In L. J. de Vin, A. H. C. Ng, P. Thorvald, W. G. Sullivan and M. Ahmad (Eds.)
Proceedings of the Eighteenth International Conference on Flexible Automation and Intelligent Manufacturing, FAIM, Vol. 1, pp. 187-196. University of Skovde, Sweden, 30 June-2 July 2008

Multi-objective Optimisation of Product Modularity

Mike Lee[1*], Prof. Keith Case [1], and Dr. Russell Marshall [2]

[1]Wolfson School of Engineering
Loughborough University,
Loughborough, Leicestershire,
LE11 3TU

[2]Dept. Design and Technology
Loughborough University,
Loughborough, Leicestershire,
LE11 3TU

ABSTRACT

The optimal modular configuration of a product's architecture can lead to many advantages throughout the product lifecycle. Advantages such as: ease of product upgrade, maintenance, repair and disposal, increased product variety and greater product development speed. However, finding an optimal modular configuration is often difficult. Finding a solution will invariably mean trade-offs will have to be made between various lifecycle drivers. One of the main strengths of a computerised optimisation is that trade-off analysis becomes simple and straightforward and hence speeds up the product architecture decision making process. However, there are a lack of computerised methods that can be applied to optimise modularity for multiple lifecycle objectives. To this end, an integrated optimisation framework has been developed to optimise modularity from a whole lifecycle perspective, namely, design, production, use and end of life. For each lifecycle phase there are two modularity criteria- module independence and module coherence. The criteria that fall under the category of module independence evaluate the degree of coupling between the products components, coupling can be physical, functional or design based. Criteria under module coherence, evaluate the similarity of modular drivers between components. The paper will examine the developed optimisation framework and software prototype. The prototype software uses a number of matrixes to represent the product architecture. A goal based genetic algorithm is used to search the matrixes for modular configurations that most satisfies the criteria of the four lifecycle phases. Sensitively analysis is carried out by changing the goal weights.

1.0. INTRODUCTION AND LITERATURE

Ever decreasing product lifecycles are leading to considerable changes in the way products are being designed. Central to this is the notion of modular product design. The benefits include, shortening design time, improved reliability, reduced construction costs and simplified service and repair. Modular design therefore represents an important means of producing competitive advantages in fast growing and changing markets

Ulrich and Tung (1991) define modularity in terms of two characteristics of product design: similarity between the physical and functional architecture of the design and the minimisation of incidental interactions between physical components

There are many more product modularity definitions in the literature. What is generally agreed from the literature is that product modularity is the arrangement of a product's components into clusters. The clusters contain stronger component interactions and similarities within clusters than between clusters. These interactions and similarities include those which arise from the component's physical and functional interactions and those which arise from the various processes the components undergo during their lifecycle. The choice of which lifecycle processes to concentrate on as the main drivers for modularity will depend upon the type of product.

There have been many previous modular design techniques that have mainly attempted to optimise one modularity objective. Methods that use clustering heuristics have been developed; these techniques only optimise modularity, for one objective, for example, functional interactions (Pimmler and Eppinger, 1994), and testability (Kusiak, 2002). Single objective optimisation models have also been developed. Slahieh and Kamrani's (1999)

---

method aims at optimisation of component similarities. Gu and Sosale (1999) have developed a heuristic and non-linear optimisation model to optimise lifecycle objectives. Manual heuristic based methods have also been developed. Erixon and Ericssion's Modular Function Deployment (MFD) uses a comprehensive list of modular drivers which can be used to evaluate modules. Stone et al (2000) work from a functional basis using energy, signal of material flows between components and use a set of heuristics to form modules.

The main problems with the previous methods include: lack of structure; poorly defined modularity evaluation guidelines; no modularity criteria weighting guidelines; suffer from pareto dominance during optimisation; have poorly designed optimisation algorithms which can get stuck on local optimal; and don't allow sensitively analysis

Therefore the main contribution of this research is to address the associated problems and create a computerised multi-optimisation framework for product modularity across the whole product lifecycle. This paper will look at the developed optimisation criteria and optimisation model.

2.0 MODULARITY OPTIMISATION CRITERIA FOR THE WHOLE PRODUCT LIFECYCLE

By evaluating modularity from each of the four product lifecycle viewpoints modularity optimisation becomes more organised and logical, this gives rise to several advantages over other methods. Firstly, the organisation of modularity criteria into lifecycle phases creates a clearly defined optimisation problem that can be efficiently handled by the multi-objective algorithm.

Next, the importance of each modularity optimisation criteria becomes easier to quantify. The importance of each lifecycle phase can be directly linked to the type of product and its characteristics. For example, for a high tech product that has a fast rate of evolution, the design phase will be of high importance as the product will likely undergo many design changes during its planned lifespan. In contrast, for a mature, high volume product, the production phase will be of utmost importance.

Lastly, sensitively analysis can be carried out. By varying the considered importance of the optimisation criteria the designer is able to study the effects and arrive at the most suitable modular design for the product that is being designed/ redesigned. The results of a sensitively analysis can be difficult to analysis if the optimisation problem is poorly defined and structured. The organisation of modularity criteria into lifecycle phases creates a clearly defined optimisation problem. By adjusting the optimisation goals of each lifecycle phase the effects on the other phases can be analysed. This is handed efficiently using a goal programming approach to the multi-objective optimisation problem.

The criteria for modularity optimisation across the four stages of the product lifecycle can be seen in figure 1. The modularity criteria are split into two categories, module independence and module coherence- see figure 1. The criteria that fall under the category of module independence evaluate the degree of coupling between the products components, coupling can be physical, functional or design based. Criteria under module coherence, evaluate the similarity of modular drivers between components. The two criteria can be conflicting and an optimal modular product architecture for any given lifecycle phase will often be a compromise between module independence and module coherence

	Module Independence Criteria	**Module Coherence Criteria**
Design Phase	Engineering Metrics	Technology Change
	Functional Interactions	Design Carryover
Production Phase	Component Attachment	Current Product Variety
	Component Alignment	Make or Outsource
Use Phase	Component Attachment	Maintenance and Service
	Component Alignment	Component Life
End of Life Phase	Component Attachment	Reuse
		Recyclability

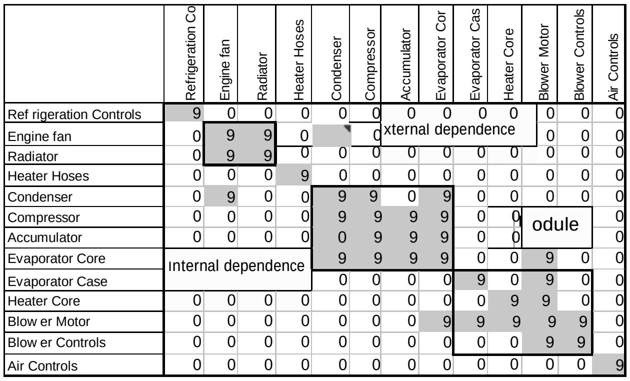
Figure 1: Modularity Analysis Criteria for the Whole Product Lifecycle

2.1 *MODULE INDEPENDENCE*

Module independence is defined as ‘Each module within the product architecture should have a minimum amount of coupling between other modules’. Coupling can be defined as functional, physicals or design information based interactions between modules.

Therefore, the key to obtaining high module independence is to obtain modules that have stronger component couplings within modules and weaker component couplings between modules. Figure 2 illustrates this principle in a design structure matrix (DSM) representation of the product architecture.

Module independence is evaluated by using the appropriate module independence evaluation guidelines to evaluate the coupling between all component pairs. The results are then stored in a DSM matrix, known as a module independence matrix.



	Refrigeration Co	Engine fan	Radiator	Heater Hoses	Condenser	Compressor	Accumulator	Evaporator Cor	Evaporator Cas	Heater Core	Blower Motor	Blower Controls	Air Controls
Refrigeration Controls	9	0	0	0	0	0	0	0	0	0	0	0	0
Engine fan	0	9	9	0		0					0	0	0
Radiator	0	9	9	0	0	0	0	0	0	0	0	0	0
Heater Hoses	0	0	0	9	0	0	0	0	0	0	0	0	0
Condenser	0	9	0	0	9	9	0	9	0	0	0	0	0
Compressor	0	0	0	0	9	9	9	9	0	0			0
Accumulator	0	0	0	0	0	9	9	9	0	0			0
Evaporator Core					9	9	9	9	0	0	9	0	0
Evaporator Case					0	0	0	0	9	0	9	0	0
Heater Core	0	0	0	0	0	0	0	0	0	9	9	0	0
Blower Motor	0	0	0	0	0	0	0	9	9	9	9	9	0
Blower Controls	0	0	0	0	0	0	0	0	0	0	9	9	0
Air Controls	0	0	0	0	0	0	0	0	0	0	0	0	9

Figure 2: Clustered DSM example

2.2 *MODULE COHERENCE*

Module coherence is defined as ‘For each phase of the product lifecycle, the modular product architecture should obtain maximum possible coherence of a modular drivers within each module’.

To evaluate the product modularity according to module coherence one must analyse the modular driver scores for all components in the product. The modular driver scores are established from the evaluation guidelines presented in this chapter. Components with similar modular driver scores will have high module coherence and should be placed into the same module during optimization.

2.3 *MODULARITY OPTIMISATION CRITERIA FOR THE DESIGN PHASE*

There are two main aims of modularity at the design stage. The first aim is to split a product into modules that can be designed in relative independence from one another, enabling the associated benefits of concurrent design. The second aim is to allow the effects of design change to be isolated within modules. The next section will look at these principles in more detail.

2.3.1 *MODULE INDEPENDENCE CRITERIA FOR THE DESIGN PHASE*

A good modular structure at the design phase should be created in a manner that ensures a minimal amount of design coupling between modules. A low level of design coupling between modules will reduce the amount of cross design team interactions that takes place. This enables each design team to produce their allocated module in relative independence from each other, speeding up product development. Therefore it can be said that module independence at the design stage aims to create design independence of modules by minimising the level of design coupling between modules. Design coupling will be looked at from two perspectives: design coupling due to engineering metrics and design coupling due to functional interactions.

DESIGN COUPLING DUE TO ENGINEERING METRICS

During a new product design or product redesign the products engineering metrics (EM) are likely to undergo changes, these changes will perpetrate throughout the product, affecting numerous components. If many different sub-systems or modules are affected this can be very disruptive, as design efforts will need to be coordinated between the various design teams that are reasonable for the affected product sub- assemblies or modules. To minimise the effects of changing EM one can optimise the product architecture by placing components affected by the same EM into the same modules. This means that the effects of the changing EM will be isolated within one module, reducing the amount of design information passing between teams, decreasing complexities and costs and increasing the level of design concurrency able to take place during a new product design/ re-design.

To evaluate the effects of EM on components a QFD based approach is used. Firstly, all current and future customer needs are listed and there associated EM are drawn up. The EM is then mapped to the physical components. For each component the amount of design effort needed to accommodate a possible change of the EM is estimated. Components that are highly affected by the same EM should be grouped into the same module.

DESIGN COUPLING DUE TO FUNCTIONAL INTERACTIONS

The second aspect of modular coupling at the design phase is that of functional interactions. For this criterion the optimisation aim is to maximise functional interactions within modules and minimise functional interactions between modules. Components that have functional interactions are likely to be strongly coupled at the design stage. Therefore by keeping functional interactions between modules to a minimal, design team interactions are also kept to a minimal, in turn minimising cross team information flow and redesign iterations. For this research the functional interactions method of Pimmler and Eppinger(1994) has been adopted.

2.3.1 MODULE COHERENCE CRITERIA FOR THE DESIGN STAGE

In order to create high module coherence at the design stage the components within modules must have similar future redesign needs. Therefore to evaluate module coherence at the design stage the two modular drivers of interest are that of technology change and design carryover.

TECHNOLOGY CHANGE

The technology change modular driver evaluate that rate of technology change in the each component. During optimisation, components with new or fast evolving technologies, should not be placed with components that are unlikely to undergo any technology change.

DESIGN CARRYOVER

The design carryover modular driver serves to identify product components that can be reused across the next generation of products. During module formation these components can be isolated into modules to improve design reuse efficiency. This is done by evaluating which of the customer needs and associated engineering metrics are likely to change for the next generation of products.

2.4 MODULARITY OPTIMISATION CRITERIA FOR THE PRODUCTION PHASE

The aim of product modularity at the production phase is to create modules that can be manufactured and assembled as efficiently as possible.

2.4.1 MODULE INDEPENDENCE CRITERIA FOR THE PRODUCTION PHASE

For high module independence at the production phase one must aim to minimise the physical interactions between modules. If the physical coupling between modules is too high it may not be possible to manufacture and assemble modules concurrently. Therefore it is highly desirable to group components that have strong physical relationships into the same module and components with weak relationships between modules. Physical couplings between components are evaluated in terms of the geometric attachment and alignment between components. The strength of attachment is dependent upon how the two components are linked together e.g. bolt fit, screw fit, snap fit. Alignment depends upon the estimated level of precision needed to align the two components to form an assembly.

2.4.2 MODULE COHERENCE CRITERIA FOR THE PRODUCTION PHASE

CURRENT PRODUCT VARIETY NEEDS

Components that are variants are best separated from components that are common across the product range in order to add efficient production. Separation of a product into variant and common modules benefits the production phase in a number of ways: Firstly, the common modules can be assembled first then variants modules can be added later in the production cycle. This is known as delayed product differentiation, which reduces production lead times. Secondly, production inventory is reduced, as common modules can be assembled and used across a number of product families.

The decision on whether components should be common or variant is not an easy choice; it will often involve a trade-off between the incurred costs of added variety and the advantages that the added product variety will offer product marketability. Product platforms and product variety is a vast area of research in it own right, therefore a detailed analysis of this area is out of the scope of this thesis. Product variety is only one part of the modularity analysis. Therefore for this research, one simply wishes to identify components that are likely to be variant or common. The basic idea is to look at each customer need and the associated engineering metric and decide whether it will be likely to need variety. Some variety will be essential in order for the product family to serve the required market segments whilst other variety may only be desirable. This desirable variety should be decided by considering the effects on the associated engineering metrics. I.e. adding a certain type of variety may mean many components would need to be redesigned to accommodate it; therefore the company may choose not to implement it as it would be too costly.

MAKE OR OUTSOURCE

The second criteria for module production coherence is that of make or outsource. Components should be grouped into the same module based on whether they will be made in house or outsourced to a supplier. If this is not know for the design i.e. if the design is in the conceptual stages, then the make or outsource decision should be based on whether components have a high intellectual property or if the company has the competences and production facilities to produce the components.

2.5 MODULARITY OPTIMISATION CRITERIA FOR THE PRODUCT USE PHASE

During the products use phase a product is likely to undergo some amount of repair and maintenance. Therefore an optimal modular structure for this phase can be achieved by grouping components that have similar maintenance and service needs and similar component lifes into the same module. At the same time each modules must remain as independent as possible, with minimal physical interactions between modules. Thus ensuring that worn out modules can be replaced or repair efficiently.

2. 5.1 MODULE INDEPENDENCE CRITERIA FOR THE PRODUCT USE PHASE

Module independence at the use life phase shares the same aims to that of the production phase, that of minimising the physical interactions between modules. To reinitiate, physical interactions are defined as the geometric mating and alignment needs between components.

2.5.2 MODULE COHERENCE CRITERIA FOR THE PRODUCT USE PHASE

MAINTENANCE AND SERVICE

Components that have a high likelihood that maintenance and service will be required and have similar maintenance requirements should be isolated into modules to enhance module coherence for the use stage. The service and maintenance analysis is carried out by listing all the known or expected service and maintenance operations and mapping them to the effected components. Components that have the same maintenance or service operations should be placed into the same module. Figure…shows an example with its resultant coherence matrix. The coherence matrix (a pair-wise comparison matrix) is generated from the service and maintenance components mappings.

COMPONENT LIFE

During the use phase it is important that components within modules have similar wear out times. Components that are likely to wear out quickly can be isolated into modules that can then be replaced with minimal impact on the rest of the system.

2.6 MODULARITY OPTIMISATION CRITERIA FOR THE END OF LIFE PHASE

Creating optimal modularity at the end of life phase requires that components are grouped according to there reusability and recyclability. However, at the same time, one must also ensure that the physical coupling between modules remains low to ensure that disassembly effort is minimised.

2.6.1 MODULE INDEPENDENCE CRITERIA FOR THE END OF LIFE PHASE

Module independence at the end of life phase shares similar aims to that of the production phase, that of minimising the physical interactions between modules. However, there is one fundamental difference, module independence at the end of life only considers the mating criteria. The alignment criteria is not necessary because at the end of the products life modules and components will not be reassembled like the use phase.

2.6.2 MODULE COHERENCE CRITERIA FOR THE END OF LIFE PHASE

REUSABILITY

Components that can be reused or remanufactured should be placed into the same module. Decisions on component reuse should be based on whether there are financial incentives and the desirability to remanufacture. High value components that need little processing, components with a high wear out life and components unlikely to be redesigned for future variety needs make ideal candidates for reuse.

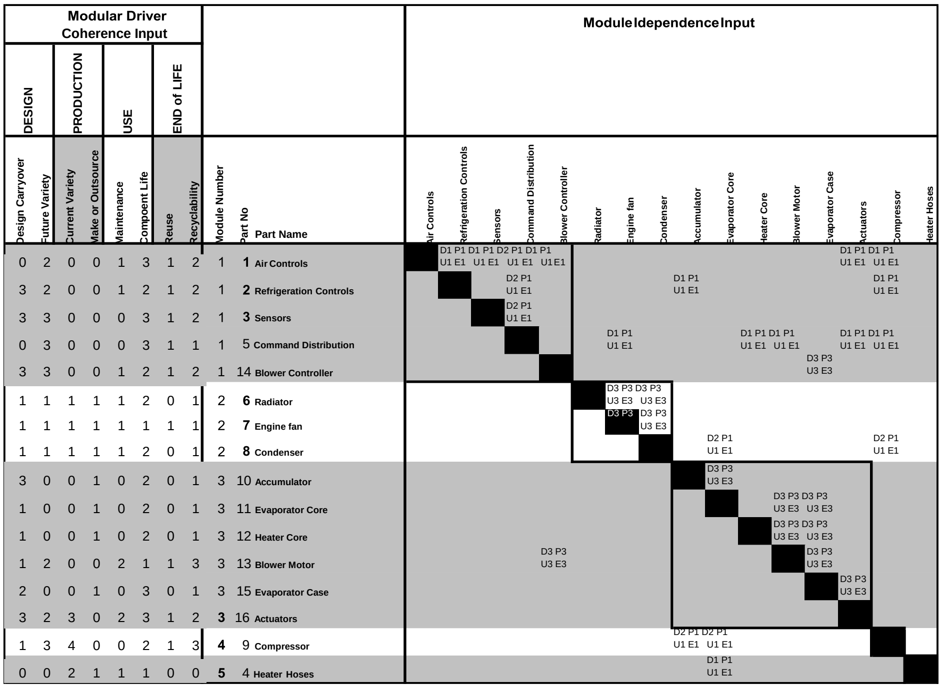
RECYCLABILITY

Components that have high material homogeneity should be grouped into the same module to make the recycling operations easier and more cost effective to perform. Similarly, components with hazardous materials should be isolated into modules. To evaluate the similarity of the

3.0 SOFTWARE OPTIMISATION- MODULE FORMATION

The main goal of the software optimisation is to provide a designer with optimal groupings of components that have maximum module independence and coherence across the whole product lifecycle.

The prototype software has been created in an excel environment using VB coded macros to create a problem specific genetic algorithm (GA) based optimiser and a VB programmed user interface. The screenshot in figure 3 is the main user input screen. On the right of the screen a design structure matrix is used to enter functional, physical and design coupling between all components- therefore this matrix represents module independence. The columns to the left of the matrix are the corresponding modular driver scores for each component. The software creates a design structure matrix (modular driver coherence matrix) from each of the modular driver columns. The similarity between each pair is calculated based on a driver value similarity scoring system.



Modular Driver Coherence Input											Module Idependence Input															
DESIGN		PRODUCTION		USE		END of LIFE																				
Design Carryover	Future Variety	Current Variety	Make or Outsource	Maintenance	Compoent Life	Reuse	Recyclability	Module Number	Part No	Part Name	Air Controls	Refrigeration Controls	Sensors	Command Distribution	Blower Controller	Radiator	Engine fan	Condenser	Accumulator	Evaporator Core	Heater Core	Blower Motor	Evaporator Case	Actuators	Compressor	Heater Hoses
0	2	0	0	1	3	1	2	1	**1**	**Air Controls**	■	D1 P1 U1 E1	D1 P1 U1 E1	D2 P1 U1 E1	D1 P1 U1 E1									D1 P1 U1 E1	D1 P1 U1 E1	
3	2	0	0	1	2	1	2	1	**2**	**Refrigeration Controls**		■		D2 P1 U1 E1					D1 P1 U1 E1						D1 P1 U1 E1	
3	3	0	0	0	3	1	2	1	**3**	**Sensors**			■	D2 P1 U1 E1												
0	3	0	0	0	3	1	1	1	5	**Command Distribution**				■			D1 P1 U1 E1				D1 P1 U1 E1	D1 P1 U1 E1		D1 P1 U1 E1	D1 P1 U1 E1	
3	3	0	0	1	2	1	2	1	14	**Blower Controller**					■								D3 P3 U3 E3			
1	1	1	1	1	2	0	1	2	**6**	**Radiator**						■	D3 P3 U3 E3	D3 P3 U3 E3								
1	1	1	1	1	1	1	1	2	**7**	**Engine fan**							D3 P3	D3 P3 U3 E3								
1	1	1	1	1	2	0	1	2	**8**	**Condenser**								■		D2 P1 U1 E1					D2 P1 U1 E1	
3	0	0	1	0	2	0	1	3	10	**Accumulator**									■	D3 P3 U3 E3						
1	0	0	1	0	2	0	1	3	11	**Evaporator Core**										■		D3 P3 U3 E3	D3 P3 U3 E3			
1	0	0	1	0	2	0	1	3	12	**Heater Core**											■	D3 P3 U3 E3	D3 P3 U3 E3			
1	2	0	0	2	1	1	3	3	13	**Blower Motor**					D3 P3 U3 E3							■	D3 P3 U3 E3			
2	0	0	1	0	3	0	1	3	15	**Evaporator Case**													■	D3 P3 U3 E3		
3	2	3	0	2	3	1	2	**3**	16	**Actuators**														■		
1	3	4	0	0	2	1	3	**4**	9	**Compressor**									D2 P1 U1 E1	D2 P1 U1 E1					■	
0	0	2	1	1	1	0	0	**5**	4	**Heater Hoses**										D1 P1 U1 E1						■

Figure 3: Modular design software -main input screen

The goal of the GA based optimiser is to find the optimal modular structure from within the generated matrices. This is achieved by adjusting module size and membership until product modularity is optimised. However there are problems:

Firstly, attempting to maximise one lifecycle phase’s modularity will often mean a reduction in another phase’s modularity. If the optimisation technique is not properly designed this can lead to a pareto dominated solution. That is a solution that is dominated by one or more objectives. For example, consider the following two modular optimisation results. For the first, the design phase reaches a score of 90, production 40, use 40, end of life 70, giving a total score of 240.for the second the design phase reaches a score of 60, production 60, use 60, end of life 60, giving a total score of 240. Both solutions have the same overall objective score, however the later is the more favourable as it contains no dominance of objectives.

Secondly, a conflict exits between module independence module and module coherence. Module independence favours fewer modules whilst module coherence favours a larger number of modules. An optimal optimisation should be a balance between the two.

To address these problems goal programming is used. The goal programming technique provides a balanced and controllable optimisation, Figure 4 outlines the application of the goal programming and GA based optimisation method. As seen in step 2 the method first performs a GA optimisation for each phase of the lifecycle, where the goal is to find the optimal balance between the corresponding module independence and coherence criteria by using equation 1 and 2. Once this has been preformed maximum fitness scores for each lifecycle phase will be known and the GA is then run to optimise modularity for the whole lifecycle. The maximum fitness scores for each phase will affectively become the goal maximums. The goal is then to minimise all deviations from each goal's maximum using equation 3. By adjusting goal weights it is easy to perform a sensitively analysis of various modular architecture alternatives to enable the designer to consider the merits and trade-offs between different solutions.

Step 1. Set the required number of modules
Step 2. Optimise modularity for each lifecycle phase

- *2.1 For lifecycle phase n, run the GA with module independence coherence and module goals weights set to equal*
- *2.2 Examine the module grouping results and if satisfactory then go to step 2.5*
- *2.3 Adjust the goal deviation weights for module independence and module coherence and rerun the GA.*
- *2.4 Examine the module grouping results and satisfactory then go to step 2.5*
- *2.5 Repeat steps 2.1 to 2.5 for lifecycle phase n*

Step 3. Use the optimal modularity results from step 2 set the goal maximums for each lifecycle phase.
Step 4. Run the GA to minimise goal deviations from the goal maximums
Step 5. perform sensitivity analysis by adjusting the goal deviation weights for the four lifecycle phases and repeating step 4
Step 6. Examine the module grouping results and if a satisfactory solution exists then end the process.
Step 7. Adjust the required number of modules and go to step 2

Figure 4: Application of the goal programming and GA based optimisation method

Equation 1 calculates modular driver coherence within modules

$$MC = \frac{CI_{\max}}{\sum_{m}^{1} Ci_{\text{internal}}} \qquad (1)$$

Where: lcp=lifecycle phase, m= module number Ci= module coherence interactions, w= weight given to lifecycle phase

Equation 2 calculates modular driver independence

$$MI = \frac{Ii_{total}}{\sum_{m}^{1} Ii_{external}} \qquad (2)$$

Where: lcp=lifecycle phase, m= module number Ii= module independence interactions, w= weight given to lifecycle phase

Equation 3, the total module goal to minimize

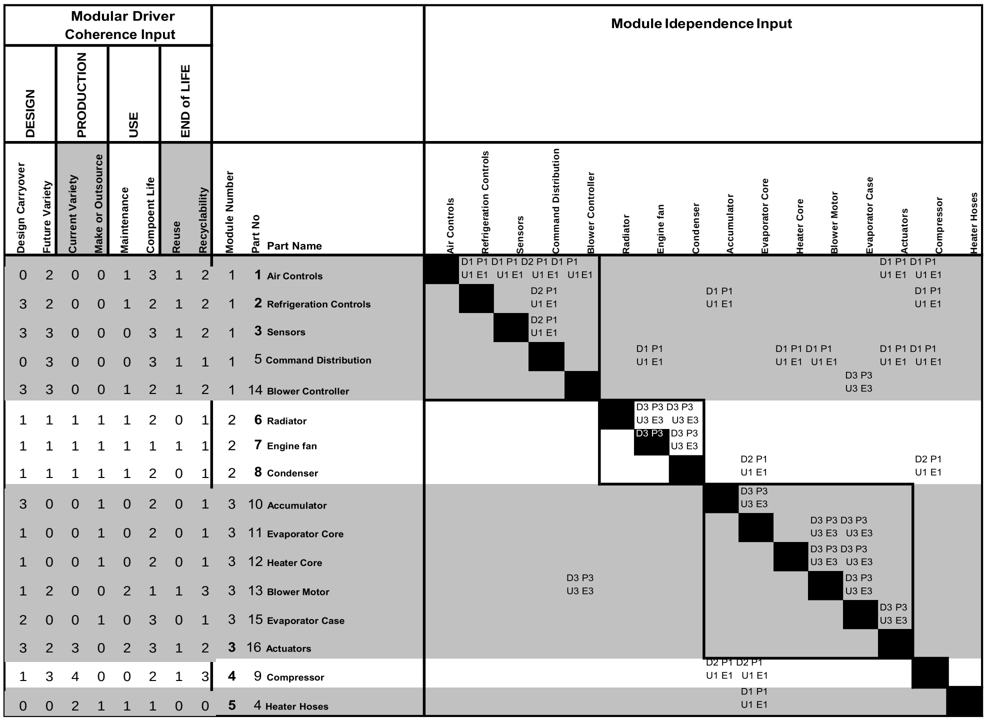
$$TM = \min\Big((Gd_{design} \times w), (Gd_{production} \times w), (Gd_{use} \times w), (Gd_{eol} \times w)\Big) \qquad (3)$$

Where Gd_{mi} = deviation from goal and w= goal deviation weight

4.0 EXAMPLE CASE STUDY- CAR CLIMATE CONTROL SYSTEM

The car climate control system has been used in various studies, so makes an ideal case to make comparisons with. The aim of the case study is merely to demonstrate the potential of the method as a means of optimising multiple modularity objectives. Therefore the modular driver scores entered into the software are by no means completely accurate and are based on the author's best judgements so will need to be quantified by further research. However the functional and physical interactions were based on the previous the work of Pimmler and Eppinger(1994), so may be considered more accurate.

Example results of the software optimisation can be seen in figures 5. Figure 5 shows the modularisation of the product with the lifecycle deviation goal weightings set equally. By changing the goal deviation weightings of the four lifecycle phase's sensitively analysis was preformed- partial results of which can be seen in figure 6.



Modular Driver Coherence Input											Module Idependence Input															
DESIGN		PRODUCTION		USE		END of LIFE																				
Design Carryover	Future Variety	Current Variety	Make or Outsource	Maintenance	Compoent Life	Reuse	Recyclability	Module Number	Part No	Part Name	Air Controls	Refrigeration Controls	Sensors	Command Distribution	Blower Controller	Radiator	Engine fan	Condenser	Accumulator	Evaporator Core	Heater Core	Blower Motor	Evaporator Case	Actuators	Compressor	Heater Hoses
0	2	0	0	1	3	1	2	1	1	Air Controls	■	D1 P1 U1 E1	D1 P1 U1 E1	D2 P1 U1 E1	D1 P1 U1 E1									D1 P1 U1 E1	D1 P1 U1 E1	
3	2	0	0	1	2	1	2	1	2	Refrigeration Controls		■		D2 P1 U1 E1					D1 P1 U1 E1						D1 P1 U1 E1	
3	3	0	0	0	3	1	2	1	3	Sensors			■	D2 P1 U1 E1												
0	3	0	0	0	3	1	1	1	5	Command Distribution				■			D1 P1 U1 E1				D1 P1 U1 E1	D1 P1 U1 E1		D1 P1 U1 E1	D1 P1 U1 E1	
3	3	0	0	1	2	1	2	1	14	Blower Controller					■								D3 P3 U3 E3			
1	1	1	1	1	2	0	1	2	6	Radiator						■	D3 P3 U3 E3	D3 P3 U3 E3								
1	1	1	1	1	1	1	1	2	7	Engine fan							D3 P3	D3 P3 U3 E3								
1	1	1	1	1	2	0	1	2	8	Condenser								■		D2 P1 U1 E1					D2 P1 U1 E1	
3	0	0	1	0	2	0	1	3	10	Accumulator									■	D3 P3 U3 E3						
1	0	0	1	0	2	0	1	3	11	Evaporator Core										■		D3 P3 U3 E3	D3 P3 U3 E3			
1	0	0	1	0	2	0	1	3	12	Heater Core											■	D3 P3 U3 E3	D3 P3 U3 E3			
1	2	0	0	2	1	1	3	3	13	Blower Motor					D3 P3 U3 E3							■	D3 P3 U3 E3			
2	0	0	1	0	3	0	1	3	15	Evaporator Case													■	D3 P3 U3 E3		
3	2	3	0	2	3	1	2	3	16	Actuators														■		
1	3	4	0	0	2	1	3	4	9	Compressor									D2 P1 U1 E1	D2 P1 U1 E1					■	
0	0	2	1	1	1	0	0	5	4	Heater Hoses										D1 P1 U1 E1						■

Figure 5: Modular architecture of car climate control system with equally weighted goal deviation for each lifecycle phase

Figure 6: results of a sensitivity analysis, adjusting the design goal deviation weight

Design		**Production**		**Use**		**End of Life**	
Goal Deviation weight	Goal Deviation	Goal Deviation weight	Goal Deviation	Goal Deviation weight	Goal Deviation	Goal Deviation weight	Goal Deviation
50%	42%	100%	22%	100%	21%	100%	23%
100%	25%	100%	28%	100%	29%	100%	26%
150%	20%	100%	35%	100%	34%	100%	37%
200%	16%	100%	38%	100%	36%	100%	38%

5.0 CONCLUSIONS AND FURTHER WORK

It has been seen that optimisation of a products modularity is a desirable but often complex task. However using the proposed computerised methodology, modularity optimisation is a less laborious and time intensive task, making it more approachable for the designer or organisation to consider. Future work will focus on further assessment and refinement of the technique though case studies.

REFERENCES

[1] Ericsson, A. and Erixon, G., 1999, Controlling design variants: Modular product platforms. ASME Press. New York, NY

[2] Gu, P., & Sosale, S. (1999). Product modularization for life cycle engineering. Robotics and Computer Integrated Manufacturing, 15, 387-401

[3] Kusiak, A. (2002,) Integrated product and process design: a modularity perspective, Journal of Engineering Design, 13, 223-231

[4] Pimmler, T. U. and Eppinger, S. D., 1994, Integration Analysis of Product Decompositions, ASME DesignTheory and Methodology Conference, Minneapolis, MN

[5] Salhieh, S. M., & Kamrani, A. K. (1999). Macro level product development using design for modularity. Robotics and Computer Integrated-Manufacturing, 15, 319-329.

[6] Stone, R. B., Wood, K. L., & Crawford. R. H. (2000). A heuristic method for identifying modules for product architectures. Design Studies, 21, 5-31

[7] Ulrich, K., and Tung, K., 1991, Fundamentals of product modularity. In Issues in Design/ Manufacture Integration 1991, edited by A. Sharon (New York: ASME), DE 39