Open access • Journal Article • DOI:10.1080/00207540903104210

Methods for the capture of manufacture best practice in product lifecycle management — Source link

Anantharajah George Gunendran, Robert I. M. Young

Institutions: Loughborough University

Published on: 01 Jan 2010 - International Journal of Production Research (Taylor & Francis)

Topics: Process development execution system, Product engineering, Integrated Computer-Aided Manufacturing, Manufacturing execution system and Product design specification

Methods for the Capture of Manufacture Best Practice in Product Lifecycle Management

Anantharajah George Gunendran, R I M Young

Methods for the Capture of Manufacture Best Practice in Product Lifecycle Management

Journal:	*International Journal of Production Research*
Manuscript ID:	TPRS-2008-IJPR-0518.R2
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	21-May-2009
Complete List of Authors:	Gunendran, Anantharajah; Loughborough University, Wolfson School of Mech. and Manu. Eng. Young, R I M; Loughborough University, Wolfson School of Mech and Man Engineering
Keywords:	MANUFACTURING MANAGEMENT, MANUFACTURING STRATEGY, MANUFACTURING INFORMATION SYSTEMS
Keywords (user):	Manufacturing best practices, PLM, UML

Methods for the Capture of Manufacture Best Practice in Product Lifecycle Management

A.G. GUNENDRAN*§, R.I.M. YOUNG§

§ Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, Leics., LE11 3TU, UK.

Anantharajah George Gunendran

Wolfson School of Mechanical and Manufacturing Engineering,

Loughborough University,Loughborough,

Leics.,LE11 3TU,UK

Tel: +44(0)1509227684

Fax: +44(0)1509227648

Email: A.G.Gunendran2@lboro.ac.uk

Robert Ian Marr Young

Wolfson School of Mechanical and Manufacturing Engineering,

Loughborough University,Loughborough,

Leics.,LE11 3TU,UK

Tel: +44(0)1509227662

Fax: +44(0)1509227648

Email: R.I.Young@lboro.ac.uk

* Corresponding author: Email: A.G.Gunendran2@lboro.ac.uk

Methods for the Capture of Manufacture Best Practice in Product Lifecycle Management

A G Gunendran* and R I M Young

Wolfson School of Mech. & Manu. Eng., Loughborough University, Loughborough, LE11 3TU, UK

Abstract:

The capture of manufacturing best practice knowledge in product lifecycle management systems has significant potential to improve the quality of design decisions and minimise manufacturing problems during new product development. However, providing a re-useable source of manufacturing best practice is difficult due to the complexity of the viewpoint relationships between products and the manufacturing processes and resources used to produce them. This paper discusses how best to organise manufacturing best practice knowledge, the relationships between elements of this knowledge plus their relationship to product information. The paper also explores the application of UML-2 as a system design tool which can model these relationships and hence support the reuse of system design models over time.

The paper identifies a set of part family and feature libraries and, most significantly, the relationships between them, as a means of capturing best practice manufacturing knowledge and illustrates how these can be linked to manufacturing resource models and product information.. Design for manufacture and machining best practice views are used in the paper to illustrate the concepts developed. An experimental knowledge based system has been developed and results generated using a power transmission shaft example.

Keywords*:* Manufacturing best practices, information and knowledge organisation, system design, UML

1 Introduction

Minimising the time of the product development phase and ensuring effective support for the service phase of the product lifecycle is a long standing problem (Sudarsan et al. 2005, Ming et al. 2008). This is partly due to

* Corresponding author Email: A.G.Gunendran2@lboro.ac.uk

the lack of communication between different actors involved during the product lifecycle (Schuh et al. 2008, Tang, Qian 2008). This paper explores a methodology to improve the communication of information between designers and manufacturing engineers as to the consequence of design decisions on the manufacturing methods required and the associated manufacturing problems which design changes may bring. Typical computational tools use feature technology as the basis to communicate and bridge design and manufacturing domains (Durr, Schramm 1997, Gadh, Prinz 1995, Gao, Zheng & Gindy 2004) where feature technology is used to link geometric shape to manufacturing processes (Abouel Nasr, Kamrani 2006, Szecsi 2006). However maintaining information across multiple feature domains is a problem (Gunendran, Young 2006) which requires different feature sets to be defined along with the relationship knowledge which bridges across these domains (Gunendran et al. 2007). The relationships between the design and manufacturing sets of features need to be better understood to improve the sharing between the domains of design and manufacture.

Further, the support for product development activities can be improved if the manufacturing methods of all previously manufactured products have been organised in a Product Lifecycle Management (PLM) system to be accessed by designers and manufacturing engineers. PLM is a key technology to support the communication between actors (Stark 2005, Guerra-Zubiaga et al. 2006). Present PLM systems provide the communication management functions, however, they are limited in supporting communication beyond the level of metadata as well as a certain level of geometrical information (Ming et al. 2008). Further, there is a problem in how to organise the required information to support specific activities of designers and manufacturing engineers without overloading them with unnecessary arrays of information. A clear grouping method is necessary to organise the manufacturing information of specific type products if we are to ensure that only relevant information is to be fed back to designers.

One way of grouping parts and features is based on their commonalities, for example the parts that share common manufacturing methods can be grouped and defined as manufacturing part families. Traditional

Group Technology methods provide ways to group parts and form part families based on common manufacturing methods (Yang, Yang 2008, Seifoddini, Tjahjana 1999, Sarker, Mondal 1999, Baykasoglu, Gindy 2000, Mckay 2003). However, traditional methods consider the manufacturing phase of products and offer solutions for the best way of utilizing manufacturing resources for parts which have already been designed (Angra, Sehgal & Samsudeen Noori 2008, Collaine, Lutz & Lesage 2002, Cheng et al. 2008). The approach described in this paper also explores a grouping method of parts based on common manufacturing methods. Further, the capture of best practice manufacturing information, based on features and part families is not new (Li, Ong & Nee 2002, Martinez, Favrel & Ghodous 2000, Sanderson, Uzumeri 1995). However, the key issue is identifying the relationships which exist between the range of part families and features for a particular product range. For example, captured manufacturing methods can be re-used to provide manufacturing consequences during product design. This requires that the relationships between manufacturing part families and design part families must be understood. Section 2 briefly explains the current state of the art in product and manufacturing modelling and argues the need to extend this to include a library of best practice manufacturing knowledge. It also discusses the system modelling method used and argues the case for the use of UML-2. Section 3 explores the required information for product realisation and identifies the complex conditional relationships between the required information sets in order to organise information for reuse. Section 4 demonstrates how manufacturing best practice methods with the complex conditional relationships can be captured using UML-2. Section 5 provides an example application of proposed methodology to model and capture manufacturing best practice methods for design realisation. Finally, we draw some conclusions and give a brief outline of our plans for future work.

2 Manufacturing Information and Knowledge Modelling for Product Design Realisation

2.1 Key information models for product design realisation

A model is a representation of the characteristics of a system. The system could be physical, conceptual or analytical. Commonly, the model for representing something physical is a prototype; for something conceptual is a scheme or a method for something analytical is an equation (Prasad 1996). According to ISO (ISO 10303-214 1997), a model is a representation or description of an entity or a system, describing only the aspects considered relevant in the context of its purpose. Generally the purpose of modelling is to predict the behaviour of the system, but in the context of computational tools, information and knowledge models should provide the required information and knowledge in a computer understandable format to support specific computational activities (Gunendran 2004).

This paper proposes a design realisation system with three information and knowledge models; a product model, a manufacturing model and a library model as shown in Figure 1. The product model provides a database of product related information such as geometry, product architecture, product functionalities, characteristics, etc while the manufacturing model provides a database of manufacturing capabilities, resources and processes available to an enterprise. The library model provides a database of best practice manufacturing methods i.e. knowledge of how best to apply manufacturing resources and processes. The combination of three models can provide manufacturing feedback to design and manufacturing engineers. The manufacturing feedback is generated for either new or modified products based on best practice manufacturing knowledge from the library model and the available manufacturing capabilities captured in the manufacturing model. This paper focuses primarily on the library model as this provides the source of best practice knowledge to support design realisation. The library model approach is a Knowledge Based System (KBS) approach, but varying from traditional KBS approach as typical KBS implementations are limited to support single knowledge base applications (Antony, Batra & Santhanam 2005, Sapuan 2001). The library model supports the capture and utilisation of multiple context knowledge including the complex relationships between these multiple contexts. In previous work the manufacturing model has been seen as the resource of all knowledge and information related manufacturing capability (Guerra Zubiaga, Young 2006, Guerra Zubiaga, Young 2007, Guerra Zubiaga, Young 2008). This is because of the complexity of the relationships between manufacturing knowledge and their relationships to manufacturing resources and product information.

[insert Figure 1 here]

The approach presented in the paper utilises an object oriented approach to capture best practice knowledge. The crucial part of the problem is defining structures to capture the relevant information and knowledge relationships. There are several methodologies available to define and represent object oriented structures with Unified Modelling Language (UML) being probably the most generally accepted method. UML is a modelling methodology commonly used in object oriented software programming to support planning and code generation activities (Thomas 2003). Further, UML is a comprehensive tool for system modelling. There are several kinds of diagrams available in the UML methodology (Hadar, Hazzan 2004), but class, and object diagrams are utilised in this paper because we are addressing information organisation issues. The class diagram graphically illustrates the database structure while the object diagram graphically illustrates the instances of the classes with the actual relationships. An application of the object diagram is given in section 4.1 and the class structures of the product and manufacturing models are given below.

[insert Figure 2 here (Young et al. 2000)] With kind permission of Springer Science and Business Media.

The product model is a source and repository of information for many applications, and such a model allows information to be shared between many users and the software components of the CAE system (Shaw, Bloor & De Pennington 1989, McKay 1993). Figure 2 shows a top level product model structure to represent any type of product (Young et al. 2005). The top level product model structure contains a product class which captures the general information about the product. The ’Product’ class has relationships with ‘Architecture’, 'Views’, ‘Characteristics’, and ’Purpose’ classes. These top level classes have different sets of child classes to capture all relevant product information. The ‘Architecture’ class and its sub classes provide a way to capture the structure of products including assemblies, sub-assemblies, and components. The ‘Views’ branch of classes facilitate the capture of product information according to viewpoint considerations such as design, manufacture, and assembly. The ‘Characteristics’ class branch captures product information such as geometry, and material property. The ’Purpose’ branch of classes facilitates the capture of product functionalities.

[insert Figure 3 here (Young et al. 2005)] With kind permission of Springer Science and Business Media.

The manufacturing model provides a structure to capture processes, resources, and manufacturing strategies (Young et al. 2000). The ‘Processes’ branch of classes captures information about the possible manufacturing processes while the ’Resources’ branch of classes captures information about the available manufacturing resources. The ‘Strategies’ set of classes captures the knowledge about the utilisation of processes and resources to perform manufacturing activities. Figure 3 shows the top level class structure of a traditional manufacturing model.

The strategy set of classes in the traditional manufacturing model provides the knowledge about how the processes and resources can be used for product manufacture. However, the applicability of the strategy class knowledge for a particular product requires additional knowledge of the product context. Therefore, the application of the knowledge resides in the strategy set of classes for a particular product requires relationship with the product model. Further, the strategy class does not capture the information on how the processes and resources are applied to products. However, this is a particularly useful knowledge requirement for the many industries whose new products are based on developments from similar existing products. Therefore, a new model, called library model, has been introduced to capture the information on how the processes and resources are best used based on knowledge from past products within the specific industry. Hence, the combination of these three models should facilitate the capture of known manufacturing knowledge in an organised way.

2.2 Requirements for modelling manufacturing best practices and the capability of UML-2

The library model is proposed to capture the identified best practice to support product lifecycle activities, and to improve the re-usability of successful past decisions. Best practices can be identified in all phases of the product lifecycle; design, manufacturing, service, and disposal, but here we focus on manufacturing best practice. The paper uses manufacturing best practices to explain the concept of modelling complex conditional relationships. The aim of the library of manufacturing best practices is to provide support to designers and manufacturing engineers for decision making.

Several conditions such as part dimensions, feature relative positions, and intersecting features need to be considered for the modelling of manufacturing best practices. Consider manufacturing methods of a simple key shaft and the position of a 'key_way' feature as an example. The 'key_way' feature can be machined either by milling or slotting, if the 'key_way' feature is placed at the end of the shaft. But the slotting machining method is not feasible, if the ‘key_way’ feature placed in any other positions, because the slotting process requires at least one open end for the reciprocating movement of the slotting tool. Hence the manufacturing method of the key shaft varies depending on the position of the ‘key_way’ feature. The relationships between the ‘key_way’ feature and the ‘shaft’ part can be represented in the traditional UML using ‘part-has-feature’ association relationship as shown in Figure 4. However, the traditional UML representation is not capable of modelling the position of the ‘key-way’ feature on the shaft. Therefore, the ‘key_way’ feature can be anywhere on the shaft as shown in Figure 4. Hence, defining an effective manufacturing method representation for the key shaft is a problem without knowing the position of the ‘key-way’ feature. This problem can be overcome by using UML-2 for modelling (Bock 2004, UML-2 2006). The UML-2 class diagrams are capable of representing conditional relationships between different information sets using an association class (Berardi, Calvanese & De Giacomo 2005). In the key shaft example, the position of the key way can be modelled with the association class as shown in Figure 4. Hence, the manufacturing method for the key way shaft can be modelled properly with UML-2. The application of the UML-2 in this research is explained in section 4.

[insert Figure 4 here]

The UML graphical class representation can be used to generate software codes for programming applications . However, the traditional UML class diagrams are limited in the capability of representing complex information relationships. Therefore, the codes generated using traditional UML class diagram, which model the complex conditional relationships, require manual interaction to add the codes of complex conditional relationships. UML-2 is capable of capturing conditional relationships through the use of association classes in addition to the traditional representation of different information sets as normal classes. Therefore, the UML-2 class diagram can be utilised to generate software codes for applications with conditional relationships. Hence, UML-2 helps to both design the information models and generate code for complex conditional relationships between different information sets.

The code generated using UML-2 class diagrams provides the way to capture the conditional rule values as instances, therefore extending the capability of traditional UML 'hard coded' and inflexible method of modelling complex relationships. Therefore, systems based on UML-2 allow the changes in the association behaviour, since the relationships are controlled by the association class instances. However, a rigours representation of association class instances is necessary to deal with the multiple systems based on different platforms as detailed in section 2.2.

2.3 Rigorous representation of association class instances using OCL

The UML-2 approach provides a basis for knowledge capture during system design rather than knowledge addition through hard coding during system implementation. The UML-2 class structure can therefore be used to update knowledge relationships over time and maintain the knowledge as systems are upgraded over time, thereby providing an element of system life-proofing. The life-proofing of relationship knowledge has two levels: maintaining within a system and maintaining between systems. The case study example given in section 5 explains the life-proofing of relationship knowledge within a system but the application of other level of life-proofing is not dealt with in this paper. The life-proofing of complex conditions can be further improved by representing rules using the Object Constraint Language (OCL), where the OCL provides an way of constraining objects. OCL is a component of UML-2. Section 4.2 explains the application of OCL with an example, which shows the life-proofing of association class objects.

Section 3 explores the organisation of manufacturing best practices based on features and part families using a simplified shaft manufacturing method to illustrate the concepts. Section 3.2 identifies the key sets of relationships between product and manufacturing information sets in order to organise and capture known manufacturing best practices.

3 Identifying key relationships between product variants and their manufacturing methods

There is a broad range of manufacturing knowledge available in most businesses which is essential to support effective design realisation. However, tapping knowledge relevant to a particular design situation requires an understanding of many complex relationships between the type of part being designed and manufactured. Examples are the manufacturing facilities, the materials to be used and the knowledge of relevant manufacturing methods used in the past.

The use of feature-based approach is generally is limited by these inter-relationships. For example design features provide a design view on a specific product while manufacturing features provide a manufacturing view on a product; a 'key-way' design feature contains functional and geometry information while the 'key-way' manufacturing feature captures the manufacturing methods and instructions. However, there is still a need to provide an understanding of how these different contexts can be related to one another and how knowledge systems should be designed to capture the constraints that exist between these relationships.

3.1 Features in context

Associating manufacturing instructions to features which can be used in design is not necessarily a straight forward task. Specific feature manufacturing methods will be influenced by the type of part on which it is to be applied. For example, a 'spline' feature can be produced by several manufacturing methods including milling, hobbing, cold rolling, and broaching. The selection of which combination of manufacturing methods should be used will be influenced by the part geometry, the purpose of the part and the overall manufacturing operations which are expected to be used for part manufacturing. Figure 5 illustrates two different spline manufacturing methods on a shaft. The spline feature on shaft A of Figure 5 is produced by milling because it is the optimal process to provide the required strength and it is a suitable process for the part geometry. The spline on shaft B of Figure 5 is manufactured by cold rolling, because the manufacturing method produces high strength splines which are needed for that part. Hence, defining manufacturing instructions for a feature must be related to the part context.

[insert Figure 5 here]

Parts can be grouped based on the commonalities in several factors such as functions, geometrical shapes, and manufacturing methods with parts based on functions and manufacturing methods and termed respectively as design and manufacturing part families. For example, power transmission shaft products can be grouped as low, intermediate, and high power transmission shaft as design part families. These power transmission shafts can be manufactured mainly by turning, milling, broaching, and cold rolling. These different manufacturing methods provide a way to group the power transmission shafts as different manufacturing part families. Figure 6 illustrates these design and manufacturing shaft part families and possible relationships between the part families. The manufacturing part family can be defined as a collection of parts that utilises the same manufacturing operations, setups, machine tools and fixtures for manufacturing. However, the variation in machining operation sequence is allowed within a part family in order to reduce the number of different part families.

[insert Figure 6 here]

Manufacturing features can be grouped in a similar way as part families, but the feature manufacturing methods need to be defined in the context of the part. Therefore, there is a need to identify the relationships between features and part families in order to group and define manufacturing best practices. The following section introduces the key relationships identified between part families and features.

3.2 Relationships between feature & part families to define manufacturing best practices

Three different kinds of relationships have been identified between features and part families as: relationships between features, relationships between part families, and relationships between features & part families.

If the relationships between design and manufacture features is understood, the known manufacturing difficulties and limitations can be shared during design realisation. As explained in section 3.1, the manufacturing methods of a manufacturing feature are dependant on the part context and cannot be identified directly from the design feature, without considering the part context. Therefore, the relationships between design and manufacturing features are controlled by part families. Hence, the relationships between features and part families need to be defined first in order to construct and control the relationships between design and manufacturing features. The other key relationship is between part families and particularly between design and manufacturing part families. Therefore, these three kinds of relationships are necessary to organise product development knowledge to support new product development. The following section explains the modelling of manufacturing best practices based on part families and features using UML-2.

4 Modelling best practice manufacturing knowledge

4.1 The structure of the model

Figure 7 illustrates a top level UML-2 class structure to represent the above mentioned three main relationships between part families and features to capture manufacturing best practices. The top level class structure contains three relationships as:

Part family – feature relationship

Part family – part family relationship and

Feature – feature relationship

[insert Figure 7 here]

Each relationship contains an association class, which controls the relationships between the associated classes and facilitates the capture of the conditions of the relationships between the instances of the related classes. The relationships between the instances of 'PartFamily' and 'Feature' classes can be controlled by the instances of the association class 'PFamily_FeatRel'. Further, the inheritance functionality of UML can be applicable to the association relationships. Hence, the relationships between the sub classes of the associated classes can be modelled and controlled by the top level association relationship. For example, the top level relationship class 'PFamily_FeatRel' facilitates the capture of the relationships between following sub classes of 'PartFamily' and 'Feature':

- design part family- design feature,
- manufacturing part family- manufacturing feature, and
- design part family-manufacturing feature relationships.

In a similar way the association class 'PFamily_PFamilyRel' facilitates the capture of relationships between different types of part families and 'Feat_FeatRel' class facilitates the capture of relationships between different types of features. Moreover, the association class 'PFamily_PFamilyRel', which is controlling the relationships between the sub classes of the 'PartFamily', is influenced by the 'Feature' class and vice versa as illustrated in the Figure 7. The development of this high level set of relationships into the sub-classes used to represent manufacturing best practices is shown in Figure 8. This is shown in the context of best practice methods for the product life cycle activities of design, manufacture, and service although these have not been explained here.

An important result of the research has been to define a class structure which supports the organisation of manufacturing knowledge in terms of machines, fixture, setups, cutting tools which may be used in producing parts and features as illustrated in figure 8. The part family manufacturing method is defined by a machining sequence. For example, a shaft with a spline feature can be manufactured by turning and milling operations. Therefore, the machining sequence for that particular shaft family is turning and milling. Each of these machining operations can be performed by lathes and milling machines respectively. Each machining operation requires specific setups and fixtures. Within a particular setup specific machining steps can be performed using the required cutting tools. In a similar way, feature manufacturing methods can be defined, taking into account the part context. For example, a spline feature on a spline shaft can be manufactured in two stages as turning and milling. The spline feature manufacturing method is influenced by the shaft manufacturing method and the machine tools used for the machining operations. Each stage requires specific machining steps and each Step links a machining step geometry to a tyoe of cutting tool. These relationships between different types of information have been captured in the UML-2 class structure as described in the following paragraphs.

[insert Figure 8 here]

The ‘ManuPartFamily’ class illustrated in Figure 8 facilitates the capture of instances of manufacturing part family. The ‘ManuPartFamily’ class is related with ‘MachiningOperationSequence’ class using ‘has’ relationship. Therefore, the machining best practices of a manufacturing part family can be defined as a sequence of machining operations. The relationship between ‘MachiningOperationSequence’ and ‘MachiningOperation’ classes facilitates the capture of sequences of machining operations. Machine tools are required to perform machining operations. Class ‘MachineTool’, which is related with the ‘MachiningOperation’ class, facilitates the capture of required machine tool information. Each machining operation may require a specific setup sequence. The setup sequence information can be captured with ‘SetupSequence’ class that are related with the ‘MachiningOperation’ class. The setups of a setup sequence can be captured with the ‘Setup’ class while the fixtures required for setups can be captured with the ‘Fixture’ class. One or more machining steps can be performed with a single setup depend on the part machining method. The step information can be captured with ‘StepSequence’ and ‘Step’ classes. Each step is linked with a cutting tool and the ‘CuttingTool’ class facilitates the capture of cutting tool information.

The feature machining method can be captured in a similar way using ‘StageSequence’, ‘Stage’, ‘StepSequence’, ‘Step’, and ‘CuttingTool’ classes as illustrated in Figure 8. The feature machining process may undergo different stages that are dictated by the machining sequence of the part. Therefore, the machining stages of a feature type may vary depend on the part machining sequences. Stages capture the geometry to be machined on Machine Operations.

The above mentioned are all potential relationships between different sets of information, but depending on the specific parts some of the relationships will not exist. Hence, the relationships need to be captured with constraining conditions to be effective in capturing the manufacturing information and supporting product development activities. The following section briefly explains how the association classes can be used to constrain relationships between classes by utilising a shaft part family example.

4.2 An example application of UML-2 and object diagram

The UML-2 class structure defines the type of objects and their relationships. Different object instances can be populated according to these classes. The populated instances and their relationships can be clearly illustrated using UML-2 object diagrams. Further, the object diagram can be used to illustrate the influence of association class instances in defining the actual relationships between the instances of related classes.

Figure 6 shows an example of spline shafts in 3 different part families; high power transmission shaft, milled spline shaft and cold rolled spline shaft. The high power transmission spline shaft can be manufactured by two main methods; milling and cold rolling.

[insert Figure 9 here]

The selection of the best manufacturing method for a spline shaft is influenced by several factors including spline strength and shaft geometry, and especially by the adjacent features of the spline. If the spline shaft has a cylindrical feature adjacent to spline and the diameter of cylindrical feature is less than the spline minor diameter, the spline can be manufactured by milling. If the diameter of the cylindrical feature is greater than the minor diameter of the spline, the milling cutter can't travel freely over the cylindrical feature and leaves a fillet feature between the spline and the cylindrical feature. If the acceptable fillet radius is greater than the spline milling cutter, then the spline shaft can still be manufactured by milling. However, if these geometrical conditions cannot be satisfied, the spline shaft can be manufactured by cold rolling. Therefore, the geometrical shape of the spline constrains the mapping of the spline shaft design part family to the appropriate manufacturing part family as shown in Figure 9.

[insert Figure 10 here]

Figure 10 shows a table of possible options for the constraining conditions. Three influencing factors have been considered in the example to define the relationships. Each factor can either be 'TRUE' or 'FALSE'. Therefore, in the example the relationships between design and manufacturing part families are controlled by eight conditional relationship options. For example, constraint condition 2 and 4 represent the relationship condition for a low strength spline with the adjacent cylindrical feature diameter less than the spline minor diameter. Therefore, the option links the design part family to the milled shaft part family and the constraining conditional options can be represented as constrain class instance between part families.

[insert Figure 11 here]

The actual relationships between the design part family instances and manufacturing part family instances can be controlled by the association class instances as shown in Figure 11. This shows the instances of the example objects. The relationships between design and manufacturing part families can be controlled by eight association class instances. These complex constraining relationships can be represented by the UML-2 class structure but additionally the actual constraining instances need to be populated to the system. This is achieved using Object Constrain Language (OCL) as explained in the following section.

4.3 An example application of OCL

The UML-2 class structure graphically represents the constraining relationships between different classes using association classes and can be used to generate software code for different information systems. The Object Constrain Language (OCL) adds to this by providing a way to represent object constrains. The enhanced definition is particularly useful for defining the behaviour of the association class objects that controls the relationships of different types of objects. The example OCL statements given in this section represent the behaviour of the association class instances explained in section 4.2. The relationship between a spline design part family and a spline manufacturing part family can be decided based on the values of the association class attributes. In this particular example, the following knowledge is required for the decision:

If the value of the attribute '_DiaCheckPossibilityOfMill' is true and '_RequirementOfHighStrengthSpline' is false then the spline shaft can be manufactured by milling

If the value of the attribute '_DiaCheckPossibilityOfMill' is false, the value of the attribute '_FilletCheckPossibilityOfMill' is true and '_RequirementOfHighStrengthSpline' is false then the spline shaft can be manufactured by milling;

In other cases the shaft can be manufactured by cold rolling

This set of knowledge can be represented and captured with OCL statements as follows:

```
context Des_MfgPFRel::_MfgPartFamily:string
if Des_MfgPFRel._DiaCheckPossibilityOfMill()
  if Des_MfgPFRel._RequirementOfHighStrengthSpline() then
     _MfgPartFamily "ColdRolledSplineShaftPF"
  else MfgPartFamily "MilledSplineShaftPF"
else
    if Des_MfgPFRel._ FilletCheckPossibilityOfMill()
      if Des_MfgPFRel._RequirementOfHighStrengthSpline() then
     _   MfgPartFamily "ColdRolledSplineShaftPF"
      else MfgPartFamily "MilledSplineShaftPF"
```

else_MfgPartFamily "ColdRolledSplineShaftPF"

To explain the above OCL statements in more detail, each line of the OCL statement takes the form of an OCL reserved word, and an attribute from an object.

For example, take the following lines of OCL statement from above for detail explanation.

if Des_MfgPFRel._RequirementOfHighStrengthSpline() **then**

_ MfgPartFamily "ColdRolledSplineShaftPF"

This statement is a part of the 'if' conditional statement and can be explained in English as if the value of the attribute '_RequirementOfHighStrengthSpline()**'** of class 'Des_MfgPFRel' is true assign 'MfgPartFamily' as 'ColdRolledSplineShaftPF'. The detail explanation of OCL reserved words and the applications can be found at (OCL 2006).

Having used the UML-2 tools to model the complex interdependency relationships these can then be captured in a knowledge base for use during product design realisation. How this has been achieved is described in the following section.

5 Case study: Capturing manufacturing best practices for a power transmission shaft

The paper uses snow mobile power transmission shafts as an example to demonstrate the methods to capture the manufacturing best practices. Three different power transmission shaft design part families have been considered:

- Low power transmission shaft, where the power is transmitted via the correct assembly fit
- Intermediate power transmission shaft, where the power is transmitted via a key or spline
- High power transmission shaft, where the power is transmitted via a spline

Alongside this four shaft manufacturing part families have been considered:

- Simple shaft

- Key shaft
- Milled spline shaft
- Cold rolled spline shaft

These design and manufacturing part family classifications are illustrated in the Figure 6. The manufacturing best practices of these manufacturing part families can be captured with the UML-2 class structure given in Figure 8. However, these do not have a one to one relationship with the design part families, but the relationships can be defined by the use of relationship rules. For example, the relationships between a spline design part family and its appropriate manufacturing part family can be defined based on the parameters of both the spline feature and the cylindrical feature adjacent to the spline as explained in section 4.1. In this example, the spline shaft can be manufactured by milling, if the spline feature has a cylindrical feature adjacent to the spline and the diameter of the cylindrical feature is less than the spline minor diameter. If the diameter of the cylindrical feature is greater than the minor diameter of the spline, the milling cutter can't travel freely over the cylindrical feature and leaves a fillet feature between the spline and the cylindrical features. If the acceptable fillet radius is greater than the spline milling cutter, then the spline shaft can still be manufactured by milling. However, if these geometrical conditions cannot be satisfied, then the spline shaft can be manufactured by cold rolling. In addition, the cold roll manufacturing method offers high strength splines compare to milling manufacturing method.

This conditional information can be captured with the association class as instances in order to control the relationships between design and manufacturing part families. The instances can be populated in the developed system, but during the system design the conditional instances can be modelled with object diagram as explained in section 4.1. Further, the OCL facilitates the representation of the instances as explained in section 4.2.

The power transmission shaft case study example has been tested within a design realisation support system utilising CAD and PLM environments in conjunction with a knowledge base. Unigraphics NX (NX 2008) has been used as the CAD tool and Teamcenter (Teamcenter 2008) as the PLM tool. Engineering Enterprise Knowledge System (E2KS) (E2KS 2008) has been used as the knowledge base to capture manufacturing best practices. Figure 12 shows the captured conditional instances, which controls the relationships between design and manufacturing part families of the spline shaft, in the E2KS knowledge base.

[insert Figure 12 here]

6 Conclusions and future work

This paper reports on an information organisation methodology to support product development activities. Three models; libraries of best practice, product and manufacturing models have been utilised in the concept. The paper has shown how the relationships and constraints which exist between manufacturing knowledge can be captured by organizing knowledge into part families and features and most importantly by defining the relationships between part families, between part families and features, and between features.

Further, the paper has illustrated the potential value of the UML-2 methodology as a means of life-proofing the complex conditional relationships between different sets of information. This is achieved by capturing the class relationship knowledge during system design rather than directly during system implementation. This will be a critical issue in maintaining and evolving next generation PLM systems.

A critical use of the approach is the identification of manufacturing consequences during the product design process. This can be used either to support redesign according to the identified manufacturing part family or alternatively to generate new manufacturing instructions for the part.

The system design resulting from this work has been implemented in CAD and PLM environments in conjunction with a knowledge base in order to create the design realisation support system. The implemented prototype design realisation system enables us to identify any scaling issues which may arise when using the approach in full scale industrial situations.

Future work will involve exploring issues of scale as the knowledge base is expanded to capture further best practice. Also a significant issue is to explore more extensive relationship knowledge between design function and manufacturing. A further concern of major future significance is that of supporting interoperability across groups and knowledge systems as best practice knowledge should ideally be accessible from multiple sources.

Acknowledgment

The work presented herein was undertaken under the aegis of the Knowledge and Information Management (KIM) Through-Life Grand Challenge Project (www.kimproject.org) funded primarily by the Engineering and Physical Research Council (EPSRC – Grant N° EP/C534220/1), the Economic and Social Research Council (ESRC – Grant N° RES-331-27-0006) and Loughborough University's Innovative Design and Manufacturing Research Centre (IMRC- Grant N° EP/E002323/1)

References

Abouel Nasr, E.S. & Kamrani, A.K. 2006, "A new methodology for extracting manufacturing features from CAD system", *Computers & Industrial Engineering,* vol. 51, no. 3, pp. 389-415.
Angra, S., Sehgal, R. & Samsudeen Noori, Z. 2008, "Cellular manufacturing—A time-based analysis to the layout problem", *International Journal of Production Economics,* vol. 112, no. 1, pp. 427-438.
Antony, S., Batra, D. & Santhanam, R. 2005, "The use of a knowledge-based system in conceptual data modeling", *Decision Support Systems,* vol. 41, no. 1, pp. 176-188.
Baykasoglu, A. & Gindy, N.N.Z. 2000, "MOCACEF 1.0: multiple objective capability based approach to form part-machine groups for cellular manufacturing applications", *International Journal of Production Research,* vol. 38, no. 5, pp. 1133-1161.
Berardi, D., Calvanese, D. & De Giacomo, G. 2005, "Reasoning on UML class diagrams", *Artificial Intelligence,* vol. 168, no. 1-2, pp. 70-118.
Bock, C. 2004, "UML 2 Composition Model", *Journal of Object Technology,* vol. 3, no. 10, pp. 47-73.
Cheng, T.C.E., Kovalyov, M.Y., Ng, C.T. & Lam, S.S. 2008, "Group sequencing around a common due date", *Discrete Optimization,* vol. 5, no. 3, pp. 594-604.

Collaine, A., Lutz, P. & Lesage, J.J. 2002, "A method for assessing the impact of product development on the company ", *International Journal of Production Research,* vol. 40, no. 14, pp. 3311-3336.
Durr, H. & Schramm, M. 1997, "Feature based feedback into the early stages of design", *European Journal of Operational Research,* vol. 100, pp. 338-350.
E2KS 2008, *, Enterprise Engineering Knowledge System*. Available: http://www.emergentks.com/solutions.html [2008, .
Gadh, R. & Prinz, F.B. 1995, "A computationally efficient approach to feature abstraction in design-manufacturing integration", *Journal of engineering for industry: Transactions of the ASME,* vol. 117, pp. 16-27.
Gao, J., Zheng, D.T. & Gindy, N. 2004, "Extarction of machining features for CAD/CAM integration", *International Journal of Advanced Manufacturing Technology,* vol. 24, pp. 573-581.
Guerra Zubiaga, D.A. & Young, R.I.M. 2007, " Information and Knowledge Interrelationships in a Manufacturing Model", *International Journal of Advanced Manufacturing Technology,* , pp. 1-17.
Guerra Zubiaga, D.A. & Young, R.I.M. 2006, " A Manufacturing Model to Enable Knowledge Maintenance in Decision Support Systems", *Journal of Manufacturing Systems,* vol. 25, no. 2, pp. 122-136.
Guerra Zubiaga, D.A. & Young, R.I.M. 2008, "Design of a manufacturing knowledge model", *International Journal of Computer Integrated Manufacturing,* vol. 12, no. 5, pp. 526-539.
Guerra-Zubiaga, D., Donato, L., Ramírez, R. & Contero, M. 2006, "Knowledge Sharing to Support Collaborative Engineering at PLM Environment", *Lecture Notes in Computer Science,* vol. 4333, pp. 86-96.
Gunendran, A.G. 2004, *An information and Knowledge Framework to Support Multiple Viewpoints in the Design for Manufacture of injection Moulded Products*, Loughborough University.
Gunendran, A.G. & Young, R.I.M. 2006, "An Information and knowledge framework for multi-perspective design and manufacture", *International Journal of Computer Integrated Manufacturing,* vol. 19, no. 4, pp. 326-338.
Gunendran, A.G., Young, R.I.M., Cutting-Decelle, A.F. & Bourey, J.P. 2007, "Organising Manufacturing Information for Engineering Interoperability", *Interoperability for Enterprise Software and Applications, I-ESA 2007*, eds. R.J. Goncalves, J.P. Muller, K. Mertins & M. Zelm, Springer-Verlag London 2007, , pp. 587.
Hadar, I. & Hazzan, O. 2004, "On the Contribution of UML Diagrams to Software System Comprehension", *Journal of Object Technology,* vol. 3, no. 1, pp. 143-156.
ISO 10303-214 1997, *Second Committee Draft for Industrial Automation System and Integration-Product Data Representation and Exchange – Part 214: Application Protocol: Core Data for Automotive Mechanical Design Process*, International Standard Organisation.
Li, W.D., Ong, S.K. & Nee, A.Y.C. 2002, "Recognizing manufacturing features from a design-by-feature model", *Computer-Aided Design,* vol. 34, no. 11, pp. 849-868.
Martinez, M.T., Favrel, J. & Ghodous, P. 2000, "Product Family Manufacturing Plan Generation and Classification", *Concurrent Engineering: Research and application,* vol. 8, no. 1, pp. 12-23.
McKay, A. 1993, "The Role of Product Data in Assemble to Engineering, MOSES Project Research Report (Moses-Report-Series-11)", *Presented at IMechE 'Assemble to Order' Seminar*London, November, 1993.
Mckay, K.N. 2003, "Historical survey of manufacturing control practices from a production research perspective", *International Journal of Production Research,* vol. 41, no. 3, pp. 411-426.

Ming, X.G., Yan, J.Q., Wang, X.H., Li, S.N., Lu, W.F., Peng, Q.J. & Ma, Y.S. 2008, "Collaborative process planning and manufacturing in product lifecycle management", *Computers in Industry,* vol. 59, no. 2-3, pp. 154-166.

NX 2008, , *Siemens NX6*. Available: http://www.plm.automation.siemens.com/en_us/products/nx/nx6/ [2008, .

OCL 2006 01-05-2006-last update*, Object Constraint Language*. Available: http://www.omg.org/docs/formal/06-05-01.pdf [2009, 05/02] .

Prasad, B. 1996, "Towards a Functional Design of a Concurrent Information Modelling System", *Computer Modelling and Simulation in Engineering,* vol. 1, pp. 7-29.

Sanderson, S. & Uzumeri, M. 1995, "Managing product families: The case of the Sony Walkman", *Research Policy,* vol. 24, no. 5, pp. 761-782.

Sapuan, S.M. 2001, "A knowledge-based system for materials selection in mechanical engineering design", *Materials & Design,* vol. 22, no. 8, pp. 687-695.

Sarker, B.R. & Mondal, S. 1999, "Grouping efficiency measures in cellular manufacturing: a survey and critical review", *International Journal of Production Research,* vol. 37, no. 2, pp. 285-314.

Schuh, G., Rozenfeld, H., Assmus, D. & Zancul, E. 2008, "Process oriented framework to support PLM implementation", *Computers in Industry,* vol. 59, no. 2-3, pp. 210-218.

Seifoddini, H. & Tjahjana, B. 1999, "Part-family formation for cellular manufacturing: a case study at Harnischfeger", *International Journal of Production Research,* vol. 37, no. 14, pp. 3263-3273.

Shaw, N.K., Bloor, M.S. & De Pennington, A. 1989, "Product Data Models1", *Research in Engineering design,* vol. 1, pp. 43-50.

Stark, J. 2005, *Product Lifecycle Management: Paradigm for 21st Century Product Realisation,* 1st Edition edn, Springer.

Sudarsan, R., Fenves, S.J., Sriram, R.D. & and Wang, F. 2005, "A product information modeling framework for product lifecycle management", *Computer Aided Design,* vol. 37, no. 13, pp. 1399-1411.

Szecsi, T. 2006, "Implementing manufacturing feature based design in CAD/CAM", *IPROMS 2006, Innovative Production Machines and Systems Virtual Conference*.

Tang, D. & Qian, X. 2008, "Product lifecycle management for automotive development focusing on supplier integration", *Computers in Industry,* vol. 59, no. 2-3, pp. 288-295.

Teamcenter 2008, *Teamcenter's Service Oriented Architecture White Paper*, Siemens PLM Software.

Thomas, D. 2003, "UML - Unified or Universal Modeling Language?", *Journal of Object Technology,* vol. 2, no. 1, pp. 7-12.

UML-2 2006, "Unified Modeling Language Version 2: available specifications, Object Management Group (OMG),", .

Yang, M. & Yang, J. 2008, "Machine-part cell formation in group technology using a modified ART1 method", *European Journal of Operational Research,* vol. 188, no. 1, pp. 140-152.

Young, R.I.M., Canciglieri, O., Costa, C.A., Dorador, J.M., Zhao, J. & Cheung, W.M. 2000, "Information Support in an Integrated Product Development System" in *Integrated Design and Manufacturing in Mechanical Engineering*, eds. P. Chedmail, G. Cognet, C. Fortin, C. Mascle & J. Pegna, Kluwer Academic Publishers, Netherlands, pp. 139-146.

Young, R.I.M., Cutting-Decelle, A.F., Guerra, D., Gunendran, A.G., Das, B. & Cochran, S. 2005, "Sharing Manufacturing Information and Knowledge in Design Decision and Support System" in *Advances in Integrated Design and Manufacturing in Mechanical Engineering*, eds. A. Bramley, D. Brissaud, D. Coutellier & C. McMahon, Springer, Netherlands, pp. 173-185.

Methods for the Capture of Manufacture Best Practice in Product Lifecycle Management

Figure Caption

Figures

Figure 1

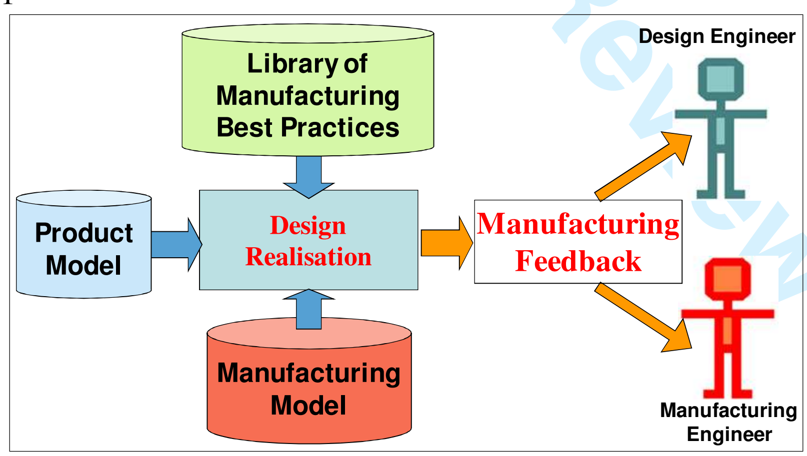


Figure 2

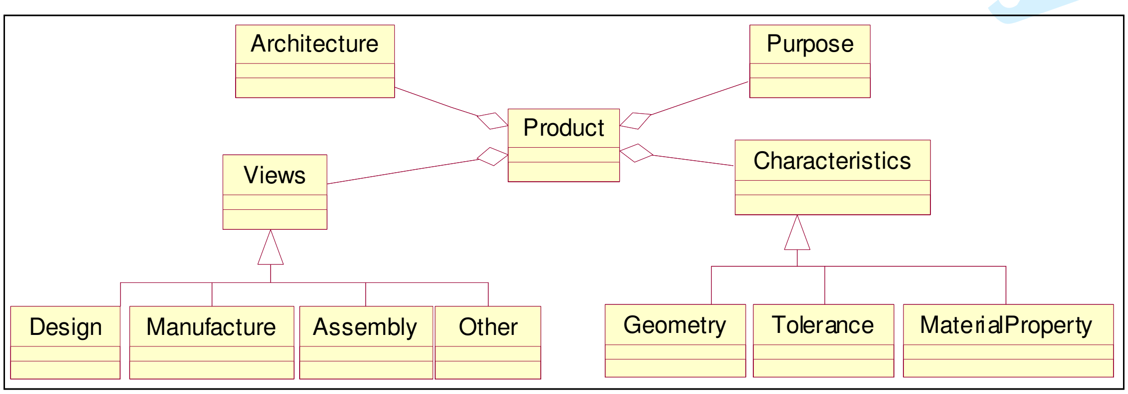


Figure 3

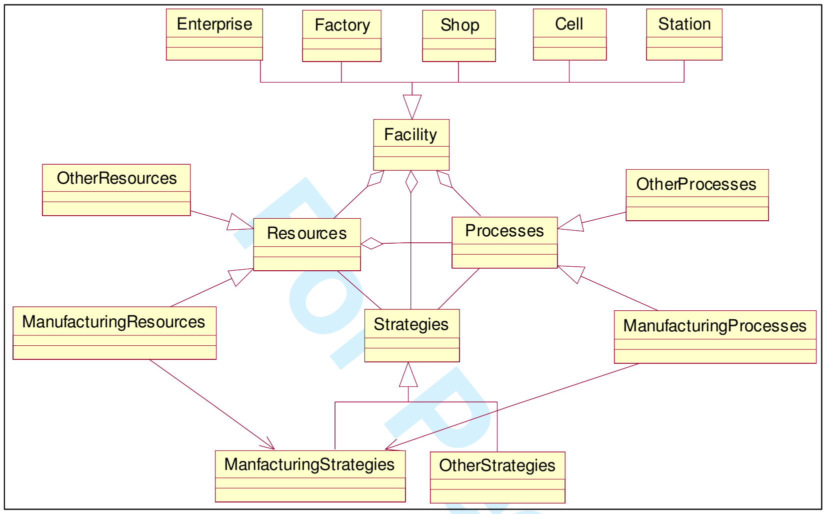


Figure 4

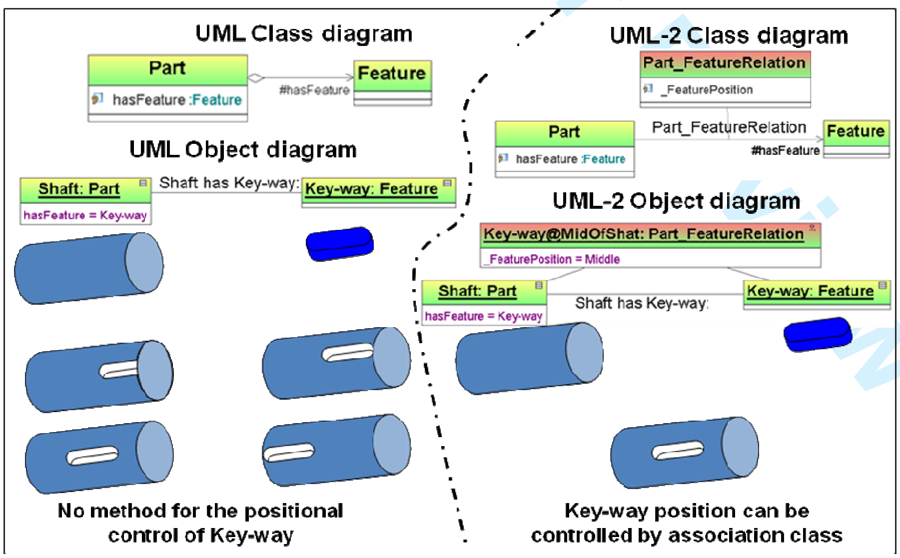


Figure 5

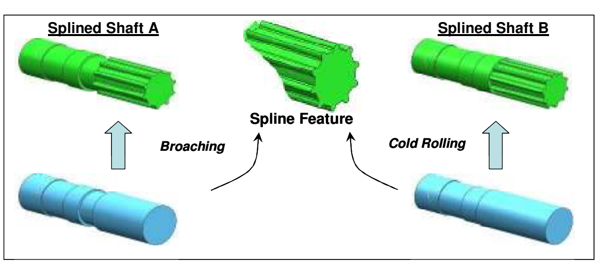


Figure 6

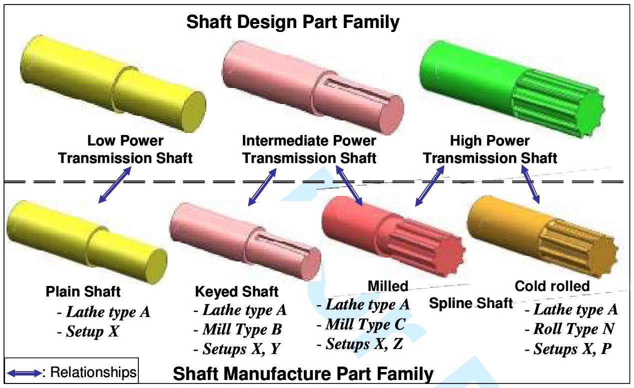


Figure 7

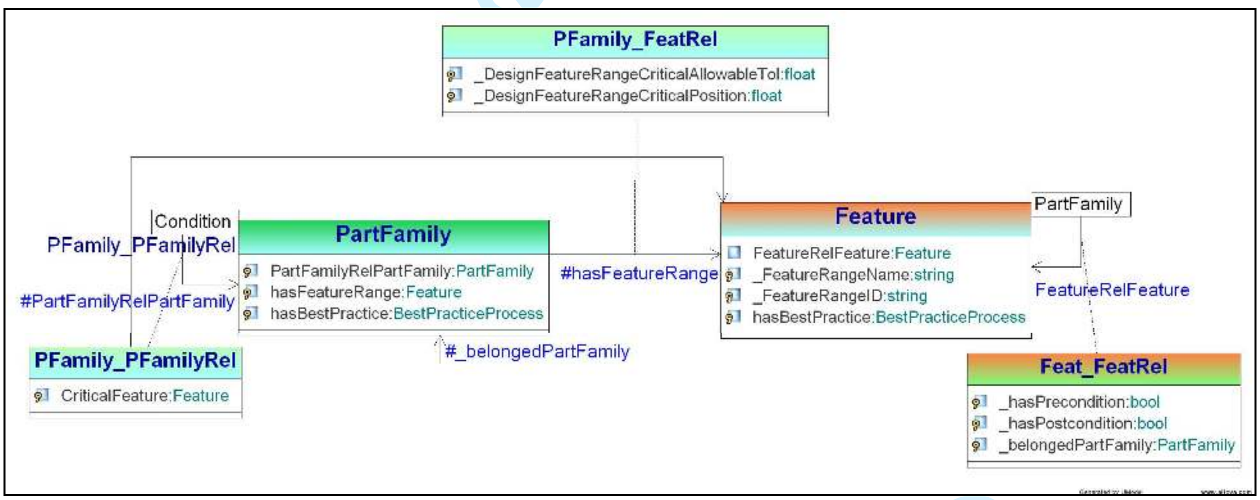


Figure 8

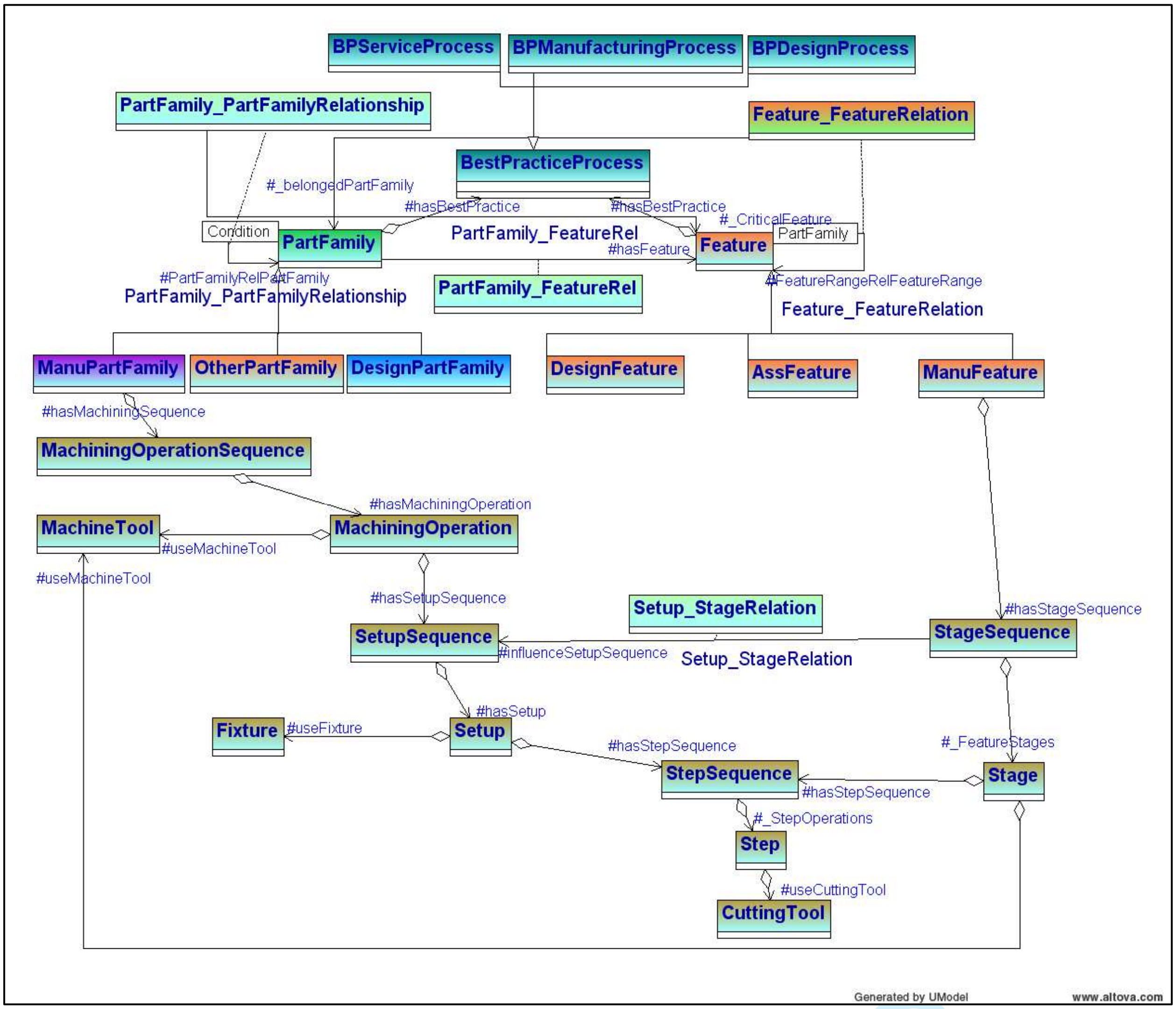


Figure 9

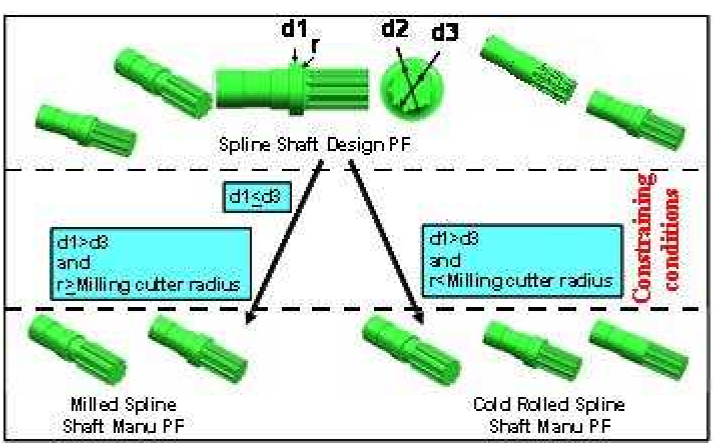


Figure 10

Option	Condition			Possible manufacturing method
	d1≤d3	R≤r	High Strength Spline?	
1	TRUE	TRUE	TRUE	***Cold Roll***
2	TRUE	TRUE	FALSE	***Mill***
3	TRUE	FALSE	TRUE	***Cold Roll***
4	TRUE	FALSE	FALSE	***Mill***
5	FALSE	TRUE	TRUE	***Cold Roll***
6	FALSE	TRUE	FALSE	***Mill***
7	FALSE	FALSE	TRUE	***Cold Roll***
8	FALSE	FALSE	FALSE	***Cold Roll***

Figure 11

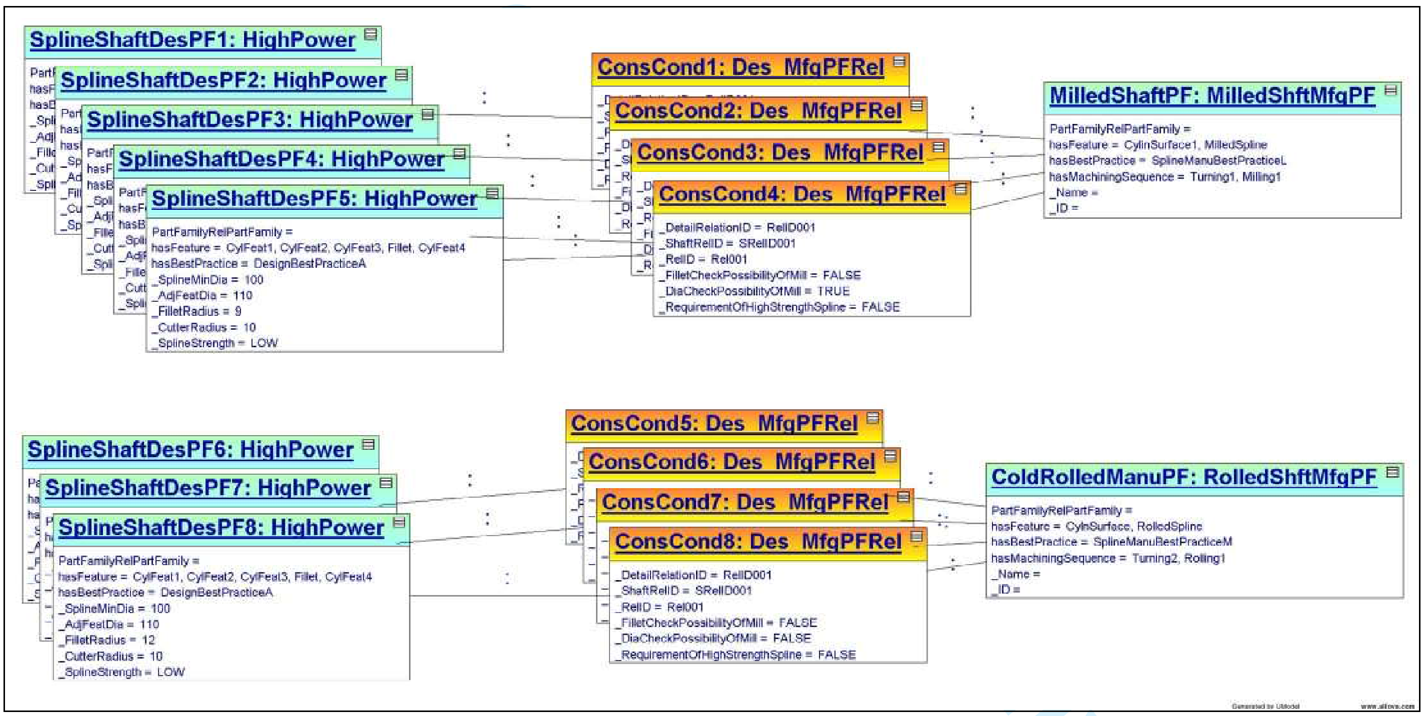
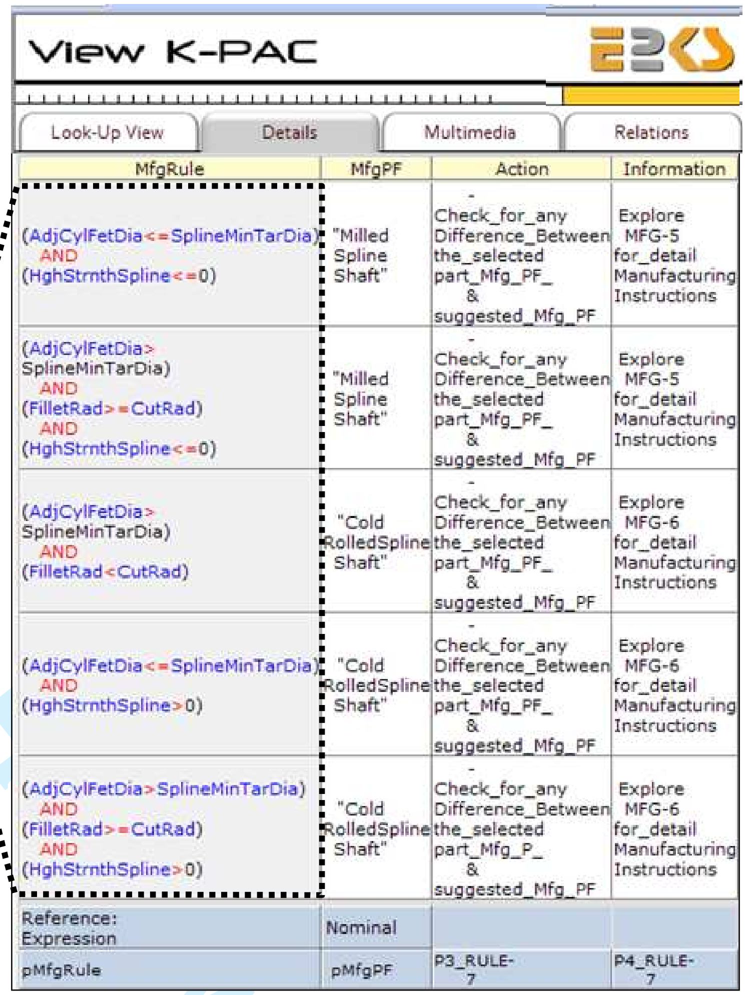


Figure 12

Option	Condition			Possible manufacturing method
	d1≤d3	R≤r	High Strength Spline?	
1	TRUE	TRUE	TRUE	*Cold Roll*
2	TRUE	TRUE	FALSE	*Mill*
3	TRUE	FALSE	TRUE	*Cold Roll*
4	TRUE	FALSE	FALSE	*Mill*
5	FALSE	TRUE	TRUE	*Cold Roll*
6	FALSE	TRUE	FALSE	*Mill*
7	FALSE	FALSE	TRUE	*Cold Roll*
8	FALSE	FALSE	FALSE	*Cold Roll*

Applying conditional relationship rules to E2KS



MfgRule	MfgPF	Action	Information
(AdjCylFetDia<=SplineMinTarDia) AND (HghStrnthSpline<=0)	"Milled Spline Shaft"	- Check_for_any Difference_Between the_selected part_Mfg_PF_ & suggested_Mfg_PF	Explore MFG-5 for_detail Manufacturing Instructions
(AdjCylFetDia> SplineMinTarDia) AND (FilletRad>=CutRad) AND (HghStrnthSpline<=0)	"Milled Spline Shaft"	- Check_for_any Difference_Between the_selected part_Mfg_PF_ & suggested_Mfg_PF	Explore MFG-5 for_detail Manufacturing Instructions
(AdjCylFetDia> SplineMinTarDia) AND (FilletRad<CutRad)	"Cold RolledSpline Shaft"	- Check_for_any Difference_Between the_selected part_Mfg_PF_ & suggested_Mfg_PF	Explore MFG-6 for_detail Manufacturing Instructions
(AdjCylFetDia<=SplineMinTarDia) AND (HghStrnthSpline>0)	"Cold RolledSpline Shaft"	- Check_for_any Difference_Between the_selected part_Mfg_PF_ & suggested_Mfg_PF	Explore MFG-6 for_detail Manufacturing Instructions
(AdjCylFetDia>SplineMinTarDia) AND (FilletRad>=CutRad) AND (HghStrnthSpline>0)	"Cold RolledSpline Shaft"	- Check_for_any Difference_Between the_selected part_Mfg_P_ & suggested_Mfg_PF	Explore MFG-6 for_detail Manufacturing Instructions
Reference: Expression	Nominal		
pMfgRule	pMfgPF	P3_RULE-7	P4_RULE-7

Relationship rules in E2KS knowledge base