INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

DESIGN OF PRODUCT ARCHITECTURES IN INCREMENTALLY DEVELOPED COMPLEX PRODUCTS

David F Wyatt[1], Claudia M Eckert[2] and P John Clarkson[1]
(1) University of Cambridge (2) The Open University

ABSTRACT
This paper seeks to understand how product architecture arises and influences the design process, using a case study of a UK-based diesel engine manufacturer. Product architecture is defined as a combination of functional and physical structure, including definitions of interfaces between components and spatial layout, at the overall product level. In the case study the minimisation of novelty was observed to be a major factor, causing the majority of the function structure and function-to-component mapping to be carried over from previous engines. New technologies are selected at an early stage without being integrated into complete engine concepts. Architecture-level decisions are then made implicitly when carrying over elements from previous products and when cascading product requirements to the component level. Elements of product architecture continue to be defined over much of the New Product Introduction (NPI) process, countering the expectation that product architecture should be defined during early design. These results suggest that support for product architecture design should take account of its incremental and extended nature.

Keywords: Product architecture, complex product, systems engineering, case study, diesel engine

1 INTRODUCTION
Complex products may have long histories. Such products are frequently designed incrementally, based heavily on previous designs, since the cost and risk of developing a new model from scratch would be prohibitive. However requirements change from version to version, leading to addition of new features, modification of existing features and removal of unnecessary elements. As a result, the architecture of such a product may be a mixture of old and new – with important implications for design change and future generations of products. The aim of this paper is to identify how product architecture is defined and the effects it has, through a case study involving a series of interviews with engineers from a single company. The approach is descriptive and reflective, comparing reality with the "textbook" approach to engineering design. The case study company designs and manufactures complex engineering products (off-highway diesel engines) that have been incrementally developed for many decades, but are coming under increasing pressure both from customers and from environmental protection legislation. The characteristics noted here challenge the textbook view of product architecture design, with potential implications for both academic research and industry practice.
The structure of the remainder of this paper is as follows. Section 2 defines product architecture and describes its importance for design. Section 3 gives details of the questions the research aims to answer, the case study chosen and the methodology used. Section 4 describes observations of factors that affect diesel engine design, both external and due to the nature of the product; Section 5 then describes how the product is designed within the company. Section 6 discusses the results, and Section 7 states conclusions and directions for further work.

2 PRODUCT ARCHITECTURE AND ITS RELATION TO DESIGN
The term "product architecture" has a range of interpretations. The most often-cited is Ulrich's three-element definition [1], focusing on the functions that make up a product (defined as a "function structure" in the sense of Pahl and Beitz [2], a set of "[verb] [noun]" subfunctions interconnected by flows of energy, materials and signals); how these are implemented by components (defined as separable physical parts or subassemblies, but the definition is extended to cover software subroutines and distinct regions of a silicon chip); and the degree of coupling and cross-compatibility of the interfaces between components. Others (for example [3]) interpret "product architecture" to mean one

element of this definition in isolation, such as the function structure. However, in other cases, "product architecture" relates to the overall structure of the product – for example in terms of component connectivity in the Design Structure Matrix field [4], as an arrangement of "organs" corresponding to clusters of related component features [5], or in terms of rough geometric layouts (which could be interpreted as structural and spatial relations between components) in more colloquial usage. For software and other (chiefly non-physical) systems, [6] gives a range of definitions for "architecture", focusing on structure or organisation in terms of component parts and interrelationships; however, the authors also point out that fundamentally "architecture is what architects produce, and […] what architects do is to help clients make decisions about building systems".

In order to encompass the majority of these interpretations, for the purposes of this paper *architecture* will be used to mean the elements of Ulrich's definition combined with a 4th point:

1. The arrangement of functional elements;
2. The mapping from functional elements to physical components;
3. The specification of the interfaces among interacting physical components;
4. The overall structure of a design in terms of component-component relations.

It is worth noting that complex products can be described at many levels of detail, each of which could be viewed as having an "architecture" in the sense of the above definition. *Product* architecture will therefore be used to refer to these four elements interpreted at the level of the item that a company sells to its customers – e.g. a complete diesel engine, rather than a cooling system or a crankshaft.

Despite this ambiguity of definition, it is generally accepted that product architecture has a strong influence on product success. Some authors have stated that as much as 80% of the cost of a product may be committed during the conceptual stage of the design process [7], where product architecture decisions should be made, according to [1]; such numerical values are debatable [8], but nonetheless serve to remind designers of the importance of paying attention to product architecture design. Ulrich [1] states that product architecture has this effect through influencing product change, product variety, component standardisation, product performance and product development management, a view reinforced by later authors [9], [10].

From the perspective of engineering design research, the fact that the product architecture is in some senses also the basis for the design process means that, for studies of design practice, it determines what observations may be made at which stage. Additionally, in as much as the overall goal of engineering design research is to enable designers to produce better products via better design processes [11], understanding how and where product architecture is determined may allow for more appropriate support for all stages of design and better timing of the application of such support.

Given the importance of product architecture, it is not surprising that "textbook" engineering design processes emphasise the importance of attempting conceptual design in full before beginning detailed design, although they acknowledge the possibility of iteration. An example is Pahl and Beitz's "Systematic Design" [2], shown in the left half of Figure 1.

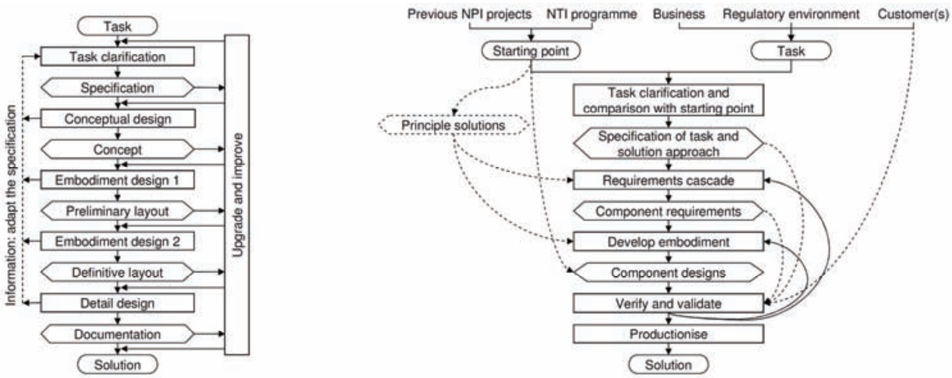

Figure 1. On the left is a summarised version of the Pahl and Beitz design process model [2]. On the right is a modified version, representing the design process followed by the case study company. Solid arrows indicate workflow, while dotted arrows indicate information flow.

In this model, the product architecture corresponds to the "Concept" which is defined as including the function structure, "principle solutions" (means to implement subfunctions), and "working structure" (an idea of how the principle solutions will be integrated). Ulrich [1] also suggests that architecture should be defined at the outset of a design project, bearing in mind its wide-reaching effects. However, in practice, design is frequently incremental: many products evolve and/or have parts of their design reused in later products [12], and even when there is opportunity for more radical innovation, engineers may be "fixated" on the previous solution and unable or unwilling to consider alternatives [13].
Having considered the "textbook" perspective, the remainder of this paper examines how product architecture is designed and influences design in practice in the case study company. For comparison, the right half of Figure 1 shows a flowchart representing the design process used by the case study company (discussed in more detail in Section 5.3).

3 METHODOLOGY

The case study was carried out as part of a project to investigate generative computational tools for supporting product architecture design [14], in order to understand the context and requirements for such tools. The researchers' initial expectations were shaped by the "textbook" view of product design outlined in Section 2: it was assumed that there would be a formal definition of product architecture and that its design would be a central but concise and well-defined stage in the overall design process. This was found in due course not to be the case.

3.1 Research questions

The research sought to address the following questions:
1. What is the case study company's interpretation of product architecture?
2. How do external factors affect the design?
3. What is the company's design process and organisation?
4. How does the design process determine the product architecture?
5. How does the product architecture affect later design?

3.2 Case study target

The case study was undertaken at a UK-based manufacturer of medium-scale off-highway diesel engines, which in the last decade became vertically integrated into a multinational engineering group after decades of independent operation. The company was chosen because diesel engines share characteristics with many other complex products, including complexity, safety-criticality, strong legislative pressure, requirements for high reliability, and potentially contradictory market demands (in this case, increased performance and reduced environmental impact within a fixed physical footprint). In addition, the authors had previously established a relationship with the company through studies on engineering change, facilitating access to the appropriate personnel.

3.3 Data collection

The case study took the form of 13 semi-structured interviews with 8 engineers from the company, held at the company's headquarters over a period from May to December 2008, as shown in Table 1. See Section 5.2 for explanations of the engineers' roles.

Table 1. The personnel interviewed at the case study company.

Engineer	Role	Interview dates					
		19 May 2008	4 June 2008	11 June 2008	20 June 2008	7 October 2008	16 December 2008
1	Platform	■				■	■
2	Platform (NPI team)	■					■
3	Platform (design architect)			■			
4	Platform				■	■	
5	Platform (design architect)					■	■
6	CPPD		■				
7	CPPD					■	
8	CAE						■

Each interview lasted between 45 minutes and two hours. Interviews were carried out by the first two authors, making opportunistic use of prepared questions aligned with the research questions stated in Section 3.1. All interviews were recorded, then transcribed and reviewed to uncover common themes. In addition, comments from other employees of the company were collected in an informal manner, and the researchers were given sample items of design documentation: a design brief for a top cover, and a Pugh analysis of options for mounting of the engine components in electrical generation applications. After the initial series of interviews from May-October 2008, a follow-up session was organised in December 2008 to test conclusions drawn.

4 PRODUCT ARCHITECTURE AND EXTERNAL INFLUENCES

4.1 Product architecture of diesel engines

As a product, diesel engines are complex and interlinked, becoming more so with each new generation of engines through advances in technology. This complexity is only feasible to achieve through incremental design – as stated by Engineer 1, "for the core engine we start where we left off with the previous generation". In order to meet new requirements there has recently been an influx of new technologies: engines are now regulated electronically rather than mechanically, enhancing performance while reducing emissions, and it has also been necessary to add "aftertreatment" systems to neutralise particular pollutants (nitrous oxides and particulate matter), in addition to external components such as turbochargers and water cooling systems introduced in previous product generations. Such additions can have deleterious effects on other aspects of the engine: some subsystems consume part of the output mechanical power, and their physical volume must be explicitly considered in installing an engine into applications. This means that the core engine has been reduced to a subsystem of the company's product: "where a diesel engine used to be a diesel engine, it's now a chemical plant, where one of the by-products is a turning shaft" (Engineer 4).

The engineers interviewed did not themselves offer any formal definition of "product architecture", although they did use the term in discussion. In particular they did not make any reference to the concept of a function structure or a function-to-component mapping. Informally they implied that "product architecture" meant the technologies used in the engine and the key performance and geometric parameters (e.g. power output or cylinder spacing). The authors' interpretation of the academic definition of product architecture given in Section 2 in this context is as follows:

- A **function structure** of a diesel engine would consist of subfunctions such as "Mix fuel and air", "Combust fuel", "Convert linear to rotary motion", "Separate soot from exhaust" etc. All diesel engines are to a large degree similar at the functional level, apart from certain auxiliary features that were on occasion incorporated into the engine (e.g. control of starter motors), depending on market demand.
- The **mapping from functions to components** then consists of the choice of technologies used to implement these functions, e.g. use of a filter to "Separate soot from exhaust" or a crankshaft to "Convert linear to rotary motion". This is again very similar across different diesel engines. Different technologies may require different auxiliary functions to support them.
- **Specifications of interfaces between components** may include attachment, spatial, transfer, control and communication, user and environmental aspects [15]; since the coupling within the product is generally strong, modularity and cross-compatibility of interfaces is limited, but there are some options (for example, alternative mounting positions for the oil filter).
- The **overall structure** reflects the spatial layout of the engine and the component connectivity.

However, there is no clear distinction between architecture at the product level and at the next level down (subsystems), perhaps due to the recent shift in core engine status mentioned above.

4.2 External forces affecting design

The dynamics of new product introduction are dominated at present by emissions legislation, as described in [16]. This means, in particular, that new engines have a window of approximately 5 years during which they may be sold in countries where emissions are strongly regulated (the US and the EU); product performance is therefore traded off against development time. Meanwhile, technology additions have led to a significant increase in the R&D expenditure necessary to develop an engine. This increase has been mitigated, and the development time reduced, through the involvement of the manufacturing division at an early stage to reduce the number of late changes to the design, and the use of fast-turnaround manufacturing processes to produce initial production runs before the long-lead-time production processes are in place. However, in order to meet the requirements of the upcoming legislation (US EPA Tier 4 Interim/EU Stage IIIB, referred to in the remainder of this paper as "Tier 4 Interim") it has been necessary to move beyond the engine and interact with the design of the machine in which the engine will be used.

As far as the customers are concerned, the function of a diesel engine within a machine is not expected to change, nor are the key performance measures associated with the engine (chiefly power output, size, cost and reliability), although the typical values for these parameters improve as technology progresses. The company's customers are themselves machine manufacturers, either internal to the parent group (a market to which the company has privileged access) or external, varying in many respects including type of machine, production volumes, and the conditions in which the engines are used (altitude, temperature, presence of explosive gases, safety-criticality etc.). Such variation means that different customers have different priorities, forcing the company to offer a variety of options – for example, they supply several different designs of sump for different applications with requirements for structural strength, gradeability, maximum oil capacity etc. However, customers also want to be able to design their machines independently of emissions legislation changes, which encourages the company to keep the "external interface" of an engine constant. Regional variations in emissions legislation and periods of transition around legislation introduction mean that different models of engine must be manufactured simultaneously.

The company makes extensive use of outsourcing. Some design functions are centralised within the company's parent group, such as exhaust aftertreatment systems and electronic control hardware, and the company relies on external suppliers of components – and, more rarely, expertise – for a significant fraction of the engine (negotiation with suppliers is handled centrally within the group). The company is required to interact with the group's network of dealers as well as its own dealers.

5 DESIGNING A NEW GENERATION OF ENGINE

5.1 Novelty, reuse and changes

The development of new engines is fundamentally incremental: "the core architecture hasn't changed in 100 years, we don't think it's going to change, in the next 10-15 years [...], it would take some catastrophic event to do that" (Engineer 1). This is due to a reluctance to incorporate novelty, which may arise either from changes to the structure of a component or to the behaviour required of it. Thus, although some types of product-architecture-level change (i.e. changes to the function structure or the function-to-component mapping) inevitably cause novelty, even conserving every structural detail of a component may involve novelty if its interface specifications change. Somewhat counter-intuitively, a carried-over component or subsystem operating outside its prior performance envelope may be considered more novel than a new design that has conservative performance goals.

Interviewees stated six reasons for minimising novelty:

- **Reliability history:** Previous designs are either known to be reliable or have known problems that can be fixed – important in a market where reliability is at a premium, and where designs must be certified before they can be sold. However, such historical information can become obsolete as technologies are added to the engine and operating conditions for components are modified.
- **Development effort:** Change to the design costs money and takes time.
- **Capital cost of change:** Certain aspects of the company's organisation are specific to a given model of engine, including manufacturing and test facilities and the dealer network.
- **Optional components:** Making a design change may require redesigning many different components used to fulfil the different options, with potentially significant cost implications.
- **Economies of scale:** Development costs are amortised over large numbers of components by sharing the design between several engines.
- **Manufacturing and service complexity:** Since a new design of engine will be manufactured and maintained concurrently with older (and newer) engines, it is important that the physical structure and assembly of an engine adheres to a common pattern. However, where a difference is necessary, it should be sufficient in order to prevent accidental misassembly.

In concrete terms, the Programme Objectives specify the reuse of specific component designs. A percentage "New Content" is calculated for each engine, which is required to be between 18-25% –

the lower the better. This value is then used to estimate development costs for the new design and to determine qualification requirements.

Where changes are made to the design, they may be required either in order to improve the performance of carried-over parts, to meet new requirements or, in the later stages of design, to remedy emergent problems. However, changes are hindered by the numerous hard constraints on global parameters of the engine, particularly size and weight: for instance, the engine must be small enough that the final machine allows sightlines for the operator down the side of the engine cowling, fits inside a standard shipping container, and may be transported by road without an escort. Changes that would cause violation of such constraints then result in further changes to compensate. Due to the complexity of the product (and in particular to the fact that certain components may be “frozen”, prohibiting further changes), the change may be applied to a different subsystem from that which causes the problem; as stated by Engineer 4, “That’s the other thing with engines: they’re almost organic, like all complicated things are, in that there’s always an angle you can take off them”.

Changes can be quantitative, merely altering the dimensions of a component, or qualitative, involving changing or adding to the technologies used in the engine. Such qualitative changes also added complexity and cost, so were generally viewed as undesirable, but were sometimes necessary in order to meet requirements. Examples include exhaust gas recirculation, aftertreatment and electronic control systems to reduce emissions while improving performance, and CCTV to replace sightlines obscured by a larger engine cowling.

5.2 Design process participants

Within the company, the groups that participate in the New Product Introduction (NPI) process are:

- Platform team: the division of the company responsible for co-ordinating the development of specific engines and gathering input from customers. Specific roles within it include:
    - Products Director: recognises the need for a new product and initiates NPI.
    - NPI team: specific to a particular engine, this is a multidisciplinary team consisting of design engineers (concerned with both the physical design of the engine and its performance) and representatives from manufacturing, after-sales service and the programme team (responsible for schedule and cost adherence).
    - Design architect: also specific to a particular engine, the architect is involved during the majority of the NPI process. They both carry out early design and co-ordinate later design.
- CPPD (Continuous Product-Process Development) teams: these have responsibility for developing and supplying components of a particular type (e.g. sumps and top covers), meeting the specifications defined by that engine’s platform team.
- CAE (Computer-Aided Engineering) teams: commissioned by CPPD teams to carry out modelling and simulation to predict the behaviour of the engine (performance, mechanical stress, thermal flows, vibration levels etc.). CAE is involved from the early definition of the engine onwards, although they do not own the design at any stage.
- Manufacturing: responsible for the manufacture and assembly of an engine.

5.3 The New Product Introduction (NPI) process

Given the importance of delivering a new product on time, because of the strict emissions legislation timetable described in Section 4.2, the company has implemented a highly structured stage-gate process for NPI. It has eight “gateways” (“Launch” and numbers 1-7) spanning the period from start of development through to a year after the start of production. A schematic of the process is shown in Figure 2, constructed from comments made by interviewees. The figure gives a description of each gateway and, where relevant, activities in the intervening phases (the interviewees defined the process with respect to the gateways, rather than assigning a distinct identity to each phase). It also indicates the involvement of different participants in the design process and the ownership of the design at each stage. The first 6 gateways of the process constitute the design phase; within this period, Figure 3shows when the different items of design documentation are first produced and how the elements of the engine’s product architecture (as defined in Section 4.1) is designed over a prolonged period.

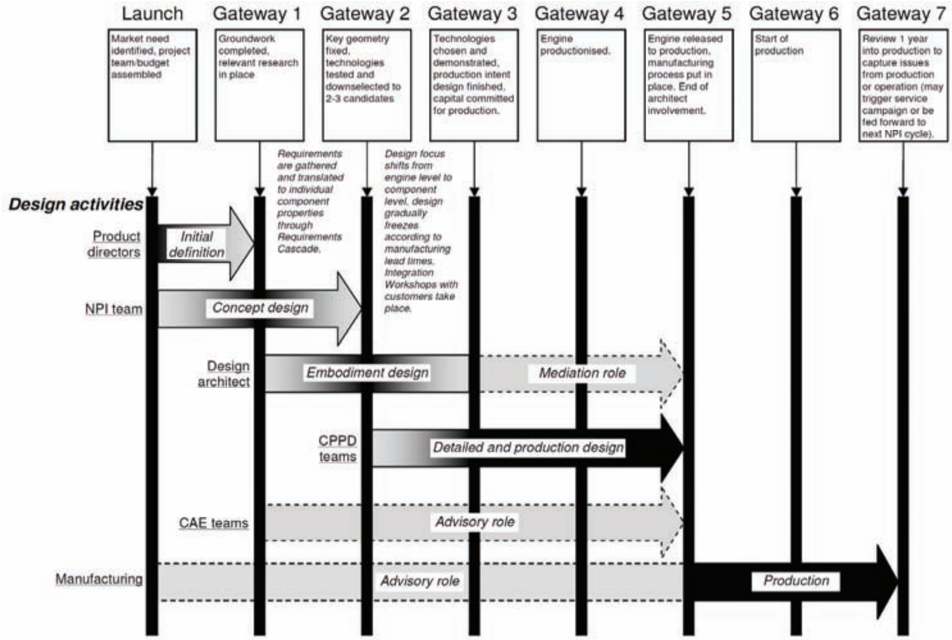

Figure 2. An outline of the company's stage-gate NPI process.

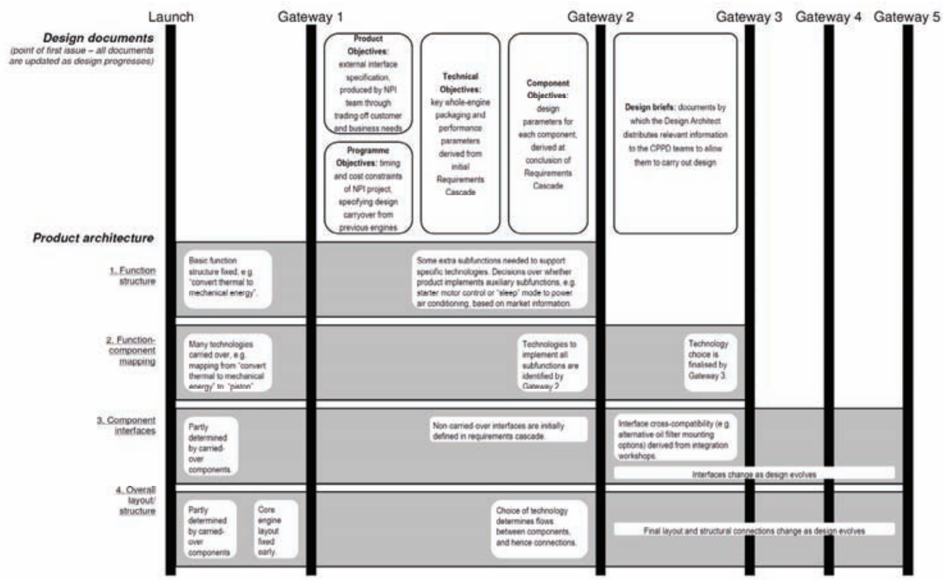

Figure 3. The first six NPI gateways, showing documentation and architecture elements.

5.4 The Systems Engineering approach: gathering and cascading requirements

In the first part of the NPI process, the emphasis is on gathering both customer and business requirements for the new engine. Performance and reliability are viewed as separate from, and more fundamental than, other non-functional objectives; the former is addressed through CAE analysis of the design, and the latter through FMEA (Failure Mode and Effect Analysis) – but enumerating all potential failure modes requires significant experience on the part of the participants. Quality with respect to non-functional objectives is assessed at this stage through the involvement of representatives from the relevant divisions (Manufacturing etc.) and later, as the design becomes more mature, through formalised procedures such as serviceability audits and testing the product on a virtual production line. At the same time as defining requirements, the NPI team also establishes the different environmental conditions under which the requirements must be satisfied, principally temperature and altitude. These not only affect the performance of the engine through physical laws, but may also impact other objectives – for example, the "Arctic mitt" scenario requires specific engine components to be accessible while wearing protective clothing needed for -40°C ambient temperature.

As a result of the need for greater integration between the engine and the customers' machines for the Tier 4 Interim engine (described in Section 4.2), the company held "integration workshops" during Phase 2-3 to allow negotiations between design teams and the customers. Requirements from the workshops could be incorporated directly since the design of the engine was still relatively fluid before Gateway 3. However, inevitably different sets of customer requirements are inconsistent with each other and with the business needs, and a tradeoff is needed. While to some degree this inconsistency can be handled by increasing the number of product options offered, adding options has an associated cost which must itself be traded off against the potential profit.

Once the requirements have been gathered and formalised as the Product Objectives, their implications for the whole engine (Technical Objectives) and the parameters of the individual components (Component Objectives) must be derived through the "Requirements Cascade". In order to do this, the parameters to be defined and their interrelationships must be known, requiring either firm decisions or assumptions about the embodiment of the engine to have been made. In many cases these assumptions equate to choosing technologies to implement functions. Technology choices may be carried over from previous engines or taken from a separate New Technology Introduction (NTI) programme. NTI obtains potential technologies from fundamental research, often in academia, and then aims to develop them to a stage where they can achieve the performance that would be required in a production application, but does not consider non-performance objectives such as packaging size or reliability.

The numerical requirements cascade starts by determining the combustion parameters in order to meet output power requirements, and proceeds through the ancillary engine systems, finishing with the control system. In Tier 4 Interim the requirements cascade was carried out by the NPI team in combination with the CAE team's Systems Engineering group (headed by Engineer 8). While most CAE tools analyse individual components, this group within CAE has developed parametric mathematical models that can predict the effect of 20-25 key component parameters on the performance of not only the engine as a whole but also the machine in which it is installed. The models consider mathematical relations between performance parameters and neglect most geometrical considerations. This geometric naivety was viewed as useful, since awareness of packaging and other constraints would have made the systems engineering team overly pessimistic about suggesting new technologies. Nonetheless, the spatial layout of an engine does constrain performance, since required performance parameters indirectly determine component geometry. In order to ensure that feasible component performance targets are set during the cascade, the design architect and other members of the platform team develop the physical design of the engine in parallel with the CAE team's performance modelling. Interaction between them is intermittent but not continuous: the platform team gives geometry-derived constraints on parameter values to the systems engineering team, who in return feed back performance-derived constraints and in some circumstances the best choice of technology. The systems engineering models also allow frequent verification of the design against the requirements during the later design process.

5.5 Detailed design

Detailed design of the components begins when the architect issues design briefs to the CPPD teams between Gateway 2 and Gateway 3. These summary documents indicate the component to be designed, its specifications (through reference to the relevant Product and Technical Objectives), and (some of) the other components (and therefore other CPPD teams) with which the design must interface. According to Engineer 2 several versions of the design brief were written: an initial request for an exploration of options, progressing to a final specification for design work to be carried out. Engineer 3, however, had a very different view, describing the CPPD teams as like "a factory, or a sausage machine that, you put a definite input in, a very sort of formal input, and you get a very formal, quality product out the other end"; he claimed that "they just don't have the capacity to look at [...] what ifs". Such differences may be due to the engineers' different roles in the design process. The biggest difficulty with writing design briefs, which Engineer 2 claimed his method avoided, is writing a component specification without necessarily knowing the important parameters. Indeed, one CPPD team leader informally stated that he sometimes wrote design briefs himself, since he understood what the specification should contain, and merely asked for approval from the architect,.

Once the CPPD teams are involved, the role of the architect shifts to coordination and arbitration while the designing itself is carried out between the CPPD teams and the appropriate CAE personnel: "CPPD draws [a component] then gives it to CAE to make it work" (Engineer 1). Overconstrained specifications and product complexity, together with the restricted interaction between CPPD teams, means that architect-mediated negotiation is frequently required, both between the CPPD teams and with the engineers responsible for performance, noise, vibration etc. In such situations the architect's overview knowledge allows them to make informed tradeoff decisions: "there is an element where, with the [CPPD teams] that, if it's something completely outside their sphere of knowledge, they just take it as a requirement" (Engineer 3). Such interface tradeoffs are made using Pugh analysis, QFD, or simply by establishing the facts and debating possible solutions; in this context possible solutions are often drawn from the engineers' past experience of their own and competitors' products. Continual negotiation leads to shifting of the interfaces from those specified after the requirements cascade.

Although the design briefs are the key to the detailed design phase, since they are verbal/numerical they only supply a part of the information needed to complete the design. In order to design the geometry of the components, many assumptions must be made about spatial layout of individual parts and features on those parts. According to Engineer 3, the source for the majority of these assumptions is a "shared vision" of an engine. This incorporates the arrangement of components, their behaviour, and their interfaces with other parts of the design. He described a particular incident where the shared vision featured, when it was necessary to produce a new bracket to support the aftertreatment components: "there was some discomfort that [the CPPD team designers] really wanted to see what it was going to look like, they wanted the design brief to say something about [...] not just what it should do but, what it should [look like], because it was just such a new component".

6 DISCUSSION

6.1 Numerical and geometric design

The product is represented in two different ways as it is designed. Initially all formal design documentation is verbal or numerical, from the Product Objectives, throughout the requirements cascade, up to and including the design briefs. Geometrical information is only included as envelope dimensions, and not in diagrammatic form, although the geometry begins to be worked out by the design architect during the requirements cascade, as described in Section 5.4; in effect, two parallel requirements cascades were carried out by the packaging and performance design teams, with constraints arising from one domain informing work in the other. Only at the detailed design stage, after Gateway 2, is geometrical information produced as formal 3D models. Even then, the design must be verified against the verbal/numerical Product Objectives. The engineers involved state that they can visualise geometry based on such descriptions; however, the spatial layout element of product architecture is only decided in the final transition from verbal to geometric representation.

6.2 Role of product architecture in the requirements cascade

The design of an engine is overconstrained (like that of many engineering products [17]): it is not possible, or practical, to use all the requirements on a product to derive its design. Instead, the requirements cascade simplifies design by effectively neglecting geometric considerations and implicitly prioritising requirements: a small number of key, usually performance-related requirements (e.g. output power) are used to synthesise an initial set of component parameter values. This converts an overconstrained problem into an underconstrained one, allowing freedom to adapt a previous design to new requirements. The design is then evaluated against the full set of requirements, including higher-fidelity evaluation of the synthesis requirements, and modified as necessary.

The mathematical relationships between higher- and lower-level requirements that allow this cascade to take place, whether historical correlations or derived from physical principles, are set by the product architecture. Each instance of a technology, an "organ" in the sense of [5], has a set of associated

design variables and may span multiple components. The technology used, element 2 of the definition of architecture in Section 2, defines what the relationship between those variables will be; how the organs are interconnected, element 4 of the definition, then determines links between the values of variables in different relationships.
From this point of view, the company's engine architecture is largely fixed; in fact, the "shared vision" of the engine described in Section 5.5 may be viewed as an implicit generic product architecture. The shared vision therefore plays a crucial role in design, through unconsciously carrying-over parts of the product architecture from one engine to the next without necessarily subjecting them to detailed analysis. Conservation of the architecture of the core engine is particularly marked, due to the "investment" in both capital terms and in the business processes that are set up based on it.
The synthesis requirement/evaluation requirement distinction also sheds light on new technology introduction and reasons for design reuse. If the requirements on an engine change, the design may need to change to meet them. In the case study the NPI cycle is governed by emissions requirements because of both the importance of meeting them and the speed at which they change, while other performance measures (reliability, fuel consumption, size etc.) change more slowly. However, there are typically many values of component parameters that would meet the synthesis requirements within a fixed architecture; reuse helps constrain this underconstrained problem, and also provides confidence that at least some of the evaluation requirements (particularly geometric feasibility) will be met. This inherent flexibility in meeting synthesis requirements also explains why changes may be transferred to different engine systems. NTI activities, aimed at bringing technologies to a stage where they can be used during a design process, equate to making explicit the design parameters of the technology at the component level and how these relate to the synthesis requirements.

6.3 Comparison with the "textbook" process

There are two striking differences between the way product architecture is designed in practice and the "textbook" process described in Section 2. The most fundamental is that product architecture as defined in Section 2 arises over a long period, rather than in a single step. Practical considerations make a concise architecture design activity impossible: it would be necessary to know at the outset the final requirements, both qualitative (which functions the engine should perform) and quantitative (performance requirements), and to be able to model all aspects of all potential technologies that could be used in the engine. Although the company has made progress towards the latter through the CAE Systems Engineering group's activities, such modelling cannot capture every aspect of a design, especially when there is uncertainty about the underlying technology or the customers' requirements (changes to which are brought in through the Integration Workshops). In a sense, the architecture is only finalised once design is complete – the Section 2 definition corresponds to a "summary" rather than a "plan" for the design. The textbook process recognises that such changes may occur through iteration loops in the flowchart, but they are viewed as exceptional. Potentially, a more helpful definition of architecture might identify that which does remain constant through the NPI process; however, almost all aspects of the design were changeable, apart from the core function (converting chemical energy to rotary mechanical output) and technology (diesel internal combustion engine).
The other major divergence from theory observed in the case study is the reuse of previous designs. As stated in Section 2, many authors stress the importance of "original design" – of starting from first principles and actively rejecting preconceived notions of what a design should be. However, Section 5.1 describes six pragmatic reasons why the company finds it necessary to carry over design elements. In such circumstances it is simply not feasible to ignore what has gone before. For comparison with the "Systematic Design" process shown in the left part of Figure 1, the actual design process followed by the company was abstracted into a similar flowchart and is shown diagrammatically in the right hand part of Figure 1. This emphasises particularly the importance of the design project's starting point and the need to trade this off against the task to be performed. Such differences from theory may be common within industry practice. However, if research is to aid such industry practice, it is important to recognise the effects of these differences and acknowledge them in proposing methods for improving design.

7 CONCLUSIONS

In order to understand how product architectures are designed, a series of 13 interviews was carried out with 8 engineers from a UK-based manufacturer of off-highway diesel engines. In the course of these interviews, an insight was gained into the company's view of product architecture, its NPI process (consisting of eight gateways, outlined in Figure 2) and organisation, and how internal and external factors interact in the course of designing a new product.

In particular, answers to the research questions outlined in Section 3.1 are as follows:

1. **Definition of product architecture:** In the interviews no formal definition was given. Section 4.1 interprets the literature-based definition from Section 2 for the specific case of a diesel engine.
2. **Effect of external factors:** These are described in Section 4.2. The key factors are the emissions legislation, which requires new introduction of new technologies into the engine (modifying its architecture), and the diverse range of customers, who require stability in the engine's specifications but flexibility in terms of engine configuration options.
3. **Design process and organisation:** These are described in Section 5, Figure 1 and Figure 2. The key phases of design are the requirements cascade, where overall product requirements are translated to component-level requirements (which requires decisions about embodiment to be made); and detailed design, where problems in meeting component-level requirements can lead to negotiation and changes to the design.
4. **Emergence of product architecture during design:** In contrast to the "textbook" approach to product architecture design, there is no single step early in the NPI process where product architecture is designed through evaluation of alternatives. Rather, the product architecture as defined in Section 2 arises over an extended period in the early part of the NPI process, as shown in Figure 3; the spatial layout element of product architecture is only decided in the final transition from verbal to geometric representation, as discussed in Section 6.1. Design is dominated by minimisation of novelty leading to reuse of significant sections of the architecture, which may be whole components or individual "organs". This reuse is both deliberate, due to the "investment" in both capital and process structures described in Section 5.1, and also because of the engineers' implicit "shared vision" of the generalised architecture of a diesel engine. However, such reuse can also lead to inability to meet requirements, leading to negotiation of interfaces. Sources of reuse include both the company's own engines and competitors' products. If part of an earlier architecture cannot be reused, new concepts may come from an NTI programme or from the designers' own creativity. The effort put into designing a particular section of architecture (in terms of the number of alternatives considered, the number of designers involved and the level of detail of the analysis) depends on the impact of that section on the design.
5. **Effect of product architecture on design process:** Product architecture affects later design in a variety of ways, as discussed in Section 2. The technologies used in the engine (i.e. the function-component mapping) determine the design tasks that must be performed and the expertise required to do so. In the case study, new design teams for electronic control systems and aftertreatment were put in place at the level of the parent group rather than the company itself. The interconnections of components then determine how changes propagate through a design if modifications need to be made to a component. Technology choice and component interconnections also determine how the requirements cascade behaves, as described in Section 6.3: design synthesis is driven by considering a small number of "synthesis" requirements, after which the design is evaluated against many more "evaluation" requirements. Design reuse provides a way to apply additional constraints while guaranteeing fulfilment of other requirements. New Technology Introduction can be viewed as the process of making explicit the relationships between the objectives used in synthesis and the design variables of the component or system in question. Finally, manufacturing considers each component separately, while performance relies on interactions between multiple features on different components. Therefore, problems may result if components' manufacturing lead times do not match up with the sequence in which the engine's performance interactions are designed [18].

Although the interviews were opportunistic, interviewees spanned most stakeholders in NPI and agreed on most points. In order to establish whether the insights are in fact general characteristics of product architecture design in this domain, the case study will be triangulated through further interviews in similar companies. Additionally, the findings presented here will be used in the development of computational methods for product architecture synthesis, in order to assist designers in carrying out this important activity.

ACKNOWLEDGEMENTS
The authors would like to thank all employees of the case study company who participated in the research, and an anonymous reviewer for constructive suggestions. This research was funded by the George and Lilian Schiff Studentship and the UK EPSRC (grant reference EP/E001777/1).

REFERENCES
[1] Ulrich K. The role of product architecture in the manufacturing firm. Research Policy, 1995, 24(3), pp. 419-440.
[2] Pahl G. and Beitz W. Engineering Design: A Systematic Approach, 1996 (Springer, London).
[3] Stone R.B., Wood K.L. and Crawford R.H. A heuristic method to identify modules from a functional description of a product. In Proceedings of the ASME Design Engineering Technical Conference (DETC98), 1998.
[4] Browning T.R. Applying the design structure matrix to system decomposition and integration problems: a review and new directions. IEEE Transactions on Engineering Management, 2001, 48(3), pp. 292-306.
[5] Andreasen M.M. Designing on a 'designer's workbench' (DWB). In Proceedings of the 9th WDK Workshop, Rigi, Switzerland, 1992.
[6] Maier M.W. and Rechtin E. The art of systems architecting, 2000 (CRC Press, Florida, USA).
[7] Berliner C. and Brimson J.A. Cost management for today's advanced manufacturing: The CAM-I conceptual design, 1988 (Harvard Business School Press, Boston, Massachusetts, USA).
[8] Ulrich K.T. and Pearson S.A. Does product design really determine 80% of manufacturing cost? Massachusetts Institute of Technology Sloan School of Management Working Paper #3601-93, 1993.
[9] Yassine A.A. and Wissmann L.A. The implications of product architecture on the firm. Systems Engineering, 2007, 10(2), pp. 118-137.
[10] Crawley E., de Weck O., Eppinger S., Magee C., Moses J., Seering W., Schindall J., Wallace D. and Whitney D. The influence of architecture in engineering systems. Massachusetts Institute of Technology Engineering Systems Monograph, 2004.
[11] Blessing L.T.M., Chakrabarti A. and Wallace K.M. A design research methodology. In Proceedings of the International Conference on Engineering Design (ICED'95), 1995, pp. 50-55.
[12] Eckert C.M., Clarkson P.J. and Zanker W. Change and customisation in complex engineering domains. Research in Engineering Design, 2004, 15(1), pp. 1-21.
[13] Purcell A.T. and Gero J.S. Design and other types of fixation. Design Studies, 1996, 17, pp. 363-383.
[14] Wyatt D.F., Wynn D.C. and Clarkson P.J. A computational method to support product architecture design. In Proceedings of the ASME International Mechanical Engineering Congress and Expo (IMECE 2009), Lake Buena Vista, Florida, USA, November 13-19 2009 (submitted).
[15] Sanchez R. Modular architectures, knowledge assets, and organizational learning: New management processes for product creation. International Journal of Technology Management, 2000, 19(6), pp. 610-629.
[16] Jarratt T.A.W., Eckert C.M., Weeks R. and Clarkson P.J. Environmental legislation as a driver of design. In Proceedings of the International Conference on Engineering Design (ICED'03), Stockholm, Sweden, August 19-21 2003.
[17] Stacey M.K. and Eckert C.M. Reshaping the Box: Creative designing as constraint management. International Journal of Product Development, 2009 (submitted).
[18] Eger T., Eckert C.M. and Clarkson P.J. The role of design freeze in product development. In Proceedings of the International Conference on Engineering Design (ICED'05), Melbourne, Australia, 2005, pp. 164-165.

Contact: David F. Wyatt, Department of Engineering, University of Cambridge
Trumpington Street, Cambridge CB2 1PZ, United Kingdom
Tel: +44 1223 748 564; Fax: +44 1223 332 662; E-mail: dw274@eng.cam.ac.uk

David Wyatt is a Ph.D. student at the Cambridge Engineering Design Centre. As an undergraduate he studied Engineering at Cambridge, specialising in Information and Computer Engineering.