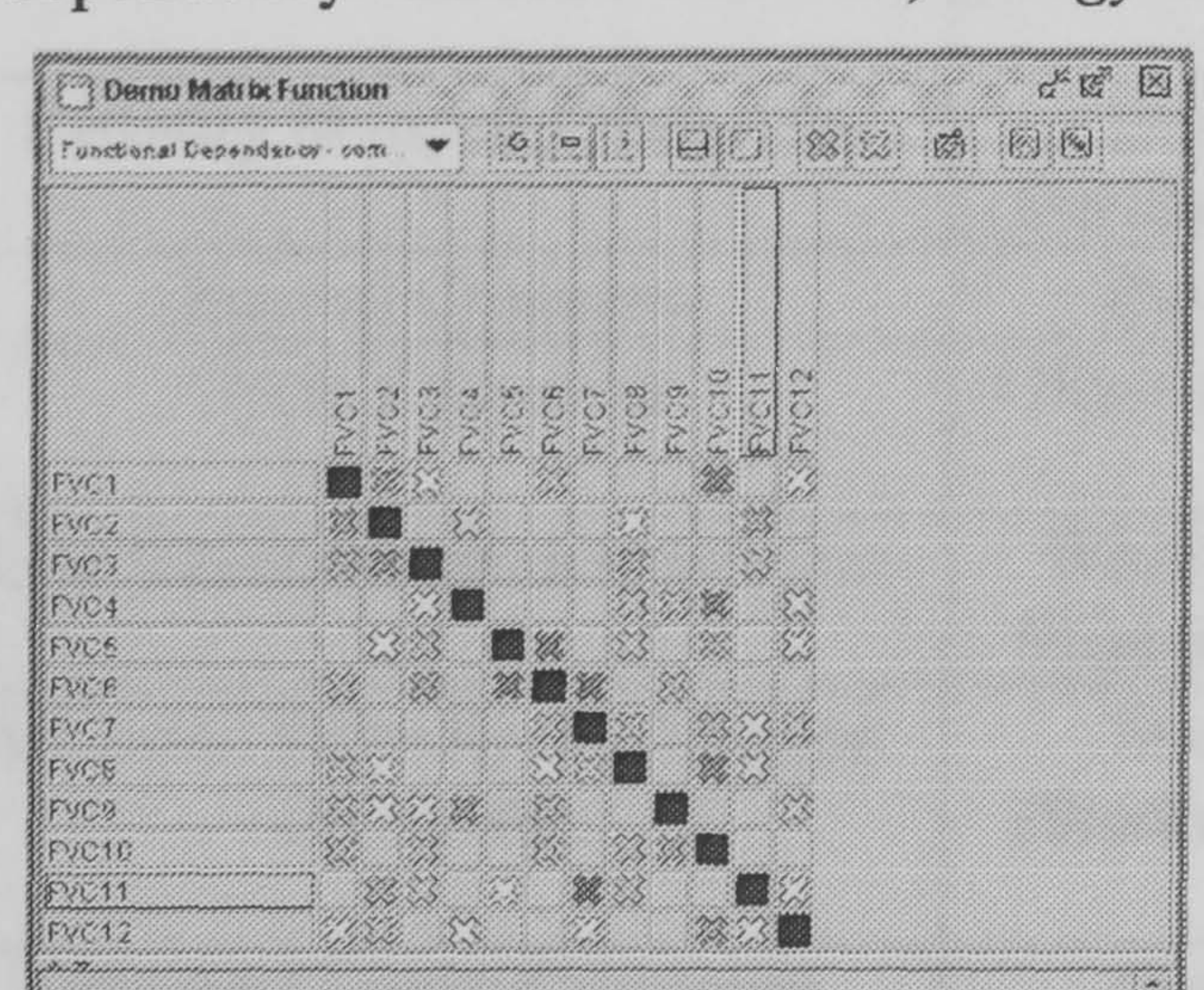


Figure 7.5: The Function Viewpoint Model depicting the combined perspective $FDP_{combined\ (material,\ information\ energy)}$

As discussed in Chapter 6, the order of the concepts is viewpoint and not perspective specific. Thus, if the user alters the order of the concepts in one perspective the system simultaneously alters the order of the components in all perspectives of that Viewpoint Model. In addition, the system automatically propagates a change to any individual perspective any combined perspective of which it is part. The system will not permit a dependency between two concepts to be violated and these are maintained regardless of the alterations in concept sequence.

The sequence of the concepts within the matrix may be managed manually, by dragging a concept in either the rows or columns into a new position. Alternatively, the sequence of the *concepts* may be optimised using one of the optimisation algorithms available within the system's optimisation module. The optimisation module of the system embodies the *Optimisation Mechanism* element of the Multi-Viewpoint MD Methodology.

7.1.3 Optimisation Mechanism

The GA in the system was programmed for its previous application in process modelling. The system's GA is generic in nature using object-oriented techniques and allows the encoding of a sequence of any type of *concept*.

As part of the GA application process crossover and mutation operations are performed. A number of crossover and mutation operations are encoded within the GA as listed within Table 7.1 and Table 7.2.

Initial	Description	Reference
1PX	One Point Crossover	Murata & Ishibuchi (Murata and Ishibuchi 1994)
2PEX	Two Point End Crossover	Murata & Ishibuchi (Murata and Ishibuchi 1994)
2PCX	Two Point Centre Crossover	Murata & Ishibuchi (Murata and Ishibuchi 1994)
2PECX	Two Point End/Centre Crossover	Murata & Ishibuchi (Murata and Ishibuchi 1994)
PBX	Position Based Crossover	Syswerda (Syswerda 1991)
IPX	Independent Position Crossover	Murata & Ishibuchi (Murata and Ishibuchi 1994)
PMX	Partially Mapped Crossover	Goldberg & Lingle (Goldberg and Linge 1985)
OX	Ordered Crossover	Davis (Davis 1985)
CX	Cycle Crossover	Oliver et al. (Oliver, Smith et al. 1987)
ERX	Edge Recombination Crossover	Whitley et al. (Whitley, Starkweather et al. 1989)
EERX	Enhanced Edge Recombination Crossover	Starkweather et al. (Starkweather, McDaniel et al. 1991)
SCX	Subtour Chunks Crossover	Greffenstette et al. (Greffenstette, Gopal et al. 1985)

AEX	Alternating Edges Crossover	Greffenstette et al. (Greffenstette, Gopal et al. 1985)
IX	Inversion Crossover	Goldberg (Goldberg 1989)

Table 7.1 Crossover operators encoded within GA.

Initial	Description	Reference
2ORS	Two Operation Random Swap	Murata & Ishibuchi (Murata and Ishibuchi 1994)
2OAS	Two Operation Adjacent Swap	Murata & Ishibuchi (Murata and Ishibuchi 1994)
3ORS	Three Operation Random Swap	Murata & Ishibuchi (Murata and Ishibuchi 1994)
3OAS	Three Operation Adjacent Swap	Murata & Ishibuchi (Murata and Ishibuchi 1994)
SOM	Shift Operation Mutation	Murata & Ishibuchi (Murata and Ishibuchi 1994)

Table 7.2 Mutation operators encoded within GA.

As discussed previously (Section 6.2.1) research has demonstrated that the parameters for the GA are intrinsically tied to the domain. For the MD application domain the optimum parameters for the GA structure have been determined to be *Two Point End* crossover and *Two Point Adjacent* mutation operators with probabilities of 80 and 20 respectively (Whitfield, Smith et al. 2002). The parameters for the genetic algorithm and the optimisation criteria may be selected using the optimiser dialog shown within Figure 7.6[30]. As illustrated in the figure, the population size, number of generations, crossover and mutation probability may be entered in the text field and the required crossover and mutation operators are selected from pull down menus.

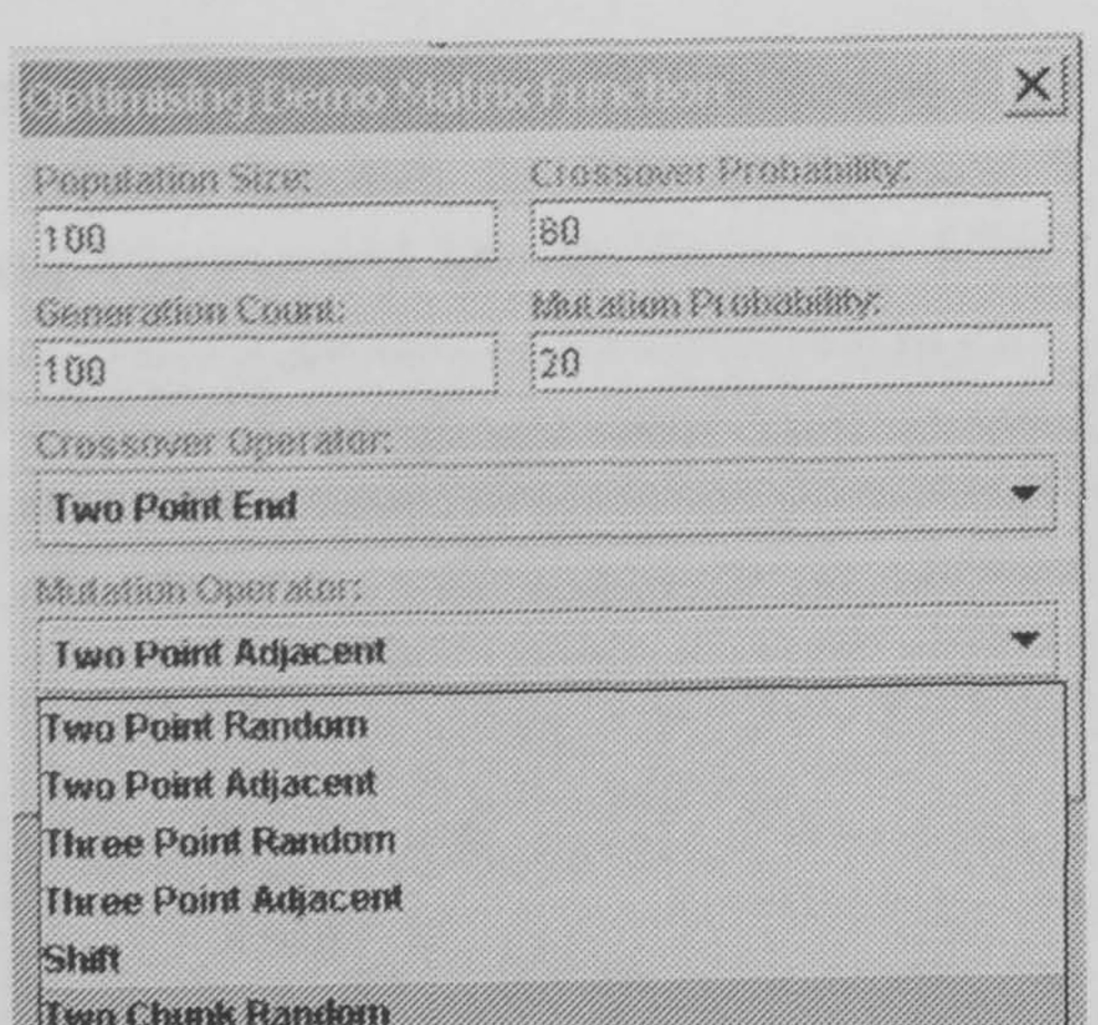


Figure 7.6: Optimiser dialog.

30 Figure 7.6 depicts an optimiser dialog box from the system with the crossover and mutation parameters optimised for the domain, i.e. set to two point end and two point adjacent with probabilities of 80 and 20 respectively.

The system is capable of single or multi-criteria optimisation where the individual objectives may be minimisation, maximisation, target value, or any combination. However, for the purposes of the work presented in this thesis the optimisation is based on a single criteria with the individual objective of minimisation. Figure 7.7 illustrates the criteria and objective selection for the modular analysis problem. Once the structure of the GA and the optimisation criteria and objective are chosen an indicator displays the genetic algorithm's progress through the evaluation of the populations. The optimisation may be run for any of the defined *perspectives* of the *Viewpoint Model* including *combined perspectives.*

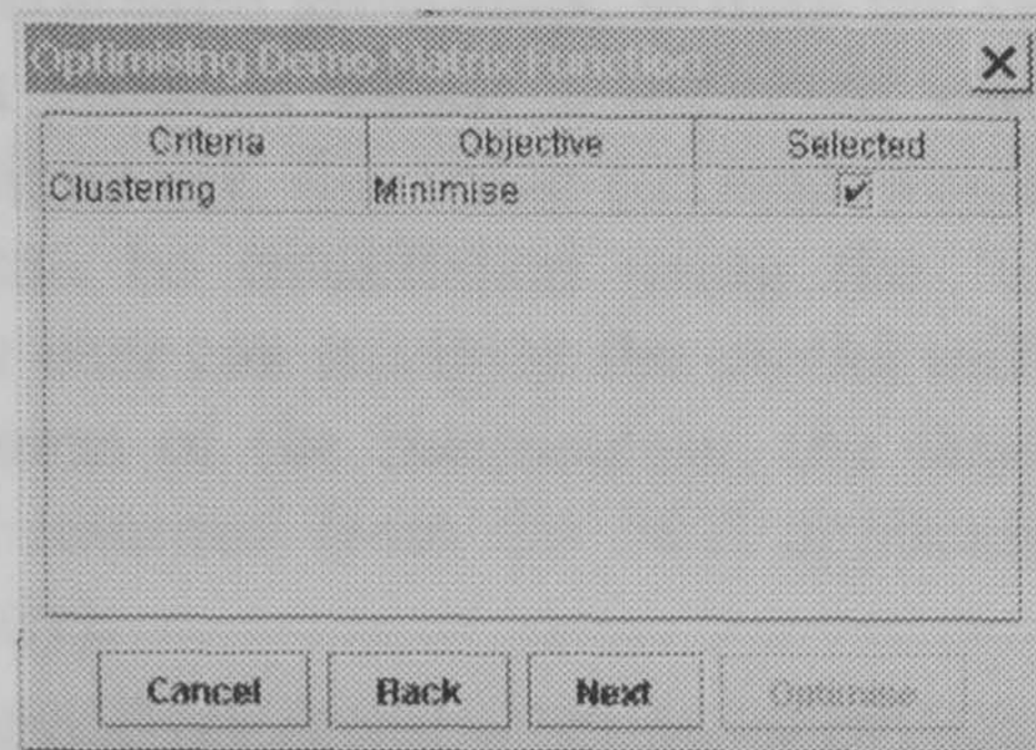


Figure 7.7: Criteria and objective selection

After completion of the optimisation process a solution table is returned which displays a list of near-optimal concept order solutions, the values for the clustering criteria, the fitness and the rank for each of the solutions. Figure 7.8 depicts the solution table for the optimisation of the Function Viewpoint Model from the perspective of $FDP_{combined}$.

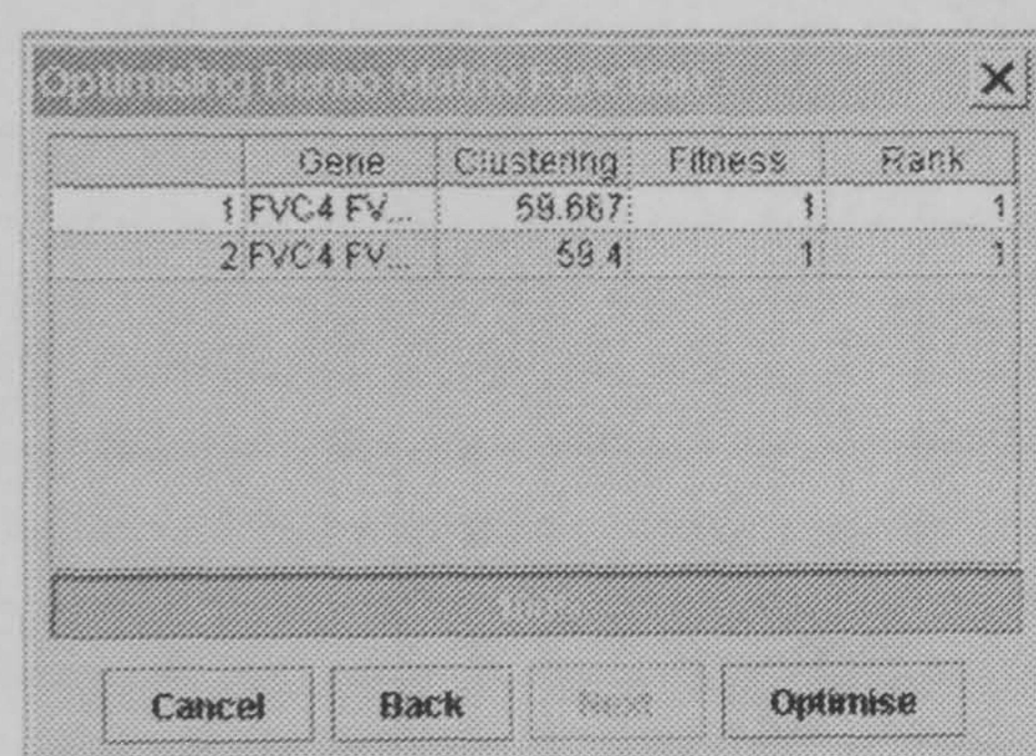


Figure 7.8: The solution table

Selecting one of the optimum solutions within the solution table will display the optimised concept ordering within the DSM, known as the *Optimised Viewpoint Model.* Figure 7.9 depicts the Optimised Function Viewpoint Model from the $FDP_{combined}$ perspective.

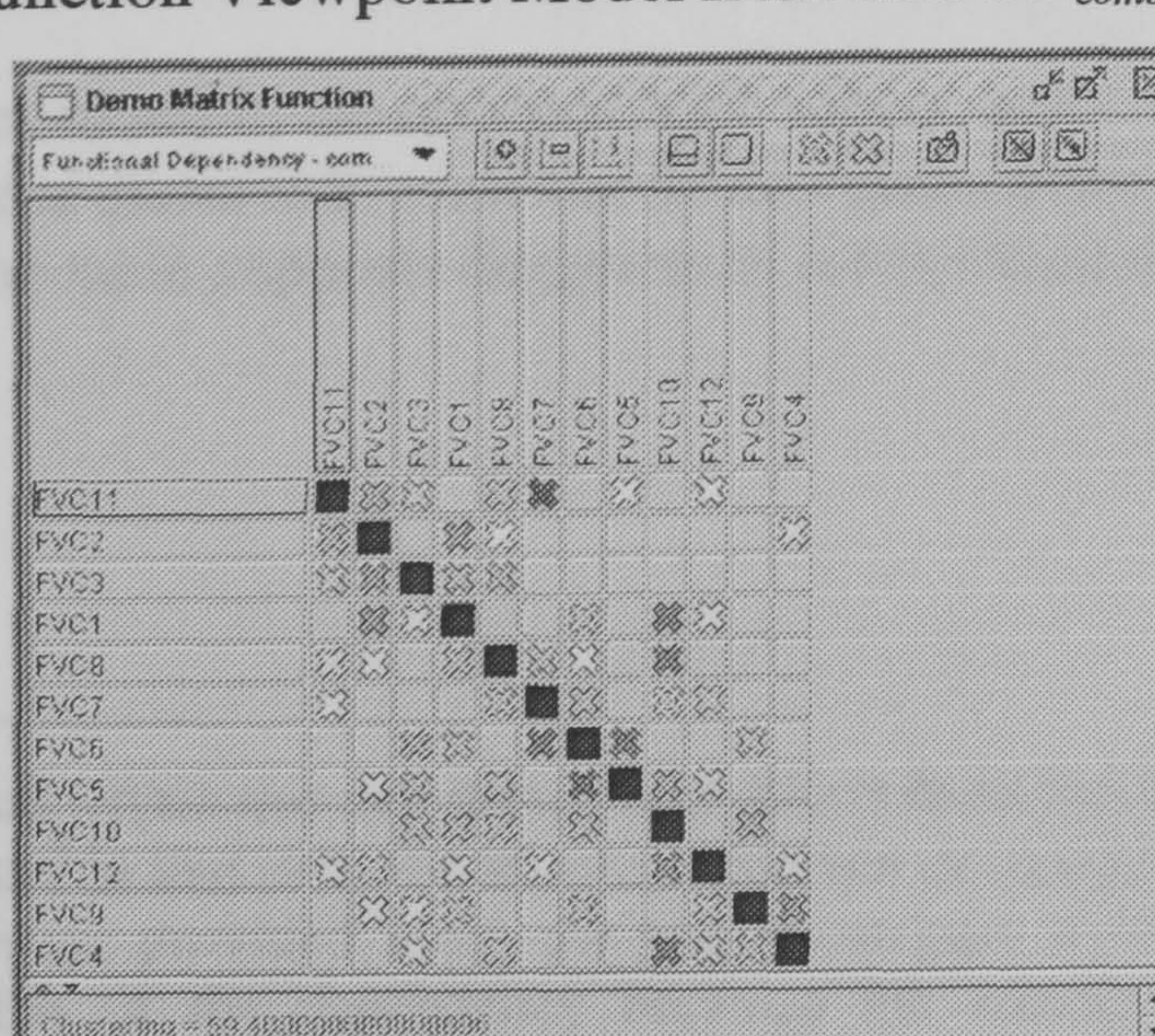


Figure 7.9: Function Viewpoint Model optimised from the $FDP_{combined}$ perspective.

To view the effect of the optimisation of one *perspective* on that of another the user selects the *perspective* of interest from the pull-down menu (see Figure 7.2).

7.1.4 Module Identification Mechanism

On selecting an Optimised Viewpoint Model, the user can choose to take an alternative view of the model, termed the Modular Structure Viewpoint Model. Figure 7.10 depicts the MSM representation of the Optimised Function Viewpoint Model $FDP_{combined}$. The MSM representation is achieved when the values returned from the application of the MSI function are graphically interpreted as coloured groupings. Within the system the user is not explicitly aware of the *MSI* application process, which is triggered when they choose to change the representation to the *MSM*. However, the value (termed weight in the dialog box) returned from the MSI application can be established using the 'Cell Details'[31] dialog box as illustrated in Figure 7.10. The user can navigate the model utilising the box to determine the *concepts* in the row and column of the intersection, the state of the dependency between these and the weight (value) returned from the MSI application (this value is rounded up when colouring the matrix body).

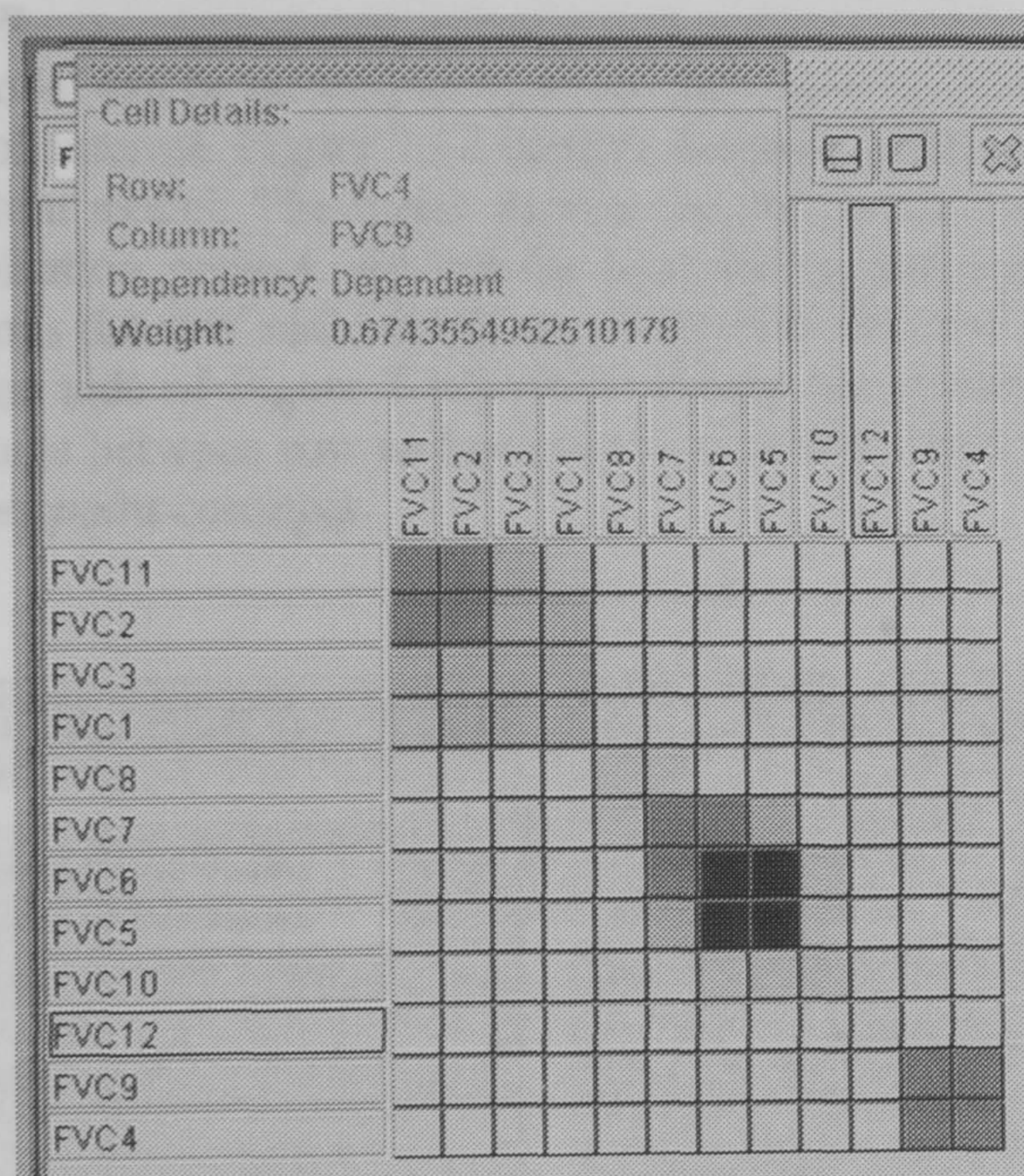


Figure 7.10: Modular Structure Viewpoint Model (Function $FDP_{combined}$)

In addition, a list modules based in the returned MSI values, and the concepts within them, is also returned when the user chooses to view the MSM representation of a *viewpoint model* as shown in Figure 7.11. The table supports designers in interpreting the MSM.

[31] The information contained within this cell details box differs from that displayed when navigating a *(Optimised) Viewpoint Model* in that only the status (dependant or independent) and not the values of the individual dependencies are provided. Here, the weight category represents the MSI value for the grouping and not the value of the dependency between the *concepts* of the row and column intersection.

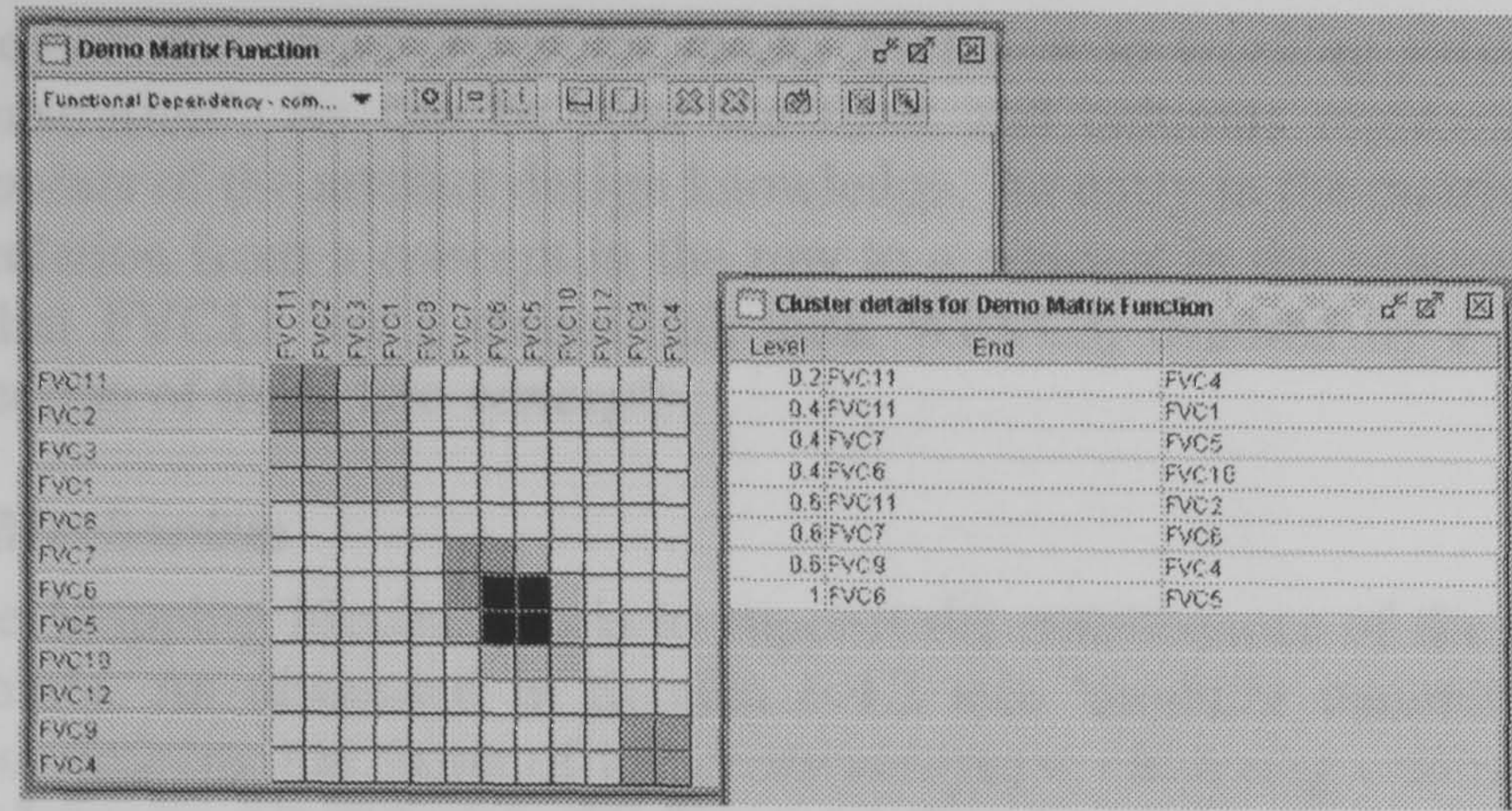


Figure 7.11: Module list (their values and the *concepts* included)

7.1.5 Viewpoint mapping

Modelling Formalism

A cross-viewpoint model can be created to represent the causal relations between *concepts* of any two *viewpoints* of interest. Figure 7.12 depicts the process through which the user can create a cross-viewpoint model. The figure depicts the dialog box for the creation of a new matrix in a modularisation project. As can be seen during the creation of an individual viewpoint the user links the row and column representation (denoted by a chain-link symbol shown of the left-hand side of Figure 7.12) whilst during the creation of a cross-viewpoint matrix the chain-linkage between row and column is broken to allow the row and column to represent different *viewpoint concepts*.

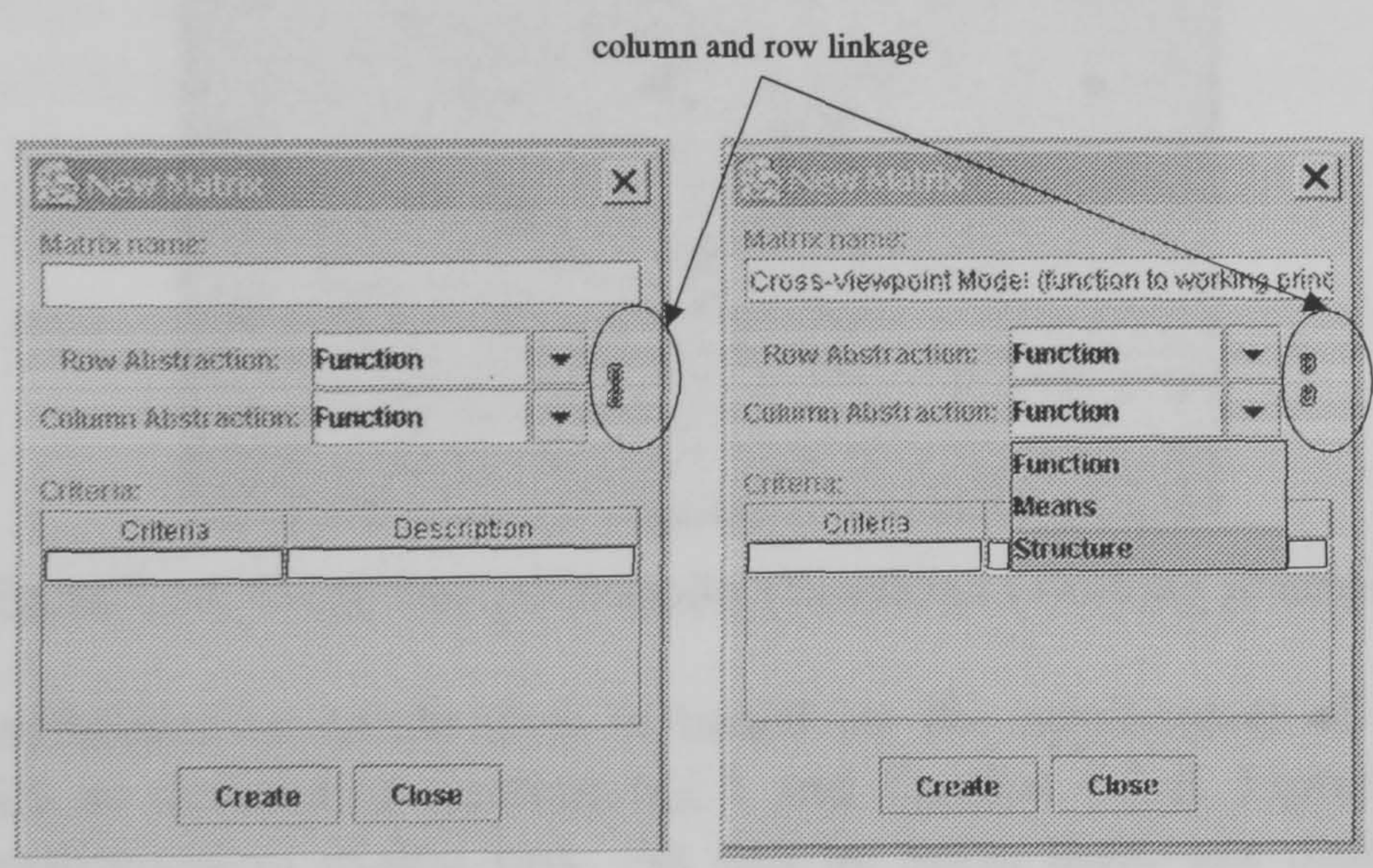


Figure 7.12: Cross-viewpoint model creation

The user can then utilise the pull-down menus (illustrated on the right-hand side of Figure 7.12) to select the *viewpoint* represented by the row and column of the *cross-viewpoint model.* The user can add concepts to either row or column by selecting the appropriate option in dialog box illustrated in Figure 7.13.

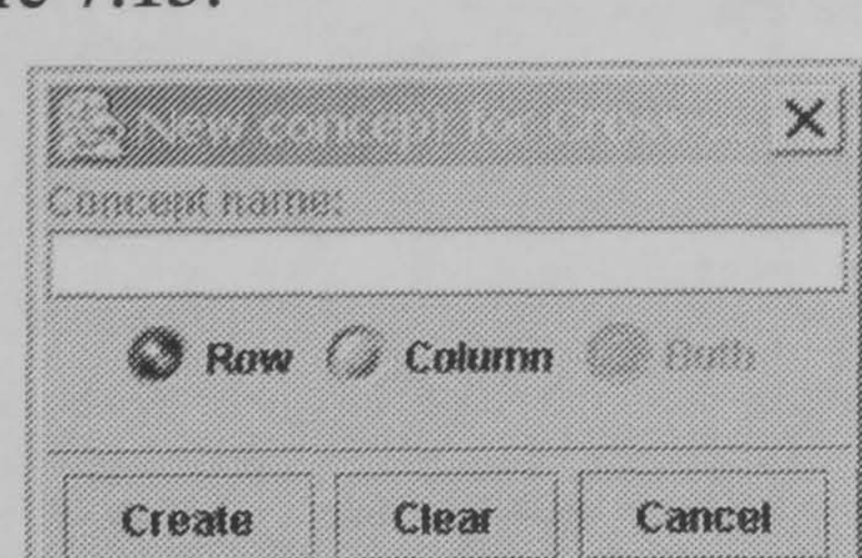


Figure 7.13: Adding a concept in a cross-viewpoint matrix

Figure 7.14 depicts the *cross-viewpoint model (function to working principle)* within the project 'demonstration'. As can be seen the rows and columns represent concepts from different *viewpoints* of the artefact design knowledge. An entry in the matrix body represents a causal-link relation from a *concept* in the row to a *concept* in the column, for example, it can be seen that FVC11 has a causal-link with WPVC3, denoted by a cross in the row column intersection of these two *concepts.*

Optimisation Mechanism

The aim of the Mapping Mechanism is to support the maintenance of the modular solution across viewpoints. As discussed in Section 6.4.2 this 'modular maintenance' process is facilitated by optimising the value of the *cross-viewpoint clustering criteria* (CVCC) in the horizontal direction i.e. $CVCC_h$ (see Section 6.4.2 Equation 6.11). The values of the $CVCC_h$ (Clustering Horizontal) $CVCC_v$ (Clustering Vertical) and the total value for the CVCC (Clustering Both) are shown at the bottom left-hand corner of Figure 7.1and are 11, 6 and 17 respectively.

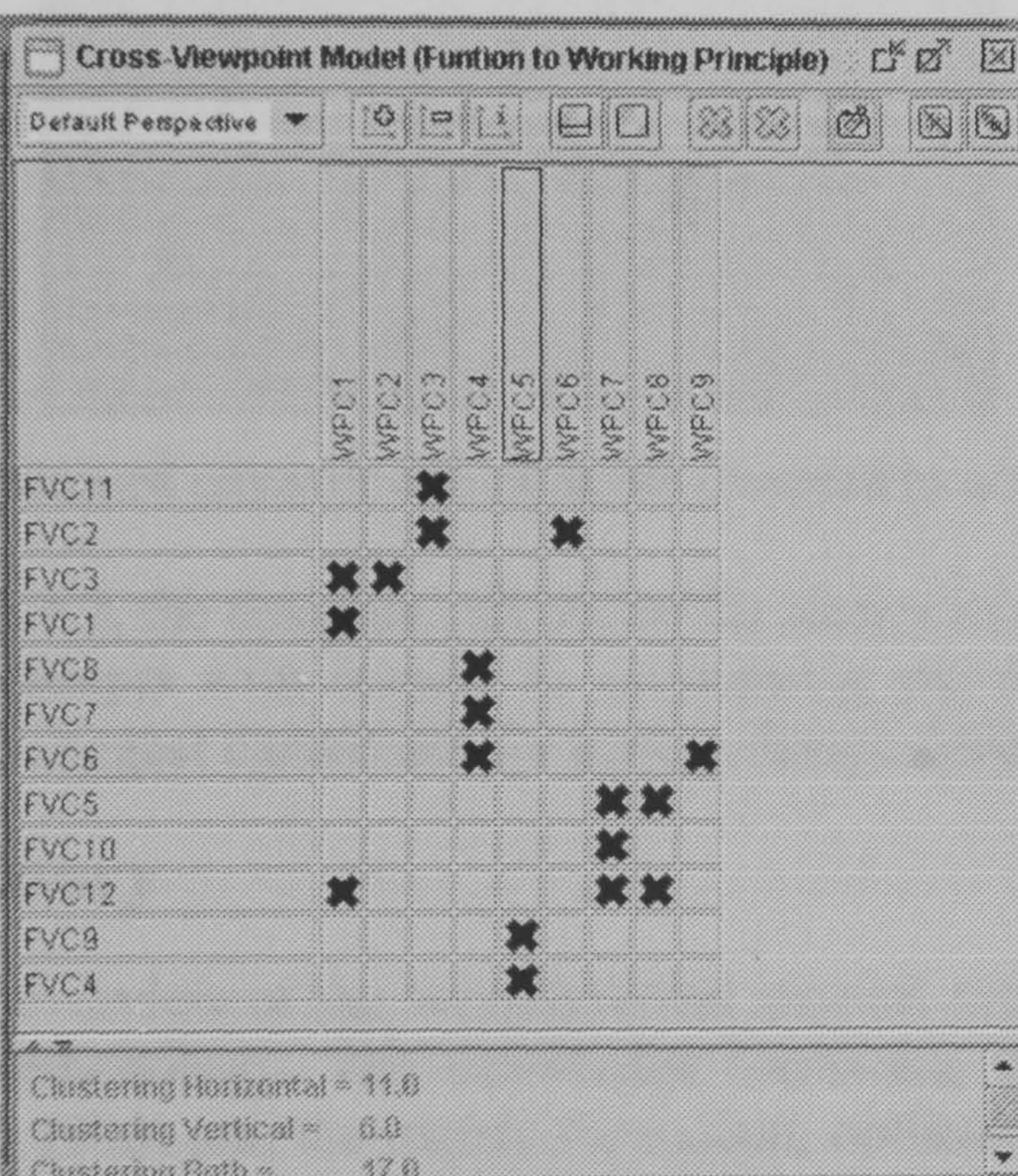


Figure 7.14: Cross-viewpoint model (function to working principle)

The mappings' optimisation mechanism is based on the application of the same general structure for a GA as defined in Section 6.2.1 and Section 7.1.3. Figure 7.15 depicts the cross-viewpoint optimisation dialog box. As can be seen, there are multiple optimisation criteria defined (as stated previously in Section 7.1.3 the system is capable of multi-criteria optimisation) though for the purposes of this 'modular maintenance' problem the objective is to optimise the $CVCC_h$. As such, $CVCC_h$ is selected (as denoted by the tick in the selection box in Figure 7.15) with the objective of minimisation.

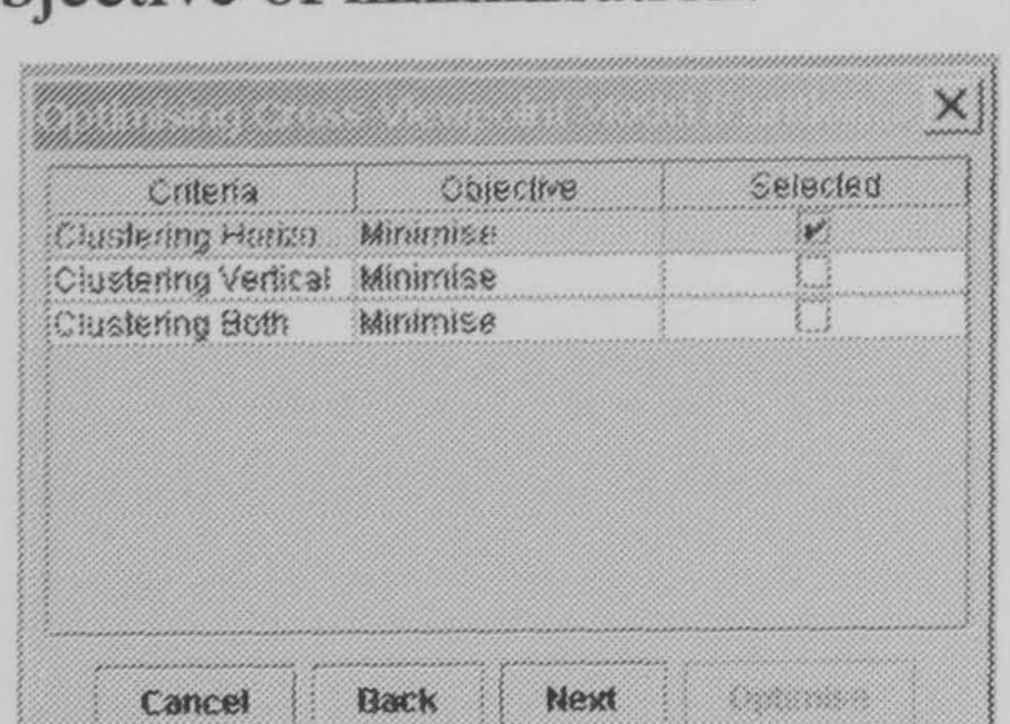


Figure 7.15: Cross-viewpoint optimisation dialog

Again, similar to that discussed in Section 7.1.3, the optimisation mechanism returns a solution table, of near-optimal *concept* order solutions, from which the user can select an appropriate model. Figure 7.16 depicts the *optimised cross-viewpoint model (function to working principle)* selected from those returned by the mapping elements' optimisation mechanism.

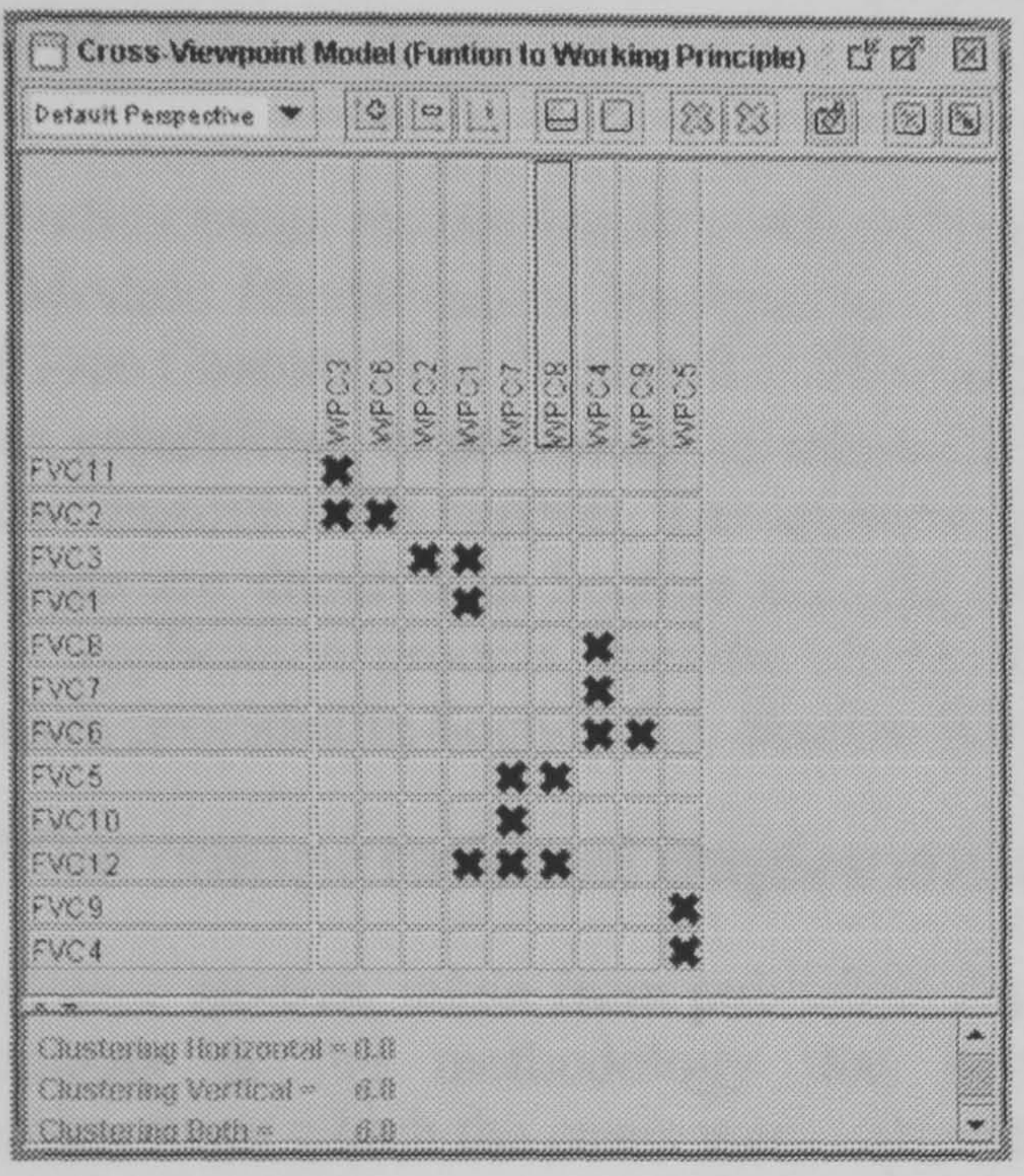


Figure 7.16: Optimised cross-viewpoint model (function to working principle)

As discussed in Section 6.4.2, the designer can utilise the matrix to assess the performance of the *working principle concepts* with respect to maintaining the modularity defined in the *function viewpoint* and to suggest candidates for further design development work.

7.2 Chapter summary

The chapter provided an overview of the computational support available for elements of the Multi-Viewpoint MD Methodology. The knowledge formalism is supported by a system named DENOTE (Zhang 1998) and the matrix formalism, optimisation mechanism, module identification mechanism and elements of the mapping mechanism are embodied in a DSM modelling and analysis system. A number of additions and alterations have been made to the DSM system, originally defined for process optimisation, to allow it to support the methodology. The two systems are currently separate, though future work includes the integration of these. The DSM system was developed to support the methodologies evaluation in a number of case studies.

8 Evaluation

The computational realisation of the methodology was discussed throughout Chapter 7. Chapter 8 utilises the computational implementation to aid the evaluation of the methodology through four studies, two extracted from previously published literature (Pimmler and Eppinger 1994; Sosale, Hashemiam et al. 1997) and two industrial based evaluations.

The two literature based studies focus on the initial verification of the 'clustering criteria' and the elements of the 'Module Identification Mechanism', which are embodied by the computational support tool (see Chapter 7) in Section 8.1. The industrial studies evaluate the overall methodology and its capabilities in providing an articulate procedure to support MD in practice and consequently improve EDR support. The application and resulting findings of the industrial implementations are detailed in Section 8.2. An the overall evaluation with respect to the methodology application process, specific elements of methodology and the ITM designers requirements is provided in Section 8.3. Section 8.4 summarises the chapter.

8.1 Verification of the computational implementation -

The following presents two case studies, taken from published literature, aimed at verifying the utility of those elements of the methodology that have been computationally implemented. The publications from which the cases were extracted both contained findings that included a list of modules and their constituent parts. As such, it is suggested that an evaluation of the published results against those obtained by the Multi-Viewpoint MD Methodology application provides a basis for the initial verification of the clustering and module identification elements of the methodology. Thus the methodology elements, embodied within the computational support tool, were applied to evaluate:

- the effectiveness of the defined clustering criteria as a means to cluster *concepts* such that it provides a model from which to interpret modularity,
- the appropriateness of the MSI function (and its application process) as a means to measure the modularity of *concept* clusters within the product structure, and
- the utility of the MSM representation of the DSM.

Section 8.1.1 discusses the application of the computational realisation of the methodologies matrix formalism, optimisation mechanism and module identification mechanism to two case studies extracted from published literature(Pimmler and Eppinger 1994) (Sosale, Hashemiam et al. 1997). The results and their evaluation is presented in Section 8.1.2.

8.1.1 Application

The two studies represent a climate control system (Pimmler and Eppinger 1994) and, an alternator (Sosale, Hashemiam et al. 1997). The cases presented cover the 'structural' viewpoint of design i.e. parts and components. The climate control example depicts the components and their dependencies from a material perspective. The alternator case depicts dependencies based on the objectives of the re-use[32] and recycling life-phase of the artefact life.

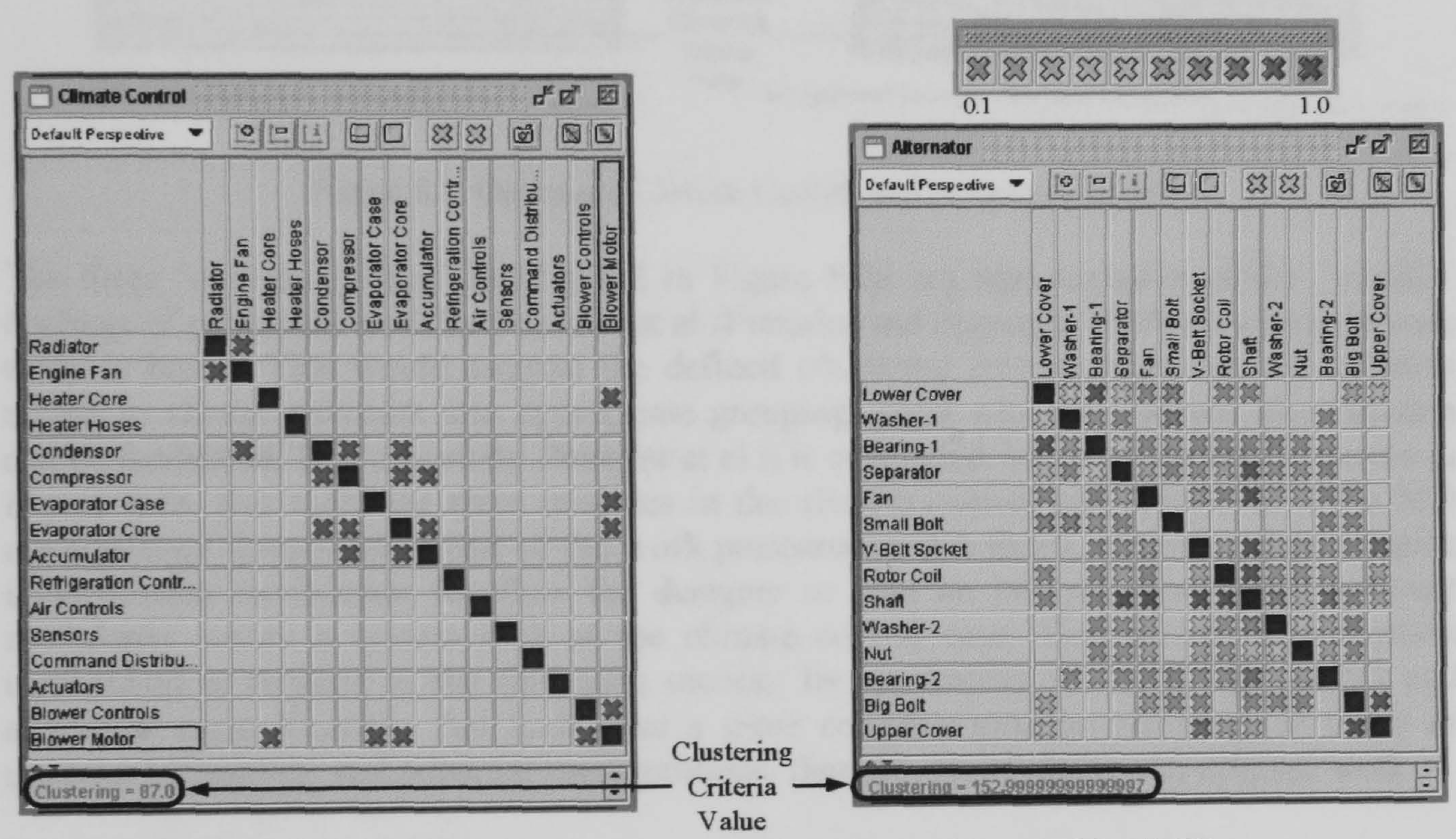
[32] Here, re-use refers to the use of material and/or component parts obtained from the artefact after retrial and not design knowledge re-use as defined in Chapter 2 of this thesis.

Viewpoint models

Figure 8.1a depicts the *Structure Viewpoint Model FDP* $_{(material)}$ for the climate control system and Figure 8.1b the *Structure Viewpoint Model FDP* $_{(reuse\ and\ recycling\ characteristics)}$ for the alternator case.

Each system's parts are represented by an arbitrary sequence. In both cases the sequence is the same in both row and column. The crosses in the row and column intersections of the matrix represent the dependencies between the components. In the case of the climate control system all the dependencies are equally weighted (with a value of 1.0, depicted as red crosses) whilst the climate control case has different weighted dependencies (represented by the range of coloured crosses). The dependency box above the climate control case depicts the coloured dependency weight range with the weight of the dependencies increasing in incremental values of 0.1 from 0.1 to 1 (with the left-hand most dependency (green) being 0.1 to the right-hand most dependency (red) being 1.0).

The current value for the clustering criterion is shown in the left-hand bottom corner of the box. The value for the clustering criterion for the climate control system is 87 and for the alternator is 153. The alternator case represents a more highly constrained problem than that of the climate control system, as indicated by the more densely populated matrix.



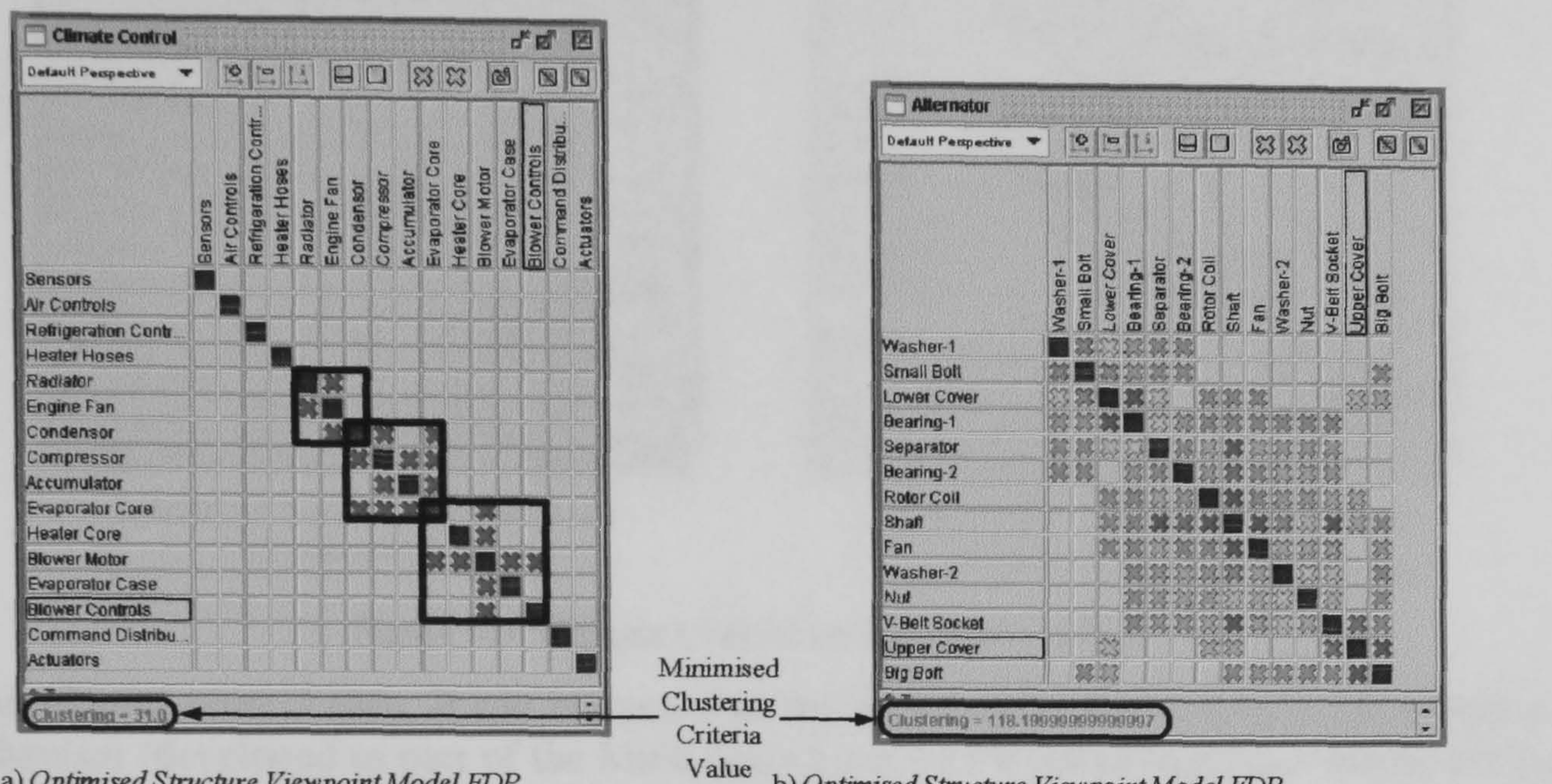
a) *Structure Viewpoint Model FDP* $_{(material)}$ b) *Structure Viewpoint Model FDP* $_{(reuse\ and\ recycling\ characteristics)}$

Figure 8.1: Climate Control and Alternator Viewpoint models

Optimisation Mechanism

The optimisation procedure is run for both examples with the population and generation size both set at 100. The crossover and mutation operator parameters were set at 'two point end' with a probability of '80' and 'two point adjacent swap' with a probability of '20' respectively.

Figure 8.2a depicts the *Optimised Structure Viewpoint Model FDP* $_{(material)}$ for the climate control problem and Figure 8.2b the *Optimised Structure Viewpoint Model FDP* $_{(re\text{-}use\ and\ recycling\ characteristics)}$ for the alternator problem. The clustering criterion has been reduced by approximately 65% (from 87 to 31) and 23% (from 153 to 118) respectively. It is also apparent that the dependencies within the matrix are now closer together and are clustered into a number of groupings around the diagonal, indicating potential modules. It can be seen from Figure 8.2a that there are three potential module groupings in the climate control example (depicted by the overlaid boxes). However in the case of the alternator (Figure 8.2b) it is difficult to clearly identify any significant component groupings within the optimised DSM without further consideration.



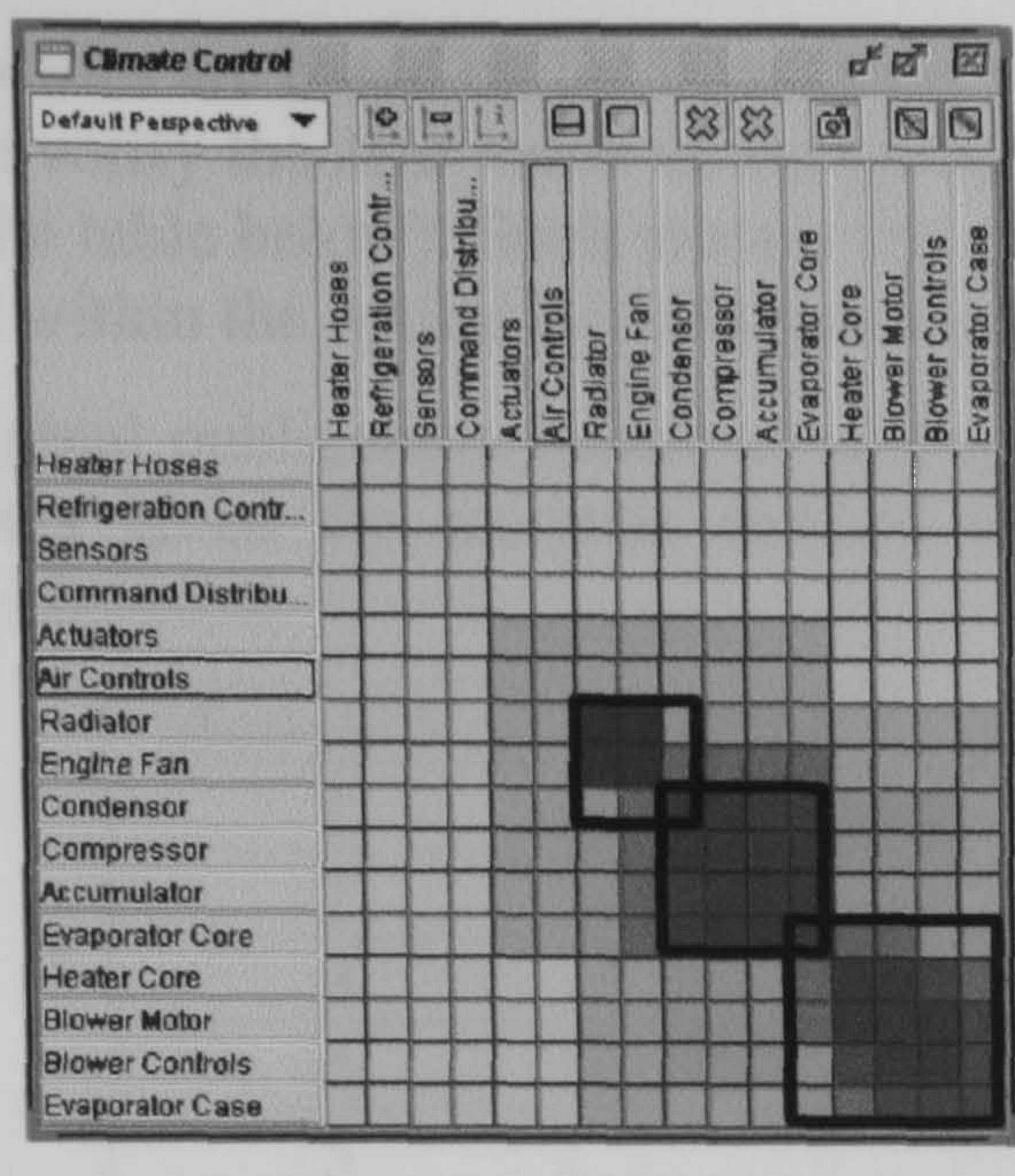
a) *Optimised Structure Viewpoint Model FDP* $_{(material)}$ b) *Optimised Structure Viewpoint Model FDP* $_{(reuse\ and\ recycling\ characteristics)}$

Figure 8.2: Optimised Climate Control and Alternator DSM

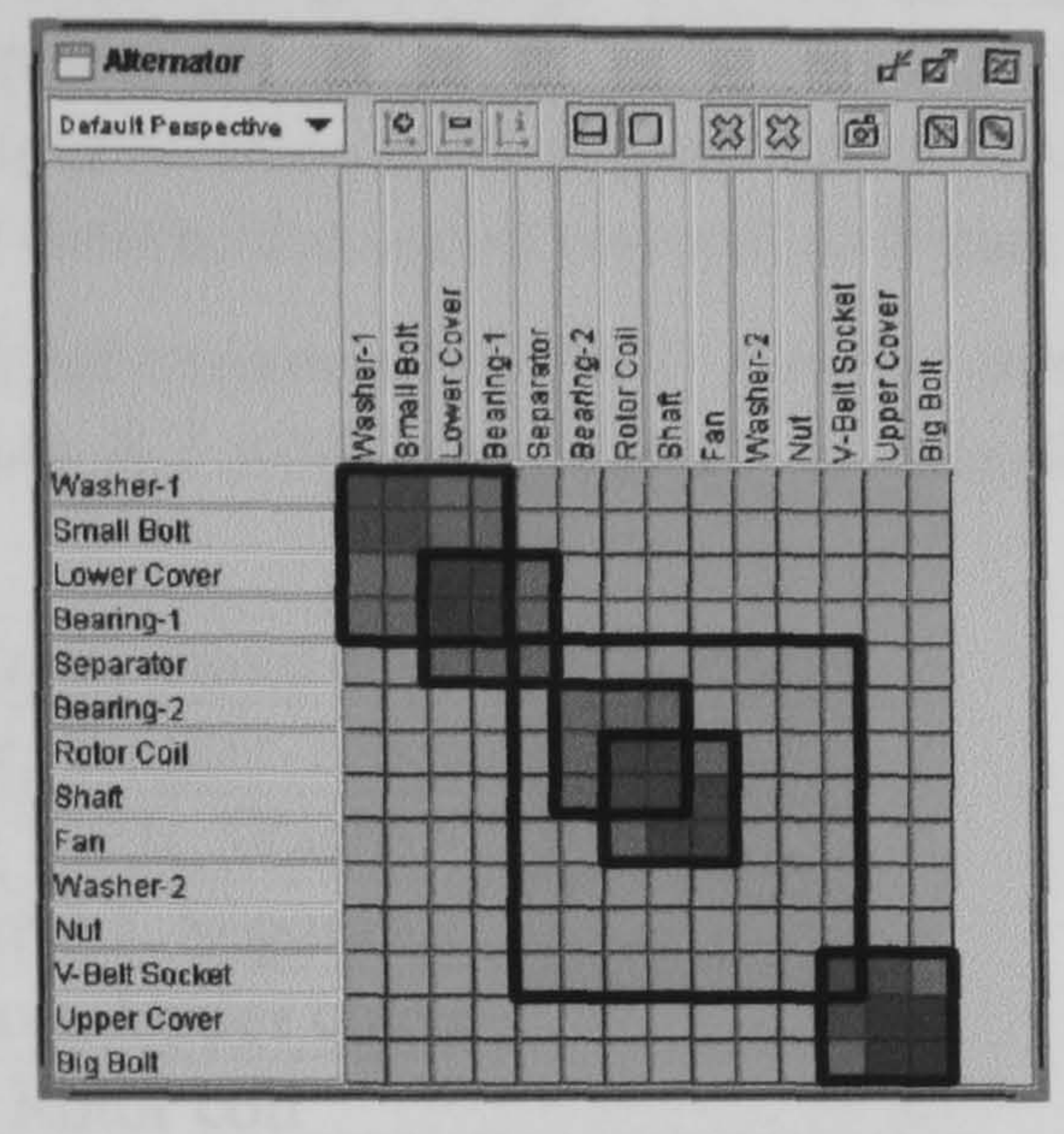
The three 'concept groups' highlighted in Figure 8.2a are representative of the 'module' findings of published work by Pimmler et al (Pimmler and Eppinger 1994) on which the case study is based. This would suggest the defined *clustering criteria* represents an effective means to cluster *concepts* into appropriate groupings from which to module identification can be facilitated. In the work by Pimmler et al it is concluded, based on the matrix shown in Figure 8.2a, that there are three modules in the climate control case. However it the MD methodology, presented as part of the work presented in this thesis, utilises a novel module identification mechanism to allow the designer to gain an interpretation of the inherent modularity within a system such as the climate control case. This module identification mechanism is detailed in the following section. Its application to the climate control and alternator cases illustrate that both have a more complex modular structure, in terms of modular hierarchies and potential configurations, than that concluded in the original work.

Module Identification Mechanism

The MSI function (see Equation 6.7) is applied to both cases (following the previously discussed procedure – see Section 6.3.1) and the resulting *MSM* representations are depicted in Figure 8.3. The climate control *Modular Structure Viewpoint Model(Structure* $FDP_{material}$*)* is shown in Figure 8.3a and the alternator *Modular Structure Viewpoint Model (Structure* $FDP_{re\text{-}use\ and\ recycling\ characteristics}$*)* for the case in Figure 8.3b.



a) *Modular Structure Viewpoint Model (Structure FDP$_{material}$)*



b) *Modular Structure Viewpoint Model (Structure FDP$_{reuse\ and\ recycling\ characteristic}$)*

Figure 8.3: Climate Control and Alternator MSM

In the climate control case, it can be seen that the application of the Module Identification Mechanism (developed as part of the Multi-Viewpoint MD methodology) has reinforced the findings deduced from perusal of the optimised DSM in that it highlights that there are 3 to 4 main component groupings (modules) within the system (depicted by the overlaid boxes in Figure 8.3a). However, in the alternator case the application of the *Module Identification Mechanism* has illustrated that there are actually a number of strong module candidates within the artefact (a sample of which are depicted within the overlaid boxes in Figure 8.3b).

8.1.2 Results and evaluation

Table 8.1 catalogues the potential modules within the alternator structure at the differing module strengths highlighted by the MSM.

Modular Strength	Module	Components
1.0	M_1	Upper Cover, Big Bolt
0.9	M_2	Bearing-1, Lower Cover
0.8	M_3	Rotor Coil, Shaft
	M_4	Shaft, Fan
0.7	M_5	Washer-1, Small Bolt
	M_6	V-Belt Socket, Upper Cove
0.6	M_7	Components within M_2 and M_5
	M_8	Components within M_2 and Separator
	M_9	Components within M_3 and Bearing-2
	M_{10}	Components within M_4 and Rotor Coil
	M_{11}	Components within M_1 and M_6
0.5	M_{12}	Components within M_7 and M_8
	M_{13}	Components within M_9, M_{10} and Separator, Nut, V-Belt Socket
0.4	M_{14}	All Components
	M_{15}	Components within M_{11}, M_{12} and Bearing 1
	M_{16}	Components within M_{11}, M_{12} and Upper Cover
0.3	M_{17}	All

Table 8.1: Module catalogue for Alternator.

These modules are based on the boundaries of the coloured groupings. The cell details box is utilised to verify the *MSI* value of the groupings (these are rounded to the nearest 1 decimal place in the table below). Such modularity is not immediately evident from the dependencies displayed within the optimised DSM as originally determined and illustrated in Figure 8.2b.

In the original publication, Sosale et al (Sosale, Hashemiam et al. 1997) claimed that based on the data given the modules defined in Table 8.2 could be formed to support their recycling objective.

Module No.	Components
1	Lower Cover, Washer-1, Small bolt
2	Fan, V-belt, Shaft, Washer-2, Nut
3	Bearing-1, Separator
4	Big bolt, Upper cover
5	Rotor coil
6	Bearing-2

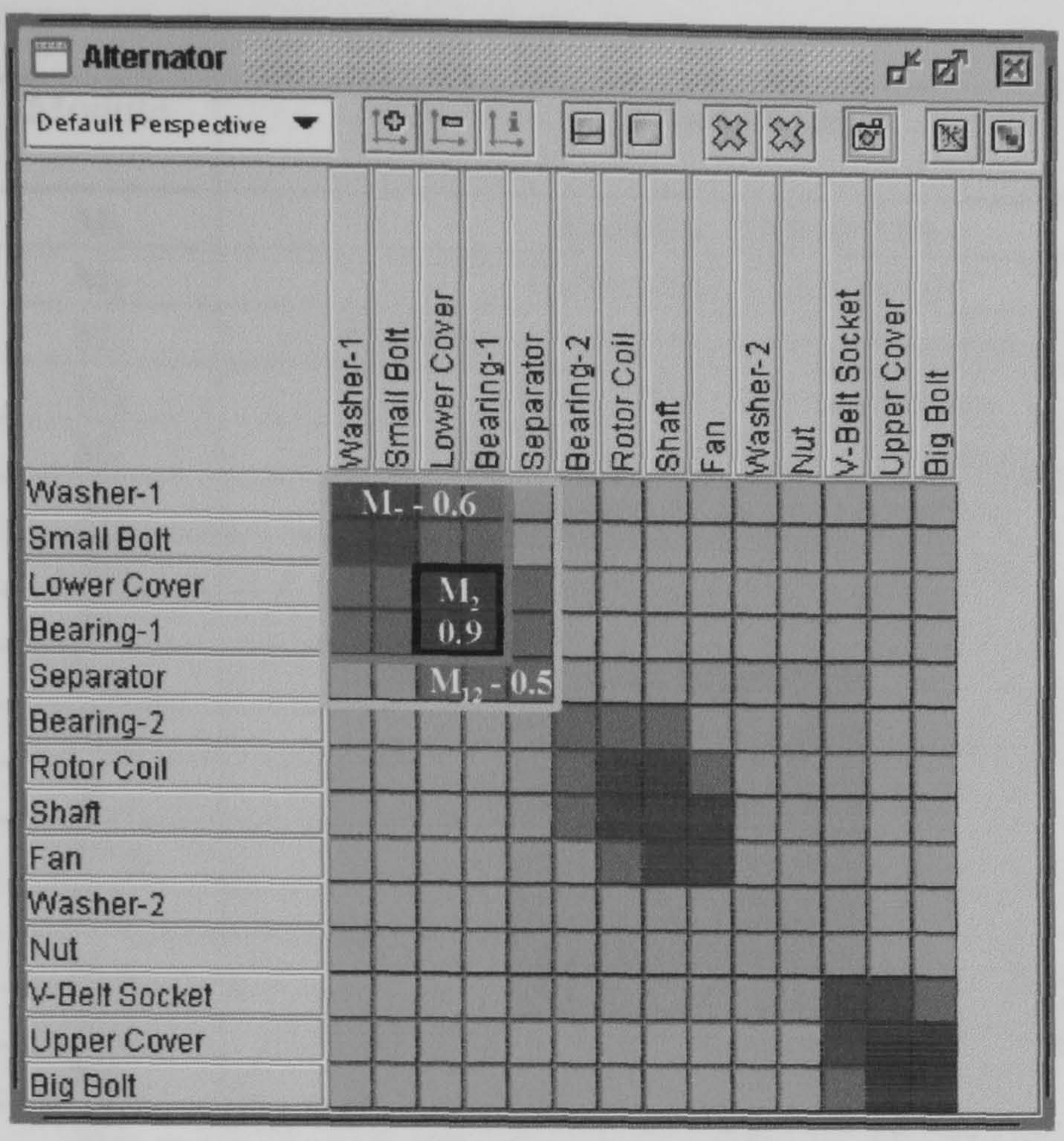
Table 8.2: Modules formed for recycling objective (Sosale, Hashemiam et al. 1997)

Sosale et al go on to explain the module groupings, which are based on the application of further domain knowledge, such as:

'the reason for separating the component (rotor) is that the rotor has copper windings which are worthwhile to recover and has high recyclability'

However, the model shown in Figure 8.2b represents the dependency data provided by the publication (where a similar matrix is presented). It can be clearly seen from Figure 8.2b that the rotor coil has strong dependencies with, amongst others, the 'shaft' component and has therefore clustered with additional components as reflected in both Figure 8.3 and Table 8.3) for example, the 'rotor coil' component being grouped into M_6 (a module with a relatively strong *MSI* value 0.7 and containing components 'shaft' and 'rotor coil'). This may be attributed to that the original dependency data that could have been defined to better reflect the rotor coils re-use and recycling characteristics.

The application of the *Module Identification Mechanism* has resulted in the exposure of the inherent hierarchical modularity within both cases. However, the findings of both the original publication were presented in a list structure as illustrated in Table 8.2(Sosale, Hashemiam et al. 1997). The list structure represents each separate module and its constituent components. The modules are standalone entities, each containing a set of components. The components are a member of one module only. However, Figure 8.3 illustrates that the modularity in both examples exists over different hierarchical levels of the product structure. For example, if we take the components 'bearing-1' and 'lower cover' we see from Figure 8.4 that they form a grouping with a high *MSI* value (depicted by the black box in Figure 8.4, module M_2, value 0.9 – see Table 8.3), i.e. they represent a strong module candidate. However, it is also clear that the two components group into another module candidate with the components 'washer-1' and small bolt' albeit with a lower *MSI* value (depicted by the dark grey box in Figure 8.4, module M_7, value 0.6). Again the components in M_7 can be seen to group into another module candidate M_{12} (depicted by the light grey box in Figure 8.4, *MSI* value 0.5) with the additional component 'separator'. Thus, modules are not stand alone entities and can be seen to form part of other modules at different hierarchical levels of the identified modularity. The *Module Identification Mechanism* has resulted in an interpretation of the hierarchal modularity of the artefact, which is not previously apparent from the results presented in the original publications.



Modular Structure Viewpoint Model (Structure $FDP_{reuse\ and\ recycling\ characteristic}$)

Figure 8.4: Hierarchical modular structure

In addition, it can be seen from the findings presented in Figure 8.3 and Table 8.3 that unlike in the original publications a component may cluster into more than one grouping. For example, if we take the 'shaft' component it can be seen that it is a member of two groupings with equally strong *MSI* values (M_3 and M_4, value 0.8 – see Table 8.3). Again it can be seen that M_3 and M_4 cluster into groupings M_9 and M_{10} respectively. Thus, unlike in the findings of the original publication, where components were assigned to one particular module, the Module Identification Mechanism results in the exposure of the different modular configurations that exist for the artefact.

The designer is free to adapt the module configuration within the boundaries of the inherent modularity identified in the MSM. Thus, based on the knowledge gained from the MSM, in terms of the module boundaries, differing configurations and hierarchical modularity, the designer can define an appropriate modular structure based on their specific design requirements and domain knowledge. As an example, Table 8.3 catalogues all available modules within the climate control structure at the differing module strengths highlighted by the MSM. Example of the hierarchical modular configurations which the designer may adopt based on the available module catalogue is given are Table 8.4 and Table 8.5 and illustrated in Figure 8.5.

Modular Strength	Module	Components
1.0	M_1	Radiator, Engine Fan
0.9	M_2	Condenser, Compressor
	M_3	Compressor, Accumulator, Evaporator Core
	M_4	Heater Core, Blower Motor
	M_5	Blower Motor, Blower Controls
0.8	M_6	Components within M_2 and M_3
0.7	M_7	Components within M_4 and M_5
	M_8	Components within M_5 and Evaporator Case
0.6	M_9	Components within M_6 and Engine Fan
	M_{10}	Components within M_4 and Evaporator Core
	M_{11}	Components within M_4 and M_8
0.5	M_{12}	Components within M_1 and M_6
	M_{13}	Components within M_6 and M_7
	M_{14}	Components within M_{10} and M_8
0.4	M_{15}	Components within M_{12}, Actuator and Air Controls
	M_{16}	Components within M_{12} and M_{14}
0.2	M_{17}	All

Table 8.3: Module catalogue for Climate Control system.

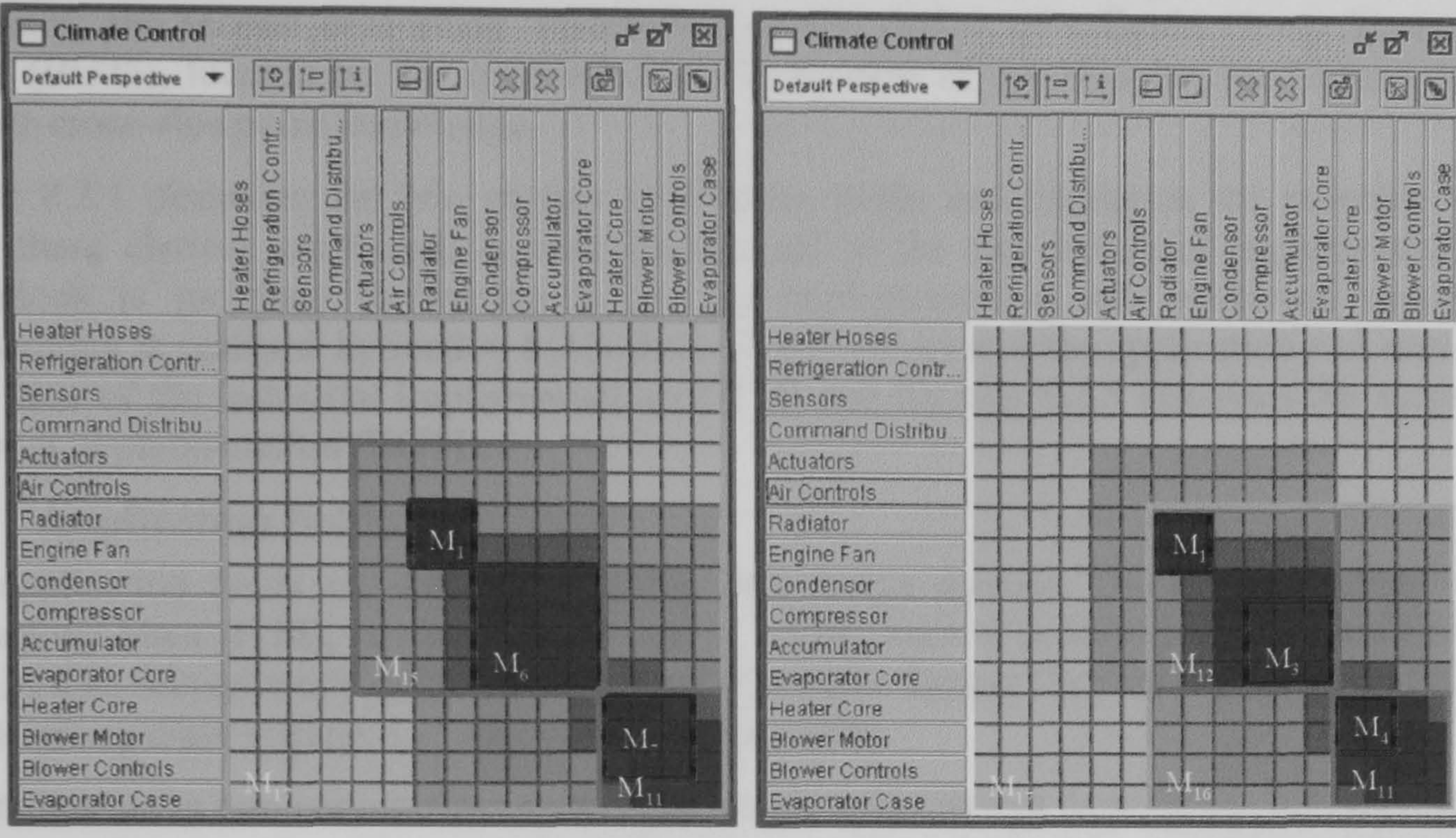
Configuration A	Module	Components
Level i	M_1	Radiator, Engine Fan
	M_6	Condenser, Compressor, Accumulator, Evaporator Core.
	M_7	Heater Core, Blower Motor, Blower Controls
Level ii	M_{11}	M_7 and Evaporator Case
	M_{15}	M_1, M_6 and Blower Controls
Level iii	M_{18}	M_{11}, M_{15}, Heater Hoses, Refrigeration Controls, Sensors, and Command Distributor.

Table 8.4: Hierarchical module configuration A

Configuration B	Module	Components
Level i	M_1	Radiator, Engine Fan
	M_3	Compressor, Accumulator, Evaporator Core
	M_4	Heater Core, Blower Motor
Level ii	M_{11}	M_4, Blower Controls and Evaporator Case
	M_{12}	M_1 , Condenser and M_3
Level iii	M_{16}	M_{11} and M_{12}
Level iv	M_{17}	M_{16}, Heater Hoses, Refrigeration Controls, Sensors, Command Distributor, Actuators and Air Controls

Table 8.5: Hierarchical module configuration B

Figure 8.5 illustrates the above hierarchical configurations through overlaid boxes, which represent each selected module, in the *Modular Structure Viewpoint Model (Structure* $FDP_{material}$*)*. The module number indicator (harmonised with those defined in Table 8.4 and Table 8.5) can be found in the bottom left hand corner of each box. The colours of the boxes represent the their level in the hierarchy with level i depicted as a black box each successive level represented by a lighter colour of grey to level iv being white.



Configuration A Configuration B

Figure 8.5: Hierarchical module configuration A and B illustrated

The clustering criteria has been shown to be effective in that it clustered the components in the climate control example into very similar groupings as those that had been defined as 'modules' in the original publication. Despite returning some similar groupings as defined in the alternator case there were some significant differences. However, these can be attributed to the application of domain specific knowledge to the groupings in the initial publication. It is suggested that had such knowledge been better represented within the initial definition of dependency knowledge (in the original publication) that the optimisation mechanism, utilised within the work represented in this thesis, would have returned more appropriate component groupings.

The MSI function was deemed appropriate in that it confirmed that the component groupings, which were defined as modules in the original publications, represent strong module candidates, i.e. have a high MSI value. In addition, the MSI application provided a measure of the modularity for all component grouping in the model, which is consistent with the hypothesis that modularity is a relative measure (on a scale from modular and integral) and not an absolute. Thus, the MSI calculation and its application process were shown to represent an effective means to measure the modularity of *concept* clusters within the product structure.

The MSM interpretation of the DSM was shown to be an effective means to represent the returned MSI values that provided a mechanism on which to interpret the structures' modularity and facilitate the selection of hierarchical module configurations by the designer.

8.2 Industrial evaluations

The following evaluates both the proposed methodology and its application process in an industrial setting. The focus of the first case study, undertaken on the design of an Integrated Technology Mast (ITM), is on the application and evaluation of the methodology to the knowledge from viewpoints of a design. In addition to knowledge from individual viewpoints, the second case study, undertaken on the design of a Battle Group Thermal Imager (BGTI), focuses on the application and evaluation of the mapping mechanism that supports cross-viewpoint knowledge.

Section 8.2.1 discusses the two chosen application fields and highlights the rationale on which these choices are based. A brief background to the two industrial implementation evaluations is provided in Section 8.2.2. An implementation and evaluation process methodology is outlined in Section 8.2.3. Section 8.2.4 discusses the application and results of the first of the industrial implementations (ITM) and Section 8.2.5 discusses the second industrial implementation (BGTI).

8.2.1 Application fields

Both application fields are within the engineering design domain. The first, an Integrated Technology Mast (ITM), is a conceptual design for the mast of a destroyer sized multi-role warship. The second study involved the design of a Battle Group Thermal Imager (BGTI) for a battle group class vehicle, i.e. armoured vehicles, tanks.

The studies are in two very distinct areas of the engineering design domain, i.e. the naval and surface combat domains. However, both domains are characterised by a number of factors, specific to military engineering design. These characteristics make this an appropriate area for evaluating the methodology, and include:

- One-off and batch design

  In the military domain a product is, in general, designed for application in a very specific role, i.e. troop and/or equipment manoeuvre, mine clearance, or casualty treatment. The equipment role imposes a number of specific constraints on the design such as equipment, noise/vibration, and communication requirements whilst all products are constrained by the requirements associated with their defensive role. In addition, many of the designs are constrained by the buyer and in consequence its location in service, i.e. international and national standards and environmental conditions. In the case of the naval domain there can be as little as one product in a batch. In addition, there is often no guarantee of a repeat or similar order, i.e. one order may be for a destroyer class ship for the Royal Malaysian Navy whilst the next may be an aircraft carrier class for the British Royal Navy. Thus, unlike in domestic engineering design, such as the automobile or white goods industries, there is little scope or justification for the development of product families and generations. As such, fields such as the military domain require a MD methodology that allows them to attain the life-cycle benefits of modularisation based on the design individual products. Therefore methodologies associated with such principles as platform design and product family generation have limited applicability in this specific domain. As discussed in Section 1 the methodology developed within this work focuses on the exploration and identification of modularity in individual products to support the structuring and enhancement of the knowledge associated with that product for re-use. Long-term utilisation of the methodology may support the re-use of design knowledge to generation of a design; however, this is not the current focus of the work presented in this thesis. The current focus is to define the declarative and procedural knowledge to provide a coherent framework to support the designer in evolving a modular solution in the first instance.

- System integration

  Products in the military domain often require the integration of a number of relatively complex systems to realise the final artefact, for example, propulsion, waste water, navigation, communication, and combat systems are integrated within the overall ship system. Thus, each individual system must be designed to not only provide its individual functionality but to integrate with the entire system as a whole. These products can be described as 'mechatronic' and combine multiple views and/or disciplines during their design. As such, the military domain represents an appropriate test bed to explore the concept of modularisation over different *viewpoints* and *perspectives*, i.e. the impact of one form of modularity on another.

- Long lead times

  The constraints imposed on products in the military design domain result in longer lead times due to increased certification requirements in terms of safety, reliability and environmental impact. The long lead times and certification requirements result in a need for improved knowledge structuring and audit-ability.

- Long life in service

  The initial capital costs of military products are very high, in comparison with a similar domestic product (armoured vehicle as opposed to a four wheel drive jeep). As such, the expected 'life in service' of military products exceeds that of domestic products and can often be measured in decades. This results in an emphasis on: the robustness of components, the minimisation of retrofit, ease of maintenance and retrofit (if required). As such, the capture, exploration and maintenance of knowledge related to how and why the design evolved into the final artefact definition is significant factor in maintain essential functionality during the design of product upgrades. As such, it represents an appropriate test bed to explore the functionality of a modular design methodology focussed on exploring, and identifying the evolution of the design solution from the abstract to the concrete and the maintenance of its associated knowledge.

and, a characteristic specific to the marine domain;

- Ship class maintenance

  In naval ship designs there is a requirement that all ships within a class are of similar outfitting. This ensures integration of the ship systems across the entire class fleet, i.e. in terms of communications, weaponry, and control system. The design and manufacture of a class of ships can span decades and the design must take this into account. This is an especially pertinent issue where technologies are subject to rapid development i.e. processors, visual display technology, digital media, and control mechanisms. As such, it represents a test bed to explore the capabilities of modular design to support the achievement of life-cycle objectives such as ease of maintenance, ease of retrofit matters related to the technology life-cycle of components etc.

The emphasis on one-off products, the need to structure and maintain knowledge related to the product and the fulfilment of a number of life cycle objectives make the military domain an ideal basis for the evaluation of the methodology presented in this thesis.

8.2.2 Background to industrial implementations

The design of both products utilised as the basis to evaluate the work presented in this thesis were subject to specific design requirements, for example, both BGTI and ITM designs focussed on improved maintenance in service. These design requirements resulted in both design teams viewing MD as a realistic solution to their problems. However, both found there was a lack of formal support available to aid in the MD process, especially in the early stages of design, and their design process had an emphasis on a 'trial and error approach' and relied on personal experience [33]. In the case of the ITM design this lack of support resulted in a requirement to have a number of 'manpower intensive' iterative reviews to 'ensure the robustness and integrity of the rapidly evolving modular solution'. Despite the lack of formal support the ITM project resulted in a predominantly modular solution. The ITM designers identified a number of criteria that they deemed important for an industry centred MD methodology (Smith, Robb et al. 2001). Their requirements centred on a generic MD approach which complimented their work practices and was applicable across a wide variety of design domains. The methodology should be capable of supporting design from project inception whilst integrating a range of different technologies from diverse fields such as mechanical, electrical and optical. In addition, the designers required a methodology and/or tool which could support their modular decision making process which is currently based on personal experience. The definition of the Multi-Viewpoint MD Methodology aimed to fulfil these criteria, and those defined in Chapter 4, whilst providing an articulate procedure for designers to follow. The methodology was implemented in two industrial studies to evaluate its performance with respect to these aims.

8.2.3 Modular design implementation and evaluation process methodology

The following section outlines the process undertaken in the following two case studies. While implementing the initial ITM study, the author noted that there was no methodology (within the literature covered in Chapters 3 and 4), which defined the elements required to implement a modular design analysis in an industrial setting. As such, the initial implementation (ITM) was carried out through a trial and error approach and through the formalisation and rationalisation of the activities undertaken during this implementation a MD implementation and evaluation process methodology was developed. The resulting methodology is depicted in Figure 8.6.

---

33 In the ITM case, the lack of support and their adopted design process is detailed in the publication shown in Smith, J. S., M. D. Robb, et al. (2001). An Experience of Modularity through Design. 13th International Conference on Engineering Design, Glasgow, UK, I Mech E..

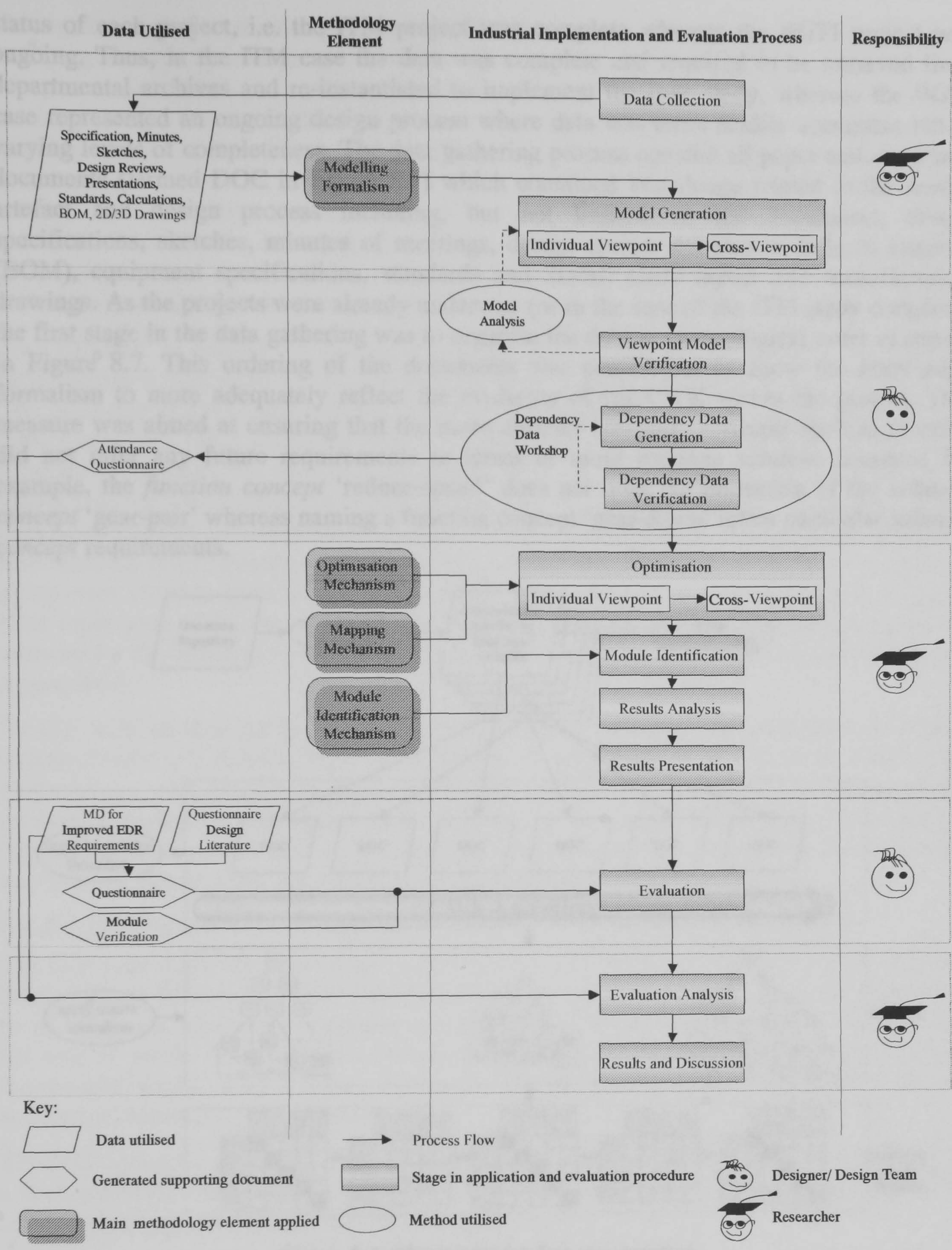


Figure 8.6: Modular Design industry implementation and evaluation methodology

The figure depicts the data utilised, the applicable Multi-Viewpoint MD methodology elements, the documentation generated and the responsibilities of the researcher and the designer/design teams at each stage of the industrial implementation and evaluation. The formalised methodology illustrated in Figure 8.6 was utilised as the basis to implement the second industrial implementation (BGTI) and had the effect of significantly reducing the implementation and evaluation process duration from around 12 months (ITM study) to 10 weeks (BGTI study).

As illustrated in Figure 8.6 both case studies began with a period of data collection. The only difference in the data collection procedure for the two studies is attributable to the different status of each project, i.e. the ITM project was complete whereas the BGTI project was ongoing. Thus, in the ITM case the data was complete and required to be retrieved from departmental archives and re-instantiated to implement the case study, whereas the BGTI case represented an ongoing design process where data was more readily accessible but at varying levels of completeness. The data gathering process covered all paper and electronic documents (termed DOC in Figure 8.7) which contained knowledge related to the design artefact and design process including, but not limited to; bid documents, design specifications, sketches, minutes of meetings, design review documents, bills of material (BOM), equipment specifications, standards and 2D/3D CAD layout and manufacturing drawings. As the projects were already underway (or in the case of the ITM study complete) the first stage in the data gathering was to organise the data in chronological order as shown in Figure 8.7. This ordering of the documents was carried out to allow the knowledge formalism to more adequately reflect the evolution of the CWK within the project. This measure was aimed at ensuring that the more abstract *viewpoint concepts* such as *function* did not infer any future requirements in terms of more concrete *solution concepts*, for example, the *function concept* 'reduce-speed' does not infer the utilisation of the *solution concept* 'gear-pair' whereas naming a function concept 'gear-down' infers particular *solution concept* requirements.

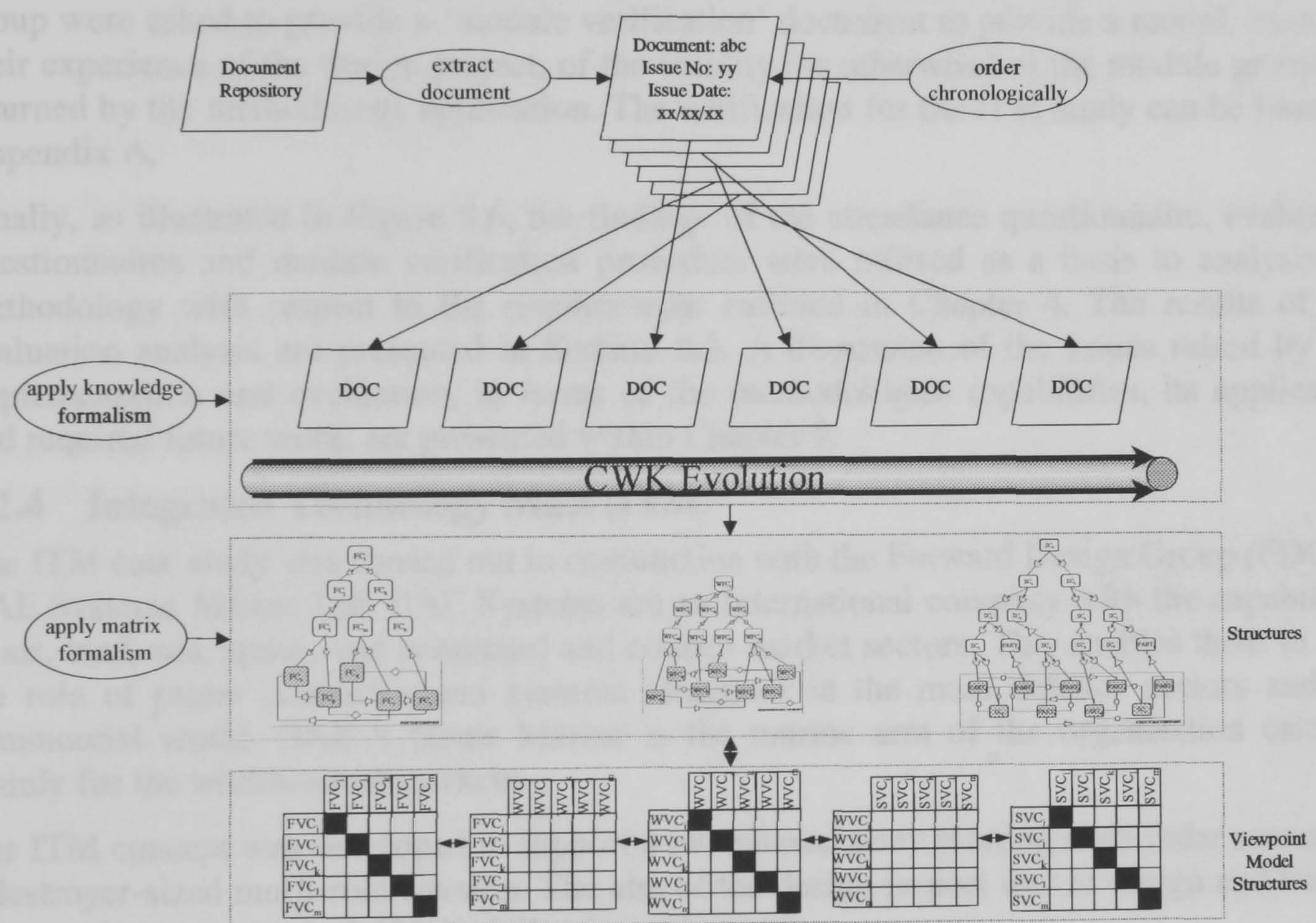


Figure 8.7: Chronological data organisation

As shown in Figure 8.6 the next step in the case study process was to apply the knowledge formalism. This knowledge formalism is applied across three views of design to create *structures* as illustrated in Figure 8.7. The matrix formalism is applied to these *structures* to generate five *viewpoint models* as shown in Figure 8.7. The *concepts* within these models represent the most concrete and detailed *concepts* defined in the knowledge *structures.* For each case study, three individual viewpoint models (function, working principle and structure) and two cross-viewpoint models ('function to working principle' and 'working principle to structure') were generated.

Figure 8.6 highlights that the dependency data generation and verification was the responsibility of the design team members and was achieved through a workshop event.

During the workshop the designers were asked to complete a matrix body by defining and verifying the type and weight of the dependencies that existed between the *concepts* in each *viewpoint model*. This dependency generation task was carried out in a number of small working groups of two to three designers. On completion of each matrix, another of the working groups verified the entries. The bi-directional knowledge flow between the *structures* and *viewpoint models* (illustrated in Figure 8.7) represents the fact that the researcher could utilise the dependency data generation and verification results to both enhance and certify the knowledge structures and vice versa. The workshop participants in each study completed a brief questionnaire (Smith 2002), the results of which were later utilised as part of the evaluation process.

As illustrated in Figure 8.6 an analysis phase, during which the researcher applied the methodologies optimisation, module identification and mapping mechanisms, followed the data gathering and workshop phases. The resulting findings were presented to the workshop participants and other interested parties within each industrial group. During the results presentation session the attendees completed an evaluation questionnaire. The evaluation questionnaire (Smith 2002) was based on the MD requirements, defined in Chapter 4, and developed through literature based research in survey design (Heather and Stone; Bradburn and Sudman 1979; Fowler 1993; Aldridge and Levine 2001). In addition, each industry group were asked to provide a 'module verification' document to provide a record, based on their experience of the design project, of the validity (or otherwise) of the module groupings returned by the methodology application. The verification for the ITM study can be found in Appendix A.

Finally, as illustrated in Figure 8.6, the findings of the attendance questionnaire, evaluation questionnaires and module verification procedure were utilised as a basis to analysis the methodology with respect to the requirements outlined in Chapter 4. The results of this evaluation analysis are presented in Section 8.3. A discussion of the issues raised by this implementation and evaluation, in terms of the methodologies capabilities, its application and required future work, are presented within Chapter 9.

8.2.4 Integrated Technology Mast (ITM)

The ITM case study was carried out in conjunction with the Forward Design Group (FDG) at BAE Systems Marine Ltd. BAE Systems are an international company with the capabilities in air, land, sea, space, and command and control market sectors. This enables them to take the role of prime contractor and systems integrator in the main defence sectors and the commercial world. BAE Systems Marine is the marine arm of the organisation catering mainly for the worlds naval markets.

The ITM concept was developed to support a two phased array rotating main radar sensor for a destroyer-sized multi-role warship. The aim of the design project was to design and build a concept demonstrator to fulfil the following objectives:

- Supporting the specified antenna (which was itself in the research and development stage),
- Reducing the ships radar signature,
- Reducing the number of external aerials,
- Improving the maintenance capabilities,
- Improving equipment availability to facilitate the multi-role status of the ship platform, and
- Improving sensor performance.

The decision to focus on producing a predominantly modular structure was influenced by a number of factors including:

- Distribution of Involved Business Groups

  The teams would have to undertake design tasks in remote locations, across the UK, whilst ensuring integration with the overall system design and specification.

- Tight Timescales

  The teams would have little chance to test the compatibility of designed components and tight delivery schedules meant many components would not be physically together in one location until the final assembly phase.

- Improved Maintainability Requirement

  A key objective in the design concept that meant related issues of removal, replacement, and accessibility of components of varying technologies required to be addressed.

- Improved Role Flexibility Requirement

  A similar requirement to that of improved maintainability, requiring the removal and replacement of components with those of a differing functionality and the maintenance of individual sensor and overall system performance.

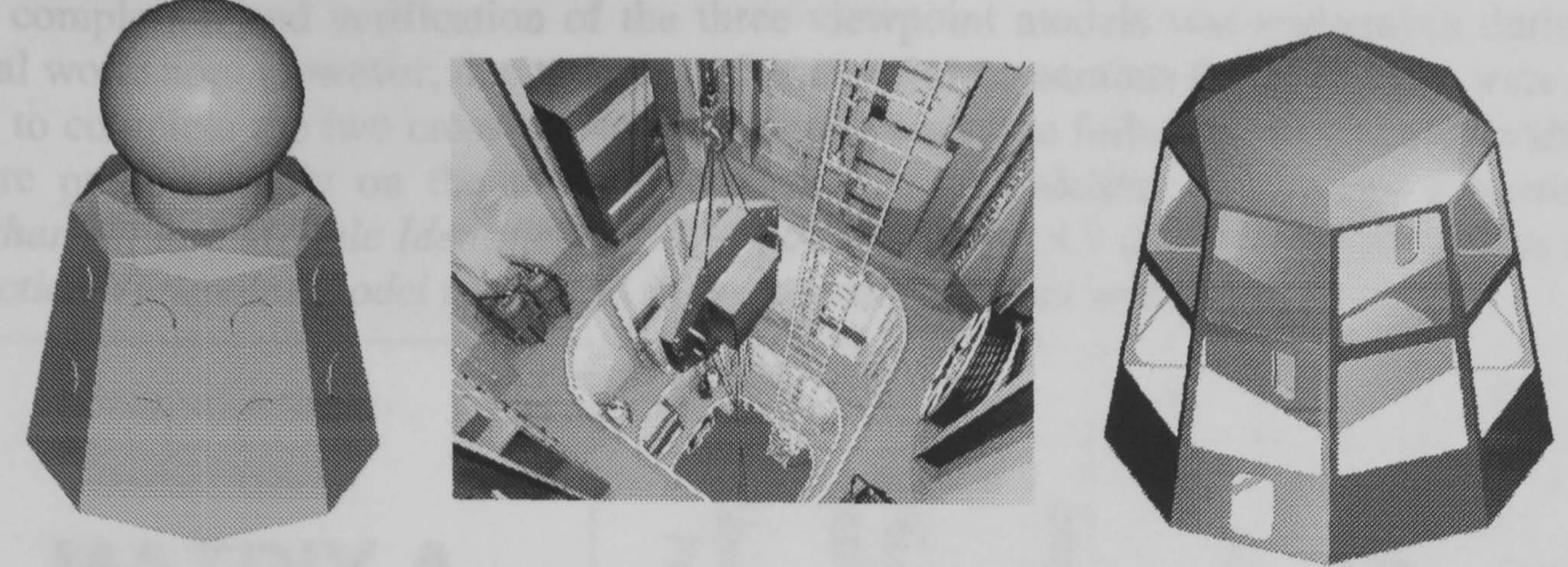
- Rapid Response to Repair of Battle Damage

  A key requirement of the concept was the ability to rapidly reconfigure and replace sensors based on various battle damage scenarios.

Thus, the application of modular design, as a principle synonymous with developing products with distinct detachable modules to support rapid product development, efficient upgrades and reconfiguration (Gu, Hashemian et al. 1997), became a practical solution to overcome such issues. However, the original design experienced problems in that they found no formal procedure or process was available to support modularisation (Smith, Robb et al. 2001). This lack of support was most noticeable in the initial design stages where the module boundaries were determined and tasks allocated to the individual teams (Smith, Robb et al. 2001). The initial module definition phase was carried out by the FDG with the resulting modules being allocated for detail design to various distributed design teams.

The project itself resulted in a highly flexible modular structure, which successfully fulfilled the above objectives, and proved the feasibility of the design concept[34]. Due to confidentiality reasons the photographs of the actual demonstrator cannot be pictured which was constructed full scale (approx. height 11m, width 7.5m). However, Figure 8.8a shows a 3D visualisation of the technology demonstrator, Figure 8.8b shows a 3D visualisation of a 'modularised sensor' replacement being hoisted into position through the internal structure and Figure 8.8c shows the centre column which supports the main radar and the external framing to which the exterior panels are fastened.

34 For an in-depth review of the ITM modular design process, experiences and outcomes the reader is referred to Ibid.



a) Mast demonstrator b) Replacing a 'modularised sensor' c) External panels and structure

Figure 8.8: ITM concept design

The Multi-Viewpoint MD Methodology was developed in conjunction with the FDG at BAE Systems. In addition, a group of experienced designers, across a number of distributed teams within BAE Systems, were specifically selected to design a modular product, the ITM. The ITM design involved a number of parties with the FDG at BAE Systems responsible for structural design, manufacture, final assembly and overall general project management. As such, the ITM design represented an appropriate initial test bed for the implementation of the methodology to evaluate both its performance and the resulting design's modularity. That is how will the results of the methodology application and those achieved by the designers, based on their own experience, compare.

A key point to note is that the original definition of the modular boundaries required a series of manpower intensive reviews and the team were keen to evaluate the methodologies performance with regards to this aspect of the modular design process.

ITM methodology application

The Multi-Viewpoint Modular design methodology was applied to the ITM case study as defined in Section 8.2.3. Five models were developed defining three viewpoints and two cross-viewpoints. The three viewpoint models had various perspectives associated with them as shown in Table 8.6. The table also defines the number of concepts in each model[35].

Matrix	Viewpoint(s)	Perspectives	No of Concepts
Matrix A –	Function	Energy Material Information	52
Matrix B	Working Principle	Energy Material Information	51
Matrix C	Structure	Energy Material Information Physical Link	58
Matrix E	Function to Working Principle	Causal-Link	52 by 51
Matrix D	Working Principle to Structure	Causal-Link	51 by 58

Table 8.6: ITM models

[35] The number of possible combinations of the *Function Viewpoint Model* (Matrix A with 52 concepts) can be defined as $52! = 8^{67}$

The completion and verification of the three viewpoint models was undertaken during the initial workshop. However, due to time and manpower constraints the FDG team were never able to complete the two cross-viewpoint models. Thus, the following results and evaluation centre predominantly on the methodology elements: *Modelling Formalism, Optimisation Mechanism* and *Module Identification Mechanism*. Figure 8.9 depicts a small portion of the *Function Viewpoint Model* as utilised in the dependency data workshop.

MATRIX A Function Concepts		1 (Mechanically) Support Periphery Sensors	2 (Mechanically) Support Main (topmast) Sensor	3 Interface Main Sensor	4 Interface Periphery Sensors	5 Support Satellite Communication	6 Support U/V/HF Communication
1 (Mechanically) Support Periphery Sensors	E				0.2		
	M		0.1		0.2	1	1
	I						
2 (Mechanically) Support Main Mast Sensor	E			0.2			
	M	0.1		0.2			
	I						
3 Interface Main Sensor	E		0.2				
	M		0.2				
	I		0.2				
4 Interface Periphery Sensors	E	0.2				0.5	0.5
	M	0.2				0.2	0.2
	I	0.2				0.5	0.5
5 Support Satellite Communication	E				0.5		
	M	1			0.2		
	I				0.5		
6 Support U/V/HF Communication	E				0.5		
	M	1			0.2		
	I				0.5		

Figure 8.9: Sample of the ITM viewpoint models

The twelve workshop participants took around 5 hours to complete and verify the three viewpoint matrices. However, it is envisaged that the models would (in practice) be completed and maintained electronically as in the BGTI case (see Section8.2.5). In addition, it is unlikely that all the matrices would be completed at one time, as in this case, as it is designed to be applied at incremental stages of the evolving design process to maintain the modular solution (as illustrated in Chapter 5, Figure 5.16)

ITM results

The following highlights the main results from the methodologies implementation in the ITM design case.

The paper-based models were utilised as a basis to develop the computational support tool for the methodology. They were converted to computational models to support their subsequent analysis. Table 8.7 depicts the resulting clustering criteria values for the ITM case study. The GA parameters are illustrated in the top row of the table and are a population of 200 with 200 generations, a two-point end crossover operator with a probability of 80, and a two-point adjacent mutation operator with a probability of 20[36]. The *Optimised Viewpoint Models* and *Modular Structure Models* for the ITM analysis set, defined in Table 8.7, can be found in work by Smith (Smith 2002).

GA Parameters: 200,200,80,20, 2 point end, 2 point adjacent							
		Clustering Criteria Values					
Viewpoint	Perspective	Pre-optimised	When Optimised For:				
A			Material	Information	Energy	Spatial Rel	Combined All
Function	Material		1054	2204	1687		1276
	Information		1096	202	592		341
	Energy		2036	3091	1202		1291
	Combined All		1395	1832	1160		969
B							
Working Principle	Material		1133	3079	3175		1611
	Information		1112	288	472		362
	Energy		1506	9993	443		689
	Combined All		1511	1582	1433		847
C							
Structure	Material	4067	1959	5607	6187	2625	2093
	Information	205	419	132	775	687	237
	Energy	124	252	110	14	195	83
	Spatial Relations	2639	2914	6144	6281	1934	1360
	Combined All	1758	1340	2991	3391	1360	1102

Table 8.7: The clustering criteria values for the ITM analysis

The rows in Table 8.7 represent the viewpoints and the perspectives of these viewpoints, whilst the columns indicate which perspective has been optimised. The entries in the row and column intersections represent the returned clustering criteria for each optimised perspective (depicted within the black boxes in the table). The effect of the optimisation of one perspective is illustrated in alternative entries in the column for that optimised viewpoint and perspective combination (depicted by the coloured columns). That is, for the viewpoint and perspective optimisation of the *Function Viewpoint Model* $FDP_{material}$ (depicted in the cyan coloured column) we see that the clustering criteria is 1054 and that this optimised sequence has the effect on the *Function Viewpoint Model* $FDP_{information}$ of increasing the clustering criteria by approximately 540% from its optimum from 202, depicted in the dark box of the blue column to 1096, depicted in the cyan column.

The optimisation of the combined perspective (depicted in the salmon coloured column) for each viewpoint and perspective combination illustrates the trade-offs if the optimisation takes into consideration an average of all perspectives, for example, the *Optimised Function Viewpoint Model* $P_{combined\ all}$, has a clustering criteria of approximately 969. This sequence effect on each individual perspective is depicted by the other values in this function viewpoint section of the column (i.e. salmon coloured). As can be seen all the effect on each individual perspectives has been that these near their optimum value i.e., the *Function Viewpoint Model* $FDP_{material}$ has a value approximately 21% greater than its optimum (1275 as opposed to 1054), the *Function Viewpoint Model* $FDP_{information}$ approximately 31% greater than its optimum (341 as opposed to 202) and the *Function Viewpoint Model* FDP_{energy} approximately 7% greater than its optimum (1291 as opposed to 1201) with the combined perspective optimised. Thus, suggesting that, in this case, that energy dependencies have the most impact on modularity from the viewpoint of *function* as it nearest its optimum value.

36 These parameters are consistent with previous findings for the GA parameters in such a domain Whitfield, R. I., J. S. Smith, et al. (2002). Identifying Component Modules. Artificial Intelligence in Design (AID '02), Cambridge, UK..

The following analysis concentrates on the structural viewpoint, as this is the view taken during the product's original modularisation by the ITM design team. The original modular design solution was determined from the physical link perspective[37] of the structural (solution and part) concepts of the design. As can be seen in Table 8.7 the ITM structure viewpoint has an additional column termed 'pre-optimised'. This column depicts the values of the existing product structure (depicted as a specific concept sequence in the *Structure Viewpoint Model*) as defined by the original ITM design team and its resulting clustering criteria values for each perspective. The concept sequence was determined from analysis of the original BOM, design drawings and input from the designers themselves. Thus, providing a basis to evaluate the performance of the Optimisation Mechanism and the Module Identification Mechanism.

Figure 8.10[38] depicts the original product structure as defined by the ITM design team. The clustering criteria value for this sequence is 2639.

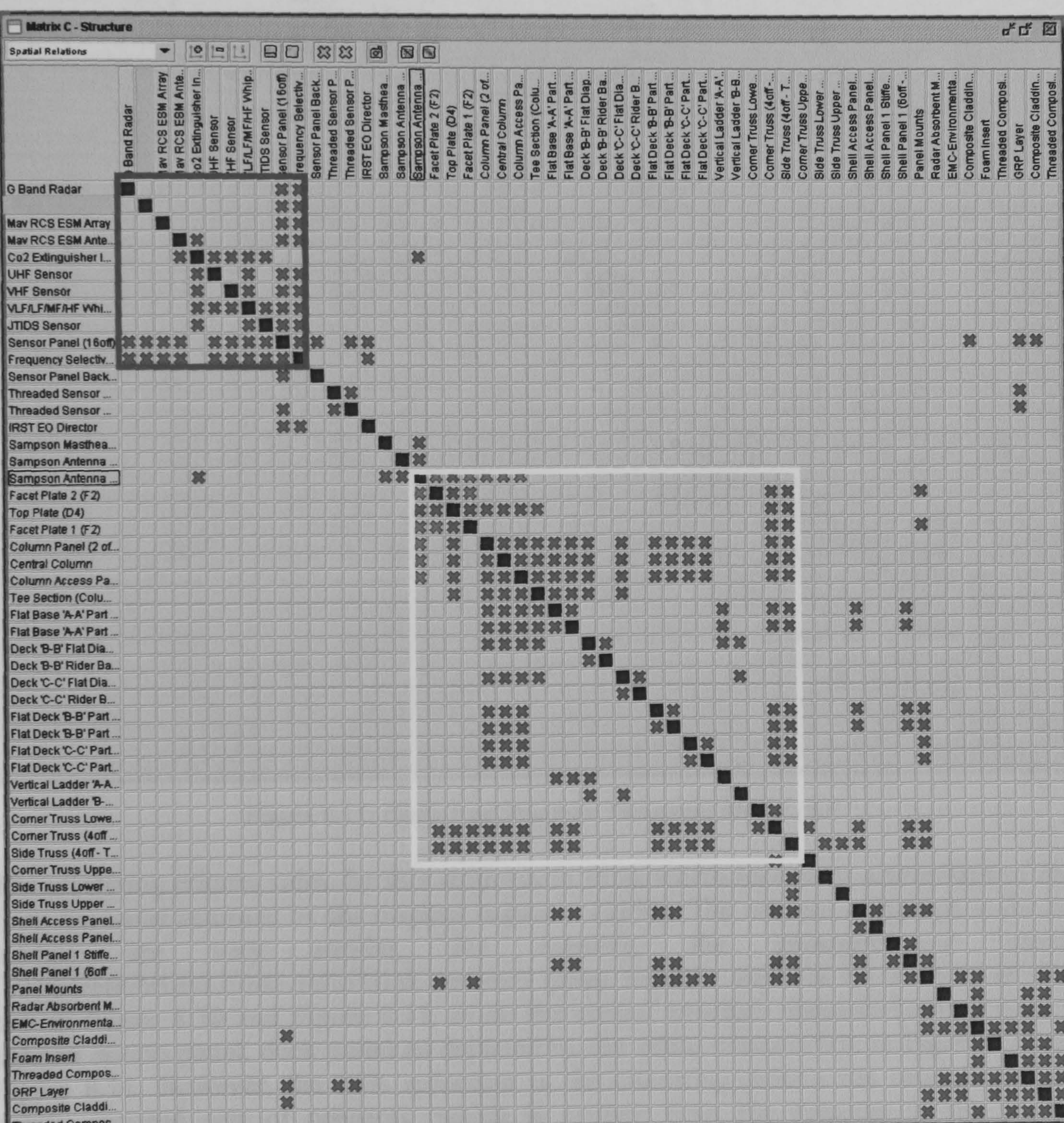


Figure 8.10: Original *Structure Viewpoint Model PLP*

37 Termed 'spatial relations' in the software implementation.

38 Some concepts have been blocked out due to confidentiality reasons

Figure 8.11 depicts the *Structure Viewpoint Model PLP* after the application of the methodologies' *Optimisation Mechanism.*

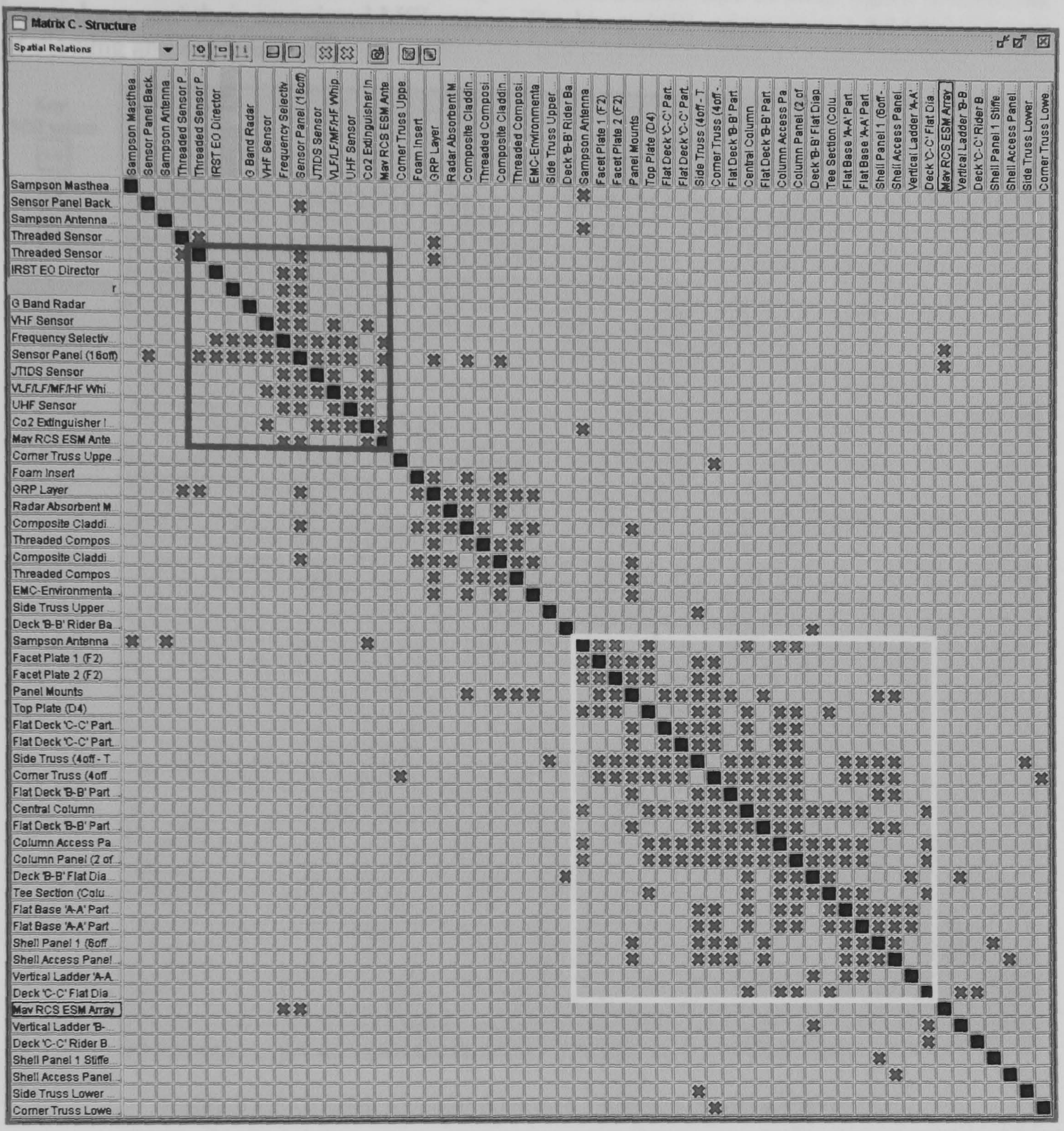


Figure 8.11: *Optimised Structure Viewpoint Model PLP*

The clustering criteria value for this sequence is 1934, which represents a reduction of 26% from the value returned by the original sequence (2639). The concept groupings in the two models are very similar, for example, the communication sensors (VHF/UHF/VLF/LF/MF/HF/JTIDS sensors) group together depicted by the blue overlaid box in Figure 8.10 and Figure 8.11, as do the central structure elements (central column, column access panel, deck parts) as depicted by the white overlaid box in Figure 8.10 and Figure 8.11. However, the 26% reduction in the clustering criteria has resulted in more tightly grouped sets of concepts, which is evident from the outcome of the Module Identification Mechanism application to both sequences, i.e. *Modular Structure Viewpoint Models* (MSVM). Figure 8.12 depicts the *Modular Structure Viewpoint Models (Structure $_{PLP}$)* returned for the original structure (shown in Figure 8.10) and

Figure 8.13 depicts the *Modular Structure Viewpoint Models (Structure $_{PLP}$)* returned for the optimised sequence (shown in Figure 8.11). Both MSVM have a series of coloured boxes overlaid over them depicting the potential modules at varying MSI values from the value 1.0 (maximum module value) to 0.6. The key on the left-hand side of each figure denotes the box colours and their associated MSI values. The lower MSI values are not depicted for ease of viewing and explanation[39].

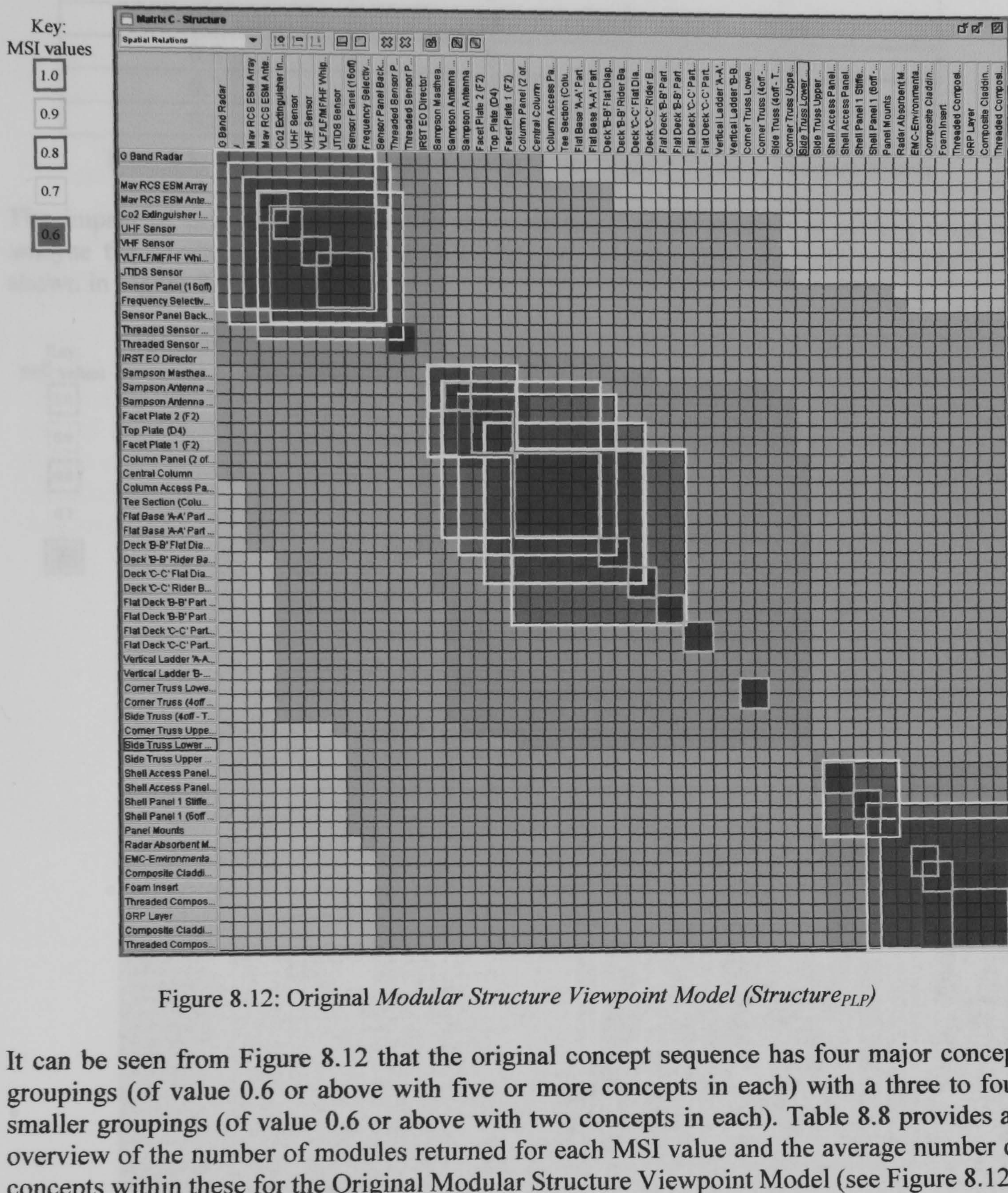


Figure 8.12: Original *Modular Structure Viewpoint Model (Structure$_{PLP}$)*

It can be seen from Figure 8.12 that the original concept sequence has four major concept groupings (of value 0.6 or above with five or more concepts in each) with a three to four smaller groupings (of value 0.6 or above with two concepts in each). Table 8.8 provides an overview of the number of modules returned for each MSI value and the average number of concepts within these for the Original Modular Structure Viewpoint Model (see Figure 8.12).

---

[39] The entire module catalogue for the *Optimised Modular Structure Viewpoint Model (Structure $_{PLP}$)* illustrated in Figure 8.13 can be found in work by Smith Smith, J. S. (2002). Integrated Technology Mast Multi-Viewpoint Modular Design Methodology Implementation Results. Glasgow, UK, University of Strathclyde: 44..

MSI Value	No. of Modules	Average no. of concepts in each module
1	1	2
0.9	13	2.29
0.8	5	3.8
0.7	7	7.57
0.6	10	9.7

Table 8.8: Overview of concept groupings in the Original Modular Structure Matrix

The impact of this 26% reduction on the modularity of the product is evident when we analyse the Modular Structure Viewpoint Model, returned from the optimised model, as shown in Figure 8.13.

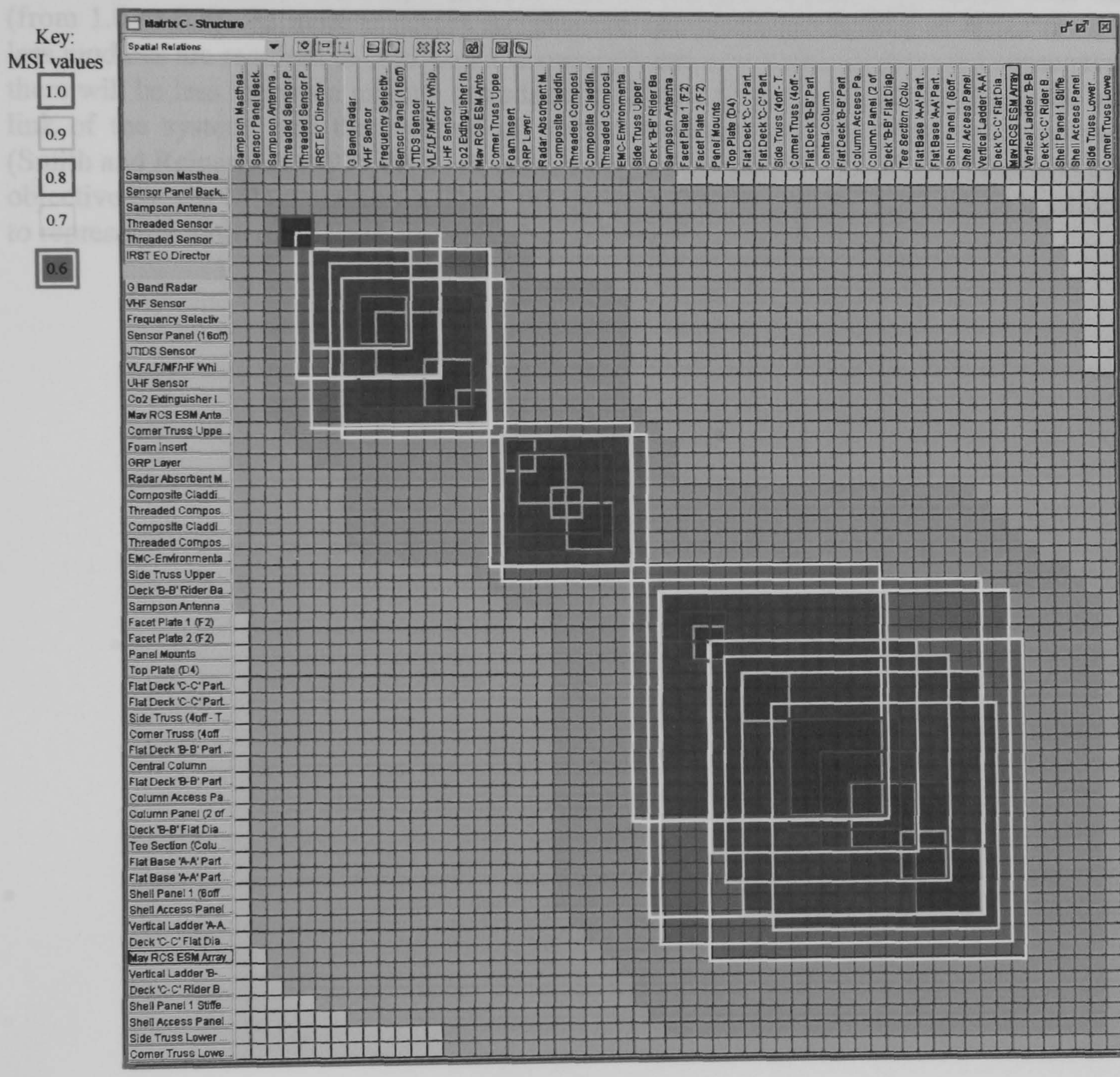


Figure 8.13: Optimised *Modular Structure Viewpoint Model (Structure$_{PLP}$)*

At first glance, it can be seen that the concepts have grouped into three major groupings (of value 0.6 and above) with no smaller groupings as in the original example depicted in Figure 8.12). Table 8.9 provides an overview of the number of modules returned for each MSI value and the average number of concepts within these for the optimised Modular Structure Viewpoint Model (Figure 8.13).

MSI Value	No.of Modules	Average Components in each module
1	1	2
0.9	14	3.2
0.8	10	5.3
0.7	9	11.1
0.6	9	14.1

Table 8.9: Overview of concept groupings in the Optimised Modular Structure Matrix

Comparing the resulting module groupings of those in the optimised model (Table 8.9) with those in the original model (Table 8.8) we see that the number of concept grouped into modules for each consecutive value of MSI is consistently greater in the optimised version as opposed to the original (with MSI value 1.0 being the exception). The optimised version has an average of 34% more concepts grouped into the modules at each incremental MSI value (from 1.0 to 0.6). As more concepts are grouped into a smaller number of major modules, less modules are required to configure the concepts into a modular solution and consequently there will be less 'between module boundaries'. Module boundaries 'are frequently the weak link of the system' and the creation of 'robust interfaces are generally more expensive' (Smith and Reinertsen 1997). As such, minimising the number of module boundaries is a key objective for the MD practitioner. Thus, the optimised version of the ITM can be considered to represent a more robust and cost effective solution.

In addition, the outcome of the Module Identification Mechanism depicts a hierarchy, which was not explicitly defined during the original ITM design activity. The hierarchy may be utilised to support a number of design activities, for example, 'assembly process design'. Figure 8.14 depicts a module configuration overlaid on the Optimised Module Structure Viewpoint Model (Figure 8.13) and extracted to illustrate how this may be used to configure as an assembly design process.

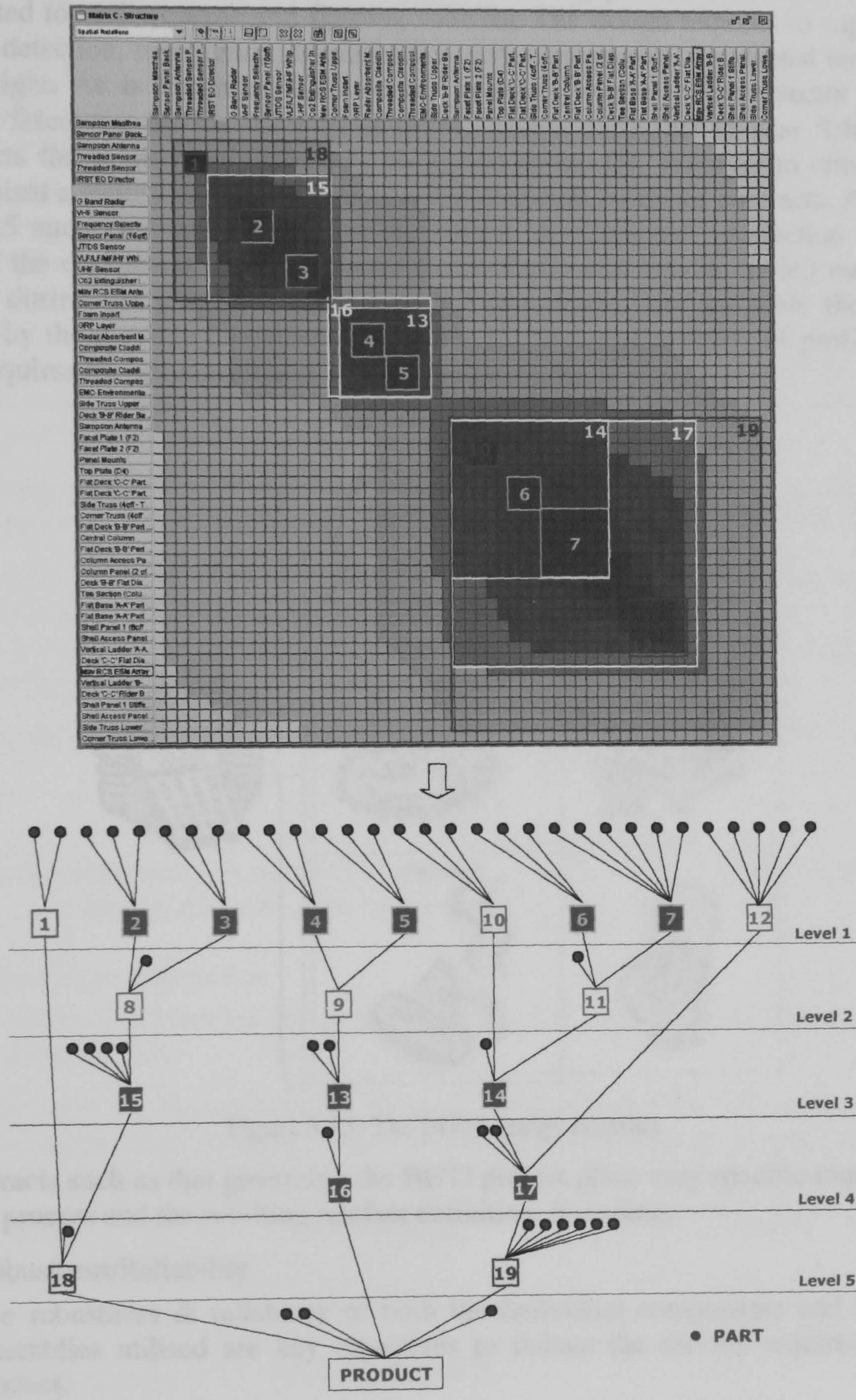


Figure 8.14: Possible assembly process design configuration extracted from the Optimised Modular Structure Viewpoint Model

8.2.5 Battle Group Thermal Imager (BGTI)

The BGTI case study was carried out in conjunction with the Battle Group Thermal Imager design team at Thales Optronics Ltd. Thales is a global electronics company serving

Aerospace, Defence, and Information Technology markets worldwide. Thales Optronics Ltd have played critical roles, meeting a broad range of requirements, including night vision, reconnaissance, target identification and precision weapon guidance. For ground forces, Thales produces a full range of day/night thermal cameras for all types of platform to perform tasks including surveillance, fire control, target designation and search and track.

The BGTI system design, the 'gunners sight', is being produced to replace the current version fitted to various armoured fighting vehicles. The design requires to support passive and direct detection, recognition, identification and engagement of potential targets by both day and night. As is becoming increasingly common in the military sector the contract included a fixed cost for the support of the equipment during its 15-year 'life-in-service'. This reflects the Ministry of Defence's obligation to the UK taxpayer to obtain value for money against a history of regularly published overspends on major contracts. As detailed in Figure 8.15 such contracts encompass the design, development, production and lifetime support of the equipment. Thus, the more maintenance and repairs carried out and spares consumed during the product life cycle until disposal, the less profitable the contract as illustrated by the possible outcomes in Figure 8.15. The maximisation of profits on such a contract requires the optimisation of support requirements.

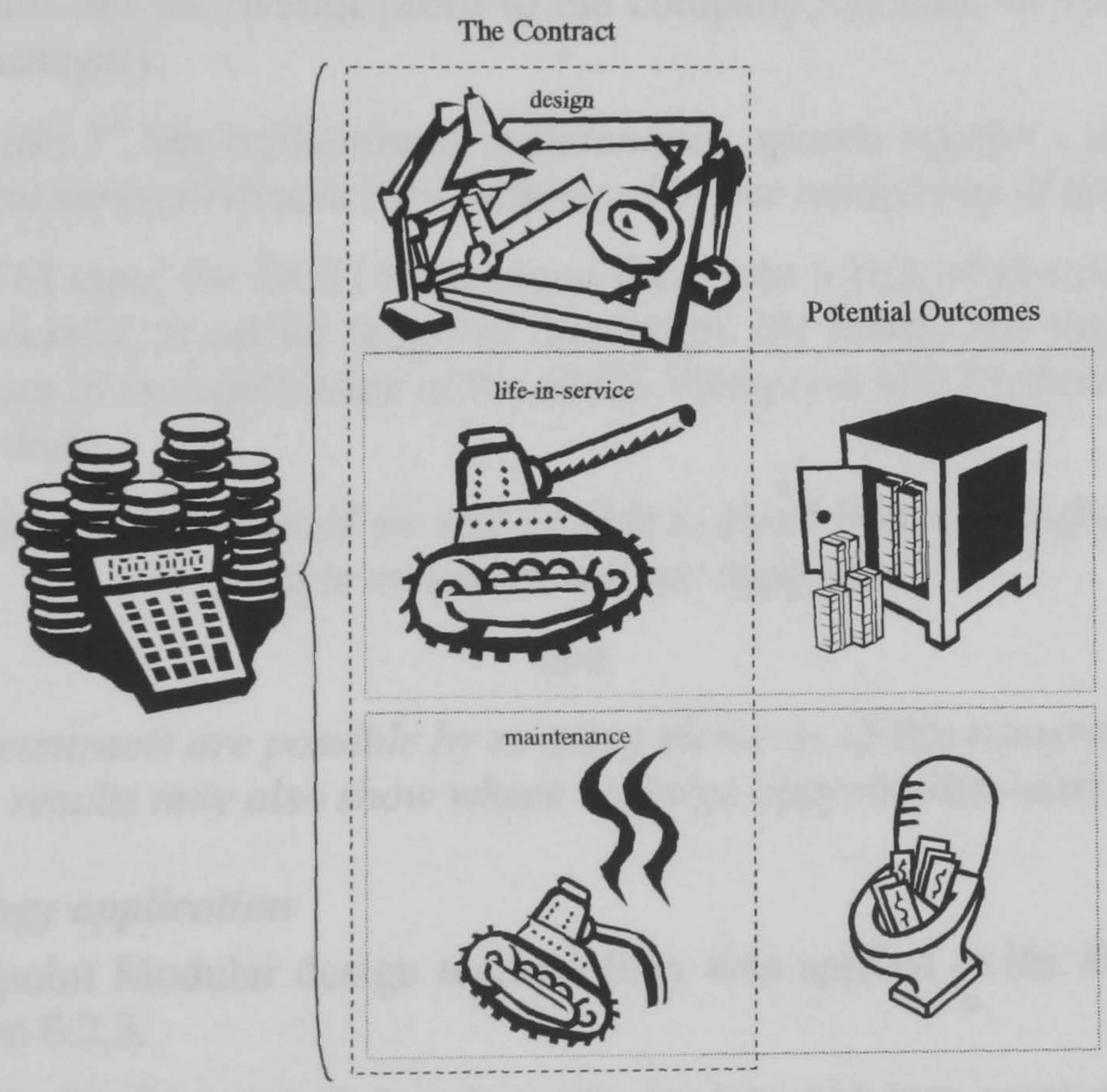


Figure 8.15: The BGTI design contract

Thus, contracts such as that governing the BGTI project place very specific requirements on the design process and the resulting artefact definition, including:

- Robustness/Reliability

  The robustness & reliability of both the individual components and the designed assemblies utilised are key objectives to reduce the service requirements of the product.

- Expected life-in-service

  The expected 'life-in-service' of major components should be equal if not greater than the length of the service contract.

- Part availability

The long-term availability of parts and technologies is a key issue in ensuring the efficient replacement of a part or assembly in the event of failure.

- Ease of replacement

  The ease in which a part or assembly can be replaced is a key factor to both minimise the downtime of the vehicle and reduce associated maintenance costs in terms of requirement for skilled labour and time considerations.

The Thales Optronics team developed a strategy to embody each of these requirements. The robustness/reliability issues were predominantly addressed through the development of rigorous specifications and testing procedures. As a number of components were 'bought-in', the expected 'life-in-service' issues were defined in initial specifications and the requirement for conformance documentation with parts or assemblies. The part availability issue was addressed through utilisation of a company wide 'parts catalogue' and an alliance with a parts replacement company. This is known as "Contractor Logistics Support". The strategy to facilitate ease of replacement, in the event of failure, was defined as a combination of the alliance with the parts replacement company and the design of easily replaceable units (i.e. modules). Thus, modular design became a key issue in fulfilling the objectives of the design project and maximising the overall profit to the company, because as stated by Martin Orr (BGTI support manager):

'things like 1st line replacement, maintenance, spares, repairs – especially repair turnaround time are all influenced by the modularity of the design'

Similar, to the ITM case, the BGTI team found there was a lack of practical support for the MD process, especially, at earlier stages of the design. Mr Martin Orr expresses the interest of Thales Optronics in the application of the Multi-Viewpoint MD Methodology to the BGTI example, stating that:

'as the design is advanced we will be able to check how successful we have been in modularising our equipment'

and

'further contracts are possible by re-using elements of this equipment and the results may also show where redesign opportunities exist.'

BGTI methodology application

The Multi-Viewpoint Modular design methodology was applied to the BGTI case study as defined in Section 8.2.3.

Five models were developed defining three viewpoints and two cross-viewpoints. Due to time restrictions the three viewpoint models had only one perspective associated with each as shown in Table 8.10.

Matrix	Viewpoint(s)	Perspectives	No of Concepts
Function Gunners Sight	Function	General Functional	67
Working Principle Gunners Sight	Working Principle	General Functional	40
Structure Gunners Sight	Structure	General Functional	53
Causal Link A	Function to Working Principle	Causal-Link	67 by 40
Causal Link B	Working Principle to Structure	Causal-Link	40 by 53

Table 8.10: BGTI models

This default perspective was defined as a general functional dependency perspective. The BGTI team defined functional dependence as their main interest due to the ease of replacement requirements of the design i.e. they required modules to have self-contained functionality to facilitate their replacement in the field and at repair facilities. The table also defines the number of concepts in each model.

The completion and verification of the five models was undertaken over a period of a week after an initial workshop to discuss the matrix content and completion process. The following covers only the main findings of the viewpoint matrices, as the ITM case has already focussed on these, and centres predominantly on the results and evaluation of the cross-viewpoint models and the methodology element *Mapping Mechanism.*

BGTI results

The following highlights the main results from the methodologies implementation in the BGTI design case. The focus is predominantly on the application and evaluation of the cross-viewpoint models.

Table 8.11 depicts the resulting clustering criteria values for the BGTI case study. The GA parameters are illustrated in the top row of the table and are a population of 250 with 250 generations, a two-point end crossover operator with a probability of 80, and a two-point adjacent mutation operator with a probability of 20[40]. The *Optimised Viewpoint Models* and *Modular Structure Models* for the BGTI analysis set, defined in Table 8.11, can be found in work by Smith (Smith 2002).

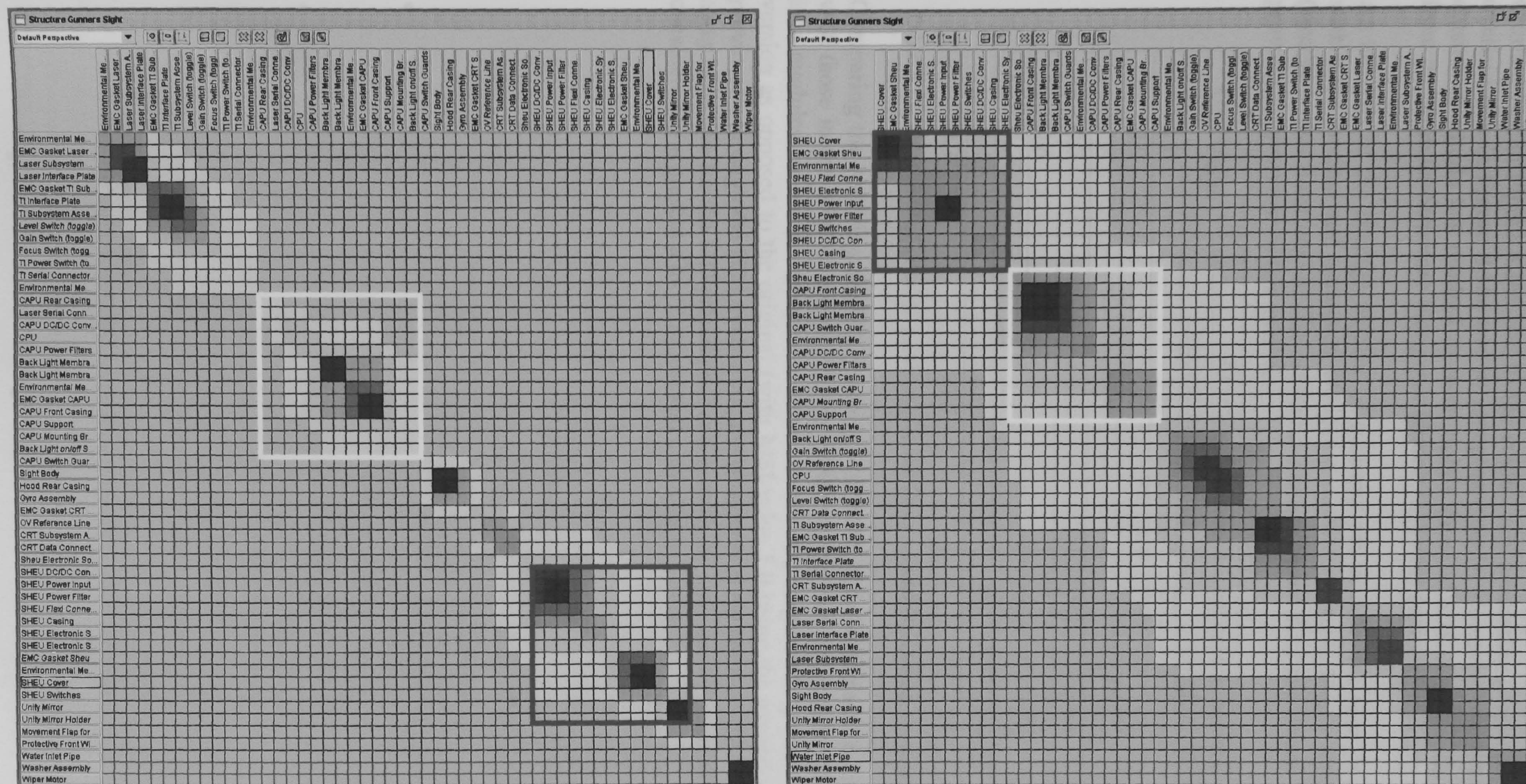
GA Parameters: 250, 250, 80, 20, 2 point end, 2 point adjacent			
Clustering Criteria Values	Viewpoint (general functional perspective)		
	FUNCTION	WORKING PRINCIPLE	STRUCTURE
Original			968
Optimised	2268	780	783
Difference			185
%			19%

Table 8.11: The individual viewpoint clustering criteria values for the BGTI analysis

Table 8.11 represents the clustering criteria values for the individual viewpoints of the BGTI implementation. A similar analysis to that carried out on the structural viewpoint in the ITM example was carried out on the BGTI case, i.e. the product's original modularisation by the BGTI design team is compared to the optimum returned through the application of the Methodology. The original modular design solution was determined based on a general functional perspective of the structural concepts (solution and part) of the design from analysis of the original BOM, design drawings and input from the designers themselves. As can be seen in Table 8.11 the original BGTI structure viewpoint modularisation resulted in a clustering criteria value of 968 whereas, based on the application of the optimisation mechanism the methodology returned a value of 783 (optimised BGTI structure viewpoint model). This can be translated into a reduction of 19% from the value of the original sequence. Figure 8.16a depicts the original BGTI structure viewpoint MSM whilst Figure 8.16b depicts the optimised structure viewpoint MSM resulting from the application of the Module Identification Mechanism both viewpoint models.

---

40 These parameters are consistent with previous findings for the GA parameters in such a domain Whitfield, R. I., J. S. Smith, et al. (2002). Identifying Component Modules. Artificial Intelligence in Design (AID '02), Cambridge, UK..



a) original structure viewpoint model

b) optimised structure viewpoint model

Figure 8.16: Original and optimised BGTI Structure Viewpoint Models

Similar to the findings of the ITM implementation, the concept groupings in the two models are very similar with the components related to electronics processing (depicted by the blue overlaid box in Figure 8.16a and b) the control and processing (depicted by the white box overlaid box in Figure 8.16a and b) grouping closely. However, the 19% reduction in the clustering criteria has resulted in more tightly grouped sets of concepts, which is evident from the *Modular Structure Viewpoint Models* (MSVM) depicted in Figure 8.16.

An analysis of the average number of concepts per module, for each returned MSI value, within the original and optimised viewpoint models shown in Figure 8.16 was undertaken (similar to that illustrated in ITM example through Figure 8.12 and Figure 8.13). Table 8.12 provides an overview of this analysis.

MSI Value	Average Number of Concepts in Module	
	Original MSM	Original MSM
1.0	2.0	2.0
0.9	---	2.0
0.8	2.5	3.0
0.7	3.0	3.0
0.6	3.5	4.4
0.5	3.5	5.6
0.4	4.0	7.8
0.3	10.5	17

Table 8.12: Overview of concept groupings in the Original and Optimised Modular Structure Matrix

Comparing the resulting module groupings of those in the optimised model (Figure 8.16b) with those in the original model (Figure 8.16a) we see that the number of concept grouped into modules for each consecutive value of MSI is, in general, greater in the optimised version as opposed to the original (with MSI value 1.0 being the exception). As explained previously, through the ITM evaluation, the reduction of the number of module boundaries is a key objective for the MD practitioner and as such the optimised version of the BGTI can be considered to represent a more robust and cost effective solution.

The BGTI implementation focussed on the application and evaluation of the mapping mechanism. Table 8.13 outlines the results of the cross-viewpoint analysis of the BGTI implementation. The analysis was undertaken to determine how well the modular solution had been maintained across the differing viewpoints of design. The table illustrates the results of two analysis scenarios: (i) mapping forward from the optimised function viewpoint model and (ii) mapping backwards from the original structure viewpoint model (see Figure 8.16)[41].

GA Parameters: 250, 250, 80, 20, 2 point end, 2 point adjacent			
Mapped Forward from Optimised Function	=	Objective (Optimise $CVCC_h$)	
	Function	Working Principle	Structure
Clustering Criteria Value	2268	1143	1975
%Difference	0%	31%	51%
Mapped Backwards from Original Structure	=	Objective (Optimise $CVCC_h$)	
	Structure	Working Principle	Function
Clustering Criteria Value	3524	1120	968
%Difference	35%	30%	0%

Table 8.13: Cross-Viewpoint mapping results

The first set of analysis results were derived based on the process illustrated in Figure 8.17.

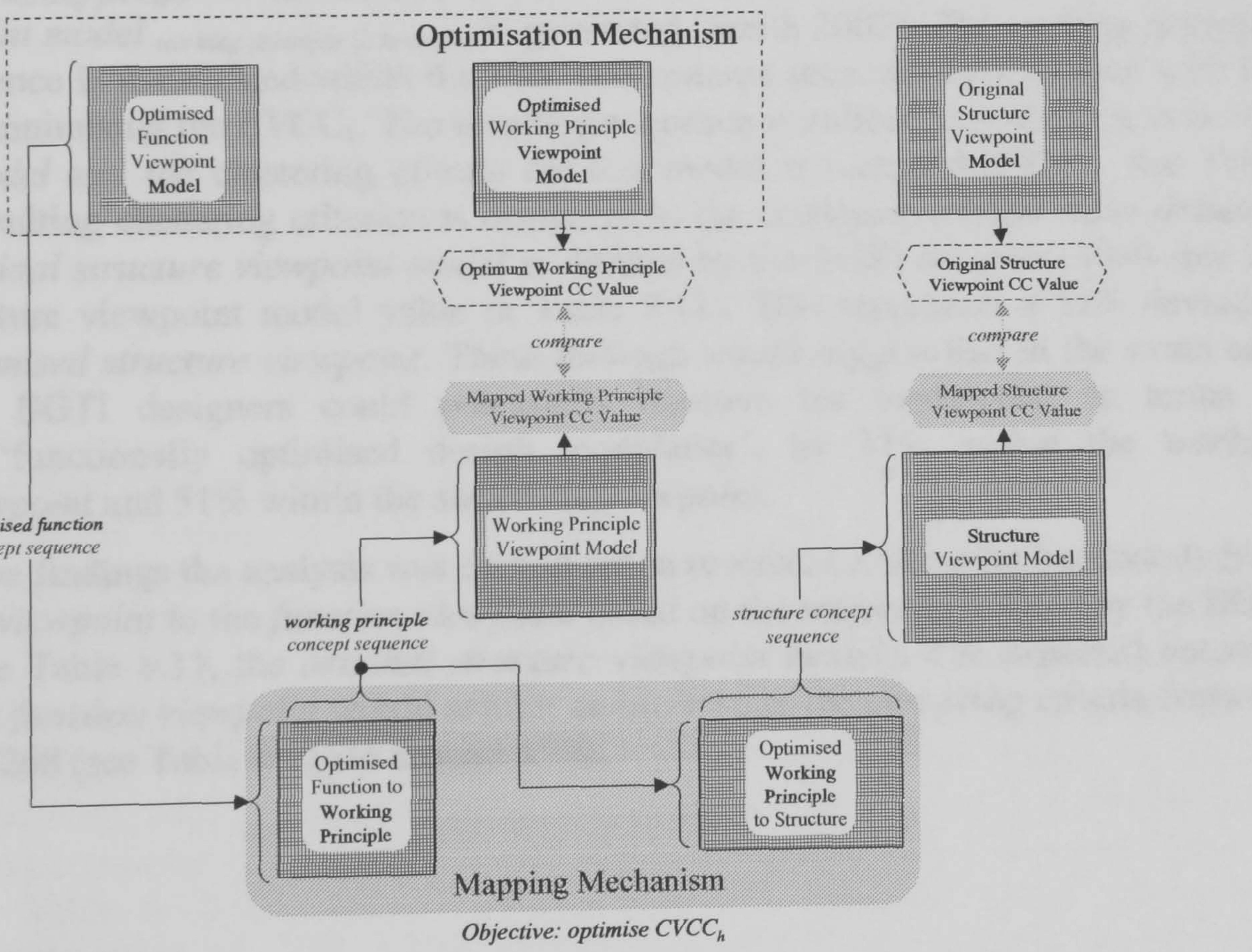


Figure 8.17: Mapping the *optimised function viewpoint* modular solution

---

[41] The results are fully detailed in the report by Smith, J. S. (2002). Battle Group Thermal Imager Multi-Viewpoint Modular Design Methodology Implementation Results. Glasgow, UK, University of Strathclyde: 17.

This process is based on the utilisation of the mapping mechanism as a means to maintain the modular solution as discussed in Section 6.4.2 and illustrated in Figure 6.30. The figure depicts the three optimised viewpoint models for which the results of the application of the optimisation mechanism on individual viewpoint model analysis are already known (Smith 2002). The optimised sequence returned for the BGTI *function viewpoint model* is utilised as the basis for a cross-viewpoint analysis. A cross-viewpoint model detailing the causal relations between the function concepts and the working principle concepts is generated. The optimised sequence of function concepts is maintained and the working principle concepts are optimised with respect to this sequence, i.e. the mapping mechanism optimisation element is applied with the objective of optimising the *horizontal cross-viewpoint clustering criteria* ($CVCC_h$).

The $CVCC_h$ is optimised as the row concept sequence (in this case the *function concept sequence*) is maintained and as such there can be no alteration in the vertical position of dependencies (see Section 6.4.2, Figure 6.30). As illustrated in Figure 8.17 the resulting working principle concept sequence is utilised as the basis for an individual viewpoint model analysis and to map to the solution viewpoint, i.e. the sequence that resulted in the optimisation of $CVCC_h$. An individual working principle viewpoint model is generated utilising the concept sequence, obtained from the *cross-viewpoint model*$_{function\ to\ working\ principle}$ analysis. The clustering criteria value for this working principle sequence is recorded. The recorded working value (1143) is compared to that obtained from the *optimised working principle viewpoint model* (780 - see the *optimised viewpoint model* values in Table 8.11). As depicted in Table 8.13 this represents a 31% deviation from the value recorded for the optimised working principle viewpoint model.

The *working principle concept* sequence obtained from the analysis of the *cross-viewpoint model*$_{function\ to\ working\ principle}$ is also utilised as the basis to map to the *structure viewpoint*. A *cross-viewpoint model* $_{working\ principle\ to\ structure}$ is generated (Smith 2002). The *working principle concept* sequence is maintained whilst the *structure concept* sequence is optimised with the objective of minimising the $CVCC_h$. The resulting sequence is utilised to generate a *structure viewpoint model* and the clustering criteria for this model is recorded (1975 – see Table 8.13). The resulting clustering criterion is compared to the clustering criteria value obtained from the *original structure viewpoint model* as defined by the BGTI designers (968- see the original structure viewpoint model value in Table 8.11). This represents a 51% deviation from the *optimised structure viewpoint*. These findings would suggest that in the worst case scenario the BGTI designers could potentially improve the modularity, in terms of maintaining 'functionally optimised design modularity', by 31% within the *working principle viewpoint* and 51% within the *structural viewpoint*.

To verify these findings the analysis was carried out in reverse, i.e. mapping backwards from the *structure viewpoint* to the *function viewpoint* based on the sequence defined by the BGTI designers (see Table 8.11, the *original structure viewpoint model*). The expected outcome being that the *function viewpoint* would exhibit an increase in the clustering criteria from the optimum of 2268 (see Table 8.11) to around 4500.

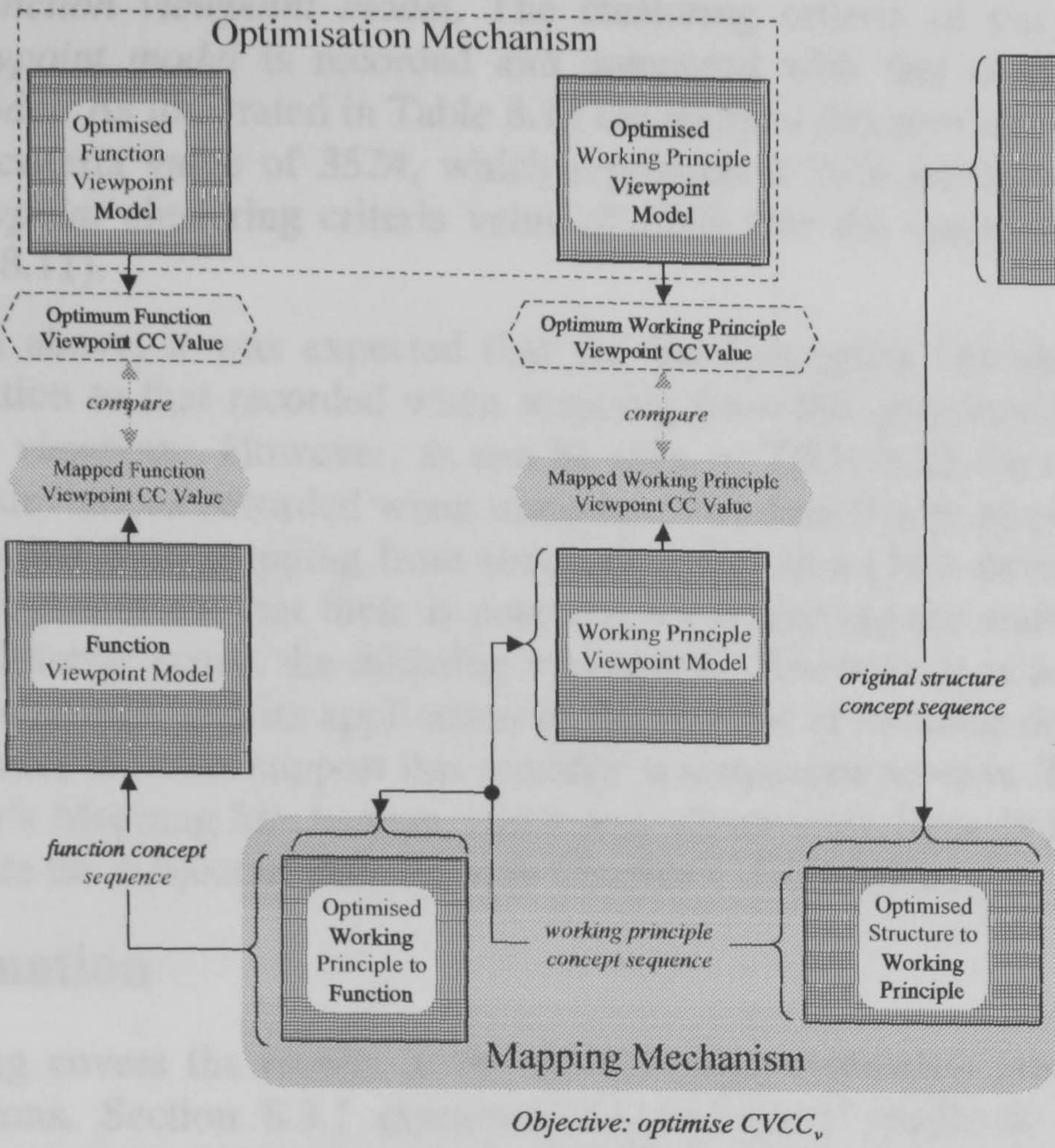


Figure 8.18: Tracing back the *original structure viewpoint* modular solution

Figure 8.18 illustrates the process undertaken to map the solution obtained from the original (BGTI team) modularisation of the *structure viewpoint* to the *function viewpoint*. The process is similar to that illustrated in Figure 8.17. However, in this 'back-mapping' scenario the objective of the application of the mapping mechanisms' optimisation element is the minimisation of the vertical *cross-viewpoint clustering criteria* (*CVCC$_v$*). In this case, the minimisation of the *CVCC$_v$* is the objective of the optimisation as it is the column concept sequence (in the first instance the *structure viewpoint* concept sequence) that is maintained and not the row sequence as in the previous forward mapping scenario. As such, the horizontal distance between dependencies cannot alter regardless of the order of the column concepts.

As depicted in Figure 8.18 the *cross-viewpoint model* $_{structure\ to\ working\ principle}$ is generated with the sequence of the *original structure viewpoint model*, i.e. that derived by the BGTI designers, being maintained in the column representation. The *working principle concept* sequence is then optimised with respect to this given *structure concept* sequence and with the objective of minimising the *CVCC$_v$*. The *working principle concept* sequence, which results in the optimisation of the *CVCC$_v$*, is then utilised as the basis for the generation of an individual *working principle viewpoint model* and the *cross-viewpoint model*$_{working\ principle\ to\ function}$. The clustering criteria of the generated individual *working principle viewpoint model* is recorded and compared with that of the *optimised working principle viewpoint model*. As illustrated in Table 8.13 the *mapped working principle viewpoint model* results in a clustering criteria value of 1120, which represents a 30% deviation form the *optimised working principle viewpoint* clustering criteria value of 780 (see the *optimised working principle viewpoint model* Table 8.11).

The *function concept* sequence of the *cross-viewpoint model*$_{working\ principle\ to\ function}$ is optimised with respect to the *mapped working principle* concept sequence and with the aim of minimising the *CVCC$_v$*. The resulting *function concept* sequence, obtained from the *optimised cross-viewpoint model*$_{working\ principle\ to\ function}$ is utilised as the basis to generate an individual *function viewpoint model.* The clustering criteria of the generated individual *function viewpoint model* is recorded and compared with that of the *optimised function viewpoint model.* As illustrated in Table 8.13 the *mapped function viewpoint model* results in a clustering criteria value of 3524, which represents a 35% deviation form the *optimised function viewpoint* clustering criteria value of 2268 (see the optimised function viewpoint model Table 8.11).

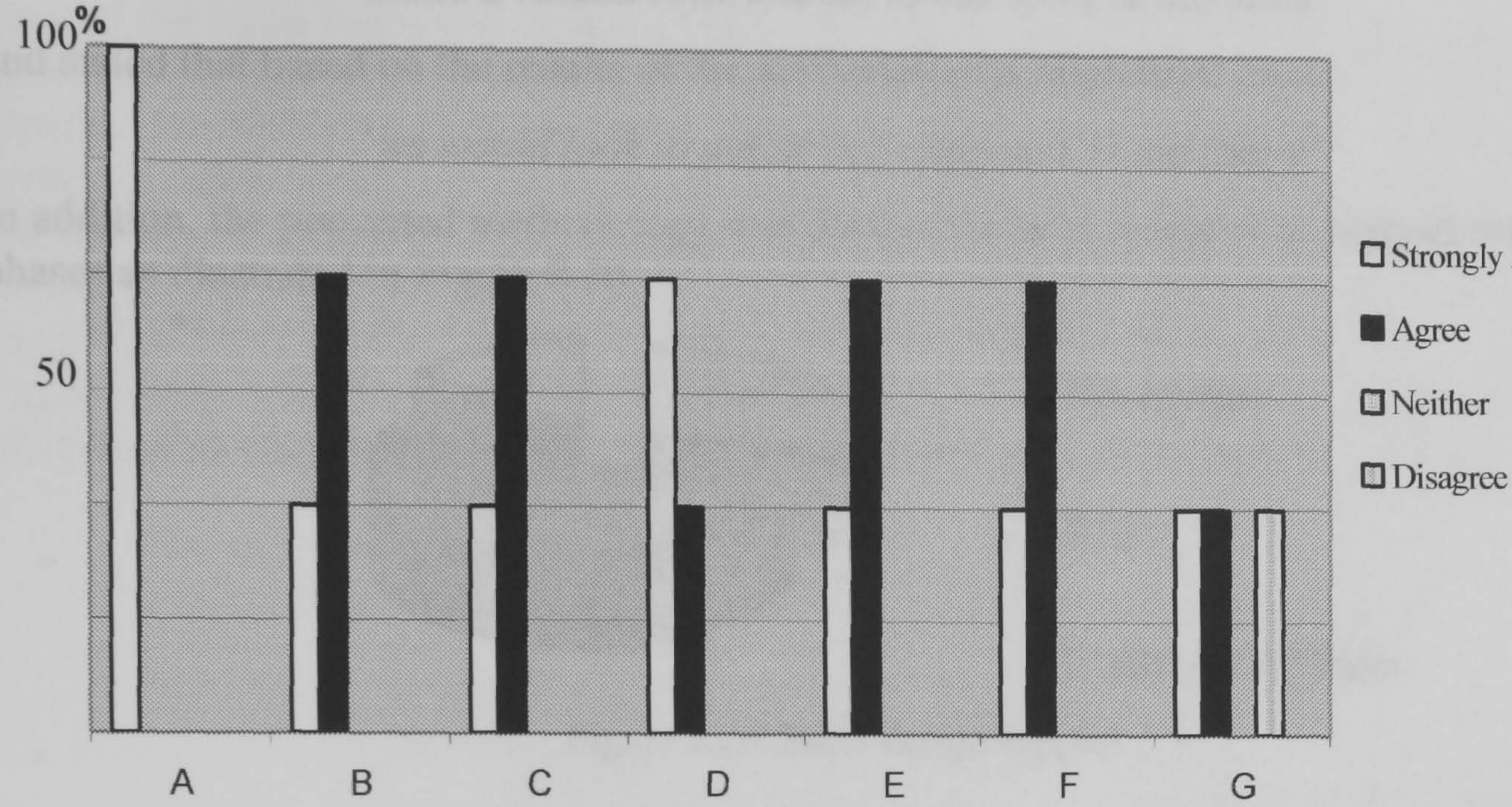
As discussed above, it was expected that the 'back-mapping' scenario would result in a similar deviation to that recorded when mapping from the *optimised function viewpoint* to the *structure viewpoint.* However, as can be seen in Table 8.13 there is a 15% difference between the deviations recorded when mapping from function to structure (50% deviation) and that recorded from mapping from structure to function (35% deviation). It is suggested that the findings indicate that there is potential for improving the maintenance of the BGTI design's modularity across the differing viewpoints. However, it is acknowledged that the Mapping Mechanism, and its application in the process of modular design, requires further research in order to better support this modular maintenance process. The capabilities of the Methodology's Mapping Mechanism, and issues which arose from its implementation in the BGTI case, are the subject of discussion in Chapter 9 (Discussion).

8.3 Evaluation

The following covers the results of the methodology evaluation process in the industrial implementations. Section 8.3.1 discusses the evaluators' feedback on the methodology application procedure, (outlined in chapter 5 of this thesis). Section 8.3.2 addresses specific elements of the methodology that were raised either through evaluation questionnaire or general discussions. Section 8.3.3 highlights the methodologies capabilities towards fulfilling the requirements identified by the ITM designers (Smith, Robb et al. 2001).

8.3.1 Methodology application

The following covers feedback on the methodology application process[42]. The evaluation process followed the procedure defined in Figure 8.6.



A	Lack of practical modular design methodologies
B	Lack of practical modular design tools to support the designer
C	Lack of practical modular design methodologies to support early design
D	Lack of practical modular design tools to support early design phases
E	Lack of training
F	Difficulties establishing what should constitute the content of each module
G	Difficulties understanding the impact of modular design on a products lifecycle

Figure 8.19: Main problems associated with industrial application of MD principles

Prior to the presentation of the industrial implementation evaluation results those participants with previous theoretical or practical experience of MD were asked to outline the main problems that they associated with the implementation of MD principles in industry.

Figure 8.19 depicts the evaluators response to the main problems associated with MD implementation in industry. The statements represented by A to G are detailed in the table below the chart in Figure 8.19. As can be seen from Figure 8.19, prior to the methodology implementation evaluation, all evaluators strongly agreed that the main problem associated with MD application in industry was a lack of practical MD methodologies. This was a problem further compounded by a lack of tools to support MD in practice with particular reference to early design support. Further, all evaluators either 'agreed' or 'strongly agreed' that they had experienced difficult in establishing what should constitute a module. It is suggested that the lack of articulate methodologies and tools for practising designer to implement MD compounds this difficult in establishing module boundaries.

[42] 12 individuals participated in the final evaluation session. The sample size for the evaluation can thus be considered to be 12.

On completion of the implementation and evaluation process 100% of the evaluation participants 'agreed' that the methodology served as an articulate procedure for the practising designer. Indeed, the BAE Systems team felt that the research and resulting methodology (see Appendix A);

'made a valued contribution to our work in this field'

and stated that based on the results of the methodologies implementation;

'we would look to utilise this approach in the future'

In addition, the presented methodology was deemed to have potential to support early design phases as illustrated in Figure 8.20.

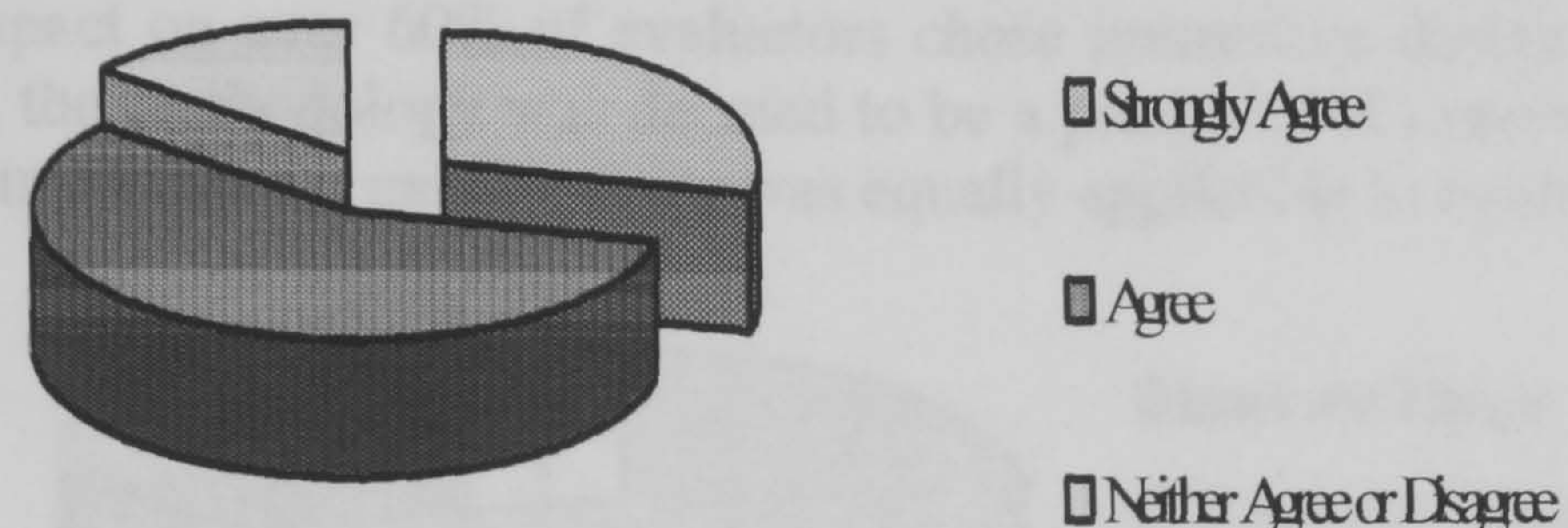


Figure 8.20: Early design support

Figure 8.20 depicts the evaluators' response with respect to the methodologies potential to support early design phases. Support for early design phases had previously been highlighted as a main problem for MD implementation in industry (see Figure 8.19). However the methodology developed and implemented as part of the work presented in this thesis was deemed by the majority of respondents (over 80%) to be able to support MD in early design phases, as can be seen from Figure 8.20. The designers within the ITM implementation were keen to ascertain whether through the implementation of the methodology the number of iterative reviews required to identify the module boundaries could be reduced. On completion of the methodology implementation and evaluation process the ITM design team agreed that they would look to utilise the methodology in future to support early design phases;

'in order to assist ourselves in clearly delineating the activity boundaries and hence reduce the number of iterations required for the whole design.'

Engineering design fields

The evaluators were from a number of different engineering design related backgrounds from design managers, systems engineers to naval architectures and support managers. All the evaluators agreed that they could envisage an application for the methodology in their area, verifying its generic nature and ability to integrate differing technological fields. The evaluators were asked to envisage the application of the proposed methodology to support in their specific fields of expertise and consider what they believed it would be utilised to support. There responses are illustrated in Figure 8.21.

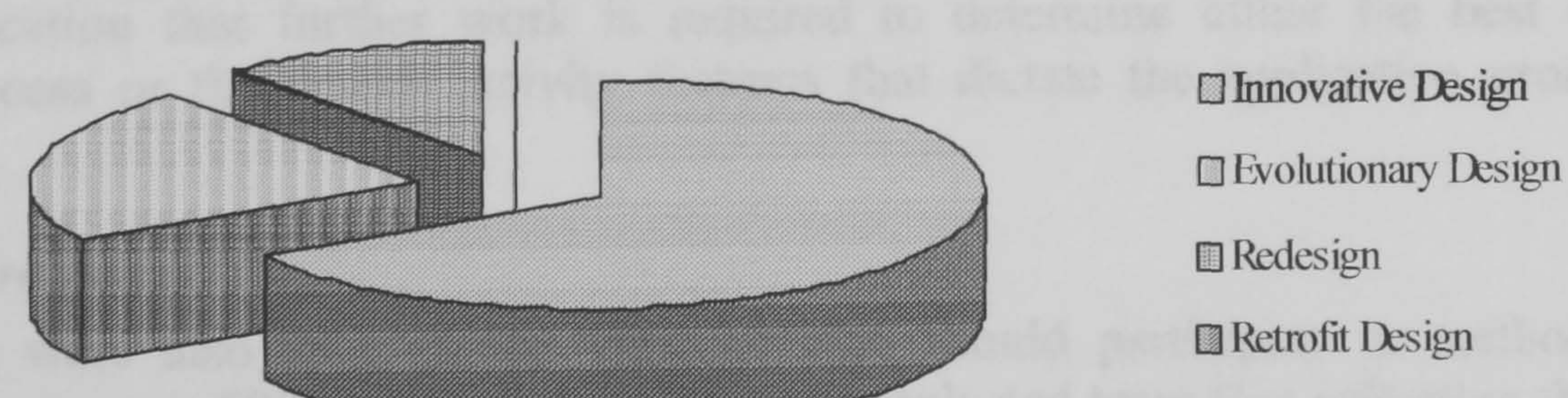


Figure 8.21: Application in evaluators' fields

Figure 8.21 illustrates that the majority of the designers would utilise the methodology to support innovative design (approx 66%) whilst 17% felt it would be most applicable to evolutionary design and 17% could envisage its application to support redesign activities.

Types of design.

One of the main perceived disadvantages of EDR is that it facilitates design fixation and stifles innovation. The methodology was specifically developed to address the requirements identified to develop a MD support for improved EDR. As such, the author was interested to ascertain, after exposure to the methodology, its application and its resulting findings, whether the evaluators would uphold this belief of EDR as an innovation suppresser. Interestingly, when asked to identify the type of design the methodology would have the most significant impact on over 60% of evaluators chose innovative design (as depicted in Figure 8.22). Thus, the methodology was deemed to be a promoter of innovation. A number of evaluators also stated that the methodology was equally applicable to evolutionary design.

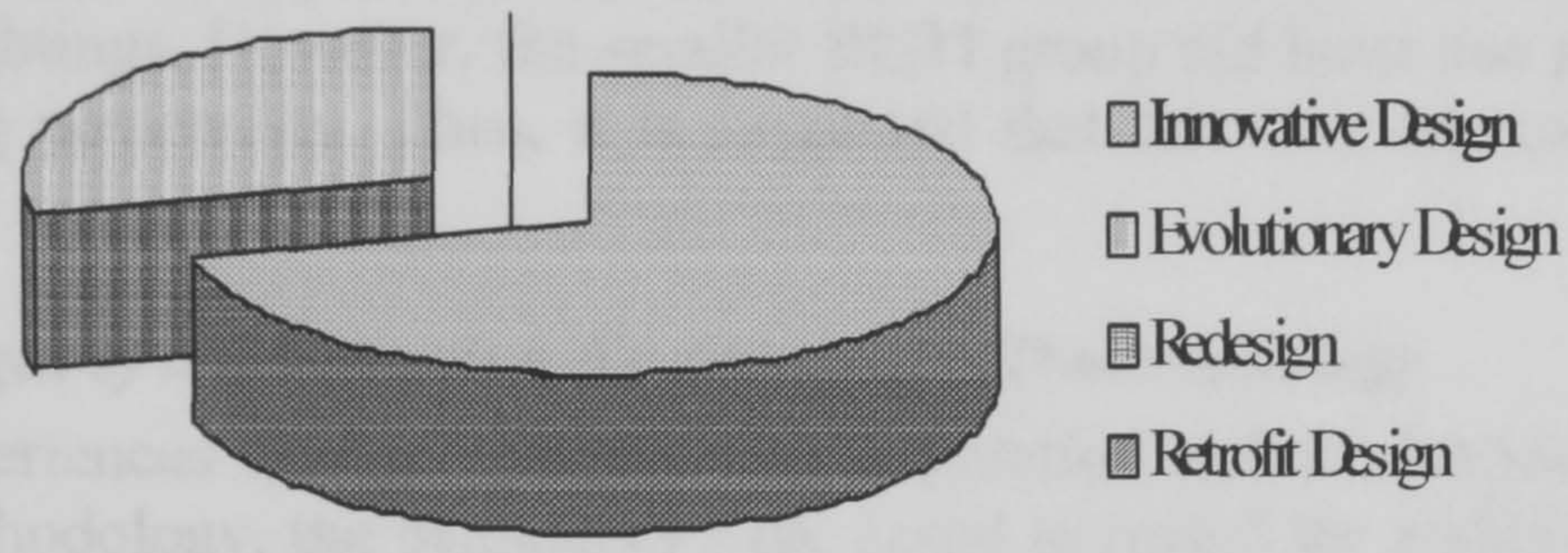


Figure 8.22: Impact of methodology on design type

Further, contrary to documented perceptions of EDR as applicable to cases of redesign (see Chapter 2), no evaluator chose either redesign or retrofit as the design type that the methodology would have the most significant impact on.

Where in the design process

Given the choice between applying the methodology to one specific *viewpoint* or throughout the design process, all evaluators felt that its strength came from its application to support an evolving design process. The design process was considered by all to cover the activities from specification to detail design (and in one case installation). This finding supports the initial review findings (Chapter 4) that current approaches fail to fulfil the potential of MD to support EDR due to their lack of consistent support for product knowledge generated throughout the evolution of the design activity.

The author's envisaged that the MD methodology be applied at discrete intervals, such as design reviews. It was envisaged that this would support knowledge modularisation, the identification of module boundaries and consequently support further design task allocation. However, the evaluators were split on the application procedure with 50% choosing to apply the methodology as suggested and 50% preferring to apply it as a continuous process (whenever new knowledge is generated for a *viewpoint*). As such, the author believes that this is an indication that further work is required to determine either the best overall application process or the design activity features that dictate the application process to choose.

Who would be responsible

The evaluators were also split on the issue of who should participate in methodology implementation process. 50% believed that a specially selected team that reflecting the core requirements of the design would be the most appropriate scenario. 50% believed that all design team members should participate. However, none of the evaluators' believed that a single dedicated user or all design team members individually (for example, in a shared computer application) were appropriate to support the methodology application. One of the reasons given for this was that:

'an individual would be unlikely to be able to reflect the design experiences of the team as a whole when gathering data or inputting data to support the analysis'

In addition, observations recorded during both the ITM and BGTI implementation do not provided any conclusive answers to this issue. For example, the ITM implementation evaluation involved the majority of the original design team whilst the BGTI implementation involved a specially selected set of participants. Both implementations resulted in the successful completion of the models. Both implementations provided sufficient material for their further analysis and there was little difference in the time taken to complete the models (though the ITM case was a one day workshop and the BGTI case was carried out in smaller units of time over a one week period) and there was no effect on the discussion times over concept and/or weightings. However, the smaller BGTI group did have one participant with a particularly strong personality. Thus, it is suggested that this may represent an area for further study.

Additional advantages of implementing the proposed MD methodology

Based on their experiences during both the implementation and evaluation of the Multi-Viewpoint MD Methodology, the evaluators were asked to reveal the additional advantages they envisaged receiving from its application. Figure 8.23 depicts their responses.

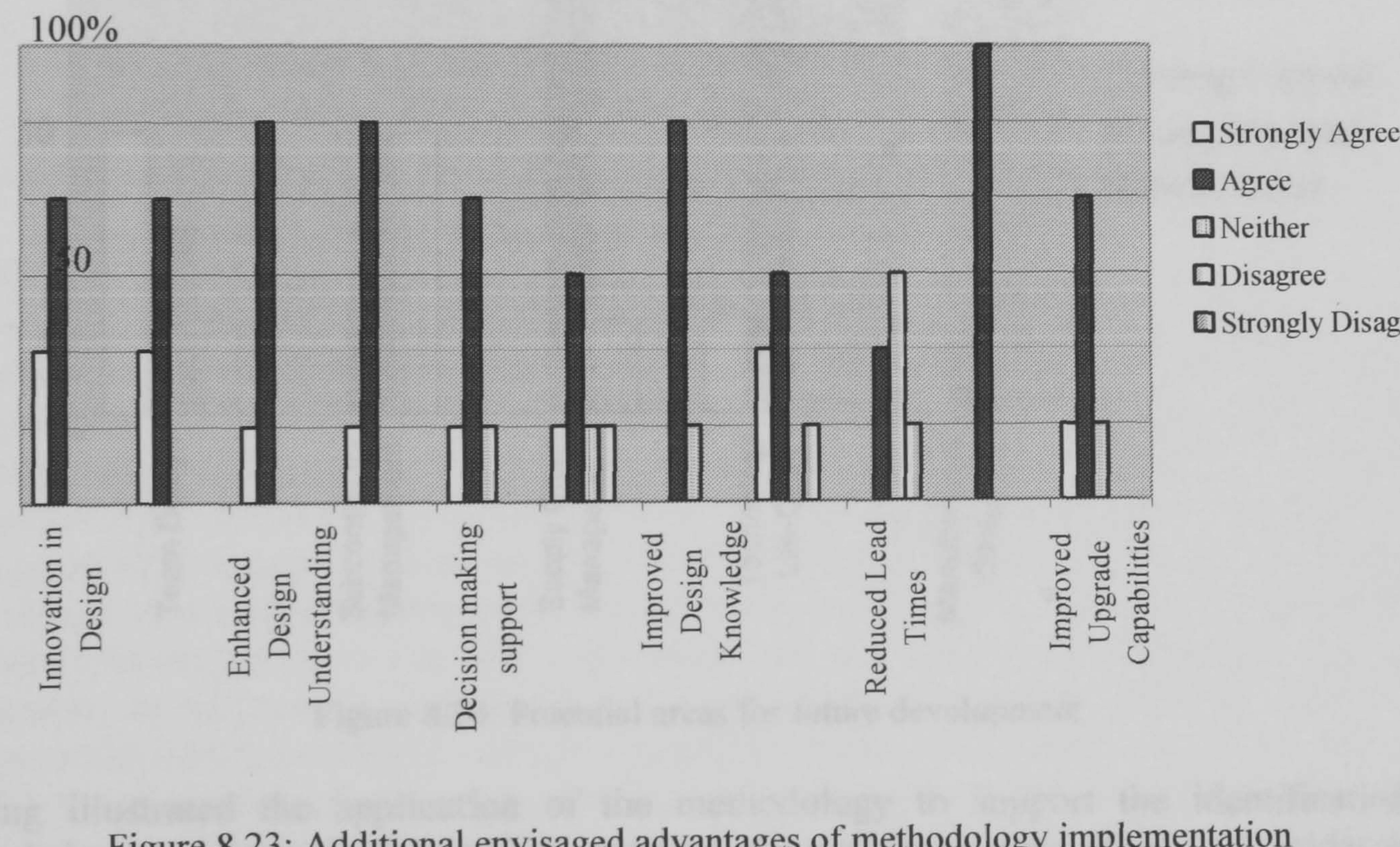


Figure 8.23: Additional envisaged advantages of methodology implementation

As illustrated in Figure 8.23 there were a number of advantages other than that of facilitating KMn that the evaluators foresaw from the implementation of the methodology. The evaluators most strongly agreed that the methodology supported the following: innovation in design (discussed above – see Figure 8.21 and Figure 8.22 above), learning in design, enhanced design understanding, decision making, design complexity reduction and upgrade capabilities improvement. For example, with respect to improving design understanding and supporting learning, during the BGTI case a novice designer (having only 2 months experience with the company) stated that:

'helping to populate the matrix with more experienced engineers, taught me more about the functionality of the subsystem as a whole'.

In addition, approximately 65% of participants agreed that the methodology supported design rationale capture, improved design knowledge structuring. These are elements discussed as central to improved EDR support in Chapter 2.

The methodologies potential implications in terms of reducing manufacture and assembly complexity, lead times and associated costs were also noted by a number of respondents. However, fewer than 35% agreed that the methodology could support cost reductions, and/or reduced lead times during design. This may be attributable to the fact that most of the evaluators felt that although *'the benefits were apparent'* the implementation of the methodology as part of their standard design practice would involve a:

steep learning curve' and result in *'high initial costs*

Similarly another evaluator commented that they;

'feel it will reduce design time for future ships – but needs investment at the right level to address the approach'

Future developments

A number of areas for future development of the methodology application were outlined as by the evaluation participants, as illustrated in Figure 8.24.

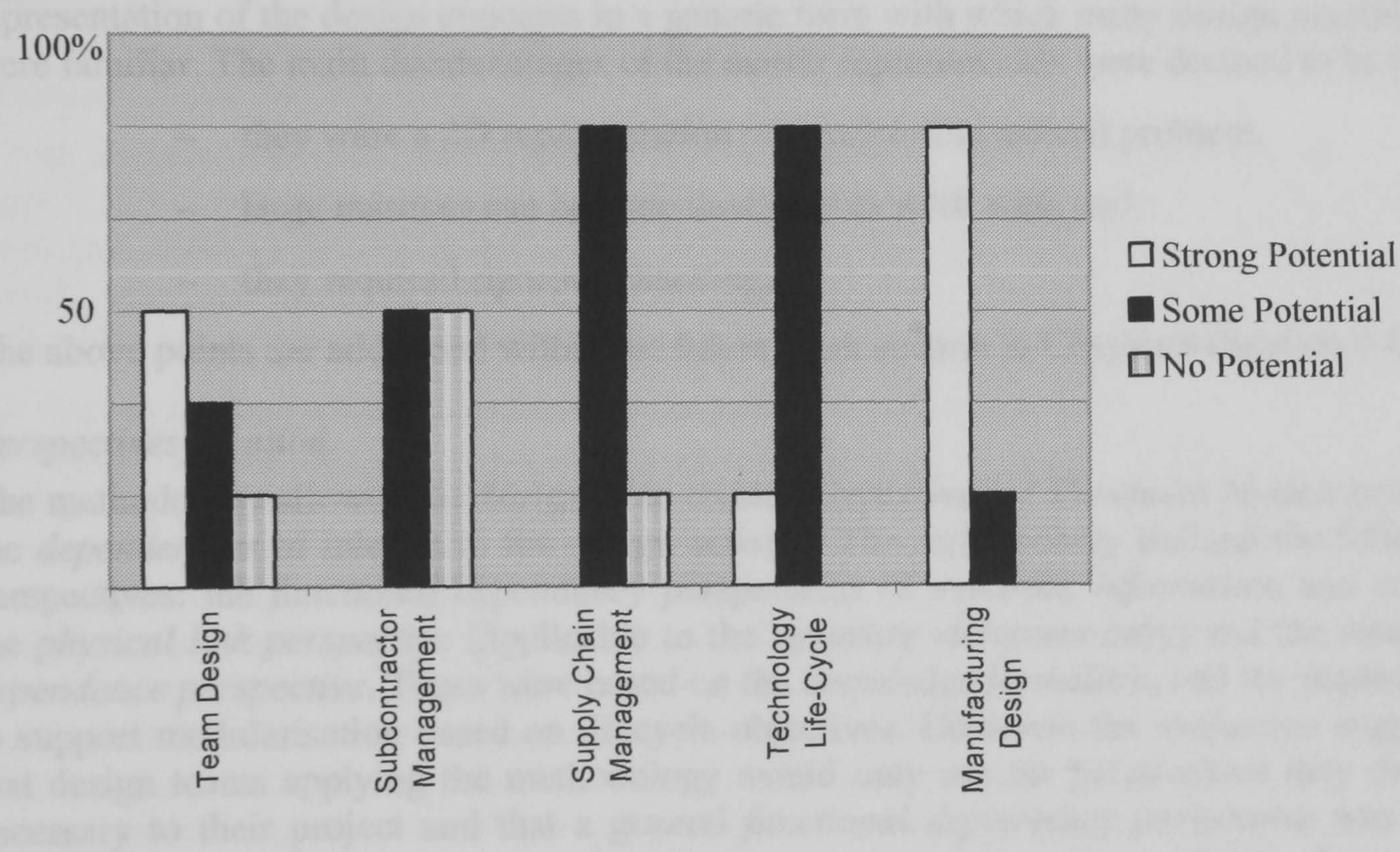


Figure 8.24: Potential areas for future development

Having illustrated the application of the methodology to support the identification of Knowledge Modularity in the product structure, the evaluators were asked to consider areas to which they believed the methodology could be adapted to support. The strongest potential areas for application were envisaged to be team design and manufacturing design. The strong potential for the area of team design, may be attributable to the fact that the designer were observed during the workshops to readily associate the grouping of related 'function concepts', required to be fulfilled by the design, to the expertise required to develop solutions to these function concepts. Thus, identifying correlations between the modularisation of function concepts and that of design team structures. The manufacturing design potential was related to the ease in which the modularisation of solution concepts was readily interpreted from a manufacturing or assembly standpoint as illustrated in Figure 8.14.

In addition, supply chain management and technology lifecycle management support were felt by many of the respondents to have some potential for support through the application of the methodology. The evaluation participants felt that a *'perspective'* could be created to represent knowledge related to these areas and used to bias the optimisation process to support these areas. For example the components sourced from a particular supplier could be clustered based on the *similarity-dependence$_{(supplier)}$* into fewer modules. As such, the designers could limit the effect of a delay in the procurement and/or delivery of those components on the overall manufacture and/or assembly process.

8.3.2 Specific elements

The following cover specific elements of the methodology raised by the company feedback.

The following discusses the *matrix formalism*, *perspectives, dependency weighting optimisation mechanism* and *module identification mechanism* and their strengths and weaknesses in terms of their support for the concept of KMn.

Matrix Formalism

A matrix formalism has been utilised to represent generated design knowledge in a form suitable for analysis. All the evaluators either 'agreed' or 'strongly agreed' that the matrix formalism, part of the methodologies *Modelling Formalism* element, provided a clear, visual representation of the design concepts in a generic form with which many design practitioners were familiar. The main disadvantages of the matrix representation were deemed to be that:

- they were a 2D representation of a multi-dimensional problem,
- large matrices can become unwieldy to work with, and
- they required rigorous checking.

The above points are addressed within the future work section in Chapter 9 (Section 9.4).

Perspectives creation

The methodology allowed the designers to create perspectives of *Viewpoint Models* based on the *dependencies* of interest to the design activity. The methodology defined the following perspectives: the *functional dependency perspectives* of *material, information* and *energy*, the *physical link perspective* (applicable to the *structure viewpoint* only) and the *similarity dependence perspective*. These were based on the *knowledge formalism*, and the requirement to support modularisation based on lifecycle objectives. However, the evaluation suggested that design teams applying the methodology would only use the *perspectives* they deemed necessary to their project and that a general *functional dependency perspective* was often preferable. In addition, a number of evaluators suggested that the *similarity dependence perspectives* which embodied customer requirements/groupings and technology lifecycles would be of interest to them but that the concept required further development and/or training to be applied in practice. However, *perspectives* in the methodology implementation are generic in that the user is not forced to define all, or indeed any, of the above *perspectives* and can choose to create as many or as few *perspectives* as deemed necessary. In addition the user can create any combination of these individual *perspectives* to support further analysis of the MD problem.

In cases where a combined *perspective* is created the evaluation process raised the requirement to be able to weight the *perspectives* in terms of their importance to the design activity objectives i.e. in defining a MD for, say, assembly the *Physical Link Perspective* may be deemed more important than, say, the *Functional Dependency Perspective $_{energy}$* and the designer may wish to reflect this in the combined *perspective* (currently defined as the average of the individual perspectives). This issue is addressed in detail in future work, section 9.4.

Dependency weighting

The *Viewpoint Models* (and *perspectives* of these) may be populated with weighted crosses that represent the *dependencies* within that *perspective* of the *viewpoint*. It is this knowledge, of *dependencies* between *concepts*, which is utilised as the means to optimise the clustering of *concepts* into potential module groupings. The *dependency* weights are an important element of this clustering process as the objective of the optimisation is to minimise a clustering criteria that is based on the position and weight of the *dependencies* with respect to the models leading line (as defined in Chapter 6). Currently, the implementation allows the dependencies to be defined as a value between 0 and 1.0 (in incremental steps of one). However, the evaluation experiences have shown that many practitioners would prefer to utilise fewer values when defining dependency knowledge as the arguments which ensue over the difference between a dependency valued at 0.8 and 0.9 can be time consuming and counter-productive. Thus, in the BGTI case study the designers chose to utilise only three quantitative dependency weights (1.0, 0.5 and 0.1) to represent the qualitative values: high, medium and low. However, the author feels this is a preference specific to each implementation and the generic nature of the current methodology implementation can support both.

Optimisation Mechanism

It was hypothesised that the application of a GA to facilitate *concept* clustering would support the optimisation of modularity. The methodologies' *Optimisation Mechanism* was developed through the definition of a *Clustering Criteria (CC)* and the utilisation of a genetic structure for a *Genetic Algorithm (GA)* (with crossover and mutation operators defined for the domain) to test this hypothesis.

The four evaluation studies illustrated the capabilities of this approach when combined with the *Module Identification Mechanism*. In all four cases, the *concept* groupings returned by the *Optimisation Mechanism* provided the basis for the identification of valid modules within the viewpoint model (see Section 8.1.2 and Appendix A). The differences highlighted in the alternator case study were attributed to the further application of domain specific knowledge (in the original publication) which they author believes could have been better represented in the initial dependency knowledge definition. A validation of both the industrial studies highlighted that the returned modules were felt to be appropriate for the application. Indeed, a 26% reduction in the clustering criteria of the ITM example was returned, which was further translated into a reduction in the number of modules and consequently the between interfaces required to fully define the products modularity. During the evaluation process all the contributors rated the *Clustering Criteria* as 4 or 5 (on a scale of 1 to 5, where 5 is excellent and 1 is very poor) for its performance as a 'measure of modularity'. The ITM implementation resulted in what the evaluators deemed to be;

'a correct representation of the modularisation of the design and manufacture of the technical demonstrator (ITM)'

The main issue with the *Optimisation Mechanism*, raised during the case studies, was the inadequacy of the Clustering Criteria to account for 'bus type' modularity. In addition, the evaluators expressed an interest in the ability to constrain the optimisation process based on their experiential knowledge. This issue is addressed in the future work section (Section 9.4).

The *Optimisation Mechanism* can be applied to any *perspective* of a *Viewpoint Model*. *Concepts* and their sequence are defined within the methodology as being *viewpoint* specific whereas the *dependencies* are *perspective* specific. This distinction supports the determination of the impact of one form of modularity on another (as depicted in the ITM example, Section 8.2.4).

Module Identification Mechanism

The clusters defined by the *Optimisation Mechanism* represent *concept* groupings from which the designer can extract module candidates. The identification of modularity within a product or system was identified as one of the main inadequacies of current approaches. The methodologies' *Module Identification Mechanism* has been defined to address this issue and support the extraction process. The *Module Identification Mechanism* provides an alternative interpretation of the *Viewpoint Model*, termed the *Modular Structure Model (MSM)*. This modular interpretation of the *Viewpoint Model* is based on the values returned from the application of a *Module Strength Indicator (MSI)* function. The *MSI* function (see Chapter 6) was developed based on a definition of modularity and in essence is a measure of the internal and external dependencies of a *concept* grouping. Based on the hypothesis that any *concept* grouping of two or more *concepts* may constitute a module, a novel application process for the MSI function is determined and thus it is applied to all possible *concept* groupings in the model. The application of the *Module Identification Mechanism* and its outcome (Modular Structure Model) has been shown to support the identification of inherent modularity in the model. As illustrated, in all four case studies the *MSM* interpretations allows the designer to not only extract module candidates but also determine different module configurations over hierarchical levels of the structure. The evaluators consistently returned an 'excellent' rating for the *Module Identification Mechanism*.

A main issue with the *Module Identification Mechanism* is that the *MSM* interpretation colours all matrix squares regardless of whether they have are dependent or not. However, the author intended the *MSM* interpretation to be utilised, along with the *Optimised Viewpoint Model*, as a means to support module extraction and not as a substitute for the model.

Mapping Mechanism

The utility of the methodologies Mapping Mechanism was addressed during the evaluation and Figure 8.25 depicts the resulting findings.

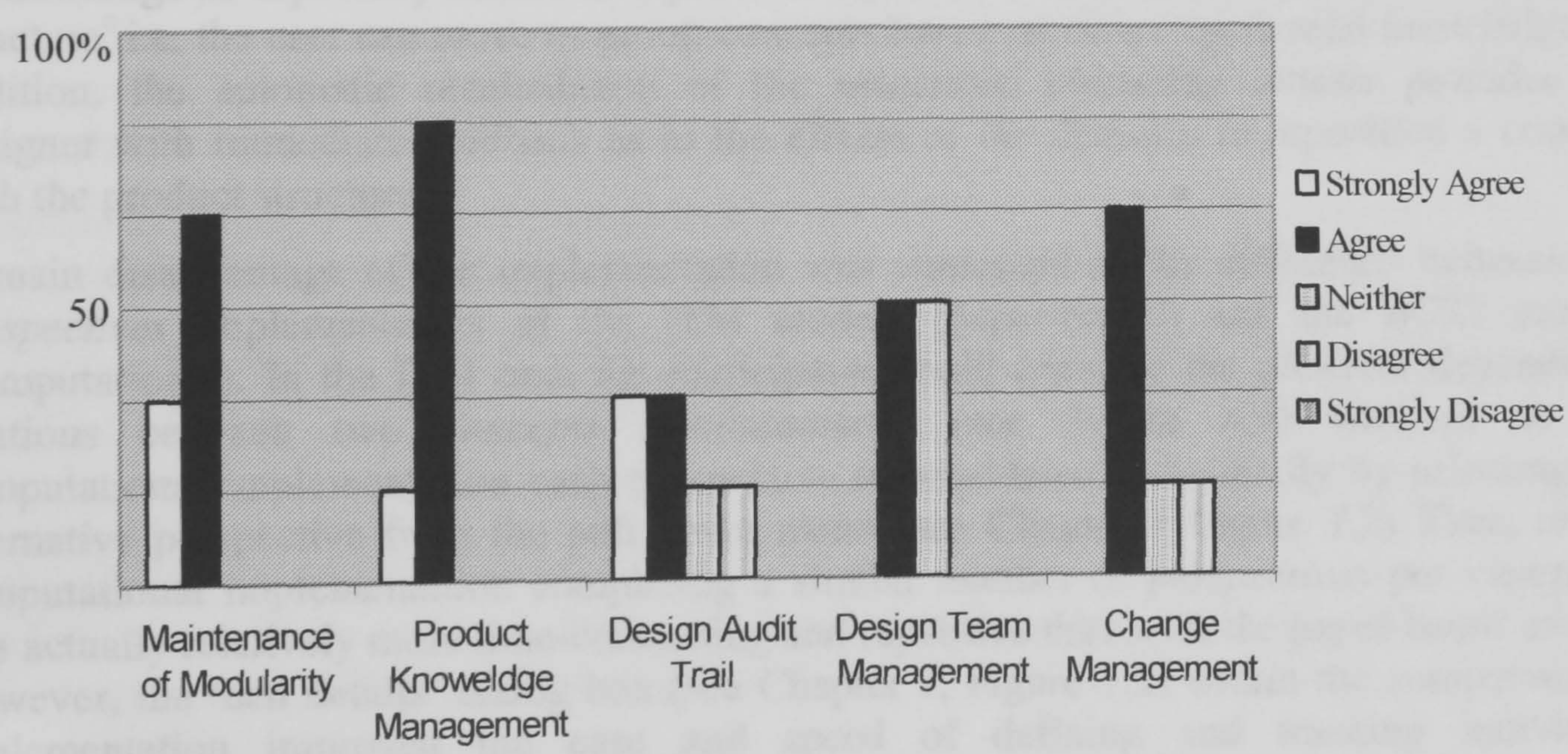


Figure 8.25: The support afforded by the Mapping Mechanism

The figure illustrates that the Mapping Mechanism was deemed to most strongly support the maintenance of modularity. This supports the reasoning behind the initial development of the multi-viewpoint approach itself. However, the BGTI implementation results indicate that this element of the Methodology requires further research and development to enhance its capabilities. In addition, product knowledge management was another area that the mapping mechanism was deemed to provide support for. A BGTI designer perceived the mapping mechanism as a mechanism to support the design audit trail. The designer explained the ability to track the purpose for a chosen working principle and/or component was especially pertinent in the military field due to the rigid specification and the conformance testing requirements.

Insights

The following covers a number of insights recorded during the ITM evaluation of the Multi-Viewpoint MD Methodology and presented to the ITM designers during the presentation of the results.

System Implementation

The evaluators were asked to rate elements of the system implementation of the approach on a scale of 1 to 5 (where 1 was very poor and 5 was excellent), with the following results:

	Excellent				Poor
	5	4	3	2	1
Concept Generation		|			
Dependency Generation		|			
The Optimisation Performance		|			
The Optimisation Speed			|		
The usefulness of the manual 'click and drag' re-sequencing		|			

Figure 8.26: Computational Implementation

All aspects of the computational realisation were rated above average. The optimisation performance, concept and dependency generation and the manual 'click and drag' re-sequencing were all rated very good to excellent. The manual 'click and drag' feature was acknowledge as especially useful as it provides the user with some control over the product structure, i.e. the user can move or group concepts based on their experiential knowledge. In addition, the automatic recalculation of the associated clustering criteria provides the designer with immediate feedback as to the effects of the decision to reposition a concept with the product structure.

A main disadvantage of the implementation was witnessed in the difference between the *perspectives* implementation of the ITM models (paper-based) and the BGTI models (computational). In the ITM case the participants could consider the different dependency relations between two *concepts* simultaneously (see Figure 8.9) whereas in the computational implementation each perspective is considered individually by selecting the alternative perspective from the pull down menu (see Chapter 7 Figure 7.2) Thus, in the computational implementation completing a similar number of *perspectives* per viewpoint was actually relatively more time-consuming and repetitive than with the paper-based model. However, the 'cell details' dialog box (see Chapter 7, Figure 7.1) within the computational implementation improved the ease and speed of defining and tracking individual dependencies between *concepts*. Thus, the author suggests that an alteration in the computationally implemented model is required to ease the *perspectives dependency* input.

Understanding viewpoint concepts

One of the main observations during the ITM evaluation was that the participants had difficulty understanding some of the concepts. This was especially pertinent in more abstract viewpoints such as function. The lack of understanding was on most occasions attributable to differences in terminology used by the author and evaluation participants, as the creation of the Viewpoint Models was carried out by the author, i.e. the author applied the knowledge formalism and matrix formalism as shown in Figure 8.6. The author believes that the integration of DENOTE with the MD system implementation would address this issue i.e. the designers themselves would generated and evolve the CWK and thus use terminology with which they are familiar. The integration is addressed in Further Work, Section 9.4.

Automatic assignment of structure to function

The participants had difficultly dealing with function without automatically assigning a structure to it, thus, narrowing the design solution space considerably and reducing creativity and innovation. The ITM designers believed that this could be attributable to the very stringent standards that are imposed on naval and marine products, in that, there is often little leeway in terms of the final design solution, for example, there are standards which govern aspects of a ships design from the dimension and position of buttons on command controls to the dimension of cabin mattresses for differing naval ranks. In addition, they are often constrained by a list of approved suppliers (approved by the MOD), which in turn, has an immediate impact on the potential solutions available to the design team. The experienced designers are often familiar with the suppliers' products and thus automatically assign a structure when they are considering fulfilling a functional objective of the ship. On considering this point one evaluator stated that the MD approach, as defined through this work, would be a *'valuable tool for research and development projects'*

Dependency generation

The participants also had difficultly in defining certain types of *dependency* knowledge in particular *viewpoints*, i.e. *material dependencies* in the *function viewpoint,* and *energy* and *information dependencies* in *structure viewpoint.* However, despite the methodology advocating the utilisation of the *material, energy, information, physical link* and *similarity dependence* perspectives, the system functionality allows the user to create only the *perspectives* of interest to them i.e. they are not forced to create any set *perspective.* Thus, the generic nature of the system would allow the user to adapt the methodology to meet their own requirements. Indeed, this is a point raised by a over 65% of the evaluators who stated that they *'would only use the perspective that they deemed appropriate to each viewpoint'* or *'would create a more general 'functional' perspective'.* The BGTI case study utilises a general *functional perspective* to support the analysis.

In addition, during the results presentation the ITM case study evaluators raised the following points (which are addressed in Future Work, Section 9.4):

- The development of capabilities that would allow the designers to constrain the re-sequencing process based on their own experiential knowledge.
- The development of some form of consistency checking mechanism.

8.3.3 ITM Designers Requirements

The following briefly outlines the capabilities of the methodology with respect to the requirements outlined by the ITM designers.

Generic and complimentary approach

Due to the characteristics of the military design domain (see Section 8.2.1), a key requirement of a MD methodology, as defined by the practicing ITM designers, was a generic nature that complimented their current design activity. The methodology defined as part of this work embodies the *MVEA knowledge formalism* (Zhang 1998) and a generic matrix formalism as the basis to produce models for analysis.

The knowledge formalism was developed to define the knowledge (concept, relations, attributes and constraints types) that is generic to the activity of engineering design. In addition, the formalism is not itself based on any specific model of the design activity and therefore does not force the designers to by a design in a specific course (Zhang 1998). Thus, the formalism supports the MD methodology in being generic and complimentary. Though the matrix formalism is generic, it can represent a departure from the current design process in that, as in the case of the ITM design, no explicit analysis was generated early in the design to support module determination. This departure is reflected in a number of evaluators' comments that stated that although

'the usefulness of the tool and the results was apparent'

they found

'completing the matrices a little difficult to understand conceptually' and felt that 'it would require retraining of existing design engineers to embrace the approach'.

Lifecycle approach

In the first instance the designers were keen that the methodology be able to be incorporated from the project inception. The methodology has been based on the MVEA knowledge formalism to support this evolution of current working knowledge throughout the entire life-phase of design. In addition, the methodology application process is designed to support its continued utilisation as the design process, and its associated *viewpoints,* develop from the abstract to the concrete.

The extension of *perspectives* of design *viewpoint dependency* knowledge was developed to allow the designer to facilitate MD based on the various life-cycle objectives of the design. Though not thoroughly evaluated during the industrial case studies, the Alternator case was based on the life-cycle objectives of recycling and re-use and the methodology was shown to be capable of both representing these and analysing modularity based on these.

Evaluators' comments on this point included, that the methodology would:

'be useful to apply from scratch'

and

'worth trying out on a new product'

Technology integration capabilities

Products in the military design domain are often based on a number of different technologies including: mechanical (levers, mechanisms), optronics (lasers, thermal imagers), and software/electronics (controllers, processors). The designers were keen to have a MD approach that could integrate these. The *viewpoint perspectives,* defined as part of the methodology presented within this thesis, facilitate modularisation of a *concepts* with respect to a specific type of *dependencies*. The ITM case represented a good example of this, in that the designers were able to modularise the concepts from an *information* perspective and thus, define the boundaries for module(s) linked to communications and information processing between concepts (potentially realisable as software modules) from as early as the functional stage. The main disadvantage of the approach was the inability to modularise 'bus' elements, such as power supply cables.

Decision support

The ITM designers felt a key feature that a MD methodology should address was decision support. This included provide some interpretation of modularity to support their decisions on their choice of module boundaries. A review of current MD approaches (see Chapter 4) also highlighted this as one of their significant inadequacies. The utilisation of different perspectives of viewpoint models and the ability to create combined perspectives to analyse the impact of these on the modularity of the product structure was deemed to;

'give the team (designers, production, customers, through-life-support, human factors) an overall perspective on the impact of design decisions'

The *Module Identification Mechanism* addressed this issue and in all four case studies provided a measure of the modular value of *concept* groupings and an interpretation of these which supported the designers decision making process in configuring a modular solution to their design problem. The results of the Module Identification Mechanism in the two industry case studies were stated to:

'clarify of the types of modules required';

be

'illustrative and informative';

and provide

'an early identification of areas of design that require further work in order to optimise the installation'.

8.4 Chapter summary

Two case studies, taken from published literature, were undertaken to support the initial verification of the effectiveness and appropriateness of: the optimisation mechanism (with specific emphasis on the utility of the clustering criteria), the MSI function and its application process and the MSM representation of the DSM. The initial verification was sought before its exposure to designers and implementation in two industrial design activities, which posed relatively more complex and larger problems.

The studies verified that the clustering criterion was effective at grouping the *concepts* to support their further modular analysis. The groupings were in alignment with the original 'module' findings. The parts (MSI function and MSM) of the Module Identification Mechanism were shown to support the previous studies 'module' findings whilst providing a more comprehensive interpretation of the modularity that exists within these cases.

As a means to evaluate the capabilities of the Multi-Viewpoint Modular Design Methodology and its application, this chapter presented two industrial evaluations. The two implementations highlighted different elements of the methodology, the ITM study emphasising the *Modelling Formalism*, *Optimisation Mechanism* and the *Module Identification Mechanism* elements, and the BGTI case, the *Mapping Mechanism*. The evaluation results were:

Modelling Formalism

The modelling formalism was shown to provide a generic approach on which to model design knowledge to support its subsequent modular analysis. The matrix formalism on which the model is based was deemed to be generic while providing a clear and concise model on which to facilitate the input dependency data by designers. The model data was shown to be appropriate for exploring and identifying modularity.

The main disadvantage of the model was that it represented a multi-dimensional problem as a two dimensional model.

Optimisation Mechanism

Through all four cases studies the optimisation mechanism was shown to result in the appropriate clustering of design concepts on which to base the identification of modules in the product structure. As such, the combination of a clustering criteria and genetic algorithm application was shown to be a suitable approach to optimise modelled design knowledge to facilitate module identification.

The main disadvantages of the optimisation mechanism was that due to the nature of the GA in that the same resulting solutions could not be guaranteed for the application of the same optimisation structure, i.e. the same population, generation, crossover and mutation parameters.

Module Identification Mechanism

The module identification mechanism was shown to aid the designer in interpreting the modularity of the optimised model. In all four cases, the formalised definition of the module (MSI function) and its novel application process was shown to facilitate the identification of inherent modularity that was not evident from the optimised models (and in the case of the literature based studies, the original published results).

Mapping Mechanism

The mapping mechanism was shown to have some potential to support the maintenance of modularity across viewpoints. The mechanisms evaluation within the BGTI implementation would suggest that there is potential for improving the 'functionally optimised modularity' of the BGTI design. However, the results were inconclusive and a requirement for further research work to support the modular maintenance problem was raised through this implementation. A BGTI team member raised the view that the current mapping mechanisms had the capacity to support the development of a design audit trail.

Methodology Application

The methodology and its application were deemed to represent an articulate procedure for designers to follow in practising modular design. The evaluation showed that, contrary to current perceptions of EDR support, the participants believed the methodology application would promote innovation. In addition, the evaluators deemed that modularising over multiple viewpoints could support a number of design related activities from 'designing' design teams to managing technology life-cycles for ease of retrofit and or maintenance.

Despite the general agreement as to the utility and potential of the methodology application which arose from the industrial implementation a number of points issues were raised which the designers believed were appropriate fields for further study including; the development of modular costing metrics, and the enhancement of the computational realisation. These are addressed through the discussion in Chapter 9.

9 Discussion

Based on the experiences during development and evaluation of the Multi-Viewpoint Modular Design Methodology, the following discusses; the resulting findings, the formalisms and mechanisms embodied within the Methodology, the research approach undertaken and lastly, suggests potential areas for future work. Section 9.1 discusses the research outcomes in terms of their ability to meet the requirements (defined in Chapter 3), and to address the inadequacies of existing MD approaches. Section 9.2 reviews the formalisms and mechanisms embodied with the Methodology. Section 9.2 appraises the research approach undertaken to facilitate the work presented in this thesis. Section 9.4 defines elements of the future work highlighted by the research undertaken as part of this work and through discussions of the methodology. Finally, section 9.5 summarises the chapter.

9.1 Research outcome

The following reviews the outcomes of the research undertaken, as part of the work presented in this thesis. Section 9.1.1 outlines the resulting Methodologies ability to meet the requirements of MD support for improved EDR and address the inadequacies of existing MD approaches in terms of providing multi-viewpoint MD support. Section 9.1.2 discusses the findings that resulted from the implementation of the Methodology.

9.1.1 Requirements for MD support for improved EDR

The Multi-Viewpoint MD Methodology is discussed in this section with reference to the requirements set out in Chapter 3 and inadequacies outlined in Chapter 4.

Support the dynamic knowledge generated throughout the design activity

The methodology utilises a previously defined knowledge formalism, (MVEA embodied by the DENOTE system) due to its capabilities in support the designer to explicitly capture design *concept* and *dependency* knowledge from different *viewpoints,* and across *viewpoints*, while modelling their evolution from the abstract to the concrete. The MVEA knowledge formalism, adopted within this work, was developed to support the construction, maintenance and further utilisation of both CWK and DK throughout the evolving design process. As such the MVEA approach, and its implementation as Denote, provides a means on which to support the storage and maintenance of knowledge for potential future re-use.

The Multi-Viewpoint MD Methodology, defined within this work, provides the declarative and procedural knowledge to support the utilisation and enhancement of the knowledge, constructed through the MVEA approach, to develop a Viewpoint Model and Cross-Viewpoint for modular analysis.

During the industrial case studies it was noted that the designers had difficulty in understanding some of the *concepts* defined in the viewpoint models. In most situations an explanation by the author clarified their meaning and the designers carried on with the dependency analysis. In a limited number of cases the designers withdrew the *concept* from the model, generally because it was embodied by another *concept.* The author believes that a significant portion of this misunderstanding can be attributed to her lack of experiential knowledge in the area, in that she was responsible for the application of the knowledge formalism to generate the Viewpoint Models (as shown in Figure 8.6). As such, the integration of DENOTE and the MD methodology implementation (see Chapter 7) is suggested as means to address such problems, in that the design team would thus generate their own CWK and can consequently utilise their own experiential knowledge and use terminology with which they are more familiar. This is covered in greater detail in future work (section 9.4)

Support Knowledge Modularisation (KM)

The methodology supports KM based on its ability to group *concepts* based on knowledge of their *dependencies* from different *viewpoints* and *perspectives* of these. The concept of KM allows the designers to modularise not only the physical component structure of a design (i.e. structure viewpoint) but also more abstract *concepts*. The KMn principle allows knowledge of the product to be utilised to support many objectives of design. The evaluation results suggest that this concept of KM, embodied within the multi-viewpoint MD methodology presented in this work, had strong potential to support such areas as team design, manufacturing design, technology life-cycle management. For example, the ITM design team had had problems in the initial design project in identifying the boundaries for each distributed team design work and that this process required a number of manpower intensive reviews. However, their evaluation stated that they believed as the methodology supports modularisation of more abstract *concepts* (than at the component level) this would provide a means to address this inadequacy of current MD support and allow them to define module boundaries at an early stage in the design without the need for so many man-power intensive reviews (see Appendix A).

9.1.2 Methodology implementations

The Methodology was implemented in two industrial projects (ITM and BGTI). Both cases resulted in the identification of modules within differing viewpoints of the product structure (function, working principle and structure). Based on specific operational requirements both designs were originally developed to embody modular design principles. In both cases the 'structural viewpoint' was the only viewpoint where the modularity of the design had been formally explored, defined and represented. The application of the Methodology resulted in an improvement of the modularity (of between 20 to 30%) of the 'structure viewpoint' of the product structure. However, based on the application of the Methodology to differing viewpoints of the design, the implementations illustrated the potential for modularity to aid the designers in:

- Structuring design knowledge, in that the Methodology supports the clustering of knowledge fragments, based on the strengths of the relations between these, and explicitly represents this 'knowledge' in a form suitable to support further design activities such as a modular analysis of the product structure.
- Early identification of module boundaries to support design task allocation, supporting the concurrent engineering initiative, i.e. different modules can be designed in parallel.
- Identification of hierarchical modularity i.e. the designers can manage the relations between each 'module' design team to support the integration of the design as a whole.
- Achieving various life-cycle objectives. For example, the evaluators indicated that they could envisage utilising the Multi-Viewpoint aspect of the Methodology to realise differing life-cycle requirements such as team design, supply chain management, technology life-cycle management and reduction of assembly and/or manufacturing complexity.
- Managing design knowledge across viewpoints of the design. The methodology provides a means to explicitly represent the evolution of design concept across viewpoints of the design. As such, the designer has knowledge of say, how *function concepts* are realised by structural components i.e. the *working principle concepts* realised by that structural component which facilitate the realisation of one or more *function concepts*. Thus, in a re-use scenario, the designer can ascertain the effect of the non-availability of say, a structural component on the functional realisation of the design. The designer can thus, look for alternative means of realising the *working*

principle previously embodied by the component to maintain the functional realisation of a design.

In general the Methodology implementation was deemed to be a successful, with the evaluators stating;

'the results showed significant improvements'

and, as stated by the ITM design implementation evaluators, its development and application represented;

'a valued contribution to our work in this field'

9.2 Embodied formalisms and mechanisms and their application

The following reviews the formalisms and mechanisms, embodied within the Methodology, and briefly discusses their strengths and weaknesses with respect to their application.

9.2.1 Modelling Formalism

Multi-Viewpoint Evolutionary Approach knowledge formalism

It was hypothesised that the utilisation of an evolutionary knowledge formalism could support the re-use of design knowledge. *Design for Re-use* was highlighted as the least supported EDR process. *Design for Re-use* was defined as the process of identification, extraction and recording of potentially re-usable design knowledge and the further enhancement of their knowledge content. As an evolutionary knowledge formalism the MVEA approach was embodied within the Multi-Viewpoint Modular Design Methodology. The parts of the MVEA formalism that were applicable to support modular design were defined and utilised as the basis from which to identify *Knowledge Modularity* within the product structure. As such, based on the utilisation of the MVEA formalism, potentially re-usable sources of design knowledge are *identified* and *extracted* during the design process. Further, based on the concept of *Knowledge Modularisation,* this source of knowledge is analysed to identify clusters of related *design concepts* whilst identifying the differing *configurations* and their hierarchical nature with respect to the overall *product structure.* As such, based on a modular analysis, the methodology explicitly represents knowledge of the product structure that is not immediately evident from the initial source of *current working knowledge* i.e. *enhances* the knowledge content of extracted fragments. In addition, as current modular design methodologies are not based on the formalism of evolutionary design knowledge they are generally limited to the analysis and enhancement of the component structure of a product which, as discussed in Chapter 2, results in an inadequate source of design knowledge to support EDR.

The industrial implementations resulted in the *identification*, *extraction* and explicit *formalisation* of *evolutionary design knowledge.* In addition, the Methodologies industrial implementation provided the designers with knowledge of the product structure that was not explicitly available from the original design documentation i.e. *enhancement* of the related knowledge content. As such, the industrial implementations verified that the utilisation of an evolutionary design knowledge formalism (MVEA) could improve EDR support by providing a basis on which to identify, extract and analyse design knowledge to enhance its knowledge content.

Matrix Formalism

The matrix formalism was deemed to be a generic and clear means to represent design knowledge *concepts* and the relations between these. During the computational implementation phase the author experimented with the use of graph-based formalisms, where the nodes represented the *concepts* and the links representing the interactions between these. However, as the complexity of the problems increased, the number of *concepts,* interactions and types of interactions increased, and the graph-based representation became more intricate and 'busy'. As a result it was difficult to gain as clear a visualisation of the problem as afforded by the matrix-based representation. As such, the matrix-based representation was implemented. However, it was criticised for being a 2-Dimensional representation of a multi-dimensional problem. It is suggested that a combination of a matrix formalism (to represent the *concepts, dependencies* and *modularity* at varying levels of a *viewpoint* of the product structure) and a graph formalism (to represent the *structural* relations between levels of the product structures' viewpoints) may be utilised to better support the multi-dimensional problem of *Knowledge Modularisation* (see Future Work below).

9.2.2 Optimisation Mechanism

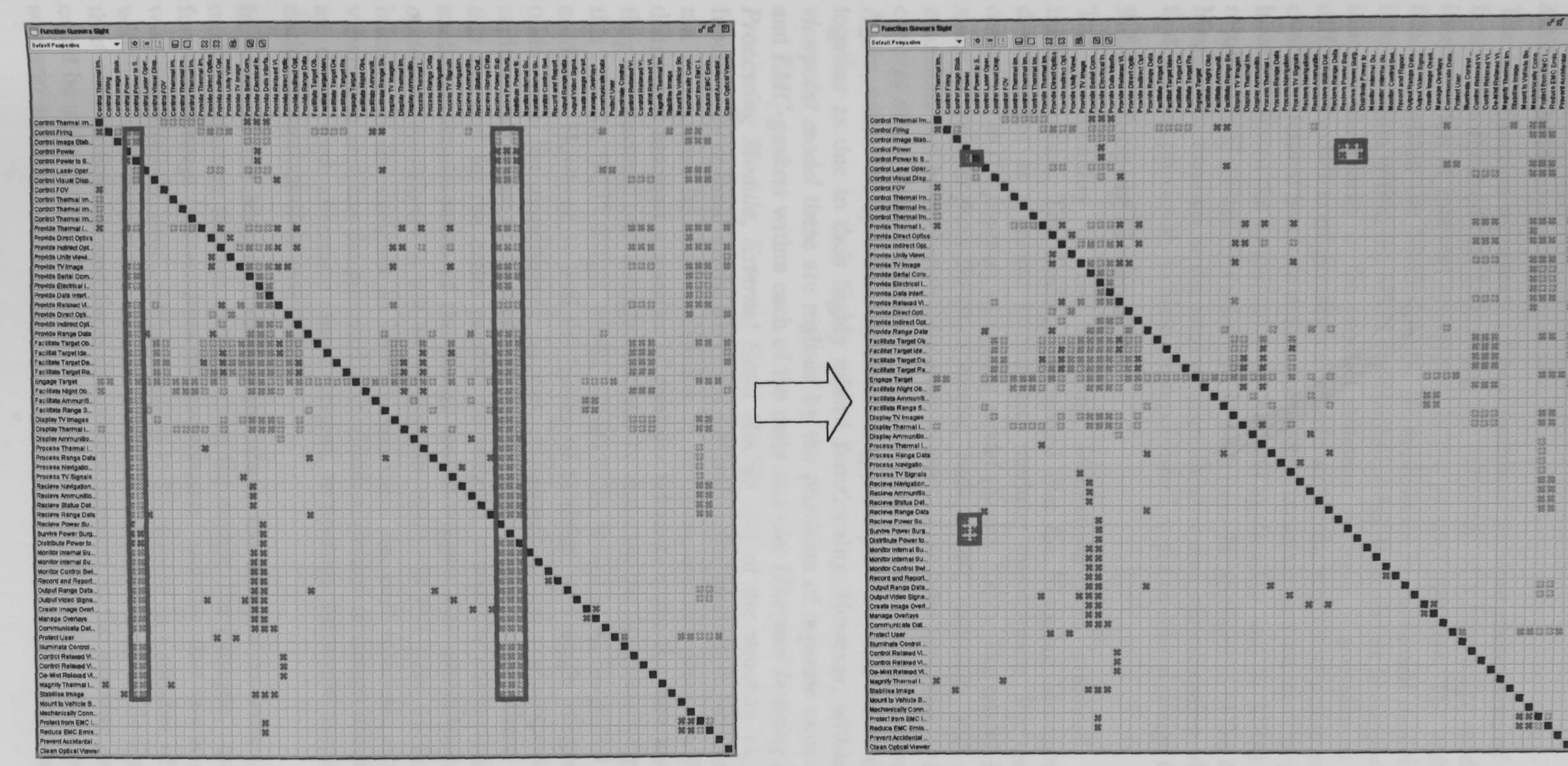
Clustering Criteria

The resulting finding of the four implementation studies would suggest that the clustering criteria represents an appropriate measure on which to determine the modularity of evolutionary design knowledge. One of the main issues with the clustering criteria was encountered during the application of the clustering criteria to the BGTI case. Figure 9.1 below depicts two examples of the BGTI *function viewpoint model.* Figure 9.1(a) illustrates the original BGTI *function viewpoint model,* as completed during the initial workshop, and Figure 9.1(b) depicts a modified *function viewpoint model.* The blue overlaid boxes in each represent the focus areas for alteration. The *concepts* within these blue boxes include; *receive power, control power, survive power surge/spike/ripple,* and *distribute power.* Due to the nature of the design, in that each *concept* of the design required the distribution of a 'clean' power source to realise its function, the designers deemed that almost all the *functions* had a form of dependency with these *concepts* (*receive power, control power, survive power surge/spike/ripple,* and *distribute power*). As such, we can see in Figure 9.1(a) that there are dependencies defined between these *concepts* and almost all other the entire length of the matrix body. The clustering criteria and module identification mechanism are based on the assumption that clusters of components around the leading line are potentially representative of modules. However, in this case, due to the number of dependencies of the *power related concepts* with other *concepts* in the matrix, the *concepts* all clustered around the *power concepts* and as such one large module of a low MSI was identified through the application of the optimisation mechanism. This would suggest that the clustering criterion is inappropriate for the exploration and identification of 'bus type' modularity, i.e. the form of modularity where a standard structure or interface (in this case the *power concept grouping*) can accept a number of different modules (in this *case control function groupings, imaging function groupings,* and *target engagement function groupings*). In the BGTI case the designers decided to remove the majority of the dependencies (those of value 0.5 or 0.1) working on the assumption that the relation was implied by the nature of the *function* itself. However, they chose to retain the dependencies of value 1.0 as these depicted the relation of *concepts* required to provide the 'clean' power source and this was deemed to be a key *function* which the designers wished to explicitly represent in any knowledge structure of that design.

The computational realisation of the Methodology has the capabilities to perform multiple-criteria optimisation. As such, it is suggested that the formalisation of other different types of 'modularity' could generate a number of 'modularisation-centred' criteria to facilitate multi-criteria modular design optimisation.

GA Application

It was hypothesised that the application of a Genetic Algorithm could support the exploration and identification of *knowledge modularity* in the product structure. The evaluations have shown that the GA application is an effective method for optimising the modularity of the product structure, resulting in a 26% and 19% decrease in the related clustering criteria for the optimisation of the ITM and BGTI cases respectively.



a) Gunners Sight

b) Gunners Sight with dependencies removed

Figure 9.1: The original and modified BGTI *function viewpoint model*

9.2.3 Module Identification Mechanism

MSI and MSM

The MSI function represents a formalisation of the core phenomena of a *knowledge module*. It was hypothesised that formalising the definition of a *knowledge module* could support the identification of inherent modularity in the product structure. The embodiment of the MSI function within the Methodology, its application to viewpoint models and its subsequent interpretation within the MSM representation has been shown to support the identification of inherent modularity in the product structure. As such, it provides an interpretation of the differing module configurations available within the product structure and how these develop over hierarchical levels to define the product structure as a whole. The industrial evaluations have illustrated the functionality of the MSI application, and the MSM representation of the resulting MSI values. For example, in each case the application of the Module Identification Mechanism resulted in the explicit identification and representation of inherent modularity that was not previously evident from published works or related design documentation.

9.2.4 Mapping Mechanism

The functionality of the Mapping Mechanism was evaluated through the BGTI implementation. The findings illustrated that the Mapping Mechanism required further development as a means to facilitate modular maintenance. For example, consider the *optimised* BGTI *function viewpoint model* (Smith 2002). The model has concepts such as such as *protect-from-the-environment*, *reduce-EMC-emissions* and protect from *EMC-emissions* that have clustered together to create a modular grouping. In the *working principle concept model* these are *realised* by such *concepts* as *environmental-seal* and *EMC-gasket.* Again, within the *optimised BGTI working principle viewpoint model* these concepts cluster together as due to their highly related functionality. However, within the *structure BGTI viewpoint model* these are realised by the provision of separate *environmental-membranes* and *EMC-gaskets* within each of the main *modules* (*Thermal Imaging Module, Control and Processing Module*, *External Hood Module etc*) of the structure viewpoint. As such the functionally optimal modularity (that defined in the function viewpoint) of the *concepts* are not maintained by their realisation in the *structure viewpoint*, in that it they are now distributed amongst a number of modules whereas in the function and viewpoint modules they are clustered into a single module. To maintain the functionally optimum modularity in this case the BGTI design would require to be sealed as an integrated unit (not as separate modules). However, to seal the 'group of modules' as a single integrated unit would be at odds with the requirement to ease maintenance and replacement. For example, the replacement of one of the modules would thus impact on all the others, as this would require that the seal for the entire product be violated, potentially contaminating all modules. As such, the requirements of the design would suggest that maintenance of the *functionally optimised modularity* is not an appropriate option in this case. However, it is still in the interests of the designer to identify and track the near optimum modularity through the viewpoints as it has the potential to facilitate a number of design life-phase objectives, such as, design task allocation (team design), technology life-cycle management and process design (manufacturing/ assembly).

In addition, the Mapping Mechanism allows the designer to track the realisation of the modular structure through viewpoints of the design. For example, consider the case of the function concept '*protect-from-the-environment*' and say, that this is realised by some the working principle '*seal*', which requires to be realised in a number of distinct applications within the *structure viewpoint*. To reduce the complexity of the design the designer would thus seek to develop one form of *solution concept* to the *working principle 'seal'* which could be utilised to realise all of its applications in the *structure viewpoint*. Thus, in a re-use scenario a change or alteration to say, the *function concepts,* could be tracked forward to identify its effect on the structural realisation of the design. For example, a change in the *'protect-from-the-environment'* (i.e. the *function concept* is further decomposed into the *concepts 'prevent-from-liquid-absorption'* and *'protect-from-vibration'*) would impact on the working principle concept. In this example the *working principle concept 'seal'* would no longer be appropriate. The *working principle viewpoint* would require further development and the definition of more concrete and appropriate *working principle concepts*. As such, the designer could track the realisation of the *working principle concept 'seal'* to the structure viewpoint. At this stage the designer would have knowledge that the change in the function concept *'protect-from-the-environment'* has an impact on a number of modules in the structure viewpoint and, as such, the requirements of each module would require to be considered when attempting to define more appropriate *working principle concepts*.

The resulting findings of the evaluation of the Mapping Mechanism indicate that it has some potential as a means to maintain the modularity of the design. However, its strength appears to lie in its ability to track concepts through the design and as a mechanism to audit the design trail, as suggested by the BGTI designer. As such, the Mapping Mechanism requires further research and development to realise its potential fully.

9.3 Research approach

The research was undertaken based on the methodology defined in Figure 1.3. The research methodology was developed specifically to address the requirements of design research where industrial application is a key factor. As such, a key element of the outcome of the research was the development of a Methodology that represented an articulate procedure for designers to follow in practice. This requirement raised a number of issues with respect to satisfying the academic research requirements whilst addressing the needs of industry, including;

- Differences in conceptualising the problem

  Based on the experiences obtained during the 12-month industrial residency, the author noted a common attribute of design practitioners in industry was that when an specific issue was identified, such as the need to support knowledge re-use, their was a tendency to address the issue in terms of a solution, i.e. we need a 'expert system to... ' or 'we need a library of standard parts'. This may be attributable to internal and external demands in terms of market demands, costs, and time. Indeed, this is especially pertinent in terms of the military engineering design domain which is often characterised by periods of 'feast or famine' in terms of contracts. However, the researchers objective is to conceptualise the problem i.e. identify the root of the issue that has arisen. This initial disparity in terms of conceptualising the problem leads to a disparity in terms of developing a solution, for example the design practitioner is keen to implement the current 'state of the art' whilst the researcher requires to define some 'novel' approach to advance the 'state of the art' and contribute knowledge to the area. In the case of the ITM design, this disparity was addressed on a number of levels.

  In the first instance, the researcher utilised literature-based sources to aid in the identification of the 'research problem' and the requirements of addressing the problem. Thus, having identified the principles of Modular Design as a potential basis for developing an appropriate solution the previous experiences of a design team, dedicated to the implementation of these principles, was utilised to 'ground' the solution development within industry design practices. As a result, the research was conceptualised in terms of the requirements of the research problem (see requirements Chapter 3), the inadequacies of the existing 'state of the art' approaches (see Chapter 4) and the requirements for an industrially applicable solution (see ITM designers requirements – (Smith, Robb et al. 2001)). As such, the solution was developed and evaluated based on the requirements of research in terms of the 'novelty' and 'knowledge contributions' whilst satisfying the requirements of industry in terms of 'solution applicability'.

- Differences in the terminology utilised

  A significant issue in terms of conducting research in industry is the discrepancy in terms of the terminology utilised to define specific practices and/or phenomena. As such, the author noted two main issues in relation to this whilst conducting the research.

  The first arose whilst defining the viewpoint model *concepts*. As indicated in Chapter 8, Figure 8.6, the researcher defined the viewpoint models and the design team members defined the dependency knowledge between these. There were a number of occasions where the author was asked to further define a *concept*. After the explanation the designers often changed the *concept* wording to better reflect their understanding of the *concept* and occasionally removed the *concept* based on the fact that it was embodied by another *concept* in the model. However, the author believes that the integration of the computational realisation of the MVEA formalism, DENOTE, would address this issues, in that the designers would build and maintain the *current working knowledge* of the design themselves and as such define the *concepts* based on the terminology specific to their industrial application.

  The second 'terminology' related issue arose when documenting and/or presenting elements of the research problem, solution development, solution implementation and resulting findings. As such, the author became aware of the requirement to 'tailor' work based on the interests of the audience, for example, a documentation for presentation to the research community would focus on the works contribution to the field and the resulting finding in terms of knowledge of that field whereas the industry version would concentrate on the practical application and the results in terms of time, cost, performance and quality of the product or related design process.

Such experiences have developed the authors understanding of the relationship between research and industry. The author believes that the rationalisation of the ITM designers modular design experiences bridged the 'conceptualisation' and allowed both parties to comprehend the solution development, and its application, in terms of their own specific objectives.

Despite the issues, which arise when conducting industry-centred research, there are a number of potential advantages, including (but not limited to):

- Based on the assumption that the research is carried out with an organisation that is actively involved in the domain of interest there is an availability of study and evaluation material. For example, BAE Systems Ltd and Thales Optronics limited are both actively involved in the engineering design domain and in both cases there were a number of potential application areas identified for the initial implementation. In addition, based on the resulting findings, illustrated in Chapter 8, a number of additional application have been identified as areas for potential further development.
- Gaining a different perspective of a problem
- The formalisation and development of state of the art approaches whilst the researcher learns more about research and development from an industrial perspective. For example, industries are often constrained by external and internal

organisation structures; legislation and working practices that have a resulting impact on the research and development activities and outcomes. For example, the military domain is subject to detailed 'ministry of defence' legislation that explicitly details many design aspects from the dimension and colour of control switches to the acceptable sub-contractors. As such, the research is developed, from the earliest conception of the problem to the detail of the solution, to account for the complexities of an industry. As such, the resulting research simultaneously addresses the research and industry requirements and which the author suggests results in a more applicable resource.

- There is the potential for the application of the research to result in tangible time or cost reductions, and/or quality or performance improvements. This can provide added incentive to the researcher and be an effective motivator.

The Methodology was successfully implemented in two industrial studies. The evaluators were unanimous in their belief that the Methodology represented an articulate procedure to follow in practice. Despite, the recognition of a requirement for more advanced training to ensure successful application and interpretation of the resulting findings of the Methodology, BAE Systems stated that they;

'would look to utilise this approach in the future.'

9.4 Future Work

The discussions on the Multi-Viewpoint Modular Design Methodology and the general feedback gained from its application to two industry-based case studies highlight a number of issues that require future development. These future developments involve theoretical improvements to the methodology as well as its further realisation in a computational environment (Section 9.4.1 and 9.4.2) and future industrial application (Section 9.4.3).

9.4.1 Theoretical Improvement to the Multi-Viewpoint MD Methodology

The following covers the envisaged theoretical improvements that could be made to the methodology.

Development of the Approach to 3-Dimensional Problem

The limitation of the matrix representation, in that it restricted to representing only 2-dimensions, has been alluded to on a number of occasions throughout the work presented in this thesis. The issue was also raised during the evaluation phase. The modular design problem can be considered to be at minimum a 3-dimensional problem in that a product (or system) may be made up of a series of modules but may itself represent a module in a relatively larger product (or system), for example, the ITM design represents a 'module' within the larger ship system but was itself the configuration of a number of smaller modules. The MSI and MSM application go somewhat towards addressing this problem in that they identify hierarchical modularity within the given model. In addition they, causal links between the modules add an additional dimension to the Viewpoint Matrices. Figure 9.2 and Figure 9.3 provide illustrate the author's current propositions for addressing such inadequacies in the current methodology.

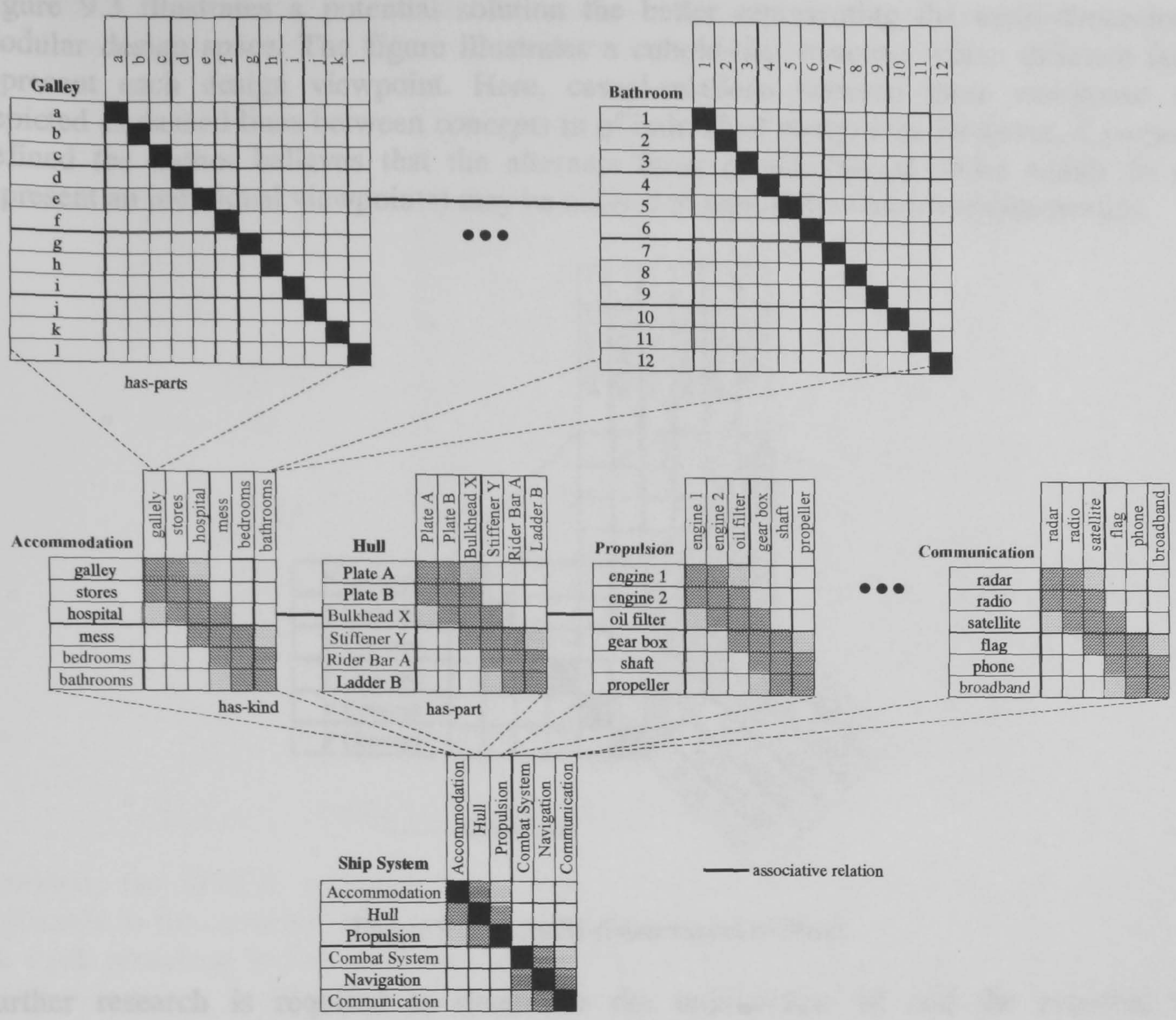


Figure 9.2: Addressing hierarchical MD

Figure 9.2 illustrates (from the structural viewpoint) a top-down hierarchy of a ship system[43]. As can be seen the top-level models represent solution concepts in a specific ship sub-assembly, i.e. the galley and bathroom. A general associative relation is defined between the galley and bathroom models (defined by the black line adjoining the models) that depicts that some form of dependency exists between these two models at a lower level in the hierarchy. Thus, these models group into the lower level model (accommodation), in which they become individual *concepts*. At this level the nature of the associative relation between these concepts is more clearly defined to support its modularisation. The dependency knowledge for this level of the hierarchy is defined. This may result in the need to either redefine the existing or addition of more general associative relations at a higher level in the hierarchy. Again associative relations between this model (accommodation) and others at this level in the hierarchy can be defined, for example, with the hull. The models continue to group into higher-level models (where they are treated as individual concepts) until the entire system is defined, i.e. in this case the ship system. The structural relations defined as part of the *MVEA* approach can be utilised to illustrate the type of relation that exists between models in successive levels of the hierarchy. For example, it can be seen that the concept 'accommodation' has a *has-kind* relation with the *concepts* galley, stores, hospital, ..., bathroom, and the galley in turn has a *has-parts* relation with *concepts* galley a, galley b, galley c, ..., galley l.

43 The parts, assemblies and identified module have been defined for illustrative purposes only and do not themselves represent a comprehensive study of a ship system.

Figure 9.3 illustrates a potential solution the better representing the multi-dimensional modular design space. The figure illustrates a cuboid-like structure where different faces represent each design viewpoint. Here, casual-relations between these *viewpoints* are depicted as dashed lines between *concepts* in of individual *viewpoints*. However, if properly defined the author believes that the alternate faces of the cuboid (those which do not represent an individual viewpoints) may be utilised to represent cross-viewpoint models.

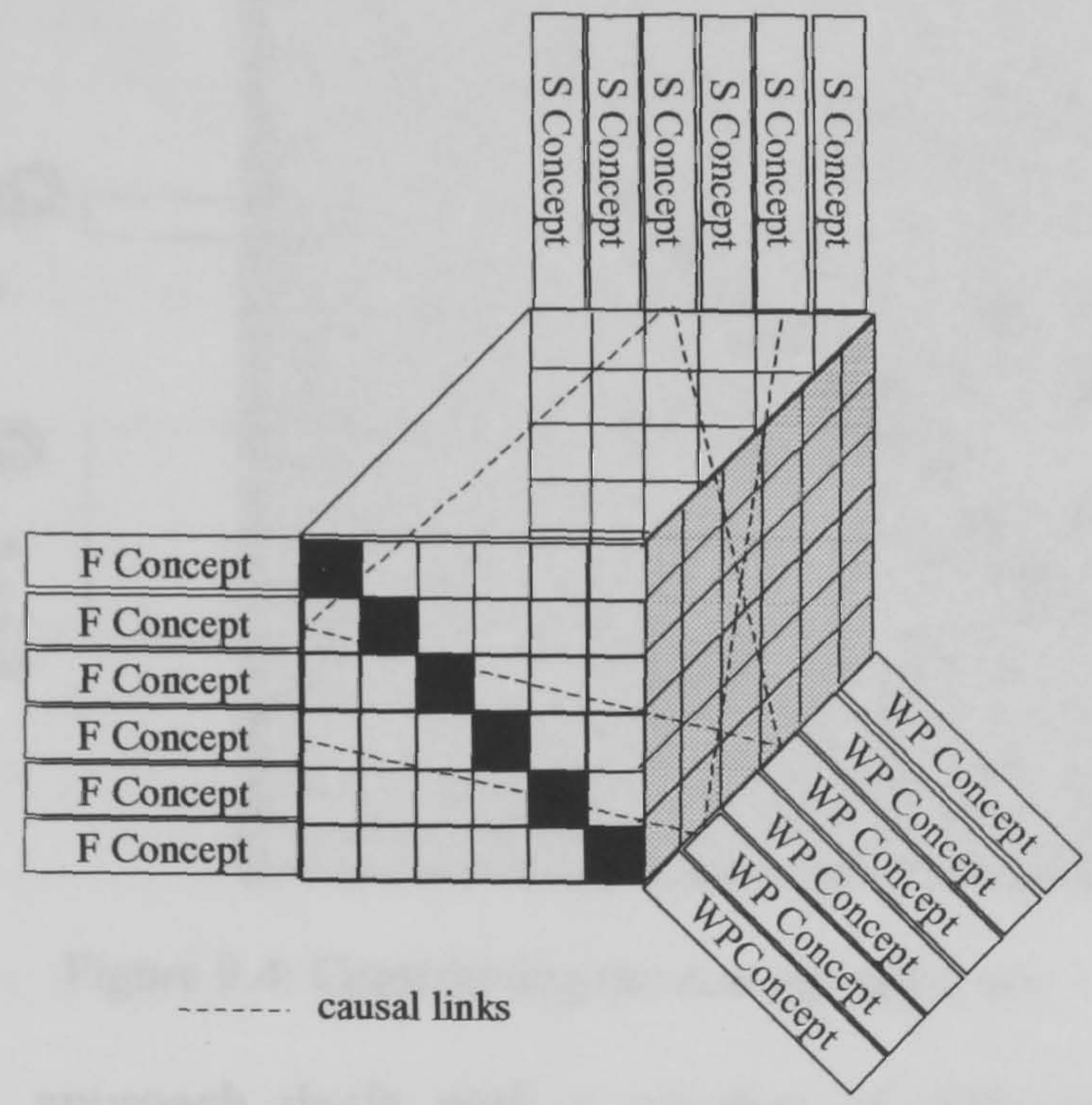


Figure 9.3: A multi-dimensional solution

Further research is required to determine the implications of and the potential for development of these solutions.

Clustering Criteria development

The clustering criterion was shown to be effective in all four cases in identifying concept clusters that can later be defined as modules. However, its limitations were noted within the BGTI case where elements of 'bus-type' were noted. This resulted in the dependencies being less symmetrical and in the clustering of almost all *concepts* around one main *concept* (in this case the power supply). Thus, the *clustering criterion* requires further development to ensure that it can cluster *concepts* appropriately. It is the author's belief that adequate support for the modular clustering of *concepts* will require the definition of a number of differing *clustering criteria,* based on different modularity types, and subsequent multi-criteria optimisation. However, although the current methodology is based on a single criterion, the system implementation was previously developed to support multi-criteria optimisation.

Introduction of constraints and constraint management

As discussed previously (Chapter 5 and 6) the current methodology does not consider the constraints explicitly. The inclusion of constraints, their management and their impact on the modular design is an issue that has been highlighted as requiring further investigation. In the first instance, the author proposes to implement a form of constraint application on the concept sequence as illustrated in Figure 9.4. The figure depicts the *Function Viewpoint Model FDP* $_{energy}$ with the 'link' (denoted by the line between FVC1 and FVC2) and the 'do not link' constraints (denoted by the dashed line between FVC6 and FVC11) applied to specific *concepts.* It is proposed that the user can choose a degree of separation (denoted by the number on the dashed line), which is equivalent to the minimum number of *concepts* that must be maintained between these regardless of the sequence.

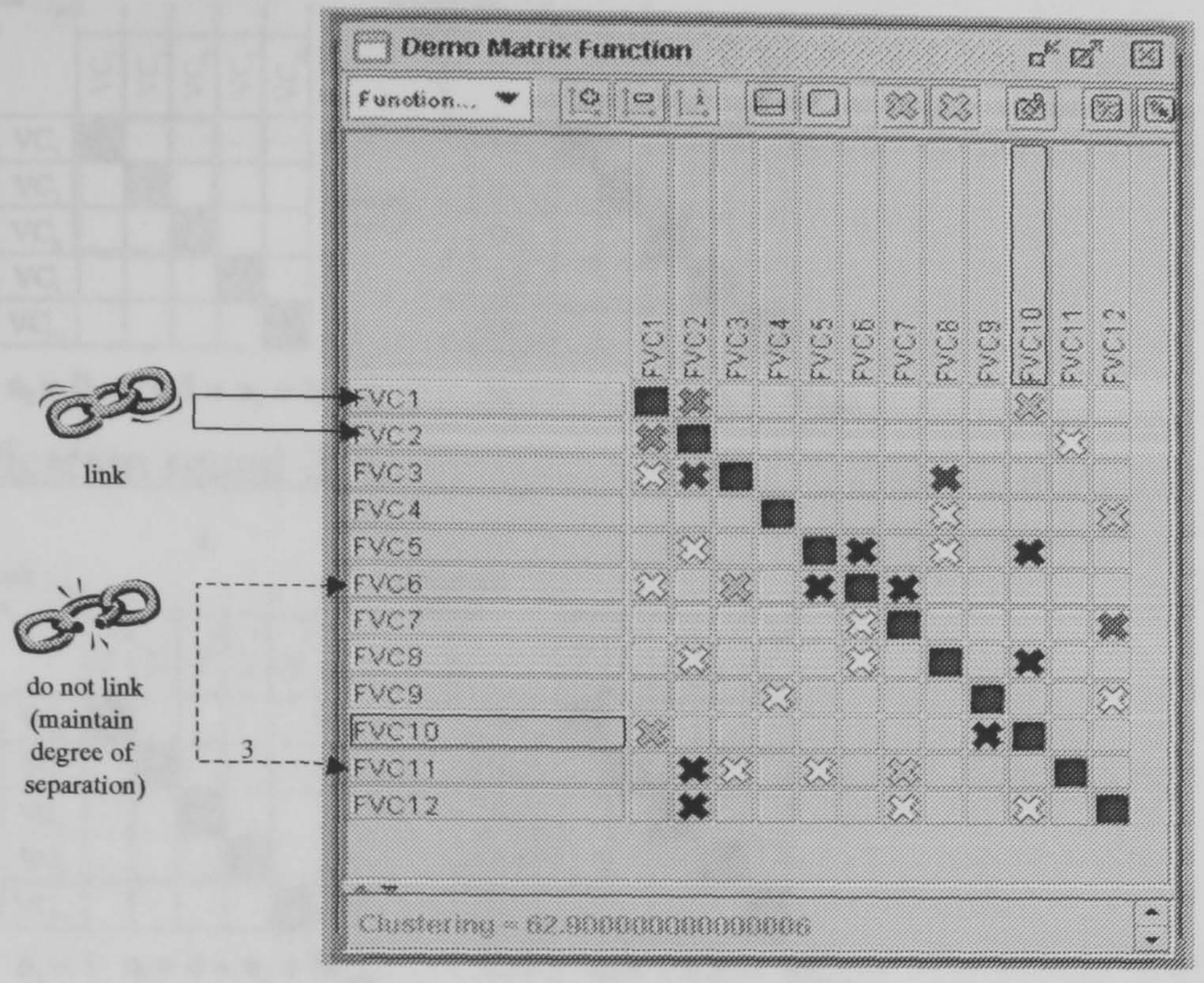


Figure 9.4: Constraining the *concept* sequence

However, the MVEA approach deals with a number of different types of constraints applicable to the *concepts*, their *attributes* and *relations* which although beyond the scope of the work presented in this thesis the author believes represents an area for future research and development. In addition, future developments include additional functionality so that the user can place upper and lower limits on the number of *concepts* that may be grouped into a module. This constraint would affect the application of the MSI function to the *Optimised Viewpoint Model.* Figure 9.5 depicts the MSI application with module grouping constraints on the minimum (M_{min}) and maximum (M_{max}) number of *concepts* allowable in a module being 3 and 4 respectively.

The MSI application differs to that depicted in Figure 6.22 which has no constraints on the minimum or maximum *concepts* allowable in a module. We can see here that the MSI application begins, as in the previous example (Figure 6.22), with $n_1 = 0$. However, based on the constraint M_{min}, the first potential module grouping is by definition $\{n_1 = 0, n_2 = n_1 + M_{min}\}$. The last potential module grouping in any round, based on the imposed constraint M_{max}, can be defined by as for (n_1, $n_2 = n_1 + M_{max}$ or until $n2 = N$, where $n_1 + M_{max} > N$). The process repeats in application rounds with n_1 rising by incremental values of 1 until $n_1 = N - M_{min}$ which represents the last potential module grouping in the model based on the constraint M_{min}, i.e. group $\{n_1 = 2, n_2 = 5\}$ in Figure 9.5.

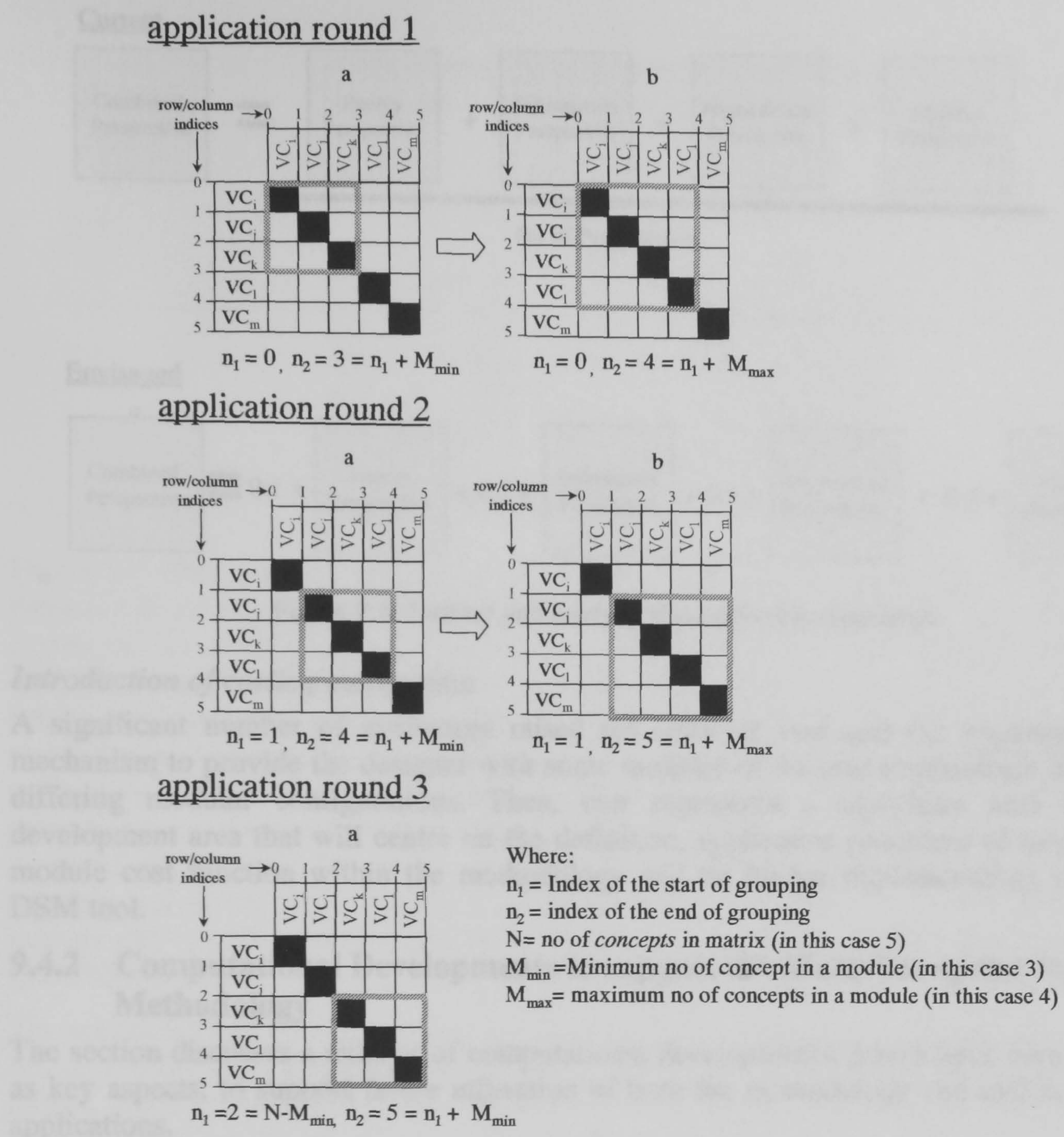


Figure 9.5: Constraining the module grouping and the MSI application

Application of weighted perspectives

The evaluation process raised the requirement for being able to 'weight' perspectives based on their significance within the design activity. Figure 9.6 schematically illustrates the current and envisaged process for creating a combined perspective. As depicted in Figure 9.6 a combined perspective is currently representative of the most concrete and detailed viewpoint *concepts* and the average weight of dependencies between these. However, one perspective may be of particular importance to fulfilling the requirements of the design activity and the designer may wish to bias the optimisation process with respect to this perspective. Thus, as depicted in Figure 9.6, it is envisaged that a combined perspective will be representative of the of the most concrete and detailed viewpoint *concepts* and the dependencies between these based on the weighting prescribed for each individual perspectives. With a maximum dependency value of 1.0 the perspective weightings would require to be normalised such that the combined dependencies are always ≤ 1.0, for example, the collective weighting of the four perspectives (energy, information, physical-link and material) in the envisaged combined perspective (Figure 9.6) is 1.0.

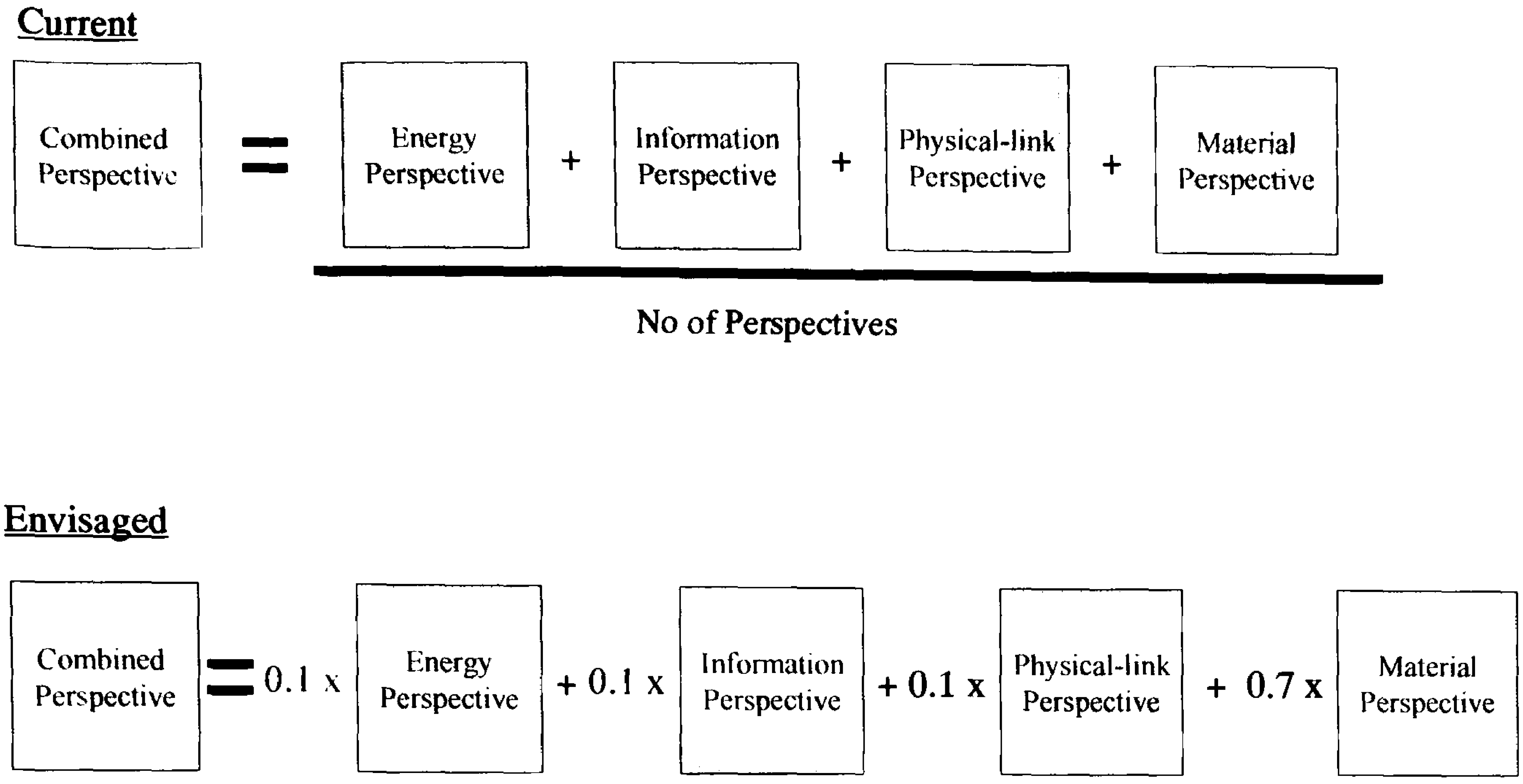


Figure 9.6: Current and envisaged combined perspectives.

Introduction of costing mechanism

A significant number of evaluators raised the issue of cost and the requirement for a mechanism to provide the designer with some measure of the cost implications of choosing differing modular configurations. Thus, cost represents a significant area for future development area that will centre on the definition, application procedure of some form of module cost function within the methodology and its further implementation in the MD DSM tool.

9.4.2 Computational Developments to support the Multi-Viewpoint MD Methodology

The section discusses a number of computational developments, which have been identified as key aspects, to support future utilisation of both the methodology and tool in industrial applications.

Integration of DENOTE

The DENOTE system embodies the MVEA knowledge formalism. The MVEA formalism for CWK has been adopted as the basis for the *Viewpoint* knowledge formalism as utilised within this work. However, DENOTE also embodies a formalism for Domain Knowledge (DK), a maintenance mechanism and utilisation schema. Thus, the integration of DENOTE, and the DSM modelling and analysis tool, presented within this work, would improve EDR support, in that this would improve the structuring, use and maintenance of design knowledge when applied over generations of design the domain of interest.

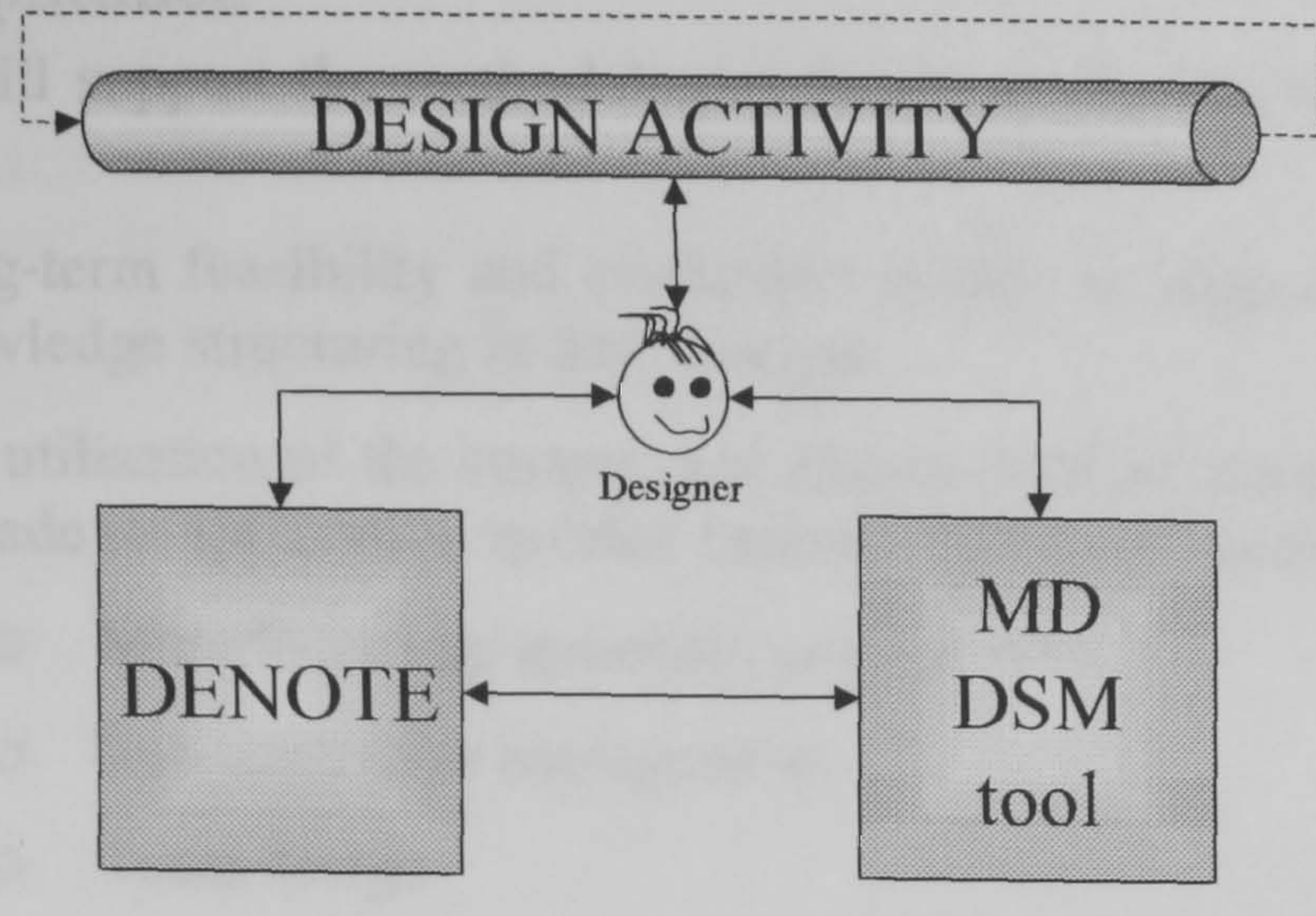


Figure 9.7: Denote and MD DSM tool integration

Figure 9.7 illustrates the integration of DENOTE and the MD DSM tool. As can be seen there is a bi-directional flow between the designer and the design activity. Here, the designer carries out the activity of design but is supported by DENOTE and the MD DSM tool. The designer may utilise the DK in DENOTE as a start point for carrying out the activity of design and in turn will generate CWK as activity of design evolves. DENOTE can be utilised to both model and maintain this knowledge. In addition, the CWK in DENOTE can be utilised by the MD DSM tool to generate Viewpoint Models. Here, the DSM system may extract the current most detailed and concrete *concepts* to generate the viewpoint model, however, the designer may prefer to support the definition of dependency knowledge (both between and across viewpoints) through the definition of entries in the matrix body of these models and in such instances, DENOTE would require to extract this knowledge to evolve the CWK model.

The results from the DSM tool can be utilised by the designer to support the evolution of the design activity and consequently the CWK of that activity. The knowledge generated from a design activity and through the application of the MD DSM tool (and its underlying methodology) can be utilised to generate DK in the form of past cases, general working principles, general function structure and knowledge modules. Thus, the integration of DENOTE and the MD DSM tool may be utilised to support: the evolution of the design activity knowledge, the maintenance of the modular solution and the utilisation of this knowledge in both a current design activity and for future design.

Dependency input – perspectives

As discussed previously the computational implementation of the perspectives results in time-consuming repetition. The author recommends that the current computational implementation be altered to reflect the form used in the paper-based models of the ITM case study (Chapter 8, Figure 8.9). A significant complication to this alteration is that the *concept* and *dependency* knowledge must remain *viewpoint* and *perspective* specific, respectively.

Checking mechanism

The evaluation raised the requirement for some form of mechanism to support the checking of *dependency* data. This may be achievable though the integration of DENOTE with the MD system, in that dependencies may be checked based on previously defined DK. In addition in may be possible where the dependencies modelled are symmetrical (as in the physical link perspective) to check for consistency either side of the leading line. However, as the approach is predominantly based on the design knowledge input from the designer, the major burden of this checking process ultimately lies with the designers themselves. As with all knowledge-based approaches, if the user input is garbage the resulting output is garbage.

9.4.3 Future Application

Such future work will support the methodologies future application to a number of areas, including:

- Long-term feasibility and evaluation studies to support modularisation and knowledge structuring in new designs.
- The utilisation of the system, and elements of its theoretical background, to cascade its application to other business functions, including:
  - Manufacturing/ assembly process flow
  - Sub-contractor management
  - Team design

9.5 Chapter summary

This chapter has discussed the strengths and weaknesses of the Multi-Viewpoint MD Methodology from the aspects of fulfilling the requirements of a MD methodology for improved EDR, addressing the inadequacies of current approaches, and fulfilling the requirements outlined by practising designers. Further, company feedback based on two industry-based case studies has been summarised and finally, these findings have been drawn together to highlight future work required to enhance the methodologies capabilities.

10 Conclusion

Based on the research methodology introduced in Section 1.3 (Figure 1.3), the work presented in this thesis has established a Multi-Viewpoint Modular Design Methodology from the perspective of improving support for *Engineering Design Re-use (EDR)*. Figure 10.1 provides a summary of the work presented in this thesis by highlighting the research contributions, the resources utilised, the design practice applications and the dependencies between these.

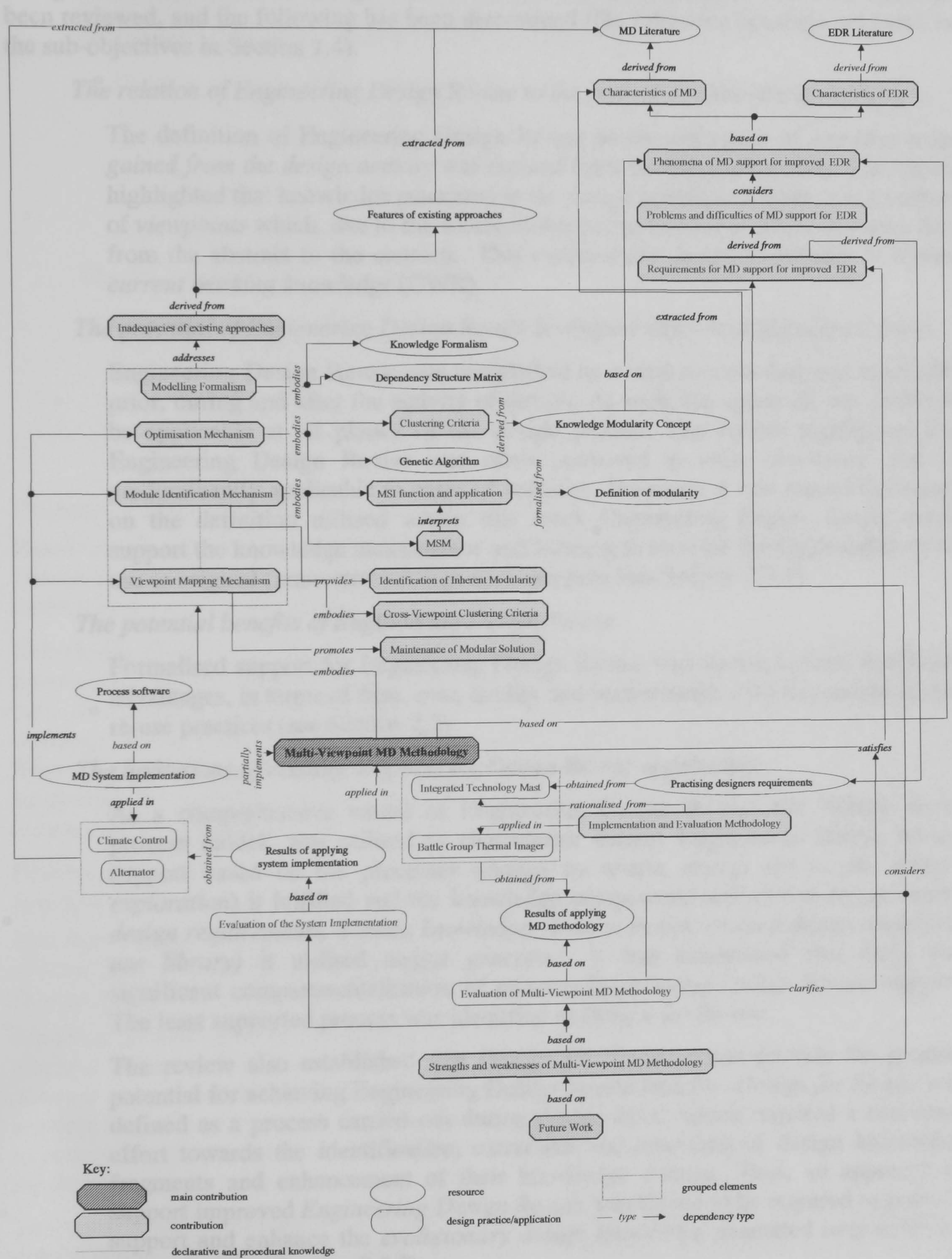


Figure 10.1: Summary of Work

The overall aim of the research was to establish a modular design methodology to support multi-viewpoint modularisation of designs within the engineering design environment. Based on the resources, research contributions and design practice/design applications, outlined in Figure 10.1, the following discusses the work undertaken to meet this aim and the resulting outcomes.

Engineering Design Re-use (EDR) and Modular Design (MD)

An objective of this main aim was to disclose the characteristics of the fields of Engineering Design Re-use and Modular Design. To achieve this objective, literature from both fields has been reviewed, and the following has been determined (the following headings are based on the sub-objectives in Section 1.4).

The relation of Engineering Design Re-use to the process and knowledge of design.

The definition of Engineering Design Re-use as *the utilisation of any knowledge gained from the design activity* was derived from the literature review. The review highlighted that knowledge generated in the design activity can exist over a number of *viewpoints* which, due to the nature of the design process itself, *evolve* over time from the abstract to the concrete. This evolutionary design knowledge is termed *current working knowledge* (CWK).

The potential of Engineering Design Re-use to support different design types/phases

Engineering Design Re-use was established as a total process that was applicable prior, during and after the activity of design. As such, the approach was shown to be applicable at all phases of the design process. The review highlighted that Engineering Design Re-use was often perceived to stifle innovation and be predominantly applicable to cases of redesign. However, it was argued that based on the definition utilised within this work Engineering Design Re-use could support the knowledge maintenance and learning to increase the applicability of re-use to original (innovative) design environments (see Section 2.3.3).

The potential benefits of Engineering Design Re-use

Formalised support for Engineering Design Re-use was shown to have significant advantages, in terms of time, cost, quality and performance, over the current ad-hoc re-use practices (see Section 2.2).

The limitations of existing Engineering Design Re-use approaches

As a comprehensive model of Engineering Design Re-use, the 'design re-use process model' was utilised to characterise current Engineering Design Re-use support based on the processes (*design by re-use, design for re-use, domain exploration*) it fulfilled and the knowledge components *(completed design model, design requirements, domain knowledge, domain model, evolved design model, re-use library)* it utilised and/or generated. It was established that there was significant compartmentalisation of existing Engineering Design Re-use support. The least supported process was identified as *Design for Re-use.*

The review also established that *Design for Re-use* could provide the greatest potential for achieving Engineering Design Re-use benefits. *Design for Re-use* was defined as a process carried out during design itself, which required a conscious effort towards the identification, extraction and recording of design knowledge fragments and enhancement of their knowledge content. Thus, an approach to support improved *Engineering Design Re-use* was shown to be required to actively support and enhance the *evolutionary design knowledge* generated over differing *viewpoints* (see Section 2.3.3).

As they provide a basis on which to support the structuring and management of design knowledge product structuring principles were reviewed with the objective of disclosing the characteristics of Modular Design (MD).

The relation of Modular Design to Engineering Design Re-use

Based on the characteristics of modular design, correlations were established between the requirements of Engineering Design Re-use and the benefits of modular design. As such, the principle of modular design was embodied as the basis on which to support improved Engineering Design Re-use.

The relation of Modular Design to the engineering design process

Modular Design (MD) was shown to be a product structuring principle that supports the structuring and management of *components* into distinct detachable modules. The modular design principle is synonymous with the creation of variety and the re-use of defined modules over generations of a product family. Module definition was shown to be dependent on the explication and exploration of knowledge of the relations between the artefact *components.*

The limitations of current Modular Design

However, the current principles of modularity focus on the *components* of a design and *Engineering Design Re-use* requires that more abstract design *concepts* from differing *viewpoints,* and their *evolution* through the design process be supported. Thus, *knowledge* related to the design *viewpoint* and its associated *dependencies* were shown to be key aspects to defining a modular design methodology to support *Design for Re-use.*

Phenomena of Modular Design support for Engineering Design Re-use

Based on the disclosed characteristics of the Engineering Design Re-use and Modular Design fields the phenomena of modular design support for engineering design re-use was derived as *Knowledge Modularity* (KM).

Knowledge Modularity defines the modularity of a design based on the *evolutionary design knowledge* generated during the design process. Thus, modularity is defined as an attribute of both the product's *components* and the associated *evolutionary design knowledge* that develops through the design activity to realise these component modules. As such, *Knowledge Modularity* structures abstract design knowledge *(concepts)* into modules whilst supporting the generation and maintenance of a modular solution over within and across *viewpoints* of the design.

Problems and difficulties of Modular Design support for Engineering Design Re-use and resulting requirements

This Knowledge Modularisation *phenomenon was utilised as the basis to fulfil the objective of identifying the requirements of MD as a mechanism to support improved Engineering Design Re-use. The following outlines the resulting findings with respect to this objective.*

Based on the concept of *Knowledge Modularity* a number of problems were highlighted. The modular design principle was required to be extended to support knowledge of a more *abstract* nature and the design *viewpoints* utilised by the designer. *Knowledge Modularity* requires a consistent approach to the capture and representation of design knowledge related to *concepts* and their *dependencies* within and across *viewpoints* throughout the evolution of the design.

The represented knowledge was shown to require some form of analysis to support the determination of module boundaries. However, *modularity* was highlighted as being a relative and not absolute property (on a scale between integral and modular). This characteristic of modularity is further complicated by the supposition that the modularity of a product is dependent on the *viewpoint* taken of that product, i.e. what is modular from the say the *viewpoint structure,* may not be from that of *function.* Thus, it was shown that *Knowledge Modularity* required an interpretation of the modularity within a particular design *viewpoint* and a means to understand the impact of this modularity on other *viewpoints.*

In addition, a set of *practising designer requirements* were obtained from observations during a twelve-month study of modular design practice (*Integrated Technology Mast* design) in an industrial setting and utilised to augment the literature-based findings.

A set of requirements for a Multi-Viewpoint Modular Design Methodology to support improved Engineering Design Re-use were defined as:

- The capture and appropriate representation of evolutionary design knowledge throughout the design activity both within and across differing viewpoints.
- The provision of mechanisms to support the exploration, optimisation and identification of *Knowledge Modularisation,* based on various design viewpoints and/or lifecycle objectives.
- Support the potential re-use of generated knowledge modules through their explicit representation and storage.

Inadequacies of existing approaches

Based upon the outlined requirements, reviewing the *features of existing approaches* extracted from *modular design literature* derived the *inadequacies of existing approaches.* These inadequacies can be outlined as:

- Insufficient support of the design knowledge, in that none of the existing modular design methodologies were able to adequately capture, model and represent evolutionary design knowledge.
    - Insufficient support of the *Knowledge Modularity,* in that none of the existing methodologies were shown to support the exploration of modularity of the evolutionary design knowledge. A particular lack of support was noted with respect to the abstract *concepts* from early design.
    - Significant limitations in the existing Module Identification (MI) processes, in that there was limited computational support for the module analysis and optimisation and the inherent modularity of the product structure was not explicitly identified or represented through existing approaches.

Multi-Viewpoint Modular Design Methodology

A novel *Multi-Viewpoint Modular Design Methodology* was been developed, based on the *phenomena of modular design support for Engineering Design Re-use,* to fulfil the objective of developing and defining a Multi-Viewpoint Modular Design Methodology. The Methodology was developed with the aim of addressing the limitations of existing modular design approaches and satisfying the identified requirements.

The overall idea of the methodology is to support the designer in creating and maintaining a modular design solution throughout the evolution of the design process. As such, the methodology embodies the concept of *Knowledge Modularity (KM).* The methodology has been developed to support the designer in modelling evolutionary design knowledge for analysis, modular optimisation to *identify inherent modularity* and map between design viewpoints to *maintain the modular solution.*

The Multi-Viewpoint MD Methodology embodies four main elements: a *Modelling Formalism,* an *Optimisation Mechanism,* a *Module Identification Mechanism* and a

Viewpoint Mapping Mechanism. The overall methodology defines the *declarative* and *procedural knowledge* required to fuse the methodology elements in a coherent framework and provide an articulate procedure for designers to follow in practice.

The *Modelling Mechanism* embodies a previously developed evolutionary *knowledge formalism* that supports *current working knowledge* and a *dependency structure matrix* to provide a representation, termed a *viewpoint model,* on which to support a modular analysis.

The *Optimisation Mechanism* was defined based on the hypothesis that the application of a *Genetic Algorithm* (GA) to cluster design knowledge would provide a basis on which to optimise modularity. The mechanism consists of a *Genetic Algorithm* and a *clustering criterion* derived from the *concept of Knowledge Modularity* and based on the *phenomenon of modular design support for improved Engineering Design Re-use*. The *optimisation mechanism* provides an optimised design knowledge model on which to facilitate module identification.

The *Module Identification Mechanism* embodies a *Module Strength Indicator* (MSI) *function* and an alternative representation of the design model termed the *Modular Structure Model* (MSM). The *Module Strength Indicator function and application* process is based on the formalisation of a *definition of modularity* extracted from the identified *characteristics of modular design*. The *Module Strength Indicator* values are interpreted and modelled to produce a *Modular Structure Model,* an alternative representation to the *viewpoint model.* The *Module Identification Mechanism* provides an *identification of inherent modularity* in the product structure.

The *Viewpoint Mapping Mechanism* embodies the knowledge formalism and matrix formalism of the *Modelling Formalism* element and the *Genetic Algorithm* of the *Optimisation Formalism* element. In addition, a *cross-viewpoint clustering criterion* embodied to assess the impact of modularity across viewpoints of the evolutionary design knowledge. The *Viewpoint Mapping Mechanism* promotes the *maintenance of the modular solution*.

Multi-Viewpoint Modular Design Methodology Evaluation

The Methodology was developed with the aim of addressing the limitations of existing modular design approaches and satisfying the identified requirements. To fulfil the objective of evaluating the functionality of the developed approach, it was partially realised within a computational environment and implemented within two industry based engineering design processes.

The *Modular Design System Implementation* was adapted from previously developed *process optimisation software*. The system was applied to two examples of design practice extracted from *modular design literature*, i.e. the *Climate Control* and *Alternator* cases. The application was aimed at an initial evaluation of the system implementation of the methodology's *Optimisation* and *Module Identification Mechanism.* The results concluded that the combination of the *Genetic Algorithm* and *clustering criterion* was an effective method to optimise design knowledge clusters on which to base module identification. Further the application of the *Module Identification Mechanism* identified a modular hierarchy and potential alternative module configurations that were not apparent from the original publications.

The complete Multi-Viewpoint Modular Design Methodology was applied to support two industry based *design practices,* i.e. the *Integrated Technology Mast* (ITM) and *Battle Group Thermal Imager* (BGTI) implementations. The *Integrated Technology Mast* implementation evaluation was undertaken on a trial and error approach. Based on a rationalisation of the activities carried out in the *Integrated Technology Mast* case an *Implementation and Evaluation Methodology* was formalised. The *Implementation and Evaluation Methodology* was applied as the basis to facilitate the *Battle Group Thermal Imager* implementation. The utilisation of *Implementation and Evaluation Methodology* resulted in a significant reduction in the time and effort required to implement the *Multi-Viewpoint Modular Design Methodology* in practice.

The results of both the *Integrated Technology Mast* and *Battle Group Thermal Imager* implementations identified potential improvements in the modularity of the designs. In addition, the methodology was deemed to provide an articulate procedure for practising designers, for example, refer to the *Integrated Technology Mast* verification certificate in Appendix A.

Strengths, Weaknesses and Future Work

An objective of the research was to analyse the *strengths and weaknesses of the Multi-Viewpoint Modular Design Methodology.* These have been determined based on the evaluations of both the system implementation and methodology itself. The features of the *Multi-Viewpoint Modular Design Methodology* that distinguish it from other modular design approaches can be highlighted as follows:

- Knowledge Modularisation Capabilities

  The incorporation of an evolutionary knowledge formalism supports the consistent capture and representation of design knowledge across *viewpoints* of design. The formalism provides an appropriate knowledge source on which to identify not only the *component* modularity, as with existing approaches, but the evolution of the modular solution based on the design knowledge associated with these components.

- Module Identification Support

  Through the formalisation of the core concept that constitutes a module in terms of the associated design knowledge the methodology provides a knowledge-based approach to module identification. This facilitates the identification of inherent modularity in the product structure, based on the dependency knowledge, between optimised clusters of design concepts. Due to the lack of formalisation of the core concept of a module previously existing module design approaches could not facilitate the identification of inherent modularity.

- Mapping Support

  Support for mapping between viewpoints allows the designer to maintain the modular solution as the design progresses from the abstract to the concrete, through successive iterations. Based on the concept of *Knowledge Modularisation* the methodology supports identification of the modular solution early in the design process. Mapping supports subsequent maintenance of the modular solution and allows the designer to both identify areas for further development and assess the impact of design decisions taken as part of the designs evolution with respect to the modularity of the design.

The main drawbacks of the *Multi-Viewpoint Modular Design Methodology* have also been identified as:

- The limitation of the matrix representation, in that it is a two dimensional representation of a multi-dimensional problem.
- The non-inclusion of costing metrics or mechanisms to allow the designer to assess the cost implications of their choice of module configuration.

The work presented in this thesis has provided a foundation for utilising modular design principles to support improved *Engineering Design Re-use* capabilities. The identified *strengths and weaknesses of the Multi-Viewpoint Modular Design Methodology* are utilised as the basis to fulfil the objectives of identify avenues of future research. Areas of future work have been identified based on the discussions of the Multi-Viewpoint Modular Design Methodology and are detailed in Section 9.4.

References

Aldridge, A. E. and K. Levine (2001). Surveying the social world: principles in survey research. Philadelphia, Open University Press.

Altmeyer, J. and B. Noll (1998). RODEO - Reuse of Design Objects: 1-2. Report, University of Kaiserslautern, Germany.

Altmeyer, J. and B. Schurmann (1996). On Design Formalisation and Retrieval of Reuse Candidates. Artificial Intelligence in Design Conference, Kluwer Academic Publishers:231-250.

Andreasen, M. M. (1991). The Theory of Domains. Engineering Design **2**(4).

Andreasen, M. M., A. Duffy, et al. (1995). Relation Types in Machine Systems. WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Andreasen, M. M., C. T. Hansen, et al. (1996). The Structuring of Products and Product Programmes. 2nd WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Andreasen, M. M. and A. Riitahuhta (1998). Considerations on metrics for modularisation and configuration. The 4th WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Berti, S., M. Germani, et al. (2001). Design of Product Families - An example within a small and medium sized enterprise. 13th International Conference on Engineering Design, Glasgow, UK, IMechE Professional Engineering Publications.

Bhatta, S. R. and A. K. Goel (1992). Discovery of Physical Principles from Design Experiences. Machine Learning - Artificial Intelligence in Design 1992, Pittsburgh.

Blackenfelt, M. (2001). Modularisation by Relational Matrices - A Method for the Consideration of Strategic and Functional Aspects. Design for Configuration - A Debate based on the 5th WDK Workshop on Product Structuring. A. Riitahuhta and A. Pulkkinen. Berlin, Springer Verlag: 134-152.

Blade, R., F. Klinge, et al. (1998). "Modular Outfitting." Journal of Ship Production **14**(3): 170-179.

Bracewell, R. H. and K. Shea (2001). CAEDRE - A Product Platform to Support Creation and Evaluation of Advanced Computer-Aided Engineering Tools. 13th International Conference on Engineering Design, Glasgow, UK, IMechE Professional Engineering Publications.

Bradburn, N. M. and S. Sudman (1979). Improving Interview Method and Questionnaire Design. London, Jossey-Bass Limited.

Brice, A. and B. Johns (1998). improving process design by improving the design process - A DRAMA white paper, QuantiSci.

Burke, G. P. and R. C. Miller (1998). "Modularisation Speeds Construction." Power Engineering **102**(2): 20-22.

Busby, J. S. (1999). "The Problems with Design Reuse: An Investigation into Outcomes and Antecedents." Journal of Engineering Design **10**(3): 277-296.

Chang and Ward (1995). Design-in-Modularity with Conceptual Robustness. DE- Design Engineering Technical Conferences, ASME.

Coates, G., A. H. B. Duffy, et al. (2000). A Generic Coordination Approach Applied to a Manufacturing Environment. Journal of Materials Processing Technology **107**: 404-411.

Council, D. (2000/2001). Design in Britain - facts, figures and quotes: 28.

Cross, N. (1998). Engineering Design Methods: Strategies for Product Development, Wiley & Sons.

Culley, S. J. (1998). Design Re-use of Standard Parts - Keynote Paper. Design Reuse - Engineering Design Conference, Brunel University, UK.

Davis, L. (1985). Job Shop Scheduling and Genetic Algorithms. Proceedings of the First International Conference on Genetic Algorithms and their Applications, Lawrence Erlbaum Associates, Hillsdale, NJ, USA.

Deneux, D. and X. H. Wang (2000). A knowledge model for functional re-design. Engineering Applications of Artificial Intelligence **13**: 85-98.

Dobrescu, G. and Y. Reich (2001). Design of a Gradual Modular Platform and Variants for A Layout Product Family. 13th International Conference on Engineering Design, Glasgow, UK, IMechE Professional Engineering Publications.

Donaldson, I. and K. MacCallum (1994). The Role of Computational Prototypes in Conceptual Models for Engineering Design. Artificial Intelligence in Design, Lausanne.

Duffy, A. H. B. (1997). The "What" and "How" of Learning in Design. IEEE Expert(May/June): 71-76.

Duffy, A. H. B. and S. M. Duffy (1996). Learning for Design Reuse. Artificial Intelligence in Engineering Design, Analysis and Manufacturing **10**: 139-142.

Duffy, A. H. B. and A. F. Ferns (1999). An Analysis of Design Reuse Benefits. International Conference on Engineering Design, Munich.

Duffy, A. H. B. and S. M. Kerr (1993). Customised perspectives of past designs from automated group rationalisations. Artificial Intelligence In Engineering Design **8**(3).

Duffy, A. H. B. and S. Legler (1998). A methodology for structuring past designs. 4th WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Duffy, A. H. B. and F. J. O'Donnell (1998). A Design Research Approach. Artificial Intelligence in Design, Lisbon, Portugal.

Duffy, A. H. B., J. S. Smith, et al. (1998). Design Reuse Research: a computational perspective - Keynote Paper. Design Reuse - Engineering Design Conference, Brunel University, UK.

Duffy, S. M. (1995). The Design Coordination Framework and the Design Complexity Map. Integrated Production Systems, IPS Research Seminar. Fuglsocentret, Fuglso, Institute for Engineering Design, Technical University of Denmark, Lyngby, Denmark.

Duffy, S. M., A. H. B. Duffy, et al. (1995). A Design Reuse Model. International Conference On Engineering Design, Praha.

Elgard, P. (1998). Industrial Practices with product Platforms in USA. 4th WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Elgard, P. and T. D. Miller (1998). Designing Product Families. Design for Integration in Manufacturing. Proceedings of the 13th IPS Research Seminar, Aalborg University, Fugsloe.

Eppinger, S. D., D. E. Whitney, et al. (1994). A Model-based method for organising tasks in a product development. Research in Engineering Design **6**: 1-13.

Eppinger, S. D., D. E. Whitney, et al. (1994). A Model-based Method for Organising Tasks in Product Development. Research in Engineering Design **6**: 1-13.

Erens, F. and K. Verhulst (1995). Managing System Design. WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Erens, F. and K. Verhulst (1996). Architectures for Product Families. 2nd WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Erixon, G. (1996). Modular Function Deployment. 2nd WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Erixon, G. (1997). Modular Function Deployment (MFD) in Practice - Industrial Examples. 3rd WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Erixon, G. (1998). Modular Function Deployment - A Method for Product Modularisation. Department of Manufacturing Systems

Assembly Systems Division. 100 44 Stockholm, The Royal Institute of Technology.
Finger, S. (1998). Design Reuse and Design Research - Keynote Paper. Design Reuse - Engineering Design Conference, Brunel University, UK.
Fothergill, P., J. A. Lacunza, et al. (1994). DEKLARE: A Methodological Approach to RE-DESIGN. Dept of Computing Science, Aberdeen University, Aberdeen.
Fowler, F. J. (1993). Survey Research Methods. London, SAGE Publications.
Galvin, P. (1999). "Product Modularity, Information Structures and the Diffusion of Innovation." International Journal of Technology Management **17**(5): 467-479.
Gao, Y., I. Zeid, et al. (1998). Characteristics of an effective design plan system to support reuse in case-based mechanical design. Knowledge Based Systems **10**: 337-350.
Gero, J. S. (1992). Creativity, emergence and evolution in design,.submitted to Knoweldge Based Systems.
Gero, J. S. and M. L. Maher (1990). Theoretical requirements for creative design by analogy, Colorado State University Fort Collins.
Gershenson, J. K. and G. J. Prasad (1997). Modularity in Product Design for Manufacturability. International Journal of Agile Manufacturing **1**(1).
Gershenson, J. K. and G. J. Prasad (1997). Product Modularity and its effect on Service and Maintenance. Proceedings of the 1997 Maintenance and Reliability Conference.
Gershenson, J. K., J. Prasad, et al. (1999). Modular Product Design: A Life-Cycle View. Journal of Integrated Design and Process Science **3**(4).
Girczyc, E. and S. Carlson (1993). Increasing Design Quality and Productivity through Design Reuse. 30th ACM/IEEE Design Automation Conference, ACM.
Glover, F. and M. Laguna (1993). Tabu search. Modern Heuristics Techniques for Combinatorial problems. C. R. Reeves. Oxford, UK, Blackwell Scientific Publications.
Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimisation and Machine Learning, Addison-Wesley, Massachusetts, USA.
Goldberg, D. E. and R. Linge (1985). Alleles, Loci and the Travelling Salesman Problem. The first international conference on genetic algorithms and their applications, Lawrence Erlbaum Associates, Hillsdale, NJ, USA.
Gonzalez-Zugasti, J. P. and K. N. Otto (2000). Modular Platform-Based Product Family Design. Computers and Information in Engineering Conference, ASME.
Gonzalez-Zugasti, J. P., K. N. Otto, et al. (2001). Assessing value in platformed product family design. Research in Engineering Design **13**: 30-41.
Greffenstette, J. J., R. Gopal, et al. (1985). Genetic Algorithms for the Travelling Salesman Problem. Proceedings of the First International Conference on Genetic Algorithms and their Applications, Lawrence Erlbaum Associates, Hillsdale, NJ, USA.
Gu, P., M. Hashemian, et al. (1997). An Integrated Modular Design Methodology for Life-Cycle Engineering. CIRP Annals **46**(1): 71-74.
Hall, P. A. V. (1992). Software Re-use, Reverse Engineering and Re-Engineering. Software Re-use and Reverse Engineering in Practice. P. A. V. Hall. London, UK, Chapman and Hall: 3-31.
Hansen, C. T. (1995). An Approach to Configuring Mechanical Systems. WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.
Hargreaves, D. H. (2001). towards education for innovation. Changing Behaviours *start here. D. Council, Design Council.
Hatton, L. (1996). Is modularisation always a good idea? Information and Software Technology **38**: 719-721.
He, D. W. and A. Kusiak (1997). Design of Assembly Systems for Modular Products. IEEE Transactions on Robotics and Automation **13**(5): 646-655.
He, D. W. and A. Kusiak (1998). Designing an Assembly Line for Modular Products. Computers and Industrial Engineering **34**(1): 37-52.
Heather, P. and S. Stone Crus Guide 5.Questionaires.

Henninger, S. (1996). Accelerating the Successful Reuse of Problem Solving Knowledge Through the Domain Lifecycle. 4th International Conference on Software Reuse, IEEE.

Herbertsson, J. (1995). Product Structuring in Design for Manufacture. WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Hofer, A. P. and M. Gruenenfelder (2001). Product Family Management based on Platform Concepts. 13th International Conference on Engineering Design, Glasgow, UK, IMechE Professional Engineering Publications.

Holland, J. H. (1962). Outline for a Logical Theory of Adaptive Systems. Joint Association of Computing Machinery **9**: 297-314.

Huang, C.-C. and A. Kusiak (1998). Modularity in Design of Products and Systems. IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans **28**(1): 66-77.

Hubka, V. and W. E. Eder (1988). Theory of Technical Systems; a Total Concept Theory for Engineering Design. Berlin, Springer Verlag.

Humphries, J. and D. Radcliffe (2001). Functional Modular Design - An Industry Case Study. the 13th International Conference on Engineering Design, Glasgow, UK, IMechE Professional Engineering Publications.

Inc, S. (1999). Synopsys Design Reuse Model - Who can Afford a $193 Million Chip?

Ishii, K., B. H. Lee, et al. (1995). Design for Product Retirement and Modularity Based on Technology Life-cycle. MH - Manufacturing Science and Engineering **3-2**: 921 - 933.

Jarventausta, S. and A. Pulkkinen (2001). Enhancing product modularisation with multiple views of decomposition and clustering. Design for Configuration - A debate on the 5th WDK Workshop on Product Structuring. A. Riitahuhta and A. Pulkkinen. Berlin, Springer Verlag: 153-168.

Jiao, J. and M. M. Tseng (1999). A methodology of developing product family architecture for mass customisation. Journal of Intelligent manufacturing **10**: 3-20.

Jones, M. E. (1995). Cost of ownership in design re-use. Electronic Product Design **16**(10): pp2.

Jones, M. E. (1995). Reusing integrated circuit designs. Computer Design **July**.

Juengst, W. E. and M. Heinrich (1998). Using Resource Balancing to Configure Modular Systems. IEEE Intelligent Systems **13**(4): 50-58.

Kamrani, A. K. (1997). Modular Design Methodology for Complex Parts. The 6th IIE Annual Engineering Research Conference, Miami Beach, Florida, USA.

Kerr, S. M. (1993). Customised Viewpoint Support for Utilising Experiential Knoweldge in Design. CAD Centre, DMEM. Glasgow, UK, University of Strathclyde: 196.

Khadilkar, D. V. and L. A. Stauffer (1996). An experimental evaluation of design information reuse during conceptual design. Journal of Engineering Design **7**(4): 331-339.

Kirkpatrick, S., C. D. Gelatt, et al. (1983). Optimisation by simulated annealing. Science **220**(4598): 671-680.

Knox, C. S. (1984). Engineering Documentation for CAD/CAM Applications, Marcel Dekker: 210 - 232.

Kolonder, J. (1993). Case Based Reasoning. San Mateo, Morgan Kaufmann.

Kusiak, A. and W. S. Chow (1987). Efficient solving of the group technology problem. Journal of Manufacturing Systems **6**(2): 117-124.

Kusiak, A. and C. Huang (1996). Development of Modular Products. IEEE Transactions on Components, Packaging and Manufacturing Technology **19, Part** A(4): 523 - 538.

Kusiak, A. and K. Park (1990). Concurrent Engineering: Decomposition and Scheduling of Design Activities. International Journal of Production Research **28**(10): 1883-1900.

Lashin, G. and M. Doblies (2001). CAX-Supported Modular System for Development and Design of Light Rail Vehicles. 13th International Conference on Engineering Design, Glasgow, UK, IMechE Professional Engineering Publications.

Lee, M. H. (1999). On models, modelling and the distinctive nature of model-based reasoning. AI Communications **12**: 127-137.

Liedholm, U. (1998). Conceptual Design of Products and Product Families. 4th WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Lloyd, P., J. Busby, et al. (1998). Reuse to overuse - a problem of fixation. Design Reuse - Engineering Design Conference, Brunel University, UK.

Logan, B. S. (1988). Representing Design Problems: formal and informal models in design,. ICAD 88.

Lubars, M. D. (1991). Reusing Designs for Rapid Application Development. IEEE International Conference on Communications, Microelectronics and Computer Technology Corp., 3500 West Balcones Center Drive, Austin, TX 78759.

MacCallum, K. J. (1995). Developing a Product Information Breakdown Structure (PIBS). WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

MacCallum, K. J., A. H. B. Duffy, et al. (1995). Design Reuse - Design Concepts in New Engineering Concepts. Glasgow, Strathclyde University.

MacCallum, K. J., A. H. B. Duffy, et al. (1987). An intelligent concept design assistant. IFIP WG5.2: Design Theory in CAD, North-Holland.

Maher, M. L. and A. G. d. S. Garza (1997). Case-Based Reasoning in Design. IEEE Expert **12**(2): 34-41.

Maher, M. L., A. G. d. S. Garza, et al. (1997). Case-based reasoning in design, Key centre for computing.

Manfaat, D., A. H. B. Duffy, et al. (1998). SPIDA :- abstracting and generalising layout design cases. Artificial Intelligence for Engineering Design, Analysis, and Manufacturing **12**: 141-159.

Marshall, R., P. G. Leaney, et al. (1998). Enhanced Product Realisation through Modular Design: An Example of Product/Process Integration. The SDPS/ASME/IEEE Third World Conference on Integrated Design and Process Technology, Berlin, Germany.

Miller, G. A. (1965). The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review **63**: 81-97.

Miller, T. and P. Elgard (1998). Defining Modules, Modularity and Modularisation - evolution of the concept in a historical perspective. Design for Integration in Manufacturing. Proceedings of the 13th IPS Research Seminar, Aalborg University, Fugsloe.

Miller, T. D. (1999). Modular Engineering of Production Plants. ICED - International Conference of Engineering Design.

Miller, T. D. and P. Elgard (1999). Structuring Principles for the Designer - Balancing Product Performance with Process Efficiency. Proceedings of the 1999 CIRP International Design Seminar: Integration of Process Knowledge into Design Support Systems. Kluwer Academic.

Mostow, J., M. Barley, et al. (1993). Automated reuse of design plans. Artificial Intelligence in Engineering Design. R. A. Adey: 2-15.

Muffato, M. (1999). Introducing a Platform Strategy in Product Development. International Journal of Production Economics **60**: 145-153.

Murata, T. and H. Ishibuchi (1994). Performance Evaluation of Genetic Algorithms for Flow Shop Scheduling Problems. Proceedings of the 1st IEEE Conference on Evolutionary Computation **2**: 812-817.

Nilsson, P. and G. Erixon (1998). The Chart of Modular Function Deployment. 4th WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

O'Grady, P. and W.-Y. Liang (1998). An Object Oriented Approach to Design with Modules. Computer Integrated Manufacturing Systems **11**(4): 267-283.

Oliver, I. M., C. J. Smith, et al. (1987). A study of permutation crossovers on the Travelling Salesman Problem. Proceedings of the Second International Conference on Genetic Algorithms and their Applications.

Ormerod, T. C., J. Mariani, et al. (1997). Supporting the process of re-use in innovative design environment. Colloquium of Institute of Electrical Engineers: 9/1 to 9/4.

Otto, K. (2001). A Process of Modularizing Product Families. the 13th International Conference on Engineering Design, Glasgow, UK, IMechE Professional Engineering Publications.

Ouyang, M., L. Chenggang, et al. (1996). Intelligent CAD approach for modular design. Proceedings of SPIE - The International Society for Optical Engineering **2644**: 614-619.

Pahl, G. and W. Beitz (1996). Engineering Design - A Systematic Approach. K. Wallace, Springer-Verlag: 407-433.

Pedersen, K., J. K. Allen, et al. (2001). Numerical Taxonomy - A systematic approach to identifying potential product platforms. 13th International Conference on Engineering Design, Glasgow, UK, IMechE Professional Engineering Publications.

Perera, S. and I. Watson (1998). Collaborative case-based estimating and design. Advances in Engineering Software **29**(10): 801-808.

Philippi, S. (1998). Modularisation of Petri Nets using Object Orientated Concepts. IEEE International Conference on Systems, Man and Cybernetics, IEEE.

Pimmler, T. U. and S. D. Eppinger (1994). Integration analysis of product decompositions. ASME Design Theory and Methodology Conference.

Preiss, K. (1999). Modelling of Knowledge Flows and their Impact. Journal of Knowledge Management **3**(1): 36-46.

Pugh, S. (1991). Total Design. Wokingham, Addison-Wesley.

Riesbeck, C. K. and R. C. Shank (1989). Inside Case-Based Reasoning. Hillsdale, New Jersey, Lawrence Erlbaum Associates.

Riitahuhta, A. and A. Pulkkinen, Eds. (2001). Design for Configuration - A debate based on the 5th WDK Worshop on Product Structuring. Berlin, Springer Verlag.

Rogers, G. C. and L. Bottaci (1997). Modular Production Systems: A New Manufacturing Paradigm. International Journal of Intelligent Manufacturing **8**(2): 147-156.

Rosen, D. W. (1996). Design of Modular Product Architectures in Discrete Design Spaces Subject to Life Cycle Issues. The ASME Design Engineering Technical Conferences and Design Automation Conference, Irvine, California, ASME.

Salheih, S. e. M. and A. K. Kamrani (1999). Macro level product development using design for modularity. Robotics and Computer Integrated-Manufacturing **15**: 319-329.

Saucier, A., C. Vargas, et al. (1994). Compared application of two knowledge modelisation methodologies on a car cylinder head design problem - Future expectations in the use of a generic design application development tool. Valenciennes, France.

Schellhammer, W. and H. Karandikar (2001). Metrics for Executing a Product Family Strategy. 13th International Conference on Engineering Design, Glasgow, UK, IMechE Professional Engineering Publications.

Shanin, T. M. M., P. T. J. Andrews, et al. (1998). A design reuse system. Design Reuse - Engineering Design Conference, Brunel University, UK.

Siddique, Z. and D. W. Rosen (2001). On combinatorial design spaces for the configuration design of product families. Artificial Intelligence in Engineering Design, Analysis and Manufacturing **15**(2): 91-108.

Simpson, T. W., J. R. A. Maier, et al. (2001). Product platform design: method and application. Research in Engineering Design **13**: 2-22.

Sivaloganathan, S. (1998). Foreword. Design Reuse - Engineering Design Conference, Brunel University, UK.

Sivaloganathan, S. and T. M. M. Shanin, Eds. (1998). Design Reuse - Engineering Design Conference. London, Professional Engineering Publishing Limited.

Smith, J. S. (2000). An Overview of Issues within Engineering Design Re-use. Glasgow, UK, University of Strathclyde. Report No: CADC/00-03/R/01, Re-use/00-02/R/01: 18.

Smith, J. S. (2002). Battle Group Thermal Imager Multi-Viewpoint Modular Design Methodology Implementation Results. Glasgow, UK, University of Strathclyde, Report No: CADC/02-04/R/04, Re-use/02-03/R/03: 17.

Smith, J. S. (2002). Integrated Technology Mast Multi-Viewpoint Modular Design Methodology Implementation Results. Glasgow, UK, University of Strathclyde. Report No: CADC/02-03/R/03, Re-use/02-02/R/02: 44.

Smith, J. S. (2002). Multi-Viewpoint Modular Design Methodology Study Attendance and Evaluation Questionnaire Samples. Glasgow, UK, University of Strathclyde. Report No: CADC/02-05/R/05, Re-use/02-04/R/04: 8.

Smith, J. S. (2002). A Review of Engineering Design Research supporting 'Re-use'. Glasgow, University of Strathclyde. Report No: CADC/02-01/R/01, Re-use/02-01/R/01: 45.

Smith, J. S. and A. H. B. Duffy (2001). Product Structuring for Design Re-use. Design for Configuration - A Debate based on the 5th WDK Workshop on Product Structuring. A. Riitahuhta and A. Pulkkinen. Berlin, Springer Verlag: 83-100.

Smith, J. S. and A. H. B. Duffy (2001). Re-using Knowledge - Why, What and Where. 13th International Conference on Engineering Design, Glasgow, I Mech E.

Smith, J. S., M. D. Robb, et al. (2001). An Experience of Modularity through Design. 13th International Conference on Engineering Design, Glasgow, UK, I Mech E.

Smith, P. G. and D. G. Reinertsen (1997). Developing Products in Half the Time - New Rules, New Tools, John Wiley & Sons: 105-117.

Smith, R. P. and S. D. Eppinger (1993). Characteristics and models of iteration in engineering design. International Conference on Engineering Design, The Hague.

Sosale, S., M. Hashemiam, et al. (1997). Product Modularization for Re-use and Recycling. DE-Concurrent Product Design and Environmentally Conscious Manufacturing **94**: 195-206.

Starkweather, T., S. McDaniel, et al. (1991). A comparison of genetic sequencing operators. Proceedings of the fourth international conference on genetic algorithms, San Mateo, CA, USA.

Steward, D. V. (1981). The design structure system: a method for managing the design of complex systems. IEEE Transactions: Engineering Management **28**: 71-74.

Suh, N. P. (1999). A Theory of Complexity, Periodicity and the Design Axioms. Research in Engineering Design **11**: 116-131.

Syswerda, G. (1991). Scheduling Optimisation using Genetic Algorithms. Handbook of Genetic Algorithms. L. Davis, Van Rostrand Reinhold: 332-349.

Takeda, H., P. Veerkamp, et al. (1990). Modeling Design Processes. AI Magazine (Winter): 37-48.

Taleb-Bendiab, A. (1992). A Concept Reuse Approach for Engineering Design Problem Solving. Waterloo University Ontario Canada.

Tichem, M. and T. Storm (1996). Issues in Product Structuring. 2nd WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Tiihonen, J., T. Lehtonen, et al. (1998). Modeling Configurable Product Families. 4th WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Tjalve, E., Ed. (1979). A Short Course in Industrial design, Newnes-Butter Worths.

Todd, D. (1997). Multiple Criteria Genetic Algorithms in Engineering Design and Operation. Engineering Design Centre. Newcastle, UK, University of Newcastle Upon Tyne.

Todd, D. (1997). Multiple Criteria Genetic Algorithms in Engineering Design and Operation. Engineering Design Centre, Department of Marine Technology. Newcastle, University of Newcastle: 279.

Trauter, R. (1998). Design-Related Reuse Problems - an experience report. 5th International Conference on Software Reuse.

Tseng, M. M. and J. Jiao (1997). Case based evolutionary design for mass customisation. Computers and industrial engineering **33**(1-2): 319-323.

Ulrich, K. and K. Tung (1991). Fundamentals of Product Modularity. Issues in Design/Manufacture Integration. A. Sharon, ASME. **DE 39:** 73-79.

Ulrich, K. T. and S. D. Eppinger (1995). Product Design and Development. New York, USA, McGraw-Hill.

Vargas, C., A. Saucier, et al. (1994). Knowledge Modelisation and Constraint Propogation in a Computer Aided Design System. Applications of constraint processing in CAD.

Vargas, C., A. Saucier, et al. (1995). A language and a tool for the development of computer aided mechanical design applications. ISATA, Stuttgart, Germany.

W.D.K. (1995). Proceedings. The 1st WDK Workshop on Product Structuring, Delft.

W.D.K. (1996). Proceedings. The 2nd WDK Workshop on Product Structuring, Delft, Netherlands.

W.D.K. (1997). Proceedings. The 3rd WDK Workshop on Product Structuring, Delft, Netherlands.

W.D.K. (1998). Proceedings. The 4th WDK Workshop on Product Structuring, Delft, Netherlands.

Whitfield, R. I., A. H. B. Duffy, et al. (2001). Efficient Process Optimisation. Submitted to International Journal of Concurrent Engineering: Research & Applications.

Whitfield, R. I., J. S. Smith, et al. (2002). Identifying Component Modules. Artificial Intelligence in Design (AID '02), Cambridge, UK.

Whitley, D., T. Starkweather, et al. (1989). Scheduling problems and the Travelling Salesman: The genetic Edge Recombination Operator. Proceedings of the third international conference on Genetic Algorithms, Morgan Kaufmann.

Yu, B. and K. J. MacCallum (1995). A Product Structure Methodology to Support Configuration Design. WDK Workshop on Product Structuring, Delft University of Technology, Delft, The Netherlands.

Zamirowski, E. J. and K. N. Otto (1999). Identifying Product Portfolio Architecture Modularity Using Function and Variety Heuristics. Proceedings of the 11th International Conference on Design Theory and Methodology.

Zhang, Y. (1998). Computer based modelling and management for current working knowledge evolution support. CAD Centre, DMEM,. Glasgow, University of Strathclyde.

Zhang, Y. (1998). Design Knowledge Structuring. Glasgow, UK, University of Strathclyde, Report No: CADC/02-02/R/02, KM/02-01/R/01: 40.

Zhou, J. and S. A. Irani Design of Modular Layouts for Fabrication-Based Assembly Facilities. 210 Baker Systems Building, Colombus OH43210, Department of Industrial Welding and Systems Engineering, The Ohio State University.

Appendix A – BAE Systems Marine Ltd Verification Letter

BAE SYSTEMS

Joanne S Smith
Department of Design, Manufacture and Engineering Management
James Weir Building
University of Strathclyde
75 Montrose Street
Glasgow G1 1XJ, UK.

23rd August 2002

Dear Joanne,

We hereby verify that the result of your modular analysis of our integrated technology mast is an accurate and correct representation of the modularisation of the design and manufacture of the technical demonstrator.

Your research has made a valued contribution to our work in this field and we would look to utilise this approach in the future in order to assist ourselves in clearly delineating the activity boundaries and hence reduce the number of iterations required for the whole design.

Wishing you every good fortune with your research.

Malcolm D Robb

Dr. Malcolm D Robb

Principal Engineer
Forward Design Group

BAE SYSTEMS Marine Limited South Street Scotstoun Glasgow G14 0XN Scotland
Telephone +44 (0) 141 959 1207 Fax +44 (0) 141 958 0642