

This item was submitted to Loughborough's Institutional Repository (<https://dspace.lboro.ac.uk/>) by the author and is made available under the following Creative Commons Licence conditions.



**CC creative commons**  
COMMONS DEED

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

**BY:** **Attribution.** You must attribute the work in the manner specified by the author or licensor.

**Noncommercial.** You may not use this work for commercial purposes.

**No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

# **Manufacturing knowledge sharing in PLM: a progression towards the use of heavy weight ontologies**

R.I.M. YOUNG\*§, A.G. GUNENDRAN§, A.F. CUTTING-DECELLE†, M. GRUNINGER‡

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

† Ecole Centrale de Paris, Laboratoire de Genie Industriel, France. Email: afcd@skynet.be

‡ University of Toronto, Department of Mechanical & Industrial Engineering, Canada. Email: gruninger@mie.utoronto.ca

Correspondence: \*R I M Young. Email: R.I.Young@lboro.ac.uk

## Abstract

The drive to maximise the potential benefits of decision support systems continues to increase as industry is continually driven by the competitive needs of operating in dynamic global environments. The more extensive information support tools which are becoming available in the PLM world appear to have great potential but require a substantial overhead in their configuration. However, sharing information and knowledge in cross-disciplinary teams and across system and company boundaries is not straightforward and there is a clear need for more effective frameworks for information and knowledge sharing if new product development processes are to have effective ICT support.

This paper presents a view of the current status of manufacturing information sharing using light-weight ontologies and goes on to discuss the potential for heavy-weight ontological engineering approaches such as the Process Specification Language (PSL). It explains why such languages are needed and how they provide an important step towards process knowledge sharing. Machining examples are used to illustrate how PSL provides a rigorous basis for process knowledge sharing and subsequently to illustrate the value of linking foundation and domain ontologies to provide a basis for multi-context knowledge sharing.

## INTRODUCTION

The relentless drive to improve manufacturing competitiveness in terms of better, faster, cheaper products has led to a range of approaches from companies including Concurrent Engineering, Lean Manufacture and Agile Manufacture. This, combined with the effects of manufacturing globalisation, is leading to greater and greater demands on information systems solutions to support the decision making processes in industry.

Although web based tools are improving global access to IT support solutions, these solutions tend, in the main, to be focused on discrete areas in the business and do not cover the broad range of information and knowledge needed to meet the needs of diverse teams of engineers working on problems in a dynamically changing manufacturing environment (Young et al, 2005). Limited success in offering some breadth of information support is beginning to appear in ERP, PLM and CRM tools (Abramovici 2002). The combination of these types of tool begins to offer the potential to provide sources of information that decision makers can begin to draw on to offer rapid and effective information support. Again, however, there are significant problems: the overhead in constructing such information systems is very high; the resulting systems tend to be inflexible to change; the ability to share diverse views of similar information, as is needed by teams of engineers, cannot as yet be supported; the ability to share information across competitive software tools is limited. It is in this last area where the identification of an effective basis for the provision of shared meaning is especially important.

Providing shared meaning is where ontological approaches have been pursued. Many research groups have pursued aspects of ontological research in many different research areas from philosophy to engineering, from web based semantics to database application inter-operability. There are many competing definitions for and approaches to ontology (Gomez-Perez 2004) and it is especially important to consider carefully what is expected from a particular approach.

There have been many definitions of ontology since the early work at Stanford (Gruber 1993). Here we follow the definition (ISO 18629-1) which states that ontology is “a lexicon of specialised terminology along with some specification of the meaning of terms in the lexicon”. A lexicon is a “set of symbols and terms” and an axiom is a “well-formed formula in a formal language that provides constraints on the interpretation of symbols in the lexicon of a language”. In simple terms we describe an ontology as providing a basis for shared meaning.

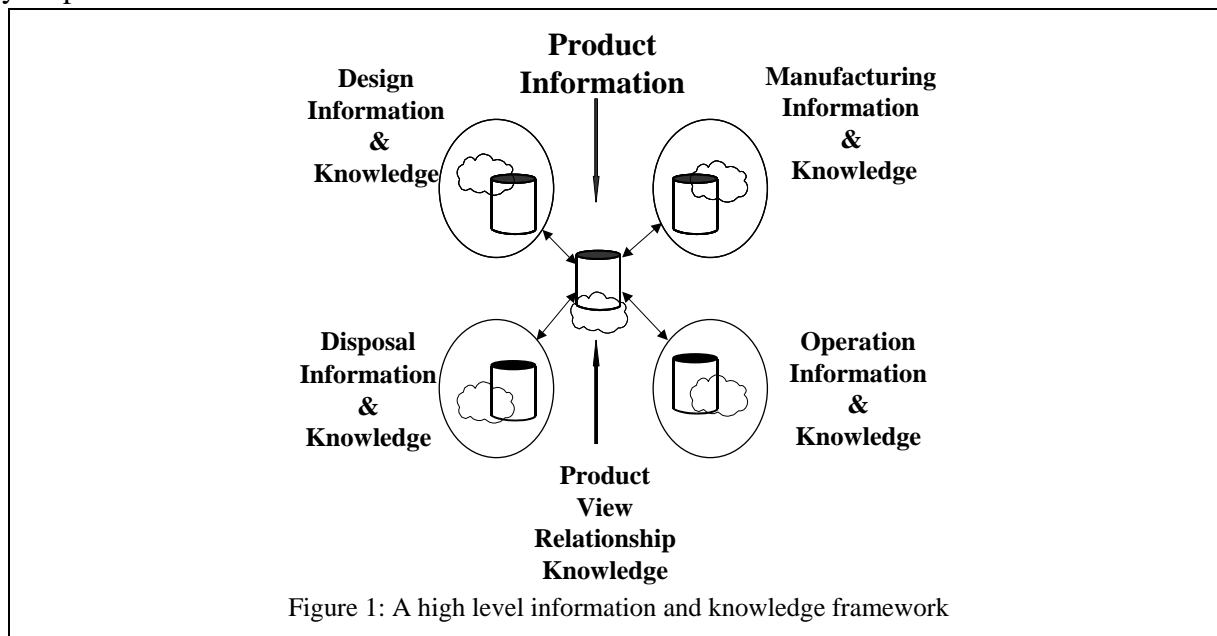
The greatest difference of significance between ontological approaches is the basis upon which the sharing of meaning is made in relation to the level of rigour with which terms are defined. This is heavily influenced by the ontological approach that is taken. Many papers arguing for an ontological approach are largely structurally based with textual definitions for terminology. These so called light-weight ontologies assume that the definition of terms is easily understood (Gomez-Perez 2004). However a more mathematically rigorous, or heavy-weight approach, is needed to provide greater confidence that the real meaning behind terms coming from different systems is the same. One of these heavy weight approaches, targeted at the discrete manufacturing domain, is the Process Specification Language (PSL) (ISO 18629-1).

This paper reflects on recent research into information support systems for new product development, with a particular focus on manufacturing information and knowledge, in order to highlight progress and to propose potential solutions for the future. In particular it compares commonly used light-weight approaches with emerging heavy-weight approaches, highlighting the potential for rigorous foundation ontologies such as PSL to support effective knowledge sharing.

## AN INFORMATION SHARING FRAMEWORK FOR DECISION SUPPORT

In this paper we focus on Product Lifecycle Management and especially the design and manufacturing part of the life cycle, with particular emphasis on the information and knowledge infrastructures which are needed to support manufacturing decision making. The need is to support globally dispersed teams involved in design and manufacture where diverse types of information are needed to support decision making, covering not just typical geometric and product information but also manufacturing resource and process knowledge, potential supply chain capability, knowledge of what has been done in the past as well as potential legislation, catalogue data and standards which may offer constraints on the decisions to be made.

While PLM offers a range of tools to support the business including the ability to manage workflows, the heart of an effective PLM system is the database at its core. The issue then is how to structure the databases at the heart of PLM in order to ensure all users have access to effective information support. The importance of product models has long been recognised in providing a core of product information to support decisions (Krause 1993). However, given the design, manufacture, operation and disposal aspects of the life cycle it is also important to support decisions with non product specific information focused on each of these areas of the life cycle. For example the manufacturing area of the life cycle should be able to offer support on manufacturing process capability and information on suppliers with resources capable of meeting specific capability requirements. This leads to the concept, illustrated in figure 1, which shows a product model at the heart of a data/knowledge based environment, with further bases of data and knowledge to support each of the life cycle phases.



This top level framework for information and knowledge can then be used as a basis to develop formal information and knowledge classifications for each of the life cycle phases. Figure 2 illustrates the framework from figure 1 at the heart of a product life cycle representation. It highlights in particular two Unified Modeling Language (UML) class structures which start to provide contexts for manufacturing knowledge sharing. The first is for manufacturing capability models which can be used to build representation of an enterprise's manufacturing ability. The second is a product model representation which goes beyond typical representations of product characteristics such as geometry and product architecture, to include other key classes relating to product purpose and views which enable life cycle contexts to be captured.

The following sub-sections explore manufacturing information organisation firstly from a life-cycle context and then from a product view context. Flexible environments which can support knowledge sharing across competing systems boundaries must meet all these requirements for effective manufacturing knowledge sharing.

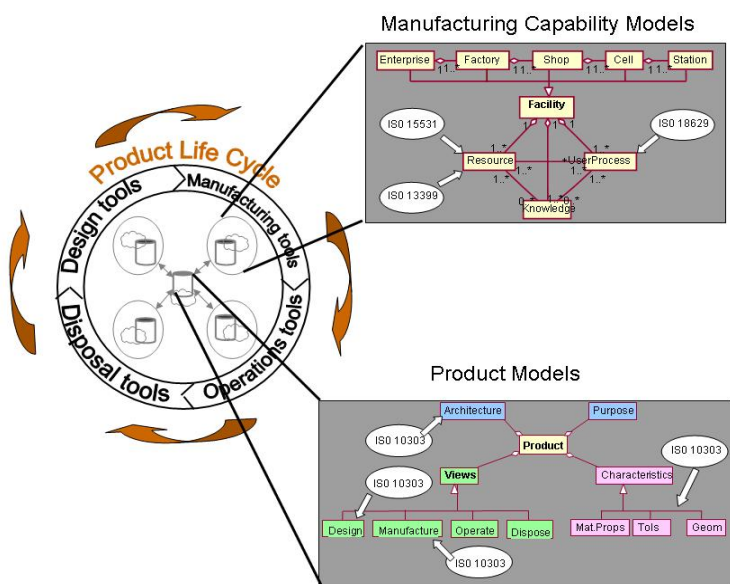


Figure 2: Product Information and Manufacturing Contexts in the Life Cycle

### ***Manufacturing Information from a Life Cycle Context***

Most PLM work appears, for historical reasons, to focus on a design perspective with at best the association of manufacturing documents for component parts. However, it is important to note that businesses have core information and knowledge on their manufacturing capability, independent of any specific product. A Manufacturing Model, representing this core manufacturing capability is just as important as any product model from the manufacturing perspective of the life cycle. A manufacturing model should identify process and resource specifications, potential methods of manufacture and best practice for manufacturing (Young 2005). These provide information which can be used in the new product introduction process. Any resulting manufacturing planning decisions can then be captured as part of a product description.

The importance of a manufacturing model is that it not only provides a common source of information to support design decisions, but it focuses the core competencies of the business so that as new understanding is generated during product manufacture, the model can be updated for future benefit. It therefore provides a clear integration link between PLM as a provider of manufacturing information and shop floor manufacturing systems in terms of data collection and feedback.

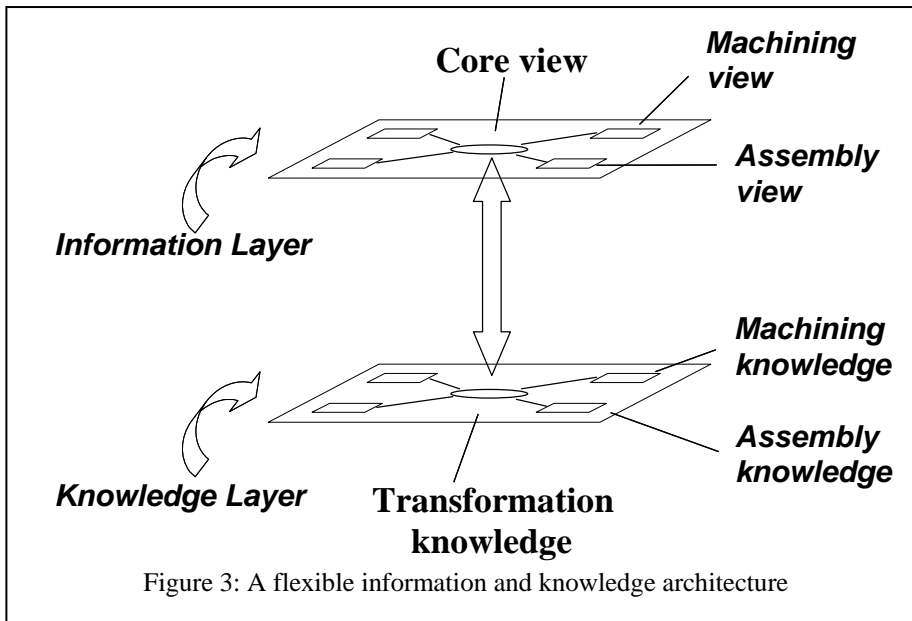
New product development involves multiple supply chain configurations in the production of products, where each manufacturing facility has its own manufacturing capability, its own software tools and its own views on how to effectively configure information systems to support its decision making needs. It is therefore highly unlikely that each company will share a common framework for the capture of their manufacturing information. However, if they are to collaborate effectively then it is critical that they have a basis upon which they can share information (Srinivas 2005, Cutting-Decelle et al 2004).

### ***Manufacturing Information from a Product Context***

Product development is typically a team based exercise where members of the team all require similar but different sets of information in order to meet their specific tasks. The interpretation of product information in a form suitable for manufacturing decision making has typically been pursued through the use of features technology and part family variants (Gunendran 2004). The results of manufacturing planning can then be captured in process plans for a specific product.

Features approaches are problematic in that they capture only a single context of information e.g. a machining feature is specific only to machining, it will not relate to assembly or to casting. It may nonetheless be useful if it provides a focused and practical set of shapes which a design team can use. There is a need for PLM systems to be able to support multiple views of information and the relationships between them. For example, a major step forward in potential functionality for manufacturing engineers would be achieved if part functional requirements could be linked to the relevant manufacturing views such as assembly, casting, forging, machining, heat treatment, grinding etc. There is therefore a requirement for methods that can support the construction of relationships between different manufacturing views. One way of dealing with this requirement is to utilise a knowledge layer which underlies an information view layer as illustrated in figure 3.

However, these ideas have been developed into solutions based on the use of UML. They therefore use a light-weight ontological approach which is inappropriate for inter-system interoperability. Our work is now addressing how languages like PSL can be used to provide a level of rigour into these concept definitions such that they can be shared with other systems as they develop.



## FOUNDATION ONTOLOGIES AS THE ROUTE TO SHARED MEANING

### *Current methods for information systems development*

The approach used in our work on developing experimental integrated systems has been to use IDEF0, IDEF3 and UML (Dorador 2000, Costa 2001). These provide a complementary set of tools to work through from requirements definition to systems design and, although there are a number of alternative methods, these provide a typical set of system design tools. This sort of approach works well in the development of single systems as it provides a logical, progressive approach which enables the various system relationships to be addressed.

One key area where such approaches are lacking is in the formal definition of the terms used to define classes and relationships in the system. Terms are defined using text based descriptions. This is critical when we need multiple systems to interoperate. Computer systems, which aim to share information, require not only a commonly agreed data structure, but common semantics before they can share meaningfully. Achieving this is fraught with problems. Firstly, independently developed applications are bound to be inconsistent with one another. If they are to inter-operate it is necessary to find a common core model. As applications develop they may add functionality that is no longer consistent with a common core model. This requires flexibility in approach. Agreeing a basis for a common core model is not easy and defining methods by which flexibly extensible models can be achieved is even more difficult. Part of the problem here is that the rate of change in systems is very high and part is identifying effective methods by which shared meaning can be defined.

Perhaps one of the best known efforts to provide common models which can provide the basis for data exchange and sharing has been through the work of the ISO in the ISO TC 184 / SC4 committee developing the ISO 10303 standard amongst others. This has been in progress for over 2 decades now, has made some significant progress, but still has much work to do. It is interesting to note that the early work focused on geometry sharing, an area where the meaning of the terminology is shared and clearly understood by the

experts involved. However as we move to areas of information where the terminology is less rigorously defined in product design and manufacture we start to have problems with the definition of terms. Either common terms are used to mean different things or different terms are used to mean the same thing which leads to potentially substantial interoperability problems (Ray 2004). The set of terms listed below (Michel 2005) illustrates a sub-set of definitions drawn from international standards which provide definitions for “process”. It is interesting to note that when comparing these definitions it is difficult with some of the definitions to tell whether they have the same meaning or not!

## **126 PROCESS**

### **126.1 PROCESS (ISO/CEN 19439)**

partially ordered set of activities that can be executed to achieve some desired end-result in pursuit of a given objective

### **126.2 PROCESS (ISO 15531-1; ISO 18629-1)**

structured set of activities involving various enterprise entities, that is designed and organised for a given purpose.

NOTE The definition provided here is very close to that given in ISO 10303-49. Nevertheless ISO 15531 needs the notion of structured set of activities, without any predefined reference to the time or steps. In addition, from the point of view of flow management, some empty processes may be needed for a synchronisation purpose although they are not actually doing anything (ghost task).

### **126.3 PROCESS (ISO 10303-49)**

a particular procedure for doing something involving one or more steps or operations. The process may produce a product, a property of a product, or an aspect of a product.

This problem highlights the need for formal, rigorous methods for the definition of semantics. The following sub-sections use a simple manufacturing scenario to compare the use of a well known method for semantic definition, the web ontology language, with PSL, which has been developed with manufacturing processes in mind. The example focuses on sequencing of processes as this is a critical aspect of manufacturing.

### ***A Hole Making Scenario to compare ontological approaches***

This scenario uses the simple hole illustrated in figure 4 which uses the following assumptions:

The hole is to be machined in a mild steel block.

The hole has a diameter of 10 mm.

The hole has a diameter tolerance range of 0.02 mm

The top face of the block has already been machined.

The length of the hole is not significant



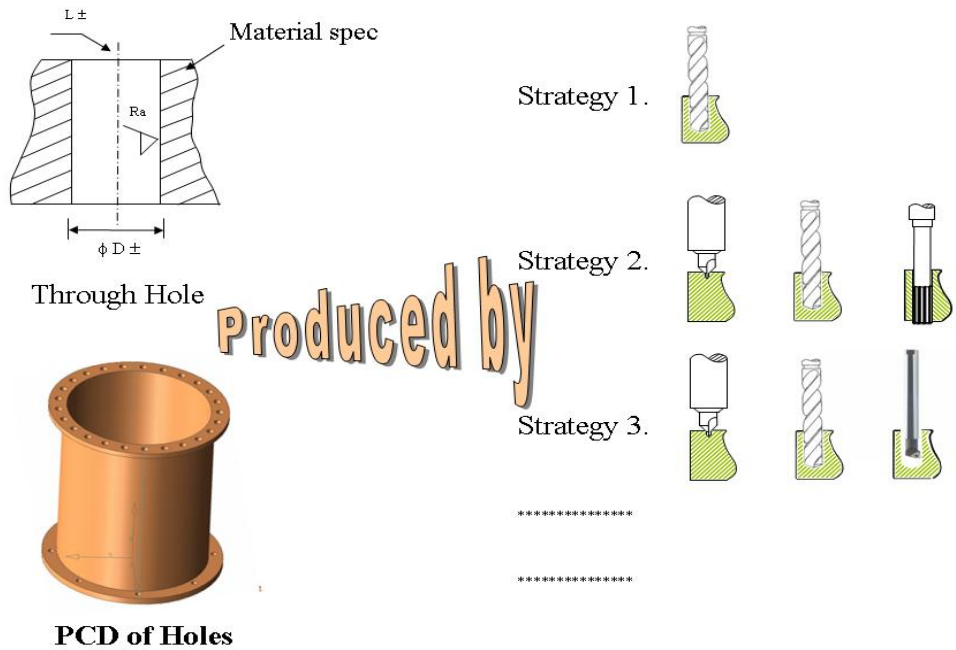


Figure 4: Hole making strategies

A number of hole making strategies are available to us for making holes, some of which are illustrated in figure 4:

- (a) drill
- (b) centre drill, drill, ream
- (c) centre drill, drill, bore

The tolerance requirements cannot be met by option (a). Both (b) and (c) are possible but let us assume we select option (b) as the simpler processing option.

Let us consider the semantics which we want to convey from “centre drill, drill, ream”. Each process has its own specific implied semantics, e.g. ‘ream’ implies a pre-existing hole with approximately a 0.2mm wall thickness and also implies a post condition tolerance range of 0.025mm. However, here we are interested in the implied semantics of the sequence centre-drill, drill, ream. This is a sequence of processes where ‘drill’ follows ‘centre drill’ and ‘ream’ follows ‘drill’. It is also important to register that it is allowable to intersperse other processes between these.

We then consider a PCD of holes also as shown in figure 4 and assume that each hole requires the same ‘centre drill’, ‘drill’, ‘ream’ sequence. Typically all ‘centre drill’ operations would be performed together, followed by all ‘drill’ operations and finally followed by all ‘ream’ operations. It is important that the semantics of the process sequencing supports this level of constraint.

### ***An OWL-S representation***

OWL-S is an extension to OWL which provides a sequencing capability. However, although this allows us to define a process sequence as shown in the OWL-S description below, it does not provide any formal way of

checking the sequence order or the constraints on the sequence. That is, using the OWL-S description we could equally well have a sequence of 'ream', 'centre drill', 'drill'. Similarly we have no knowledge of whether we can intersperse other operations between the existing operations in the sequence. The axiomatization using OWL-S is also shown below.

### OWL-S Process Description:

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:process-1.0="http://www.daml.org/services/owl-s/1.0/ Process.owl#">
<process-1.0:CompositeProcess rdf:ID="ReamHole">
<process-1.0:composedOf>
  <process-1.0:Sequence>
    <process-1.0:components rdf:parseType="Collection">
      <process-1.0:AtomicProcess rdf:about="#CentreDrilling"/>
      <process-1.0:AtomicProcess rdf:about="#Drilling"/>
      <process-1.0:AtomicProcess rdf:about="#Reaming"/>
    </process-1.0:components>
  </process-1.0:Sequence>
</process-1.0:composedOf>
</process-1.0:CompositeProcess>
</rdf:RDF>
```

### OWL-S Axiomatization:

```
<owl:Class rdf:ID="Sequence">
<rdfs:subClassOf rdf:resource="#ControlConstruct"/>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#components"/>
    <owl:allValuesFrom rdf:resource="#ControlConstructList"/>
  </owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

<owl:Class rdf:ID="ControlConstructList">
<rdfs:comment> A list of control constructs </rdfs:comment>
<rdfs:subClassOf rdf:resource="#shadow-rdf;#List"/>
<rdfs:subClassOf>
  <owl:Restriction>
    <owl:onProperty rdf:resource="#shadow-rdf;#first"/>
    <owl:allValuesFrom rdf:resource="#ControlConstruct"/>
  </owl:Restriction>
</rdfs:subClassOf>
<rdfs:subClassOf>
  <owl:Restriction>
```

```

<owl:onProperty rdf:resource="&shadow-rdf;#rest"/>
<owl:allValuesFrom rdf:resource="#ControlConstructList"/>
</owl:Restriction>
</rdfs:subClassOf>
</owl:Class>

```

### ***A PSL representation***

To define the sequence of processes for ‘centre drill’, ‘drill’, ‘ream’ PSL concepts of ‘activity occurrence’ and ‘min\_precedes’ can be used. In combination they enable the process sequence to be defined as illustrated below. Each of ‘centre drill’, ‘drill’, ‘ream’ is defined as activity occurrences and there relationships are defined using the ‘min\_precedes’ concept. This checks that each process in the correct order but allows other processes to be interspersed. The axiomatisation is also shown below. As can be seen, while the formal description of the process is quite straightforward, the underlying axiomatisation is quite complex. However, as this has already been defined for some 350 concepts these can be used directly and do not need to be redefined. The rigour provided by this approach ensures that the sequence constraints defined in the description are maintained during any further process manipulation during manufacturing planning.

### **PSL Process Description:**

```

(forall (?occ)
  (implies (occurrence_of ?occ ReamHole)
    (exists (?occ1 ?occ2 ?occ3)
      (and (occurrence_of ?occ1 Reaming)
          (occurrence_of ?occ2 Drilling)
          (occurrence_of ?occ3 CentreDrilling)
          (min_precedes ?occ3 ?occ2 ReamHole)
          (min_precedes ?occ1 ?occ1 ReamHole))))))

```

### **PSL Axiomatization:**

Axiom 1 Occurrences in the activity tree for an activity correspond to atomic subactivity occurrences of the activity.

```

(forall (?a ?s1 ?s2)
  (implies (min_precedes ?s1 ?s2 ?a)
    (exists (?a1 ?ap)
      (and (subactivity ?a1 ?a)
          (atomic ?ap)
          (subactivity ?a1 ?ap)
          (occurrence_of ?s2 ?ap))))))

```

Axiom 2 Occurrences in the activity tree for an activity correspond to atomic subactivity occurrences of the activity.

```

(forall (?a ?s1 ?s2)

```

```
(implies (min_precedes ?s1 ?s2 ?a)
  (exists (?a2 ?ap)
    (and (subactivity ?a2 ?a)
      (atomic ?ap)
      (subactivity ?a2 ?ap)
      (occurrence_of ?s1 ?ap))))))
```

Axiom 3 Root occurrences in the activity tree correspond to atomic subactivity occurrences of the activity.

```
(forall (?a ?s1)
  (implies (root ?s1 ?a)
    (exists (?a2 ?ap)
      (and (subactivity ?a2 ?a)
        (atomic ?ap)
        (subactivity ?a2 ?ap)
        (occurrence_of ?s1 ?ap))))))
```

Axiom 4 All activity trees have a root subactivity occurrence.

```
(forall (?s1 ?s2 ?a)
  (implies (min_precedes ?s1 ?s2 ?a)
    (exists (?s3)
      (and (root ?s3 ?a)
        (or (min_precedes ?s3 ?s1 ?a)
          (= ?s3 ?s1))))))
```

Axiom 5 No subactivity occurrences in an activity tree occur earlier than the root subactivity occurrence.

```
(forall (?s ?a)
  (implies (root ?s ?a)
    (not (exists (?s2)
      (min_precedes ?s2 ?s ?a))))))
```

Axiom 6 An activity tree is a subtree of the occurrence tree.

```
(forall (?s1 ?s2 ?a)
  (implies (min_precedes ?s1 ?s2 ?a)
    (exists (?s0)
      (and (initial ?s0)
        (or (precedes ?s0 ?s1)
          (= ?s0 ?s1))
        (precedes ?s1 ?s2))))))
```

Axiom 7 Root occurrences are elements of the occurrence tree.

```
(forall (?s ?a)
```

```
(implies (root ?s ?a)
  (exists (?s0)
    (and (initial ?s0)
      (or (precedes ?s0 ?s)
        (= ?s0 ?s))))))
```

Axiom 8 Every atomic activity occurrence is an activity tree containing only one occurrence.

```
(forall (?a1 ?a2 ?s)
  (implies (and (atomic ?a1)
    (occurrence_of ?s ?a1)
    (subactivity ?a2 ?a1))
    (root ?s ?a2)))
```

Axiom 9 Activity trees are discrete.

```
(forall (?s1 ?s2)
  (implies (min_precedes ?s1 ?s2 ?a)
    (exists (?s3)
      (and (next_subocc ?s1 ?s3 ?a)
        (or (min_precedes ?s3 ?s2 ?a)
          (= ?s3 ?s2))))))
```

Axiom 10 Subactivity occurrences on the same branch of the occurrence tree are on the same branch of the activity tree.

```
(forall (?a ?s1 ?s2 ?s3)
  (implies (and (min_precedes ?s1 ?s2 ?a)
    (min_precedes ?s1 ?s3 ?a)
    (precedes ?s2 ?s3))
    (min_precedes ?s2 ?s3 ?a)))
```

Axiom 11 The activity tree for a complex subactivity occurrence is a subtree of the activity tree for the activity occurrence.

```
(forall (?a1 ?a2)
  (implies (subactivity ?a1 ?a2)
    (not (exists (?s)
      (subtree ?s ?a2 ?a1))))))
```

### ***Capability and Limitations identified with PSL***

We have illustrated with the simple examples above the capability which PSL has for process sequencing over other well known ontological approaches. PSL provides comprehensive coverage for process relationships and includes concepts for concurrency, preserving order and repetitive processes. It also provides concepts to

capture state and/or time effects, either those which are pre-conditions for an activity or those which are the effects of an activity. This provides an extensive set of some 350 concepts for process description.

The limitations of PSL lie in its ability to relate to resource definitions and to the product inputs and outputs. In a manufacturing planning environment the process sequence decisions to be made depend on the resources which are available and the product features which are to be machined. In these two areas the capability of PSL is limited.

PSL's concepts of resource are limited to:

- (reusable ?r ?a)
- (possibly\_reusable ?r ?a)
- (renewable ?r ?a)
- (weakly\_reusable ?r ?a)
- (consumable ?r ?a)
- (possibly\_consumable ?r ?a)
- (weakly\_consumable ?r ?a)
- (wearable ?r ?a)

These provide an indication of the extent to which the resource will be consumed by the process. However, PSL does not provide any ontological foundation from which to specify the type of resource being used e.g. a machine tool or a cutting tool.

PSL has been extended to include the concept of inputs and outputs for a process, but typically in manufacturing systems some form of feature description or part variant description is used to drive planning decision making. PSL does not support this type of concept.

We also recognise that PSL has been developed with the discrete manufacturing domain in mind. A key issue for the future is how broadly the language can be applied to other processes. Also, assuming that one broadly based language cannot cope with the complexities of semantics within each domain there will be a need to identify effective domain boundaries and relationships between domains.

## **DISCUSSION AND CONCLUSIONS**

The potential benefits for manufacturing industry are substantial if manufacturing knowledge can be shared within and across software environments. The need to share meaning is fundamental to achieving any level of success in sharing knowledge. Many ontological approaches are being pursued to provide a basis for sharing meaning. However, we have shown that these approaches are built with varying levels of mathematical rigour and in order to share manufacturing knowledge effectively it is critical that ontologies are built on a mathematically rigorous foundation. PSL is one such ontology.

We have demonstrated by comparison the effectiveness of PSL in capturing the semantics of manufacturing process sequences and argued for its value in providing an effective foundation to capture meaning related to a broad range of manufacturing concepts. However, we have also highlighted that PSL has its limitations as a foundation for manufacturing knowledge sharing as it provides only limited support for concepts related to manufacturing resources and to workpiece relationships to process.

It is clear that ideally we would look for a single foundation ontology on which to build any domain ontology. However, given the diversity of requirements for each possible domain, it is questionable whether a single foundation ontology can be agreed upon to provide the range of concepts needed. A key issue for future work is to identify the most effective way to support the development of multiple domain environments such that effective inter-operability between them can be maintained. If a single foundation ontology cannot be achieved then an alternative would be to use multiple foundation ontologies, but this leads to a further level of problem in how to build relationships between the foundation ontologies. In our future work we anticipate an exploration of manufacturing domain scenarios would include design for manufacture, manufacturing planning, production planning, supply chain configuration as well as their links to other life cycle activities. These will involve a range of concepts related to manufacturing processes, manufacturing resources, manufacturing strategies, product architecture, product views, product features and materials. We have shown that PSL provides a rigorous foundation from which to construct manufacturing process ontologies. However there is a clear need for a foundation ontology to support manufacturing resource definition. Further, given the need to share product terms there are also requirements for foundation ontologies which can support key product classes such as architecture and views as illustrated in figure 5.

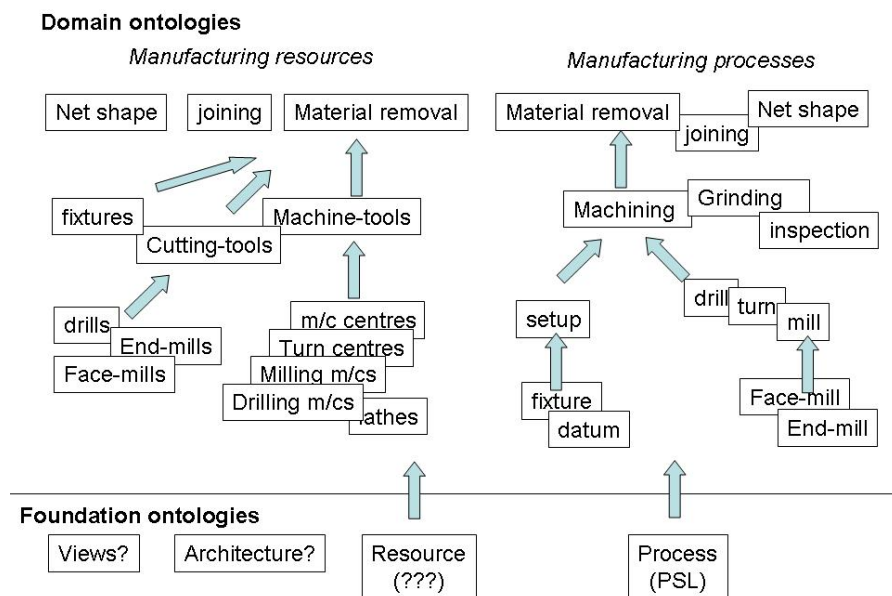


Figure 5: Foundation requirements for manufacturing process and resource semantics

The subsequent issues to be addressed are then:

- What is the most effective foundation ontology approach to support the communication requirements of manufacturing?
- If, as seems likely, a single foundation ontology is impractical what other levels of foundation are required?
- Can the methods developed support the flexible extension of current and developing manufacturing systems to give businesses a step change in knowledge sharing capability?

Work to explore these issues is currently underway and we anticipate a substantial change in future system inter-operation and knowledge sharing methods as a result.

## REFERENCES

- Abramovici M., Sieg O. (2002) Status and future trends of product lifecycle management (PLM ) technology. IPPD conference, Wroclaw.
- Costa, C.A., Harding J.A., Young R.I.M., (2001) "The application of UML and an open distribute process framework to information systems design" *Computers in Industry* Vol 46, pp33-48.
- Cutting-Decelle, A.F., Young, R.I.M., Baldwin, A.N., Anumba, C.J., Bouchlaghem, N.M., (2002), A new challenge for CE : PSL, a standardised language for an integrated supply chain management, *Advances in Concurrent Engineering*, Roy & Steiger-Garçao Eds, Swets & Zellinger, Lisse.
- Cutting-Decelle A.F., Young R.I.M, Das B., Anumba C.J., Bouchlaghem N. M., A.N. Baldwin. (2004) A multi-disciplinary representation of supply chain information in construction: an innovative approach to project management. *Tools and Methods of Competitive Engineering (TMCE)*, Lausanne, Switzerland. pp 465-474, ISBN 9059660188.
- Dorador J.M., Young R.I.M. (2000). The Application of IDEF0, IDEF3 AND UML Methodologies in the Creation of Information Models. *International Journal of Computer Integrated Manufacture*, Vol.13, No.5, pp430-445 ISSN 0951-192X
- Gomez-Perez A., Fernandez-Lopez M., Corcho O. (2004) *Ontological Engineering*, Springer-Verlag ISBN 1-85233-551-3.
- Gruber T. R., (1993), "A translation approach to portable ontology specifications", *Knowledge Acquisition*, 5, pp199-220.
- Gunendran AG, (2004), *an Information and Knowledge Framework to Support Multiple Viewpoints in the Design for Manufacture of Injection Moulded Products*, PhD Research Thesis, Loughborough University, UK.
- Michel J.J (2005) "Terminology extracted from some manufacturing and related standards" Proposal for new standardization work CEN/TC 310 N1119R3
- ISO 18629-1 (2004), ISO TC184/SC4/JWG8, *Industrial automation system and integration – Process specification language : Part 1 : Overview and basic principles*
- ISO/JTC1/SC32/WG2 (1999). *Knowledge interchange format, Part 1 : KIF-Core*,
- Krause, F. – L., Kimura, F., Kjellberg, T. and Lu, S. C. – Y., (1993). *Product Modelling*, *Annals of CIRP* 42/2 695-12.



- Ray S. R. and A. T. Jones. (2003). "Manufacturing interoperability", Concurrent Engineering: Enhanced Interoperable System: Proceedings of the 10<sup>th</sup> ISPE International Conference, pp.535-540.
- Srinivasan V., (2005) "Open Standards for Product Lifecycle Management" in Product Lifecycle Management: Emerging solutions and challenges for global networked enterprises. Ed. Bouras, Gurumoorthy and Sudarsan. Pub. Inderscience Enterprises Ltd. pp475-484, ISSN: 0-907776-18-3
- R I M Young, D Guerra, G Gunendran, B Das, S Cochrane Anne-Francoise Cutting-Decelle, (2005), "Sharing manufacturing Information and Knowledge in Design Decision Support" Advances in integrated design and manufacturing in mechanical engineering ed A. Bramley, D Brissaud, D Coutellier, C McMahon. Pub Springer, pp173-188. ISBN 1-4020-3481-4
- Young R., (2005), "Views of Manufacturing Information: Organisational issues in PLM" Manufacturing Data Technology, June 20-22, Gothenburg, Sweden, <http://www.mdteurope.com>