In M. Van Sinderen and P. Johnson (Eds.), *'Enterprise Interoperability', Lecture Notes in Business Information Processing, the Proceedings of the Third International IFIP Working Conference on Enterprise Interoperability, IWEI 2011* Vol. 76 (pp. 5-18). Royal Institute of Technology, Stockholm, Sweden, 23-24 March 2011: Springer-Verlag. doi:10.1007/978-3-642-19680-5_3

A Manufacturing Core Concepts Ontology for Product Lifecycle Interoperability

Zahid Usman, Robert Ian Marr Young, Nitishal Chungoora, Claire Palmer, Keith Case, and Jenny Harding

[1]Wolfson School of Mechanical & Manufacturing Engineering, Loughborough University, Loughborough, UK, Post Code LE11 3TU

{Z.Usman, R.I.Young, N.Chungoora, K.Case, C.Palmer3, J.A.Harding}@lboro.ac.uk

Abstract. This paper proposes a manufacturing core concepts ontology (MCCO) aimed at providing support for product life cycle interoperability. The potential focus of the work is interoperability across the production and design domains of product lifecycle. A core set of manufacturing concepts and their key relationships are identified in MCCO. Semantics are captured formally through heavyweight logic using rigorous rules and axioms. Three different levels of specialization have been identified according to the degree of specialization required. Each level provides an immediate route to interoperability for the concepts specialized from that level. MCCO enable knowledge sharing across design and production domains through core concepts. A successful initial experimental implementation has been done to demonstrate the working of MCCO.

Keywords: semantics, core concepts, interoperability, manufacturing ontology, knowledge sharing, product lifecycle, design and production

1 Introduction

Technology and knowledge have been recognized as the key factors for production [1]. Information and Communication Technologies (ICT) have become an integral part of most organizations. Manufacturing organizations have moved from traditional manual drawings and design to Computer Aided Technologies (CAx). Software based approaches like Enterprise Resource Planning (ERP), Manufacturing and Materials Resource Planning (MRP), Product Lifecycle Management (PLM) etc are being employed rapidly. ICT are key to the manufacturing competence, competitiveness and jobs in Europe[2]. With machines replacing men, a mechanism of interoperability is required for machines to communicate across different domains.

Interoperability is “the ability to share technical and business data, information and knowledge seamlessly across two or more software tools or application systems in an error free manner with minimal manual interventions” [3]. To highlight the importance of interoperability a study in 1999 at NIST showed that U.S.$ 1 billion are

spent per year by the U.S. automotive sector alone for solving interoperability problems [4]. The multiples of this amount when other sectors like, services, health care, logistics, telecom, etc are considered from around the globe, the figures would definitely highlight this as a major problem. It has also highlighted the need to minimize the cost incurred in solving interoperability problems.

To make a system interoperable it is of extreme importance to formally capture & incorporate the semantics of concepts. A survey highlighted that almost 70% of total costs of interoperability projects is spent on solving issues of semantic mismatches [5]. Semantics can be captured formally by using ontologies based on rigorously formalized logical theories [6] i.e. heavyweight ontologies. Several definitions of ontology which is a borrowed term from philosophy are found in literature. The most quoted one being "An ontology is an explicit specification of a conceptualization" [7]. The one preferred for this work though is "a Lexicon of the specialized terminology along with some specifications of the meanings of the terms involved"[8]. This definition covers both the lightweight and heavyweight ontologies. The definition and use of concepts are captured by formalizing ontologies with rigorous logic based rules and this is what makes the ontologies formal or heavy weight.

To fulfill the requirements of manufacturing knowledge sharing core ontologies are generally developed from foundation ontologies [9]. In the domains of design and production an extended heavyweight ontological foundation needs to be explored and developed further[10]. A novel method for developing a novel common semantic base in the form of a multilevel heavyweight MCCO to assist sharing knowledge across design and production domains of product lifecycle is proposed in this paper.

2 The Need for a Heavyweight Manufacturing Ontology

2.1 Lightweight Ontologies

Knowledge capturing and sharing has been done partially through database approaches like ERP, MRP, PLM, etc software tools. Limited success has been achieved in providing the information through databases [11] because they have an underlying structure based on lightweight ontologies. Lightweight Ontologies in manufacturing have loosely formalized semantics making concepts open to multiple interpretations. These are also not understood well enough by computers for interoperability. There exists a lack of generally agreed terminology and underlying concepts not being defined explicitly in the manufacturing enterprise architectures area [12].

The current major route to interoperability is to use international standards. But, when it comes to knowledge sharing across different domains they have their own issues. ISO standards relevant to the manufacturing (mainly from ISO TC 184/ SC4) are very focused on their narrow domains of interest e.g.

- ISO 10303-STEP-Standard for The Exchange of Product date model AP239-Product Lifecycle Support (PRODUCT LIFECYCLES), AP224-Feature based manufacturing and mainly machining, AP1-Overview and fundamental principles, etc
- ISO 13584-Part Library (PLIB),

- ISO 15531-industrial MANufacturing management DATa (MANDATE),
- ISO13399-Cutting Tool Standard.
- Etc.

In a very specific and narrow domain of discourse the relevant standards are very useful. Definitions of terms in a narrow domain can be loose since their meanings are already understood by the concerned community. Across a broader domain like product lifecycle where more than one standard are required, interoperability through standards becomes an issue. To share knowledge across standard a common understanding among them is required. Most of the relevant ISO standards have non formalized text based semantics. Consistency lacks not only across standards but even within the same standards as well e.g. the definition of 'component' in ISO standards:

- 'Component' definition in ISO-10303-1 "A product that is not subject to decomposition from the perspective of a specific application"
- 'Component' definition in ISO-10303-AP224-"The component specifies either a Single_piece_part or another Manufactured_assembly used to define an assembly"
- 'Component' definition in ISO 19439:2006. [general] "Entity that is part of, or capable of becoming part of, a larger whole".

The semantics being text based and different within and across standards raise an issue for knowledge sharing through standards. The design and production domains of product lifecycle would require different set of concepts but they need to have a commonly understood formal semantic base for interoperability and knowledge sharing.

2.2 Heavyweight Ontologies Approach

Heavyweight ontologies can potentially overcome this problem of standards and lightweight ontologies. Heavyweight ontologies can formally define concepts, control their use, capture knowledge and provide a route to share across design and production. They offer better reasoning capability compared to the databases with fixed form and formats. Heavyweight ontologies have the potential to provide a rigorous common semantic base. Therefore, research potential is there to work on precisely and rigorously defined manufacturing ontology as a common semantic base. No common semantic base in the form of a heavyweight core manufacturing ontology is available for interoperability across design and production.

Foundation ontologies like DOLCE, SUMO, OCHRE, OpenCyc, BFO provide the first stage of a common understanding. They provide formally axiomatised domain independent set of concept e.g. *AbstractEntity, ConcreteEntity, Endurant, Perdurant*, spatial and temporal concepts, etc. But these are developed to cover everything therefore they are broadly based and generic [13]. Thus, the common semantic base provided by foundation ontologies will be too generic for interoperability across

product lifecycle domains. Thus, concepts from foundation ontologies can serve as a basic backbone for the creation of the more specialized/viewpoint-dependent MCCO.

Heavyweight manufacturing ontologies available as of now are incomplete and do not cover the whole of product lifecycle and need to be completed and developed more [14]. Ontologies for the product lifecycle and manufacturing need to be developed further and tested[13, 15]. Also, the lack of core manufacturing ontologies to provide a common understanding for various strands of manufacturing [12, 16] needs to be overcome. Therefore, heavyweight product ontology capturing the semantics will help focus others on knowledge management issues [14].

3 Manufacturing Core Concepts Ontology

MCCO is formed by identifying a core set of concepts formalized in heavyweight logic. Three different levels with increasing degrees of specialization have identified for formalizing concepts.

3.1 Core Concepts and Relationships within MCCO

Capturing production knowledge requires different types of concepts. A detailed discussion on this is not possible in this paper. The UML diagram in fig 1 summarizes all the categories of concepts, key concepts in each category, and some of the key relationships identified between the concepts.

Features and *Part Family* category contains the most important one. *Features* and *Part Family* concepts and their specialized concepts at three different levels are key to capturing and sharing knowledge in product lifecycle [17, 18]. In this paper the feature concepts are used to show the implementation of the multi level ontology, to show implementation of core concepts, to prove their specialization and their ability to provide a route to sharing knowledge across design and production domains.

3.2 Levels of Specialization of Concepts

The domain specific concepts developed directly from the foundation ontology lack the required level of interoperability. The design and production concepts can be directly specialized from very generic foundation ontology concepts. Foundational concepts enable knowledge sharing only through a level having nothing to do directly with either design or production. Some intermediate concepts are required in addition which are more concerned with the product lifecycle and its sub domains. Design and production layers of concepts can have a common underlying layer which can provide the route to interoperability and knowledge sharing at a more specialized level. Since the layers above the foundation ontology contain concepts relatively more specialized they provide a common base for interoperability at a more specialized level.

Each intermediate level of concepts has a higher degree of specialization with concepts closer to the specific domain. Each level acts as a semantic base for the concepts specialized from that. This gradual specialization of concepts is thus required for providing a route to share knowledge at more specialized levels. Various levels are required to specialize concepts from the foundation to the specific domains. As shown in fig. 2 the number of levels identified in this research work are three [19] based on the degree of specialization required.

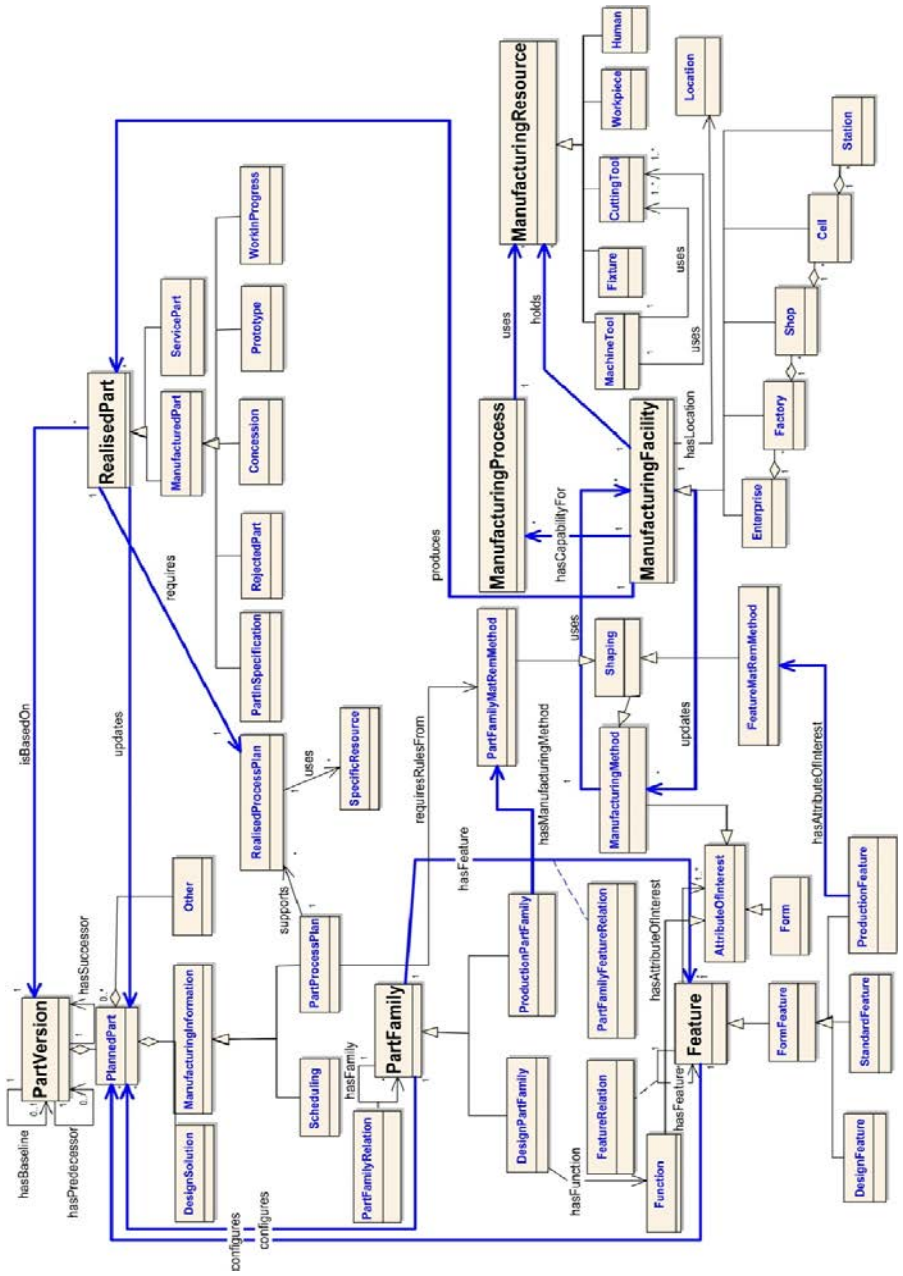

Fig. 1. Light weight representation of manufacturing core concepts ontology (MCCO): Key categories, concepts and their relationship.

Fig 2 summarizes the manufacturing core concepts ontology and its application. It shows two main layers. The bottom layer represents MCCO. While the top layer represents the implementation and evaluation. MCCO at the bottom layer with its three levels of specializations. Top Layer represents the interoperable specialized ontologies and knowledge bases (KB) developed from MCCO. This layer may contain some further specialized concepts according to requirement. The Interoperability across design and production has been tested by querying the relevant knowledge between the two domains. The exploration of core concepts and their relations in MCCO is vital for the successful implementation.

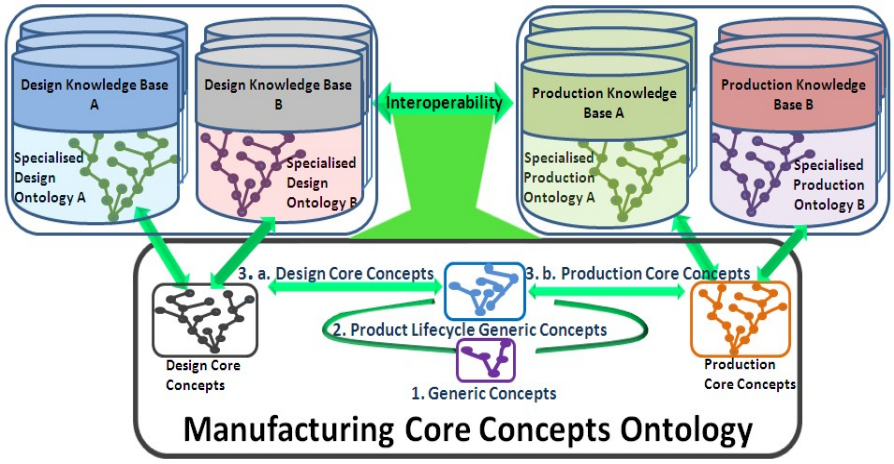

Fig. 2. MCCO, its specialization levels and its implementation scheme

3.2.1 Generic Core Concept Level

This level is to provide a link of product lifecycle concepts to other domains like business domain etc if required. These concepts are more specific as compared to foundational concepts like entity, event, quality, quantity etc. Concepts like *activity, activity occurrence, feature, dimension, tolerance, part, part family* etc, are present in this level. These concepts are applicable to any domain.

3.2.2 Product Lifecycle Generic Core Concept Level

A set of concepts generic to the product lifecycle domains are also required to act as a common level for the specific product lifecycle domains like design and production. Product lifecycle generic concepts are specialized from generic concepts and are applicable to any of the specific product lifecycle domains like design and production but not outside product lifecycle. Concepts like *ProductFeature, ProductPartFamily, GeometricDimensions, GeometricTolerance,* are some of the product lifecycle generic concept level.

3.2.3 Product Lifecycle Domain Specific Core Concept Level

This contains concepts specific individually to each product lifecycle domains like design and production. Concepts like *ProductionFeature* and *ProductionPartFamily*'

are production specialization of product lifecycle generic concepts. Similarly *DesignFeature*' and '*DesignPartFamily*' are design specializations of product lifecycle generic concepts. The design and production concepts can either be used directly for capturing knowledge or can be further specialized to develop customized ontologies.

4 An Example of Concepts Specializations

Specialization of concepts is not a simple process. Most specializations of concepts require other concepts, relations, function and rigorous rules & axioms. The three specialization levels have been elaborated by taking the *Feature* concept and showing its journey through the levels. The formalization of definitions, knowledge capture and sharing are also demonstrated. It is appropriate to use *Feature* as this is one of the key concepts for interoperability and has simpler relations and axioms. Moreover, the ontology is developed more with respect to *Feature.*

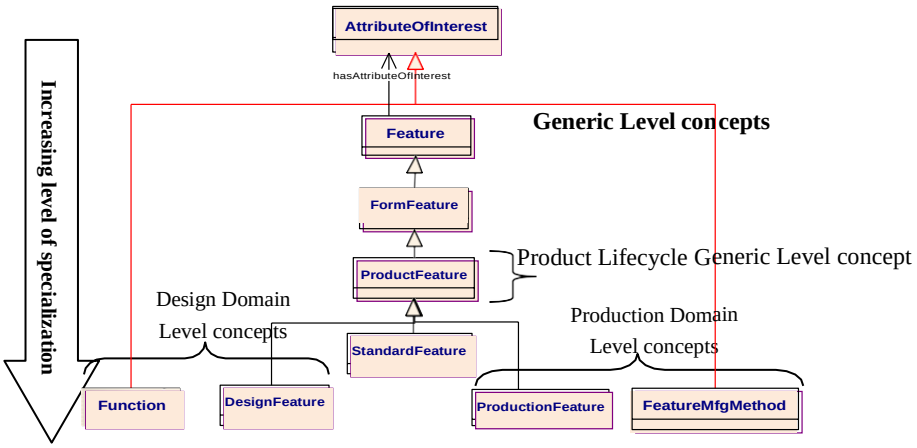

Fig. 3. Feature specializations, lightweight representation

Feature concepts start from the generic "*Feature*" concept. Feature as defined in oxford dictionary is "a distinctive attribute or aspect of something". So '*Feature*' is defined as "anything having an attribute of interest". *Feature* is at the generic level of the ontology. *Feature* thus can be the dark hair of a person, or the ability of a person to run fast etc. The *Feature* in the product lifecycle domain will have some form or shape. This leads to the specialization of *Feature* as a *FormFeature* which should have a *Form* as its *AttributeOfInterest*. A *FormFeature* may be associated to a *Product*. The *FormFeature* thus gets specialized in to a *product feature* where it has an associated *product*. The domain specific concepts *DesignFeature* and *ProductionFeature* can be direct specializations of *from form* or *product feature*. So, *Design Feature* is a *product/form feature* having *function* as a compulsory attribute of interest and *ProductionFeature* is a *ProductFeature* / *FormFeature* having *ManufacturingMethod* as an *AttributeOfInterest*. The concept of *StandardFeature* is generic to both design and production which is a *ProductFeature* / *FormFeature* having both *ManufacturingMethod* and a F*unction* as attributes of interest. Fig 3 shows the lightweight UML representation of *feature* specializations.

5 Formalization of Concepts

To formally define concepts, control their use, capture knowledge and populate facts, heavyweight logic is embedded in MCCO. Common logic (ISO/IEC 24707:200) based Knowledge Frame Language (KFL) provided by Highfleet is used for heavyweight formalization. Axioms and rules are there at all levels of MCCO. The more generic the concepts the lesser the number of constraints on them and the higher is the level of interoperability. The formalization of *Feature* concept and its specializations in KFL are elaborated.

First of all the concepts and relations are declared in the ontology e.g. the concepts '*feature'* and '*AttributeOfInterest*' and relation *hasAttributeOfinterest* which relates a *feature* to its *attribute* are declared in ontology for defining *feature.*
Similarly specializations of *feature* i.e. *FormFeature, ProductFeature, DesignFeature, ProductionFeature* and their respective attributes of interest *Form, Product, Function, ManufacturingMethod*, along with their key relationships are declared in MCCO.

The declaration of concepts, relations and functions is followed by the most important part of formalization i.e. Axiomatization, which makes the ontology heavyweight. Rules and axioms have been divided in two parts i.e. 'Semantic Rules' and 'Knowledge Rules'. Semantic Axioms formally capture and control the meanings of terms. They are subdivided into 'Defining Axioms' and Controlling Axioms'. Defining Axioms formally capture the definition of concepts e.g. to capture the definition of *Feature* following axioms is embedded in ontology.

```
(=>  (Feature ?f)(exists(?AOI)
  (and (AttributeOfInterest ?AOI)
       (hasAttributeOfInterest ?f ?AOI))))
:IC hard "Feature has an Attribute of Interest
```

The above axiom means in simple English "if there is a *feature* ?f then there has to exits an *attribute of interest* ?AOI related to *feature* by the relation *hasAttributeOfInterest*". This captures formally the definition of *feature* and puts it as a hard integrity constraint (IC) in MCCO. This would prevent loading any *feature* without its *attribute of interest*. Similarly definitions of all specializations of *feature* i.e. *FormFeature* at generic level, *ProductFeature* at product lifecycle level and *StandardFeature, ProductionFeature* & *DesignFeature* at domain specific levels can be captured. Other type of semantic axioms and rules i.e. 'Controlling Axioms' are similar and they make the facts assertion fool proof in accordance with formal definitions e.g. a *ProductionFeature* cannot be asserted with *function* as its *attributes of interest* as it belongs to *DesignFeature* and similarly a *DesignFeature* cannot have a *ManufacturingMethod* as its *AttributeOfInterest.*

Knowledge rules are divided into two parts as well. The first type is 'knowledge capturing rules' which formally capture the actual domain specific knowledge e.g. the rule below captures the knowledge relating *NeckWidth parameter* of a *feature* and the *CuttingTool* available to machine that and places it as a soft IC. This IC fires and warns the designer whether the value of *NeckWidth* is out of range with respect to available *CuttingTool*. The facts are still populated because soft ICs are there to warn only.

```
(=>(and(Groove ?g) (NeckWidth ?n)
       (hasParameter ?g ?n) (hasValue ?n (mm ?r1)))
  (inInterval ?r1 (interval in 8 12 in)))
:IC soft "NeckWidth value is out of range (8mm to 12mm)
for machining with standard tooling"
```

The second of the knowledge rules types i.e. 'Inference rules' make the inference of facts from the already loaded facts e.g. A *StandardFeature* has both *function* and *ManufacturingMethod* as its *attribute of interest.* Therefore, it has both *DesignFeature* and *ProductionFeature* in it, and they should be inferred whenever a *StandardFeature* is asserted in knowledge base. The rule below does exactly that when a *StandardFeature* is asserted in knowledge base.

```
(<=(and   (DesignFeature   (DesignFeaturefor ?sf))
   (hasAttributeOfInterest (DesignFeaturefor ?sf)?f)
   (hasAttributeOfInterest (DesignFeaturefor ?sf)?fm))
(and (StandardFeature ?sf) (Function ?f) (Form?fm)
     (hasAttributeOfInterest ?sf ?f)
     (hasAttributeOfInterest ?sf ?fm)))
```

In simple English the above rule implies: if there is a *StandardFeature* ?sf, having *Function* ?f and *Form* ?fm as its attributes of interest then infer a *design feature* 'DesignFeaturefor' having same *Function* and *Form* as those of *StandardFeature*.

6 Experimental Validation of the MCCO

The experimental implementation of MCCO is focused on the critical concepts and their relationships, on the basis that if these can be rigorously defined the rest of MCCo will be straight forward to implement. MCCO is loaded in the Integrated Ontology Development Environment (IODE). The following aspects of ontology have been tested. 1. Capturing of semantics, 2. Controlling concepts, 3. Capturing Domain Knowledge 4. Inferring Knowledge 5. Route to knowledge sharing through MCCO.

6.1 Testing Definition and Specialization of Feature Concepts

This experiments test the assertion *ProductionFeature.* This verifies two things 1. Definition of *ProductionFeature* has been captured, 2. *ProductionFeature FormFeature* is indeed a specialization of *ProductFeature* which is a specialization of *FormFeature* which is a specialization of *Feature*.

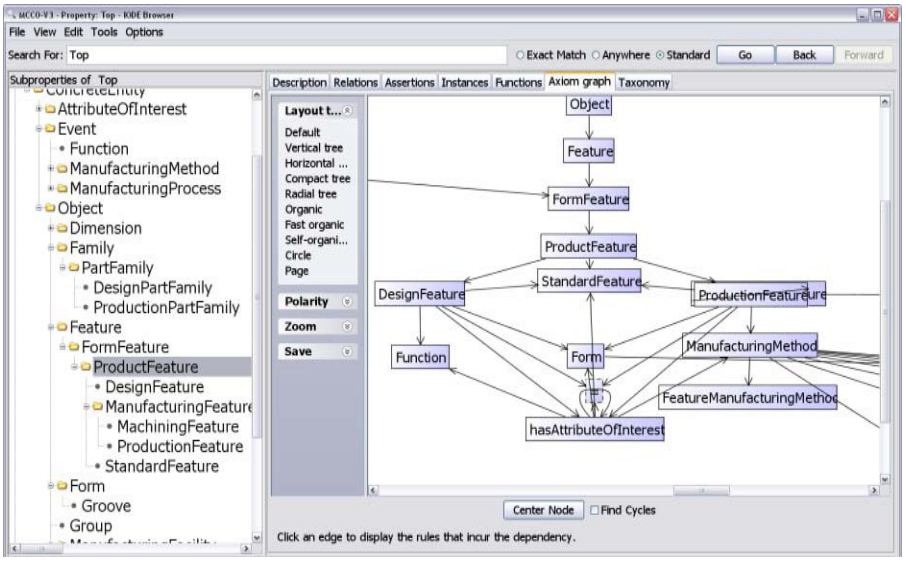

Fig. 4. A portion of MCCO ontology in IODE showing feature and its specializations

According to the definition, a *ProductionFeature* has a *ManufacturingMethod* as its *AttributeOfInterest*. First the *ProductionFeature* is asserted without *ManufacturingMethod* to test the definition. As shown in fig 5 the assertion has been cancelled. The IC cancels the assertion and notifies the user that a *ManufacturingMethod* may be defined for it. But this is not the only IC which has fired. ICs from generic concepts *Feature* and *FormFeature* as well of product lifecycle generic concept *ProductFeature* have also fired. This is due to the specialization.

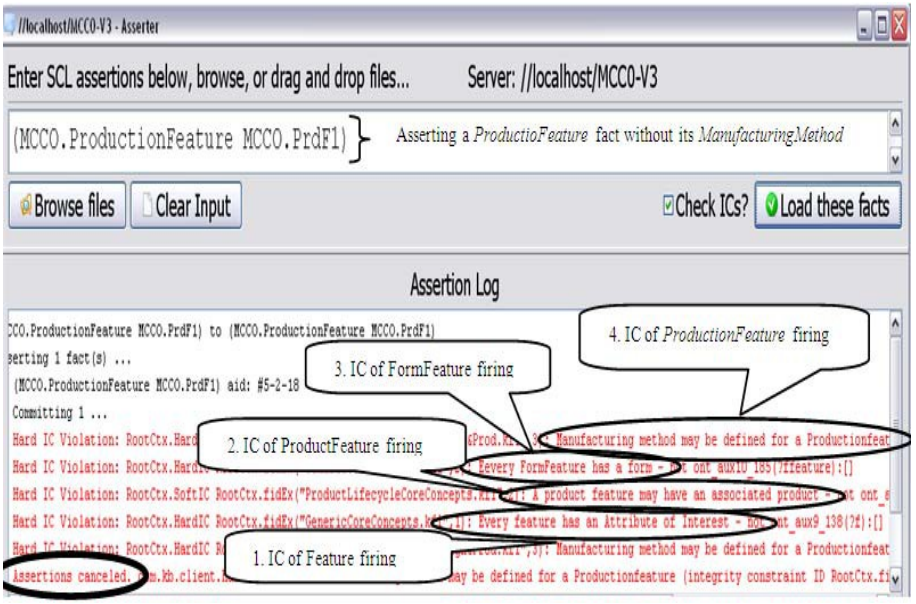

Fig. 5. Asserting a production feature without manufacturing method

A *ProductionFeature* inherits ICs from all three levels of specialization.

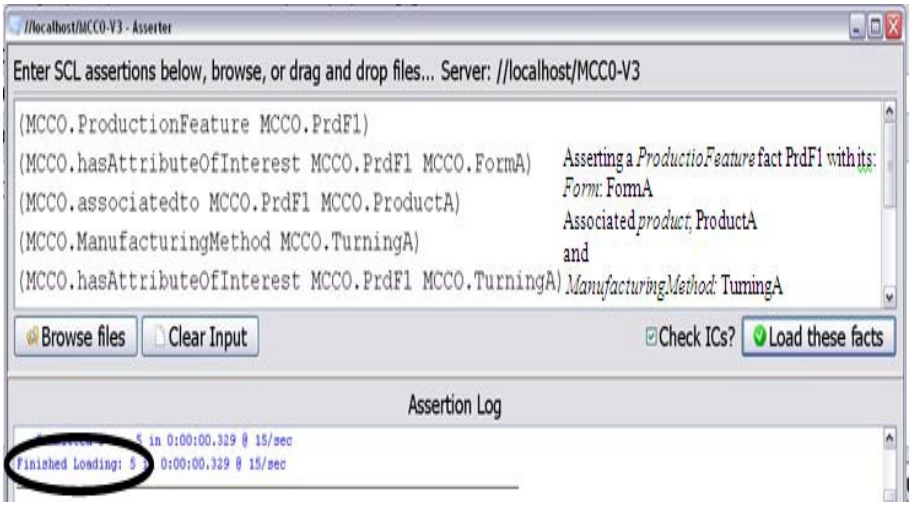

Fig. 6. *ProductionFeature* asserted with all its A*ttributesOfInterest* and associated

ProductionFeature is specialized from the product lifecycle generic concepts *ProductFeature* which is specialized from the generic concept *FormFeature* which in turn is specialized from the generic concept *Feature*. After that a production feature is asserted with *ManufacturingMethod* 'TurningA' as its *AttributeOfInterest*, 'ProductA' as it associated *Product*, and 'FormA' as it other *AttributeOfInterest.* The assertion is accepted as shown in fig 6. This is because all the ICs coming from all three levels have been taken care of and the assertion satisfies all of them. This shows that the definition of *ProductionFeature* has been successfully captured. Firing of ICs from all levels confirms the integrity driven specialization of concepts at different level.

6.2 Testing Inference of Knowledge and Route to Knowledge Sharing

A sample of DesignFeature, ProductionFeature and StandardFeature facts have been asserted in the KB having a common form. A query is run find out the *DesignFeature* and *ProductionFeature* as well as *StandardFeature* having common form. As shown in fig 7 Form 'CirGroove' is the common form for all feature facts. Once a *ProductionFeature* with common a *Form* as that of a *DesignFeature* is identified, the knowledge about the *ProductionFeature* can be queried and fed back to the *DesignFeature* with the same form. Common form comes from *FromFeature* thus the concepts *Form* and *FormFeature* provided the route to interoperability between design and production features. The *DesignFeature* 'DesignFeaturefor StdGroove' and *ProductionFeature* 'ProductionFeaurefor StdGroove' are inferred from the *StandardFeature* StdGroove. So this also showed that knowledge can be inferred automatically using MCCO rules and axioms.

Fig. 7. Route to knowledge sharing and inferred facts

7 Conclusion

The focus of the work is to provide an ontological foundation for sharing manufacturing knowledge across production design and production. However, the underlying technological base can provide an understanding of production quality, cost and timescales. This has a potential to provide further linkages to a business perspective. This approach does not ensure full interoperability but it does ensure the understanding of the extent to which interoperability is possible. A core set of concepts has been formally defined in MCCO and these concepts have been used to capture as well as share production knowledge. Three different levels of specialization i.e. generic, product lifecycle generic and domain specific level have been identified for MCCO. Feature concepts have been successfully specialized at the three levels and other concepts in MCCO are being specialized. The behaviors of concepts have been controlled. Knowledge has been captured and inferred successfully using core concepts and the expressive power of common logic. The ability of core concepts to provide a route to communicate, identify and share the knowledge across design and production domains has been demonstrated thoroughly the *Feature* concepts.

Future research direction is aimed to explore a more detailed level of Interoperability between design and production features at parametric level. Manufacturing method for features and part families from knowledge sharing context is being explored using a meta-level underpinning. Actual industrial implementation remains to be explored. The formalization of concepts in MCCO which are borrowed from various ISO- standards and encoded in common logic based formal definitions. MCCO can be extended to explore interoperability across other product lifecycle domains like services, operation and disposal.

Acknowledgements. This research work is funded by the Innovative Manufacturing and Construction Research Centre (IMCRC) under the Interoperable Manufacturing Knowledge Systems (IMKS) project (IMCRC project 253). The authors would also like to thank the research team in IMKS project for their support and cooperation.

References

1. Frankovič, B. and I. Budinská. The Role of Ontology in Building of Knowledge Systems for Industrial Applications. in 4th Slovakian-Hungarian Joint Symposium on Applied Machine Intelligence,. Herlany, Slovakia (2006)
2. ICT for Manufacturing, Report of Meeting with Group of Representatives of Five Expert Panels, Brussels (2005)
3. Ray, S. and A. Jones, Manufacturing interoperability. Journal of Intelligent Manufacturing, 2006. **17**(6): p. 681-688 (2006)
4. Brunnermeier, S.B. and S.A. Martin, Interoperability Cost Analysis of the U.S. Automotive Supply Chain, National Institute of Standards and Technology, U.S.A (1999)
5. Bussler, C., et al., Context Mediation in the Semantic Web: Handling OWL Ontology and Data Disparity Through Context Interchange, in Semantic Web and Databases, Springer Berlin / Heidelberg. p. 140-154 (2005)
6. Uschold, M. and M. Gruninger. Ontologies and Semantics for Seamless Connectivity (2004)
7. Gruber, T., R., Toward principles for the design of ontologies used for knowledge sharing, Academic Press, Inc. p. 907-928 (1995)
8. Schlenoff, C., et al., ISO-18629 The Process Specification Language (PSL): Overview and Version 1.0 Specification," NISTIR 6459, National Institute of Standards and Technology, Gaithersburg, MD (2000)
9. Young, R.I.M., et al., Manufacturing knowledge sharing in PLM: a progression towards the use of heavy weight ontologies. International Journal of Production Research, **45**(7): p. 1505–1519 (2007)
10. Chungoora, N. and R.I.M. Young, The configuration of design and manufacture knowledge models from a heavyweight ontological foundation. International Journal of Production Research (2010)
11. Abramovici, M. and O.C. Sieg, Status and development trends of product lifecycle management systems, in IPPD conference, Wroclaw (2002)
12. Chen, D., G. Doumeingts, and F. Vernadat, Architectures for enterprise integration and interoperability:Past, present and future. Computers in Industry, **59**: p. 647-659 (2008)
13. Borgo, S. and P. Leitão. Foundations for a core ontology of manufacturing, Bragança, Portugal (2007)
14. Lee, J.-H. and H.-W. Suh, Ontology-based Multi-layered Knowledge Framework for Product Lifecycle Management. Concurrent Engineering, **16**(4): p. 301-311 (2008)
15. Zhou, J. and R. Dieng-Kuntz, Manufacturing Ontology Analysis and Design: Towards Excellent Manufacturing. IEEEXplore, p. 39-45 (2004)
16. Leimagnan, S., et al. MASON: A proposal for an ontology for manufacturing domain. in Proceedings of the IEEE Workshop on Distributed Intelligent Systems: Collective Intelligence and Its Applications (DIS'06), IEEEXplore (2006)

17. Gunendran, A., An Information and Knowledge Framework to Support Multiple Viewpoints in the Design for Manufacture of Injection Moulded Products, PhD Research Thesis, Loughborough University (2004)
18. Gunendran, G. and B. Young, Methods for the Capture of Manufacture Best Practice in Product Lifecycle Management, in International Conference on Product Lifecycle Management 2008, Inderscience Publishers (2008)
19. Usman, Z., et al., A Manufacturing Foundation Ontology for Product Life Cycle Interoperability, in Interoperability for Enterprise Software and Applications, Springer: Coventry, UK (2010)