BRIDGING THE PRODUCT CONFIGURATION GAP BETWEEN PLM AND ERP – AN AUTOMOTIVE CASE STUDY

JOY BATCHELOR
Technical Specialist Product Configuration, Jaguar Land Rover, Banbury Road, Gaydon, CV35 0RR, UK. E-mail jbatche4@jaguarlandrover.com

HENRIK REIF ANDERSEN
CEO Professor Founder, Configit A/S, Kristianiagade 7, DK-2100 Copenhagen, Denmark. E-mail hra@configit.com


ABSTRACT
The development and use of product configurators has been highlighted as one of the more important IT enabled technologies to facilitate the realisation of mass customisation. However, the integration of these technologies into day to day business operations is complicated by the proliferation of configuration capabilities across a number of types of IT applications. This paper seeks to address both the theoretical and the practical challenges of product configuration in an assemble-to-order environment via an automotive case study. The purpose of the paper is to highlight a number of configuration gaps and propose an integrated approach for the consolidation of configuration across a number of business domains.

We set the context for product configuration before we move on to describe our research approach and the automotive case study. We conclude the case study with a summary of the overall requirements for a successful system to support the product configuration needs throughout the extended enterprise. We call this new area of enterprise software Configuration Lifecycle Management (CLM). Finally, we move on to explain the business benefits and the management implications of implementing a CLM solution.

INTRODUCTION
There is well established literature around mass customisation and the application of configuration technology (Blecker and Friedrich, 2007; Forza and Salvador, 2007) as a means of connecting customers to personalised products and services. Mass customisation tries to achieve both the cost efficiency of mass production from a manufacturing point of view and the desire to meet customer demands with flexible and customisable product offerings. For many goods, in particular high-valued products such as vehicles, the trend over the last three decades has been to offer ever more choice in the offerings to customers. As a consequence, almost every business process from marketing, development, manufacturing through to the delivery of such configurable products needs to be adapted to ensure both accuracy and efficiency. Increasingly, as products become more complex a key enabler for achieving this tailoring of products to customer needs is the application of computer aided configuration technologies. Such technologies enable the efficient and accurate definition of allowable configurations thereby supporting the demand for customisation and personalisation.
This paper is not about the personalisation of products per se, rather it is about the ability to define a valid configuration of a product which may be specific to a particular channel and the barriers encountered in providing multiple views of a configured product over the complete product lifecycle via the use of configuration technologies. Which variants are valid for a specific customer depends upon a range of parameters including the customer’s market and the local regulations and conformance conditions in that market. The exact time for the delivery of the product will also influence what options are available, as will the overall business decisions made from a profit perspective on what will be offered to what customers on a market by market basis. These are all aspects that complicate the definition and administration of the possible and allowable configurations of products made available for customers to order.

A further complication arises from the need to have control over the effectivity of key elements of the product definition. In the automotive industry, product offerings are always under change. Major changes are grouped into model years of a car line, which typically involves changes to all major sub-systems of the vehicles. Minor changes can occur at any time and can be due to upgrading of some of the sub-systems for various commercial reasons. All changes need to be reflected in all the key processes involved in the automotive business; developing, pricing, selling, marketing, building and the servicing of vehicles. In order to get alignment between all these processes, the exact timing of the effect of a change is of key importance. Any system that is to support the business must, therefore, have full control over the effectivity points and their interactions.

This paper seeks to address both the theoretical and practical challenges of product configuration set in the context of an assemble-to-order environment. Businesses that employ an assemble-to-order approach are based upon line assembly techniques and include the manufacture of cars, trucks, aircraft, computers and high volume machine tools to name a few. An extended case study is presented of an automotive OEM which has concurrently embarked upon the introduction of a new Product Lifecycle Management (PLM) application and a new Enterprise Resource Planning (ERP) application. The concurrent replacement of such enterprise IT applications is highly unusual in a large industrial company, however, this has provided a unique backdrop within which to examine the opportunities for the rationalisation and optimisation of product configuration capabilities to support the mass customisation of products throughout the product lifecycle.

This paper is organised into four sections. Firstly we set the context for product configuration before we move on to describe our research approach and the automotive case study. Secondly, we conclude the case study with a summary of the overall requirements for a successful system to support the product configuration needs throughout the extended enterprise. Thirdly, we call this new area of enterprise software Configuration Lifecycle Management (CLM). Finally, we move on to explain the business benefits and the management implications of implementing a CLM solution.

The contribution of this paper to theory and practice centres upon:

- The designation of a master system for the authoring of product models and configuration.
- Rationalisation of structure and content to facilitate enterprise integration of PLM and ERP.
- Opportunities for lifetime modelling to supporting the evolution of a product over time.

**SETTING THE CONTEXT FOR PRODUCT CONFIGURATION**

At the simplest level product configuration is an interactive process in the selling of a vehicle whereby a customer chooses a value or an option, which is then validated via a configuration engine before allowing the customer to make the next choice about the remaining variables to be configured. This process of choice followed by validation continues until all choices have been made and a complete and valid configuration has been found. The apparent simplicity of this process hides a significant amount of complexity to construct the rules used to drive the configuration process and it does not reveal all the processes involved in running a business with customisable products. In fact product configuration is an essential part of almost all business processes which we will categorise in seven overall areas, which underneath contain a lot more processes. They are illustrated in Figure 1.

![Figure 1. The seven main business processes in developing customisable products.](image)

Configuration is involved in all seven business processes and the text in the figure should be read as "develop products", "price products", "market products", etc. We will briefly describe the seven areas and how product configuration is an essential part of each of them.

- **Develop.** During the development of new products, decisions, have to be made on how to modularise the product so that it will be amenable to customisation. The allowable variants and options must be defined so that the necessary dependent engineering, sourcing and other business processes can be initiated. In this phase the allowable combinations forming the configuration space are being defined. This involves defining legal and technical constraints representing, for example, the permitted radio frequencies for wireless door locking in specific markets. It is important to understand the consequences for the downstream processes and
profitability of the decisions taken on the offered options. This involves analysing the space of allowed configurations.

- **Price.** Pricing is a key activity for any business. It involves analysing the cost and the market acceptable sales prices and from these deducing the expected profit margins. With customisable products this is grossly complicated because the calculations have to be made over the full space of possible configurations and not just on a list of predetermined products.

- **Market.** When marketing a customisable product on a global scale there will be differences in the offering across the geographical markets. The differences arise as a consequence of local regulations and varying customer preferences. It is necessary to ensure that the offered configurations match the needs. Sales and Marketing departments often express the mapping of offerings to markets using large availability matrices which, in essence, are configuration rules.

- **Sell.** The sales process involves the key operation of deciding on a specific configuration meeting the customer requirements. A customer will be faced with a lot of choices and it is important to ensure accurate guidance when making the selections so that the end result will be a satisfactory product at an acceptable price. The result of the sales process is a valid order to be fed into the downstream systems.

- **Source.** Manufacturing of complex products involves using parts and assemblies from sub-suppliers. When the overall product is configurable the parts and assemblies vary according to the individual configurations. It is, therefore, important to ensure that the sourcing of parts is such that correctly configured sub-assemblies are delivered in the right sequence for the production line. Configurations must, therefore, be communicated correctly to the suppliers. Other constraints to be considered are the suppliers’ yearly or monthly capacity that must be taken into account in the production planning.

- **Build.** The building of the product is defined by the configuration. Each product will have its own defining configuration that is used for selecting the correct parts to be mounted. An error in a configuration is very costly if it makes its way down to the factory floor. The result can be that the end product will be useless or that the production line must stop for a period of time while the problem is overcome. Other constraints to be considered are build constraints on the production line reflecting known line balancing issues and line side feeding constraints.

- **Maintain.** After a product is delivered it will typically go through service and maintenance where these activities are also dependent upon the configuration of the product. The configuration of the product can also be changed over time by upgrading sub-systems.

To focus on the core aspects of configuration, for the purpose of this paper, product configuration is the process through which product features (which represent both technical constraints and allowable customer choices) are modelled in a configuration application. A product model is developed which consists of a set of parameters called feature families, a set of possible values for the parameters called features, and rules describing dependencies among the features. Figure 2 presents a high level overview of the ideal product modelling process.
A product model will be created and will consist of rules to control the technical parameters of the product as well as rules that control the commercial presentation of the product. A product model is then used by a configuration engine to assist in generating a configured product. Product configuration management is, therefore, the process through which a full or partial product configuration is solved via a configuration engine to generate an analytical, a digital or a physical representation of a product.

The engineering intent of a product sets the parameters of the features available to marketing and sales by which to configure a marketed product. All legal, quasi-legal, technical, engineering and licensing constraints are encoded so that Marketing and Sales cannot overwrite these product rules and Manufacturing cannot receive orders for infeasible or unbuildable feature combinations. Examples of restricted engineering rules include all features subject to third party licensing arrangements, e.g. Bluetooth and legally required features such as day light running lights. Marketing intent sets the commercial parameters which relate to the way in which the product is to be marketed. Marketed intent is a subset of the engineering intent and working within the defined engineering capability of a product, Marketing can then configure and group features in the most appropriate manner for the market, region, trim level etc.

The development of the product models used to drive the configuration process is a highly complex process and can be divided into two distinct phases; namely the development of the models in line with the product development process, and then the maintenance of the models once the product enters production. As a consequence of these two distinct phases our experience indicates that in practice it is often difficult to designate a master system for the authoring of product models for the purpose of configuration.

Product information linked to the product development process increasingly resides within PLM systems which are design centric whilst product information used to drive transactional processes typically reside within ERP systems. As a consequence of this a configuration gap can emerge due to the duplication of configuration capabilities within
these different types of enterprise applications; a point that we shall return to in the next section.

**Evolution of Configuration Technologies**

The use of product configurators has been highlighted as one of the more important systems for streamlining the response to customers within a mass customisation environment (Trentin, *et al.*, 2011; Arana *et al*., 2007; Skjevdal and Idsoe, 2005). The evolution of configuration technologies has taken place in three waves. The first wave dates back to the 1980s where software based on ideas of *truth-maintenance systems* and the use of the Rete algorithm emerged (Doyle 1979; Forgy 1974; Forgy 1979; Forgy 1982). These first generation applications were rule-based configurators, which represented the product using a set of "If-Then" statements. The first reported system was Digital Equipment Corporation’s XCON developed to support the configuration of the increasingly complicated PDP computers (Sviokla 1990). The Rete algorithm is still used in a range of commercial systems including SAP’s Variant Configurator module.

The second wave of configuration technologies dates back to the mid 1990s, when a range of factors including the growing capacity of computers, advanced programming techniques and studies into artificial intelligence paved the way for advances in configuration technology with the emergence of *constraint based configurators* (Forza and Salvador, 2007:47; Sabin and Weigel, 1998). In these systems relationships are described as declarative constraints that should always hold (CIMdata, 2006). These types of configurators can handle both "If-Then" rules and capacity constraints. Compatibility constraints can include tabular or multidimensional relationships which must either hold or are forbidden. For example, "you can have A and B and C, but you can't have A with D, or B with E and you must have at least three of A, B, C, D, and E". This approach is currently the dominating technology in commercially available product configurators.

The third wave of configuration technologies are *compilation-based configurators* where maintenance and usability is largely improved through the use of a compilation step in which the full space of valid configurations is constructed for later efficient runtime behaviour (Møller *et al*., 2001; Subbarayan *et al*., 2004; Subbarayan, 2005). There is no standardised approach to the functionality of configurators and various classifications have been used to describe the different types of configurators currently available. The Aberdeen Group (2008) segmented configurators into three broad categories:

- **Sales Configurators** – used to define compatibility rules to enable pricing and the creation of quotes and orders.
- **Product Configurators** – used to define compatibility rules and configure the product (typically a bill of material, CAD model, and drawings).
- **Manufacturing Configurators** – used to define compatibility rules to drive manufacturing processes, scheduling, and resource allocation.

Mesihovic and Malmqvist (2000:1) also reasoned that the definition of product configuration be extended to support the configuration of products containing pre-defined, parametric and modifiable components to account for two types of configuration strategies, namely:
• **Assemble-to-Order Configurators** – used to support assemble-to-order processes

• **Engineer-to-Order Configurators** – used to support engineer-to-order processes

In June 2011 Gartner published a report on the size of the configuration, price and quotation applications (CPQ) market sector noting that:

"The size of the CPQ application suite market is still modest, with worldwide revenue estimated at approximately $200 million in 2010, an increase of approximately 20% from estimates of more than $150 million for 2009. These estimates include licenses, subscription fees, professional services fees, and maintenance and support revenue generated by the software vendors; however, they exclude sales of product configuration systems for ERP, manufacturing and fulfilment processes" (Gartner, 2011:4).

Based upon these figures alone it would be easy to underestimate the impact and growing importance of configuration technologies within producers of complex configurable products in particular. Whilst technologies to address the problem of configuration continue to evolve, the integration of these technologies into day to day business operations is complicated by the fact that product configuration is not a process in itself. Product configuration is a step within a number of other business processes such as the product design process, the sales process and the order fulfilment process. As a consequence of this, configuration capabilities can be realised via a range of specialist configuration applications. Often these are focused upon a particular phase of the configuration lifecycle, or configuration capabilities can be embedded into enterprise focused applications such as PLM, ERP and CRM systems.

Two key IT technologies, Enterprise Resource Planning (ERP) and Product Lifecycle Management (PLM), have emerged to drive business efficiencies within product development and manufacturing over the last 20 years. Each technology brings unique value to the enterprise, and when combined, ERP and PLM provide the ability to allow direct sharing of engineering and manufacturing data through automated processes. Understanding the core functionality underpinning both ERP and PLM systems is increasingly important as the boundaries between these applications begin to blur as the vendors of these systems merge functionality at key touch points. For example, all major ERP applications such as SAP, Baan and J.D. Edwards have a configuration capability in their standard application (Trentin et al., 2011:261), similarly both Dassault's Enovia PLM application and Siemens PLM solution include configuration components.

The challenge facing companies would appear to be the navigation of the myriad of applications and the avoidance of unnecessary duplication of configuration activities, contrary to the assertion that configurators centralise configuration knowledge, maintenance and management (Sviokla, 1990). The reality for many companies is that the integration of these different types of configurators is problematic. Over a decade ago Timmermans (1999) noted that integration across different configuration domains would be difficult due to the differences in the configuration paradigms employed by the different configuration technologies to solve particular types of configuration problem. AMR (2009:2) highlighted three barriers to the deployment of configuration technologies, namely:

• **A lack of technology**: the difficulty of finding a solution that meets the needs of a company.
• A lack of resources: The time and maintenance associated with the creation of product models and their rules across the engineering and marketing domains.

• The cost and complexity of integration: The complexity associated with managing and synchronising configuration master data across consuming applications such as CRM, ERP, CAD and PLM.

The integration landscape is further complicated where companies have developed their own in-house configuration technologies to meet their specialist configuration requirements. A study conducted by AMR noted that "in-house developed applications are still used by 33% of companies for some or all of the configuration process" (2009:2). To further illustrate this point Siemens conducted an internal review of the use of configurators across their business units (Falkner and Haselböck, 2012). In total 96 different configurators were identified, 67 of which were developed in-house by different business units of Siemens. Of the remaining applications bought off the shelf, 27 different vendors were identified as supplying configuration technologies to Siemens.

RESEARCH APPROACH: AN AUTOMOTIVE CASE STUDY OF JAGUAR LAND ROVER

Jaguar Land Rover (JLR), owned by Tata Motors since June 2008, is a premium vehicle manufacturer selling approximately 300,000 vehicles per year; in 150 plus markets around the world. Jaguar produces high performance sports cars and saloons (XK, XF and XJ), whilst Land Rover produces class leading 4-wheel drive vehicles (Defender, Discovery 4, Freelander 2, Range Rover Evoque, Range Rover Sport and Range Rover). Research and design activities are undertaken in the UK at two development sites whilst manufacturing is undertaken at 3 UK based plants with overseas assembly occurring in India and a planned joint venture in China is underway. Whilst the impact of the recession, which started in 2008 was severe, JLR's recovery, like the rest of the premium automotive sector, has been driven by strong demand in the developing BRIC markets.

JLR has concurrently embarked upon the introduction of both a new PLM application and a new ERP application. The concurrent introduction of such enterprise IT applications is highly unusual, however, this has provided a unique backdrop within which to examine the opportunities for the rationalisation and optimisation of product configuration capabilities to support the mass customisation of products throughout the product lifecycle. Specifically this case study seeks to identify those factors which enable the adoption of a single configuration management approach to support the enterprise over the complete product lifecycle using a single version of configuration truth.

A case study approach is therefore justified as it enables analysis of the dynamic factors which, whilst present within a single setting (Eisenhardt, 1989:534), may give indications for new avenues for future research. In this context “qualitative observation” identifies the presence or absence of something, in contrast to “quantitative observation” which involves measuring the degree to which some feature is present (Kirk and Miller, 1986:9).

Jaguar Land Rover’s varied ownership heritage has resulted in a highly challenging IT landscape. Following ownership by a number of companies including BMW and then Ford, JLR has inherited a large portfolio of applications as a consequence of various IT integration activities. From a product definition and configuration perspective information is authored and duplicated across many systems leading to wasted effort and
the increased risk of inaccurate information being published to drive configuration activities.

JLR has a 20 year history in the use and development of configuration management capabilities. Both of JLR’s existing product definition authoring tools originated in the early to mid-1990s and are built upon an object oriented approach. Both use Boolean constructs meaning that they can handle "If-Then" rules and they both have a constraints capability in that they can handle tabular or multidimensional relationships which either must hold or are forbidden. One system was inherited from Ford and is a commercial application that was popular in the late 1990s and early 2000s and this application is used to author the engineered intent.

The second application was developed in the early 1990s by Land Rover (then a part of Rover Group). Three IT applications were concurrently developed in-house; a product definition application, a bill of material system and an order scheduling system. Each system was designed to operate with one another in support of a drive to improve efficiencies to enable the company to operate as an assemble-to-order manufacturer. Whilst the bill of material system was replaced several years ago under the ownership of Ford, the product definition application used to define the marketed intent of JLR’s vehicles and the order scheduling system remain in the IT landscape today.

JLR are not the only automotive company to have a history in the development of configuration technologies. Mercedes have been engaged in research over the last 10 years (Sinz et al. 2003; Kübler et al. 2010), whilst Volvo Trucks and Fiat participated in the European funded CATER Research Project. Helo et al., 2010 present the piloting work undertaken within Volvo Trucks via the CATER research where the front-end of the configurator system acts as an intelligent decision making mechanism which is capable of suggesting best-matched product models based on preliminary customer preferences (Helo et. al. 2010: 8).

Having an in-depth knowledge of configuration technology and the experience of using this capability in an integrated IT environment caused JLR to ask the question “could it be possible to introduce a single rule authoring and configuration management application to serve the needs of the enterprise?” The ongoing evaluation of PLM applications in 2009, combined with the evaluation of a multi-phased deployment of an ERP application presented an opportunity to undertake a significant technical evaluation of the market for configuration applications. This technical evaluation finally resulted in the selection of a configuration development partner. Presented and discussed here are some of the key conclusions of the technical evaluation which highlights what we believe are areas for further research and development. The technical evaluation of the market covered a 20 month period between September 2009 and May 2011 comprising three distinct phases:

1. The desk top market evaluation exercise conducted from September 2009 to January 2010.
2. The Request for Information phase conducted from February 2010 to July 2010.

The initial desk top market evaluation exercise identified 35 specialist configuration applications, or best of breed companies, available on the market which satisfied a high level set of technical criteria. The 35 applications were identified via an extensive internet
based search and review of published reviews and technical white papers on configuration applications/technologies by the likes of Gartner, Aberdeen Group and AMR. The websites of the vendors were used to identify the nature and scope of the functionality of the applications and reference was made to any application specifications as well as other documentation relating to the configuration technology underpinning the applications. Excluded from this list of specialist configuration applications were PLM and ERP applications which contained in-built configuration capabilities.

From this list of 35 companies, 12 were judged to cover the range of technical capabilities which we wanted to review in detail and these companies were invited to undertake a Request for Information (RFI) exercise. In addition to these specialist configuration companies, two market leading enterprise application companies, one providing a PLM solution and one an ERP solution, were also invited to participate. In total 9 companies, including the PLM and ERP companies, chose to participate in the RFI review. A detailed set of technical criteria was developed and used to evaluate the configuration capabilities of the companies participating in the study. In addition to this a detailed practical use case was developed and supplied to each company to model in their applications.

The use case demonstrated a range of configuration concepts to be modelled which are representative of the complexity of JLR’s vehicles. Table 1 summarises the high level configuration concepts tested for. Upon completing the RFI phase three companies were invited to participate in the RFQ phase. During this next phase each company participated in a second series of in-depth technical evaluations and the use case was refined to demonstrate a range of more sophisticated modelling concepts, building upon the existing data.

In their own right the PLM and the ERP applications contained a degree of inbuilt product modelling and configuration capability; however, two conclusions were drawn. Firstly, it was found that the breadth of the product modelling capabilities of these types of applications were restricted and less capable than the existing technologies employed within JLR and the specialist applications. The rule authoring capabilities of both applications were only at the level of truth-maintenance systems, meaning that complex modelling constructs would be difficult to support. It was also found that both applications adopted a hierarchical tree structure for the development of product models on a vehicle line by vehicle line basis with no ability to concurrently maintain multiple model years. Product model maintenance in both approaches would, therefore, be highly complex and restricted to a model year based approach to product model maintenance contrary to a configuration approach based upon the use of effectivity.

Secondly, the scope of the configuration capabilities of both applications was found to be restricted. The configuration functionality embedded within the ERP system is designed to support the front-end sales and order processes but is not suited to the back office requirements of Engineering. Equally, the configuration capabilities of the PLM application, due to its PDM origins, are focused towards CAD, prototyping, visualisation etc. as opposed to supporting the demands of front-end sales and order management. The configuration capabilities of these applications were therefore judged to be unsuited to supporting the lifecycle configuration needs of complex products from concept to run-out via a single version of configuration truth.
Table 1. High Level Configuration Concepts

<table>
<thead>
<tr>
<th>Configuration Concepts Tested</th>
<th>Configuration Principles</th>
</tr>
</thead>
</table>
| 1. Support for effectivity | • Support for effectivity based configuration throughout the product lifecycle.  
• Effectivity is a time point or set of points which define the availability period of an object e.g. a feature, a feature rule, a feature specification etc in a product. |
| 2. Support for concurrent build periods | • Differentiation between production builds and prototype builds using effectivity.  
• Releases managed around timing points and change points, with multiple programmes of work being managed independently. |
| 3. Support of an integrated modelling environment | • Support for both perpetual and model year based releasing in the product modelling environment. |
| 4. Support for rule inheritance | • Creation of marketed intent will be defined within the bounds defined by the engineered intent. |
| 5. Support for the authoring of cross-carline rules | • Enable modular product modelling, for example, homologation rules and certain architectural technical constraints apply across multiple car-lines. |

Due to the complexity of JLR’s premium products, perpetual product models are used for modelling purposes and effectivity is used to create “virtual” model years for the purpose of releasing marketing intent to the sales systems. Information such as product and feature descriptions, price structure and feature visibility are associated at the model year variant level. A purely model year approach to product modelling is unsuited to a vehicle manufacturer with the complexity of products sold in the number of markets such as those of JLR. At any one time a vehicle line may have up to three model years to be managed concurrently; one model year in production and the subsequent models year for the next two model years in development. It should be noted that Ford North America does model on a model year basis, whilst Ford of Europe shares the same modelling philosophy as JLR due to its model and market complexity.

The concept of effective points is used to manage updates and control when changes are released into the production environment. The use of a perpetual view of a product in the modelling environment enables feature rules to flow through the model years and be inherited where required, therefore, meaning less maintenance or errors due to re-authoring. Whilst features are affected in and out as required over the life of a vehicle many features, mainly technical constraints, will remain unchanged and will cover multiple marketing model years. The best of breed configuration applications studied do not currently have a sophisticated approach towards effectivity, instead they rely upon release management processes as opposed to using a fully defined approach toward effectivity via the use of effective points associated to features, rules and build phases.

Of the 7 best of breed configuration applications reviewed, 6 used a constraints-based approach to rule definition whilst 1 used a compilation-based approach. These applications were better able to express some of the more complex feature relationships demonstrated in the use case, and two applications were able to tackle the problem of
writing cross-car line rules to be inherited across multiple vehicle lines. Due to the increased complexity of vehicles, the use of modular designs and the need to be able to define common technical constraints across vehicles/platforms means that there is a requirement to author common feature rules independent of the product ranges and then incorporate these common rules into specific product models. Such an approach we have referred to as being “modular” product modelling.

Through the adoption of modular modelling principles it is proposed that certain types of feature rules are authored as independent files which incorporate effectivity and are change managed in their own right. These independent modular files are then incorporated into the product line files that need to observe these rules. Such an approach, it is proposed, lessen the negative impact of product modelling practices that exhibit the characteristics of a tightly coupled design where feature inter-dependency is complex due to the fact that a complete vehicle line is modelled in its entirety and rules are inherited via the explicit use of structure. The concept of modular modelling appeared to present a challenge for those rules authoring applications that operated via explicit inheritance through the use of product structure, as opposed to those applications whose modelling approach is constructed around implicit inheritance. Product models, which are tightly coupled, tend to exhibit the following developmental characteristics, which are seen as disadvantages:

• A change in the rules applied to one feature/feature family can force a ripple effect of changes into other features/families and their corresponding rules.
• The compilation of a valid product model for a complete vehicle line will require more effort and/or time due to the increased inter-dependencies between features and their rules.
• A particular set of common feature rules cannot be reused because they are embedded into the complete product model.

A conclusion of this limited study was that neither the PLM nor the ERP system could supply the needed functionality for a lifecycle approach to product configuration. This can be explained by the cross cutting nature of configuration management across very many diverse business processes whereas PLM and ERP systems are focussed only on a particular subset of these processes. PLM systems have the purpose of supporting product knowledge management in the development and maintenance of products whereas the main purpose of ERP systems is to support operational business requirements in areas such as finance, manufacturing, inventory and human resources. From a product configuration management perspective they differ in a range of aspects, which is summarised in Table 2.

When working in PLM systems the basic approach is project-based centred on a particular product to be developed, whereas in ERP systems the basic approach is transactional dealing with representing the business operations taking place in the extended enterprise. Similarly, PLM systems seek to support time to market for new products whereas the time cycle supported by ERP systems is directed towards supporting the time to customer from the receipt of an order to delivery of the product. The focus of these systems is directed towards the optimisation of the business processes supported by these applications, not towards the optimisation of configuration activities to support the complete product lifecycle.
Table 2. Main aspects of PLM, ERP and CLM with Respect to Configuration Management

<table>
<thead>
<tr>
<th>Purpose</th>
<th>PLM</th>
<th>CLM</th>
<th>ERP</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Approach</strong></td>
<td>Project-based.</td>
<td>Supports both project-based and transactional view with one set of configuration master data.</td>
<td>Transaction-based.</td>
</tr>
<tr>
<td><strong>BOM focus</strong></td>
<td>Definition of an Engineering BOM.</td>
<td>Generation of solved feature strings.</td>
<td>Execution of a Manufacturing BOM.</td>
</tr>
<tr>
<td><strong>Solve performance</strong></td>
<td>BOM-explosion workload: 10’s per hour.</td>
<td>Performs many types of solves also to support, among other things, BOM explosion at a rate of 10000s per hour.</td>
<td>BOM-explosion workload: 1000s per hour.</td>
</tr>
<tr>
<td><strong>Change of product configuration offerings</strong></td>
<td>By model year.</td>
<td>Linking model years and running changes.</td>
<td>Running.</td>
</tr>
<tr>
<td><strong>Configuration space</strong></td>
<td>Partially defined configuration space biased towards technical features.</td>
<td>Fully defined configuration space linking technical and commercial features.</td>
<td>Partially defined configuration space biased towards commercial features.</td>
</tr>
</tbody>
</table>

When it comes to parts management in bills-of-material (BOM) the focus of the systems differs: PLM is supporting the development and evaluation of the Engineering BOM whereas ERP is concerned with executing the manufacturing BOM. In the middle is the need of linking commercial and technical features in fully solved feature strings to be used in driving the planning and manufacturing. Performance metrics also differ a lot. In PLM the BOM explosion is typically triggered manually as part of an engineer’s work whereas in ERP the BOM explosion workload can easily go to 1000s per hour during planning and entering of orders. On top of these are the need to make configuration solves for guiding customer and resellers using interactive ordering tools, which can easily go to the need of 10,000s per hour. However, the most limiting of all aspects when it comes to managing configuration is the lack of a total view of the full configuration space as seen by PLM or ERP. Only partial knowledge about the technical or commercial rules is accessible in any one system, however, the ability to analyse the complexity of configured products through analysis of the effects of the rules require a complete view of all rules and their interactions.

The interfacing between PLM, CLM and ERP is conveniently built on the features and the feature families. Features are the basic building blocks for defining configurations of vehicles and provide the basis for one common modelling language through which to
write rules. The features include customer-facing features used in the ordering of products as well as technical features (Forza and Salvador, 2007) which are unimportant to customers but are essential in driving the manufacturing processes. Therefore, an important shared resource is a master feature dictionary which contains the common vocabulary of all allowed features across the involved systems.

A master feature dictionary should be managed carefully to avoid the duplications of features. The master feature dictionary will evolve over time and some re-modelling of feature families is unavoidable as vehicle technologies develop and change otherwise simple features into being combinations of sub-features. (An example is steering columns. Originally there might be just two: with or without power steering. But over time new features occur such as “tilt” and “memory” for storing configurations for various drivers will be added suddenly giving rise to an increasing set of combinations for steering columns that might better be modelled from a range of features families instead of just one.)

![Diagram of relationships between core concepts]

**Figure 3.** An Overview of the Relationships between the Core Concepts.

For the synchronisation of product changes, effectivity points can be given symbolic names that can be used across systems. This gives rise to the shared resource of an event calendar. The event calendar contains named time-points on which a change in the product definition is to have effect. Therefore the master feature dictionary and the event calendar are the basic mechanisms for synchronisation between PLM, ERP and other important enterprise systems. The relationship between the core concepts are shown in Figure 3.

We would, therefore, argue that the scope and purpose of the two enterprise applications examined would mean that it is extremely difficult to maintain a master system for the authoring of product models and configuration for anything but the most simple of products. In the context of the configurable complexity presented by the automotive case study considerable gaps are perceived in the functionality of the enterprise applications and the specialist configurators using constraints based technologies. In the next section we shall review the influence of product architecture upon the need for a holistic and
integrated approach towards configuration lifecycle management through the use of configuration technologies.

**MANAGEMENT IMPLICATIONS: MASS CUSTOMISATION, MODULARISATION AND CONFIGURATION OF PRODUCT VARIANTS**

For more than 20 years the automotive industry has been characterised by increasing levels of product modularisation (Gadde and Jellbo, 2002; Sako, 2003). Ideally, the design of a product should reflect the need to configure it and the concept of modular design is intended to fulfil this need. The decision on how a product is decomposed into smaller building blocks (product partitions or modules) is core to the technical design process (Suh, 2001; Ulrich and Eppinger, 1995), whilst the degree of interdependency, as defined by product interfaces, defines the extent to which a product architecture might be described as being 'integral' or 'modular' (Ulrich, 1995).

The concept of modularity is not a new theme for the product design literature (Alexander, 1964), yet little consensus exists with respect to defining the ideal approach toward the implementation of modular design principles (Gershenson, et al., 2004), whilst the partitions within a product are problematic to define (Sako and Murray, 1999a; Gadde and Jellbo, 2002; Fixson, 2003; Florent, 2005). However, automotive architectures are examples of near-decomposable systems (Dosi et al. 2003) meaning that there is typically an overlap between systems and sub-systems as opposed to perfect decomposability.

At the highest level of configuring a vehicle it is split into largely independent features that can be combined. For example, from an engineering perspective the choice of power train (engine and transmission) is independent of the choice of internal trim (colour, seat texture, internal veneer, and other features). The design of the lower level sub-systems and their components can vary greatly with respect to the way in which functionality is mapped to product designs and the way in which interfaces determine the way in which the component designs link together (Ulrich, 1995).

What sets automobiles apart from most other products are high levels of customisable product variety combined with multiple vehicle technologies used to realise the design of a vehicle. Fine (1998) argued the case that each industry has its own evolutionary lifecycle or "clockspeed" which determines the rate at which it introduces new products, processes or organisational structures. An automobile is not subject to a single technology lifecycle, but to the multiple lifecycles of each of the major systems and sub-systems that are integrated into the design of a vehicle. Major vehicle architectures, such as the infotainment system, run to a different technology lifecycle to those of the engine and transmission systems.

Throughout the life of a vehicle platform the technologies of its major systems evolve and are up-dated according to their own evolutionary cycles. The combined development and production lifecycle of a vehicle can be anything up to eight years in duration and a lifecycle approach towards configuration covers all phases of development and change. The product modelling lifecycle has to start in the very early product development phases with the definition of the engineering constraints. Later, as the commercial definition of the vehicle evolves the market constraints will be added. Together the combination of these different types of constraints results in very high levels of product variety. The growth of product complexity, the need to observe legislation in multiple markets, combined with the pressures of product personalisation drives the need for a lifecycle
approach towards configuration with a single centralised system for managing both the
definition of the configuration space and the storage and manipulation of individual
configurations.

The calculation of product variety for automobiles remains an area of debate,
controversy and great complexity. It is also an area that we would argue that has not been
adequately addressed to date. Pil and Holweg (2004) calculated the number of automotive
product variants via the number of body styles, power trains, paint-trim combinations,
and factory-fitted options such as air conditioning, sunroofs and tinted glass and this
method has been replicated by other researchers (Scavarda et al. 2009). However, this
approach is incomplete as it is not simply a case of multiplying the number of sales
options to determine the number of buildable combinations; it is much more complex
than that (Kübler et al. 2010).

Such an approach to the calculation of buildable combinations does not take into
account the number of markets and the variation by market caused by legal and territorial
restrictions. Complex interdependencies are created and many of these interdependencies
are caused by implied technical features which are required for technical or legal reasons
to configure a valid configuration of a product. These implied features are not visible at
the point of order and, therefore, customer visible variations (observed via brochures or
manufacturer websites) are only a small subset of the variations generated by technical
and legal restrictions (Kübler et al. 2010:45). Any calculations based upon what is visible
at the point of order will be inaccurate, and most likely will under-estimate the true
number of total buildable combinations.

The number of valid combinations is also not a static number but a dynamic one that
varies over time as the technical content of the product changes and regions and markets
alter and flex their product offerings to customers. For all but very trivial configurable
products the number of valid configurations will be an incredible high number which
complicates representation and analysis. As an example, a modern car such as the Range
Rover Sport has more than $5.8 \times 10^{21}$ buildable combinations. [This number has been
found with a commercially available tool from Configit using BDD-based techniques on
a better encoding of the configuration space (Møller et al., 2001; Hadzic et al. 2004;
Subbarayan et al. 2004 ) than the one-Boolean-variable-per-feature encoding used in
(Kübler et al. 2010) where they report failure of BDD-based counting.] The growth of
software driven technologies such as telematics, infotainment and telecommunications
systems in cars means that the way in which the cars can be configured and customised
will continue to increase.

The number of buildable combinations serves to illustrate the complexity of managing
highly customisable products but is of little practical use, although studies have tried to
determine benchmarks for the auto industry (Scavarda et al., 2009). Of more practical use
is the understanding of the necessary variants and the feature drivers of the individual
components and sub-assemblies to be used in the manufacturing of the automobile. When
the number of variants of products goes up, the associated engineering and tooling costs
increase and the supply chain management becomes increasingly complex. For instance,
it is of key importance that a sub-assembly matching a vehicle’s configuration arrives at
the assembly line in the right sequence and at the right time for that particular vehicle.
With more variants the likelihood of having mismatched parts increases with the potential
consequence of incurring production losses and delivery delays.
CONCLUSIONS AND NEXT STEPS

Product configuration applications continue to be implemented via a myriad of technical solutions, via commercial sales configurators, as parts of ERP systems and more recently PLM systems or as bespoke company specific solutions. The proliferation of configuration capabilities means that configuration data is increasingly likely to be distributed across multiple applications throughout the enterprise. In the context of assemble-to-order companies we conclude that consideration needs to be given to the notion of a master system of record for product definition logic and configuration. As a consequence we have argued for the need for a configuration-centred system for Configuration Lifecycle Management (CLM).

This paper has presented the reasoning around the need for such an approach and the next natural steps with respect to research would be to consider further important aspects of a successful implementation of a CLM solution. One aspect comes from the observation that configuration knowledge is consumed by many systems and users in the extended enterprise. A complication is that all systems and users do not have the same needs for their view on a configuration. For instance, for billing it is important to know the choice of features influencing the price whereas technical features driving parts for manufacturing are irrelevant and only confusing. For a customer going to a web-site to explore the options a textual description of the features in understandable text in the local language is important whereas an engineer seeking knowledge about a product will most likely prefer a concise, easy to recognize description. Therefore, a successful CLM system must take the channel consuming the knowledge into account.

Another line of research would be to explore the potential for achieving further business value coming out of a centralised source of product definition and configuration data, for example:

- The ability to associate predicted market take rates to features and calculate part volumes compounded from those take rates.
- The ability to compare feature commonality or differences between products and product types.
- The support for architectural analysis and cross carline analysis to optimise commonality and complexity targets.
- The ability to analyse parts usage in BOMs based on what configurations are valid over time.

At Jaguar Land Rover, used in our case study, the plan is to proceed with an implementation of an integrated CLM system over the coming 18 months.

REFERENCES


