



**HAL**  
open science

# Towards Expressive Ontology-Based Approaches to Manufacturing Knowledge Representation and Sharing

Nitishal Chungoora, Osiris Junior Canciglieri, R I M Young

► **To cite this version:**

Nitishal Chungoora, Osiris Junior Canciglieri, R I M Young. Towards Expressive Ontology-Based Approaches to Manufacturing Knowledge Representation and Sharing. *International Journal of Computer Integrated Manufacturing*, 2010, 23 (12), pp.1059-1070. 10.1080/0951192X.2010.518976 . hal-00642937

**HAL Id: hal-00642937**

**<https://hal.science/hal-00642937>**

Submitted on 20 Nov 2011

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**Towards Expressive Ontology-Based Approaches to Manufacturing Knowledge Representation and Sharing**

Journal:	<i>International Journal of Computer Integrated Manufacturing</i>
Manuscript ID:	TCIM-2010-IJCIM-0056
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	25-Apr-2010
Complete List of Authors:	Chungoora, Nitishal; Loughborough University, Wolfson School of Mechanical and Manufacturing Engineering Canciglieri, Osiris; Pontifical Catholic University of Paraná, Laboratory of Automation and Systems Young, R I M; Loughborough University, Wolfson School of Mech & Man Engineering
Keywords:	AI IN MANUFACTURING SYTEMS, MANUFACTURING INFORMATION SYSTEMS, DESIGN FOR MANUFACTURE, PRODUCT MODELLING
Keywords (user):	KNOWLEDGE ENGINEERING, ONTOLOGIES



## Towards Expressive Ontology-Based Approaches to Manufacturing Knowledge Representation and Sharing

N. Chungoora<sup>a\*</sup>, O.J. Canciglieri<sup>b</sup>, R.I.M. Young<sup>a</sup>

<sup>a</sup>*Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, UK*

<sup>b</sup>*Pontifical Catholic University of Paraná (PUCPR), Laboratory of Automation and Systems (LAS), Rua Imaculada Conceição, 1155, Prado Velho, Curitiba, PR, Brazil, CEP 80215-030*

Nitishal Chungoora

Wolfson School of Mechanical and Manufacturing Engineering,  
Loughborough University,  
Loughborough,  
LE11 3TU, UK  
Tel: +44(0)7871226566  
e-mail: [N.Chungoora@lboro.ac.uk](mailto:N.Chungoora@lboro.ac.uk)

Robert Ian Marr Young

Wolfson School of Mechanical and Manufacturing Engineering,  
Loughborough University,  
Loughborough,  
LE11 3TU, UK  
Tel: +44(0)1509227662  
Fax: +44(0)1509227648  
e-mail: [R.I.Young@lboro.ac.uk](mailto:R.I.Young@lboro.ac.uk)

Osiris Canciglieri Junior

Pontifical Catholic University of Paraná (PUCPR),  
Laboratory of Automation and Systems (LAS),  
Rua Imaculada Conceição,  
1155, Prado Velho,  
Curitiba, PR, Brazil, CEP 80215-030  
Tel: +55(0)4132711304  
Fax: +55(0)4132711345  
e-mail: [osiris.canciglieri@pucpr.br](mailto:osiris.canciglieri@pucpr.br)

\* Corresponding author. e-mail: [N.Chungoora@lboro.ac.uk](mailto:N.Chungoora@lboro.ac.uk)

The present capability that ontological approaches offer to formally represent and share manufacturing knowledge is dependent on the choice of ontological formalism. Currently, there exists a spectrum of these formalisms which is being subjectively exploited across multiple domains in design and manufacture. Hence, there is an important prerequisite to achieve an understanding of which family of formalism strictly enables the expressive capture of semantics in order to progress towards meaningful information and viable knowledge sharing. This paper analyses the relative strengths and weaknesses in employing a “lightweight” ontology versus a “heavyweight” version of the ontology to represent and share knowledge between multiple domains in injection moulding design and manufacture. A pertinent direction, from an ontology perspective, is then exposed as a prescription for the improved capture and dissemination of formal semantics, to support multi-domain knowledge sharing.

Keywords: design and manufacture; lightweight ontology; heavyweight ontology; semantics; knowledge sharing; injection moulding

## 1 Introduction

Ontological approaches are nowadays increasingly being applied to support the formal capture and sharing of the meaning and intent (i.e. semantics) of design and manufacture concepts. For a particular domain, the representation of the required semantics is held in an ontology and the Knowledge Base (KB) deployed from the ontology is used to populate knowledge which should consistently derive from the semantic structures within the ontology. Represented knowledge in a KB provides useful support for key engineering decisions, for example, the ways in which a designer’s intent in the design domain could affect the selection of manufacturing processes in the manufacturing domain. Thus, expressive manufacturing knowledge refers to populated knowledge in a KB, based on the unambiguous definition of semantics structures which carry enriched formal meaning.

Unfortunately at present, the seamless exchange of design and manufacture semantics for knowledge sharing is still not achievable as a result of domain models that do not carry sufficiently-expressive semantics. This is because there are currently several ontological formalisms, of varying expressiveness (Ray, 2004) and system interaction capabilities, which do not all necessarily address the knowledge capture

and sharing needs in product design and manufacture. Consequently, there exists an ongoing requirement to refine the understanding of the level of logical expressiveness capable of semantically structuring the meaning of product lifecycle concepts (Young et al., 2009; Chungoora, 2010).

This paper investigates the capture and intra-system sharing of ontology-based knowledge using the basis of two broad categories of ontological formalisms, notably “lightweight” and “heavyweight” approaches (Gómez-Pérez et al., 2004), further explained in the next section. By understanding the implications of each approach applied to concepts in injection moulding design and manufacture, the paper contributes to a clarification of (1) the ways of expressively capturing domain semantics and (2) the mechanisms for sharing semantics across intra-system domains to support engineering decisions.

A case study has been devised in order to expose the relative strengths and weaknesses between a lightweight ontological model and a version of the model formalised using a heavyweight formalism. It has been shown that the existence of an axiom layer in the heavyweight model is paramount to capturing rigorous semantics and for prompting the potential for knowledge sharing. Moreover, certain characteristics of the lightweight model have proved to be pertinent to aiding intra-system knowledge sharing. Following this case study, a suitable ontological direction is then identified, as a benchmark for design and manufacture domains that intend to exploit expressive semantics alongside knowledge inference support.

## 2 Lightweight and heavyweight ontological approaches

### 2.1 A categorisation based on expressiveness

The requirements and preferences adopted by different communities have led to the development and utilisation of various ontological formalisms. Ontological formalisms are essentially formal languages that support the construction of ontology-based models and the encoding of the subject matter within these models. Some commonly-occurring formalisms are illustrated in Figure 1, featuring the Unified Modelling Language (UML) (UML, 2009), Frame-based languages (Wang et al., 2006) and Description Logic-based languages (Baader et al., 2007) among others. To distinguish families of ontological formalisms, the ontology community has introduced a categorisation based on the expressiveness of the subject matter contained within ontologies, and enabled via the use of ontological formalisms.

[Figure 1 to be inserted about here]

This categorisation involves the notions of “lightweight” and “heavyweight” ontological approaches, which primarily differ in the degree of formality and granularity with which they can represent the same knowledge (Gómez-Pérez et al., 2004; Casely-Hayford, 2005). Lightweight models predominantly consist of a taxonomy of concepts, with simple relationships established among these concepts and very basic constraints over the meaning of the ontological terms. On the other hand, heavyweight models, in addition to having the lightweight structures, are accompanied by a rich set of formal axioms that constrain the interpretation of ontological terms. In Figure 1, UML and Frames used on its own are examples of

lightweight ontological approaches (**A**), while DL and Frames with a First Order Logic constraint language are examples of heavyweight ontological approaches (**B**).

In the field of manufacturing engineering research, both lightweight and heavyweight methods have been utilised for the formalisation of domain models (ISO 18629, 2005; Patil et al., 2005; Kim et al., 2006; Lin and Harding, 2007). It is clear, from the extent of the exploited lightweight and heavyweight ontological approaches, that there is currently no discernable consensus on a preferred ontological direction. This is largely because of the ongoing need to establish the suitability of these approaches to meet the semantic and knowledge sharing requirements of design and manufacture. Hence, the aim of assessing the benefits and limitations of both approaches becomes a key step towards identifying the essential elements to progress towards expressive ontology-based approaches.

## ***2.2 Multi-domain knowledge representation and sharing using lightweight and heavyweight approaches***

The methodology to achieve the previously-mentioned aim is identified in Figure 2. Emphasis is placed on multi-domain knowledge representation and intra-system knowledge sharing in the context of injection moulding. The methodology involves considering a simple consumer product concept, namely a rotational container (**C**), as shown in Figure 2. Using both lightweight UML and heavyweight Frames with a First Order Logic constraint language, the product representation is first to be captured in the mouldability domain (**D**), by using the semantic structures supported in both methods.

[Figure 2 to be inserted about here]

Then, populated knowledge from the mouldability domain is to be shared with the mould design domain (**E**) for obtaining a representation of the mould product model knowledge. Following this stage, the mould product model knowledge from the mould design domain is to be shared with the mould manufacturing domain (**F**) to capture the manufacturing representation knowledge for the mould. The sharing process between domains is to be achieved by using the adequate translation / mapping mechanisms accommodated in both ontology-based approaches.

A number of reasons justify the selection of UML and Frames with a First Order Logic constraint language, as the preferred lightweight and heavyweight ontological formalisms respectively. In the first place, a range of lightweight information models has exploited UML for multi-viewpoint modelling applied to design realisation stages (Tam et al., 2000; Kugathasan and McMahon, 2001; Canciglieri and Young, 2009). Thus, by performing an assessment of UML against the exposed methodology, it becomes possible to provide an appreciation of one extensively-used lightweight formalism.

Frames with a First Order Logic constraint language as heavyweight formalism presents characteristics that overlap with a range of other heavyweight formalisms. This explains its suitability for this investigation. For example, the formalism bears several structural similarities to Description Logic-based languages, which have witnessed unprecedented relevance in product design ontologies (AIM@SHAPE, 2004; Lukibanov, 2005). Furthermore, the chosen heavyweight formalism holds key commonalities with the Common Logic Interchange Format (CLIF) (ISO/IEC 24707, 2007), which has been used to encode the Process Specification Language (PSL) ontology (ISO 18629, 2005).

### 3 Case study

#### 3.1 Overview of case study

Figure 3 identifies the case study scenario for the analysis of the selected lightweight and heavyweight ontological approaches, to support multi-domain knowledge representation and sharing. This scenario provides a more detailed view on the use of the methodology portrayed in Figure 2. Based on Figure 3, the study concentrates on the analysis of a UML multi-domain injection moulding model against a similar model formalised in Frames with a First Order Logic constraint language. A detailed understanding behind the UML development of the multiple viewpoint domains can be found in an earlier manuscript (Canciglieri and Young, 2003).

For implementation purposes, the Knowledge Engineering Methodology, prescribed by Noy and McGuinness (2001), has been adopted during ontology development. An appropriate UML tool has been utilised to formalise the lightweight UML model. Furthermore, the formalism Frames with the Protégé Axiom Language (PAL), as First Order Logic constraint language, has been used in the Protégé Frames 3.4 ontology editor (Protégé, 2009) for representing the heavyweight model.

[Figure 3 to be inserted about here]

An important facet of the case study is related to the formal representation of the semantics of multi-domain injection moulding and the corresponding populated knowledge. This involves:

- The explicit representation of the mouldability, mould design and mould manufacturing domain semantics.

- Representing knowledge that is either common across domains or needs to be translated / mapped to a different domain.
- Capturing logical pre-conditions that exist in one domain which can drive the translation / mapping of the appropriate knowledge.

The representation of the rotational product in the mouldability domain is partly comprised of internal and external profiles (**G**), that pertain to primary and transition features that form the wall of the product. The dimensional knowledge captured in these profiles is to be shared with the mould design domain (**H**). It is to be noted that emphasis is laid on the semantics of the internal profile of the rotational product, which are then used to drive the rotational core insert representation knowledge (**I**) in the mould design domain. In this case the mould design domain has been referred to as the “rotational core design domain” to clarify that the intended representation is for the rotational core component of the mould.

The knowledge shared from the mouldability domain to the rotational core design domain, is then used to disseminate additional knowledge to the rotational core manufacturing domain (**J**) (i.e. the mould manufacturing domain). The rotational core manufacturing domain represents the semantics of the rotational core insert from a machining viewpoint, for example, in terms of the types of machining features that the rotational core insert holds in the manufacturing domain (**K**).

### ***3.2 Lightweight ontology-based model***

The lightweight UML model has been previously documented (Canciglieri and Young, 2009) and therefore, this section concentrates on the most relevant strengths and weaknesses carried by the lightweight UML approach. Figure 4 provides a broad

understanding of the implementation of the lightweight ontology-based model. In the model, UML class diagrams (**L**) have been exploited to represent the necessary concepts and, to some extent, the basic semantics of each domain. These UML class representations capture domain concepts in the form of classes and arrange these classes according to a taxonomy. Relations with cardinality information (**M**) are used to formulate the key associations that hold between different classes.

The identification, retrieval and sharing of populated knowledge from the mouldability domain to the rotational core design domain and from the design domain to the manufacturing domain is formalised through UML activity diagrams (**N**). These activity diagrams enable the user to create a set of instructions on how to translate the required attributes and knowledge from one domain to another.

[Figure 4 to be inserted about here]

### *3.2.1 Strengths and weaknesses*

- UML class diagrams provide a convenient way to design ontologies, as they support a fairly rich set of graphical constructs. This can be a particularly useful means of reusing platform-independent ontologies prior to their implementation in the required ontology applications.
- The representation of multi-domain information structures is dominated by the use of UML class diagrams that involve taxonomies of classes and cardinality relations between classes, which are fundamental to any ontology. From the lightweight model explored, it has been possible to exploit UML class diagrams to capture common information content across domains.

- There are two main ways in which classes are allowed to carry some semantics namely (1) through the specification of traits that the classes possess (i.e. attributes), and (2) by specifying binary relations that hold between pairs of classes. Classes, attributes and relations in UML hold textual descriptions rather than semantic definitions. Consequently, domain concepts can only be meaningfully interpreted if the implied semantics of these concepts are understood by the user.
- UML activity diagrams allow translation / mapping knowledge to be captured and aid, at a system development level, to automatically perform information sharing procedures from one domain to another. For example, it is possible to trigger the automatic assertion of attributes and dimensional knowledge from the mouldability domain to the rotational core design domain. However, in the experiment, because UML activity diagrams depend on UML class diagrams, this implies that translation procedures are dependent on the terms carried by domain concepts rather than the semantics of these concepts.
- Although a low level of computational interpretation can be captured in UML classes purely associated to variations in class names, it is not fully possible to embed pre-conditional knowledge and intent. For example, in Figure 4, the class name “Rot\_Wall\_Par\_Part\_Line” (**O**) in the mouldability domain is used to imply a rotational primary feature which is positioned parallel to a parting line configuration (**P**). However, the condition for parallelism to a parting line cannot be formally stated in UML.

### 3.3 Heavyweight ontology-based model

The heavyweight ontological exploration using Frames with the Protégé Axiom Language (PAL) differs both in the degree of formality and granularity when compared to the lightweight approach. Figure 5 depicts the heavyweight ontological structures used to model multi-domain semantics and to identify sharable knowledge between domains.

[Figure 5 to be inserted about here]

In the heavyweight model, ontological structures consist of taxonomies of classes, relations and functions, accompanied by a rigorous logic-based axiom layer as shown in Figure 5 (Q). This layer is responsible for supporting the meaning of concepts in computational form. The axiom layer is built on top of the basic ontological structures and consists of integrity constraints and mapping rules, which are both written in PAL. This constraint language accommodates first order semantics, thereby providing considerable flexibility in specifying the conditions for semantic conformance and knowledge sharing. Integrity constraints are logical restrictions that help ensure the semantic integrity within the injection moulding domains identified in Figure 3, while mapping rules are logical conditions that help identify potential knowledge that could be shared from one domain to the other.

Figure 6 provides a screen shot of the “Mouldability Domain” (R) class taxonomy in the class browser, which at first glance is very similar to the class taxonomy from the lightweight UML class model. Other abstract classes are present namely “Rotational Core Design Domain” (S) and “Rotational Core Manufacturing Domain” (T), which contain the information structures for the rotational core insert design and manufacture respectively. The abstract class “Common Semantics” (U)

regroups reusable behaviours across domains, such as the notions of “Point”, “Axis”, “Length Measure” and “Dimensional Tolerance” among others.

An instance of the class “Rotational Mouldability Product” (**V**) is shown in the instance browser. Captured semantics for one specific instance of “Rotational Mouldability Product”, named “Product 1 - Rotational Container” (**W**), can be identified in Figure 6. These semantic structures involve, for example, the list of point profiles aggregated under the relations “holds\_internal\_profile” (**X**) and “holds\_external\_profile” (**Y**) and the list of primary and transition features aggregated under the binary relation “holds\_feature” (**Z**).

[Figure 6 to be inserted about here]

### 3.3.1 Integrity constraints

From an ontology formalisation viewpoint, PAL is used for model-checking. This implies that integrity constraints act as semantic prescriptions to ensure that populated knowledge in the KB conforms to the semantics expressed in the heavyweight model. To verify whether asserted knowledge violates or conforms to semantics, integrity constraints can be processed and a number of results are retained in the event that these constraints have been infringed. In other words, integrity constraints contribute to the semantic integrity and enrichment of the KB.

In order to account for the semantic needs of the heavyweight model, integrity constraints have been written for the multiple domains under consideration. Over 30 integrity constraints including both simple and complex ones, have been modelled for all three domains. The expression listed next gives an example of a simple integrity constraint in the mouldability domain to ensure that instances of the class “Rotational

Mouldability Product” (see Figure 6 (V)) are only allowed to hold one axis of rotation.

*(defrange ?product :FRAME 'Rotational Mouldability Product')*

*(forall ?product*

*(= (number-of-slot-values holds\_axis ?product) 1))*

If, for example, an instance of “Rotational Mouldability Product” is asserted as having more than one axis in the KB, then an execution of the PAL constraint would show that this instance is violating the fundamental semantics that a rotational mouldability product must always hold one axis. Figure 7 illustrates the result of querying an integrity constraint based on an incorrectly-populated knowledge element. The instance “Product 1 - Rotational Container” (W) is shown to be violating the integrity constraint at query time as a result of an additional “Probe Inconsistent Axis” (A1) having been assigned. The identification of inconsistent knowledge, like the one shown in Figure 7, provides a useful way of prompting the user to rectify the incorrect assertions.

[Figure 7 to be inserted about here]

An example of a more complex integrity constraint in the mouldability domain is shown in Figure 8. The axiom captures the relevant logical pre-conditions to ensure the correct specification of parting line features (see Figure 4 (P)), by using the appropriate formalised statement (B1). In the expression, the accurate definition of a “Parting Line Feature” (C1) is captured based on the known existence of some defined “Primary Feature” (D1). A similar understanding has been followed for the

specification of other simple and complex integrity constraints required for the mould design and mould manufacturing domains.

Once the mouldability domain has been modelled with its required instances populated, integrity constraints for that specific domain are processed to ensure that the asserted knowledge concords with semantics. If a constraint is violated this implies that the query response, obtained from running the constraint, points to inconsistent knowledge. The consequence of a visible inconsistency prompts the user to modify and/or assert correct knowledge. The process of checking integrity constraints is iterated until there are no violated conditions, thereby ensuring the completeness of domain semantics.

[Figure 8 to be inserted about here]

### 3.3.2 Mapping rules

The next stage following model-checking of the mouldability domain involves the execution of mapping rules. A first set of mapping rules is run with the intention of identifying sharable knowledge that needs to be communicated from the mouldability domain to the rotational core design domain. Mapping rules are written in the same way as integrity constraints and are very similar in terms of complexity. The main difference between the two lies in the specification of existential quantifiers (i.e. the First Order Logic directive called “exist”) in the consequent of the mapping rules. An example of a mapping rule is informally quoted next, together with an exemplified understanding of the implications of the mapping rule as shown in Figure 9.

*“Rotational core perpendicular straight line(s) must be specified”* informally says that:

If a parallel parting line primary feature (e.g. **(E1)**) of a rotational mouldability product has two points  $?p1$  and  $?p2$  that describe the internal profile of the product, such that only the z coordinates of the two points are different while the x and y coordinates are the same, then there should exist a core perpendicular straight line (e.g. **(F1)**) that meets the two points in the rotational core design domain.

[Figure 9 to be inserted about here]

In simpler words, logical semantic conditions arising in the mouldability domain imply the existence of similar, modified or different knowledge elements in the rotational core design domain. After sharable knowledge is processed on running the first set of mapping rules, the user then manually creates the identified knowledge for the rotational core design domain. Once this stage is performed, integrity constraints for the rotational core design domain are executed to ensure the consistency of the new knowledge input.

The next stage of knowledge sharing involves discovering mappings from the rotational core design domain to the rotational core manufacturing domain. An informally-expressed example of a mapping rule in this case is listed next, together with its corresponding explanatory diagram in Figure 10.

*“Horizontal turning feature(s) must be specified”* informally says that:

If a straight line (e.g. **(G1)**) that defines the core insert for a rotational mouldability product has two points  $?p1$  and  $?p2$ , such that only the x coordinates of the two points are different while the z and y coordinates are the same, then there should exist a horizontal turning feature profile (e.g. **(H1)**) that meets the two points in the rotational core manufacturing domain.

[Figure 10 to be inserted about here]

Results from similar mapping rules help identify new knowledge required for the rotational core manufacturing domain based on the knowledge elements found in the rotational core design domain. The user asserts the identified sharable knowledge in the rotational core manufacturing domain and ascertains that the knowledge input is consistent with domain semantics by executing the integrity constraints for the manufacturing domain.

Figure 11 illustrates the results of processing two mapping rules for each domain-to-domain sharable knowledge identification process. Overall, 18 complex mapping rules have been explored in the heavyweight experiment to obtain, from the product representation in the mouldability domain, the accurate representation of the rotational core insert in both the mould design and manufacturing domains.

In the sample results in Figure 11, it can be seen that sharable knowledge has been inferred and consists of (1) hole features from the mouldability domain that can be directly shared with the rotational core design domain (**I1**), (2) product geometry-related semantics (**J1**) from the mouldability domain which are required in the rotational core design domain and (3) geometry-related semantics (**K1**) of the design rotational core insert that are sharable with the rotational core manufacturing domain, in order to obtain a machining feature definition for the rotational core insert.

[Figure 11 to be inserted about here]

### 3.3.3 Strengths and weaknesses

- The fundamental, primarily lightweight, semantic structures of domain concepts can be readily modelled through the specification of classes and their

taxonomies, accompanied by binary relations that hold between classes, and functions which act like attributes of classes.

- The presence of an axiom layer provides the capability to support the definition of rigorous semantic structures, which complement the lightweight structures. The axiom layer accommodates a set of integrity constraints and mapping rules written in the expressive and relatively flexible Protégé Axiom Language (PAL), which is First Order Logic-based.
- The axiom layer offers the ability to formally capture logical pre-conditions. Although the integrity constraints that model these pre-conditions deserve to be carefully written and may result in an impedance in processing time as a result of complexity, yet it is seen that such constraints explicitly enable the subject matter of the heavyweight model to be represented.
- In the heavyweight approach, the use of integrity constraints has remained rigid. In other words, integrity constraints can only be specified to dictate the compulsory conformance of populated knowledge in the KB. In certain situations, it could be necessary to also support optional conformance of knowledge which is left for the user to decide, as opposed to relying on the system. This would require the ability to specify integrity constraints of lesser “strength”, while remaining traceable by the user.
- It is possible via the use of mapping rules to make inferences for aiding the identification of sharable knowledge, that needs to be mapped from one domain to another. However, in the chosen heavyweight approach, it has not been possible to automatically perform the assertion of new knowledge as this has relied on manual input, articulated through the processing of mapping rules. This drawback is due to the fact that PAL is essentially used for writing

restrictions on existing knowledge rather than for asserting new knowledge (Protégé, 2009).

- Furthermore, the heavyweight ontology development process is absent of the use of suitable ontology design schematics, which would serve as platform-independent model. This is because the ontology has been directly constructed within the implementation environment. Hence, this suggests that the heavyweight model is platform-dependent and, therefore, makes the process of interoperability between different applications a potential issue.

#### 4 Discussions

The case study has documented a set of strengths and weaknesses of lightweight and heavyweight ontological approaches, applied to multi-domain knowledge representation and sharing in injection moulding design and manufacture. One of the primary differences between the two approaches lies in the ability for the heavyweight model to accommodate an axiom layer supported by First Order Logic semantics. The axiom layer helps express the behaviours and conditions that prescribe the integrity of populated domain knowledge in KBs.

In UML 2, the Object Constraint Language (OCL) can be used to specify invariant conditions that must hold for the system being modelled (OCL, 2006). This implies that in a similar way to PAL, OCL would enable the representation of certain logic-based conditions for ensuring that accurate knowledge is populated. However, OCL does not possess the expressive power of First Order Logic. For this reason, heavyweight ontological approaches are favoured from the perspective of semantic expressiveness and interoperability. Nevertheless, lightweight UML models can still

be effective in collaborative settings, provided the concepts defined in these models are agreed and understood.

From the perspective of automating knowledge assertion processes, UML activity diagrams hold a stronger prospect of making the translation / mapping of knowledge an easier task compared to mapping rules. Thus, it can be extrapolated that by interfacing UML activity diagram mechanisms with heavyweight models, it could be possible to achieve an improved method for automatically asserting new knowledge. In the Protégé ontology environment, the Jess rule engine reasoner (Protégé, 2009) is able to interact directly with populated knowledge and could, therefore, potentially be exploited to perform the automatic assertion of inferred knowledge from PAL mapping rules.

Additional opportunities exist for extending the scope of the heavyweight ontology of injection moulding design and manufacture. For instance, the mould manufacturing domain could be broadened to include manufacturing process sequencing knowledge. This is where, in particular, the Process Specification Language (PSL) ontology would help formalise the semantics of flow models (Bock and Gruninger, 2005). Such extensions would require a heavyweight ontological approach that fully supports more intricate relations and functions, as a result of the complexity in the semantics of manufacturing process sequences. For meeting this purpose, it would be required to identify an even more expressive heavyweight ontological formalism, since employing Frames with a First Order Logic constraint language in Protégé imposes certain restrictions to using binary relations and less powerful semantic structures.

On the other hand, UML class diagrams as a conceptual modelling method presents interesting possibilities as far as platform-independent ontology design is

concerned. This is especially because currently, there is no de-facto ontology design schematic language. So, for example, a UML class can be used to represent a class in a heavyweight ontology and a UML binary association can directly map to a binary relation. Another example involves higher-arity relations, such as ternary relations, that can be represented in UML by using the construct of n-ary associations.

However, more complex heavyweight constructs like functions of multiple arities and the instantiation of meta-classes would demand an agreed mode of exploiting UML class diagrams in order to avoid ambiguity.

## 5 Conclusions

The study presented in this work has shown that there currently exists a number of benefits and drawbacks related to both lightweight and heavyweight ontological approaches. It is understood from this that the progress towards enabling expressive manufacturing knowledge representation and sharing is bound to enfold the integration of an array of ontology-based understandings and semantic technologies.

Hence, the underpinning towards expressive ontology-based approaches firstly requires enabling multiple domains to explicitly represent fundamental semantic structures. These fundamental structures should include the notion of classes and their taxonomies, together with relations and functions which can bind more than two classes together. At present, the formalism Frames with a First Order Logic constraint language is not able to capture the representation of these more complex relations and functions, while UML also falls somehow short of a direct way for so doing.

Secondly, integrity constraints need to be formalised to complement these fundamental structures, thereby semantically-enriching ontologies and ensuring the semantic consistency of populated knowledge. It is highly desirable that integrity

constraints be written using an appropriate First Order Logic-based language in order to impart the required level of logical expressiveness.

Thirdly, the process of ontology development should be accompanied by the provision of appropriate ontology design schematics. This is an essential stage to support the platform-independent representation and design of fundamental structures, prior to their implementation. Currently, Frames with a First Order Logic constraint language in Protégé is less suited for the purpose of ontology design due to platform-dependent modelling. Conversely, UML offers useful prospects for ontology design, specially as it may be possible to utilise UML in a customised way to represent more complex ontological notions.

Finally, the ability to formalise knowledge inference rules, using a suitable First Order Logic-based language, should be provided as a means of deriving new and expressive knowledge to support engineering decisions. By providing a logic-based ground for inference rules, it becomes possible to verify derived knowledge via tractable reasoning procedures. These procedures need to be accompanied by the relevant translation / mapping mechanisms so as to help perform the automatic assertion of derived multi-domain knowledge.

With the introduction of new ontological formalisms notably Common Logic (CL) (ISO/IEC 24707, 2007), it is clear that improved capabilities are emerging to address higher levels of semantic expressiveness and interoperability. Work is currently under way in our laboratory to take this aspect forward.

## Acknowledgements

The research developed in this paper has resulted from a number of strands of work which have been supported by different funding agencies. In particular we wish to thank (1) the EPSRC, who are funding a majority of our work on “Interoperable Manufacturing Knowledge Systems” (IMKS) under project 253 of the Loughborough University Innovative Manufacturing and Construction Research Centre, and (2) the Wolfson School of Mechanical and Manufacturing Engineering of Loughborough University for funding research studentships.

## References

- AIM@SHAPE, (2004), Advanced and Innovative Models and Tools for the development of Semantic-based systems for Handling, Acquiring and Processing knowledge Embedded in multidimensional digital objects. Available from: <http://www.aimatshape.net/> [Accessed March 2010]
- Baader, F., Horrocks, I. and Sattler, U., 2007. Description logics. *In: van Harmelen, F., Lifschitz, V. and Porter, B., eds. Handbook of Knowledge Representation.* Elsevier.
- Bock, C. and Gruninger, M., 2005. PSL: a semantic domain for flow models. *Software and Systems Modelling Journal*, 4, 209-231.
- Canciglieri, O.J. and Young, R.I.M., 2003. Information sharing in multi-viewpoint injection moulding design and manufacturing. *International Journal of Production Research*, 41, 1565-1586.
- Canciglieri, O.J. and Young, R.I.M., 2009. Information mapping across injection moulding design and manufacture domains. *International Journal of Production Research*, DOI: 10.1080/00207540902824974.
- Casely-Hayford, L., 2005. Environments, methodologies and languages for supporting users in building a chemical ontology. MSc Dissertation. Faculty of Life Sciences, University of Manchester, Manchester, UK.
- Chungoora, N., 2010. A framework to support semantic interoperability in product design and manufacture. Ph.D. Thesis. Loughborough University, Loughborough, UK.
- Gómez-Pérez, A., Fernández-López, M. and Corcho, O., (2004). *Ontological engineering: with examples from the areas of knowledge management, e-commerce and the semantic web.* London, UK: Springer-Verlag London Ltd.
- ISO 18629, 2005. Industrial Automation Systems and Integration – Process Specification Language (PSL).
- ISO/IEC 24707, 2007. Information Technology – Common Logic (CL): a framework for a family of logic-based languages.

- Kim, K.-Y., Manley, D.G. and Yang, H., 2006. Ontology-based assembly design and information sharing for collaborative product development. *Computer Aided Design*, 38, 1233-1250.
- Kugathasan, P. and McMahon, C., 2001. Multiple viewpoint models for automotive body-in-white design. *International Journal of Production Research*, 39(8), 1698-1705.
- Lin, H.K. and Harding, J.A., 2007. A manufacturing engineering ontology model on the semantic web for inter-enterprise collaboration. *Computers in Industry*, 58(5), 428-437.
- Lukibanov, O., 2005. Use of ontologies to support design activities at DaimlerChrysler. *Proceedings of the 8<sup>th</sup> International Protégé Conference*, Madrid, Spain.
- Noy, N.F. and McGuinness, D.L., 2001. Ontology development 101: a guide to creating your first ontology. Knowledge Systems Laboratory, Stanford University, USA.
- Object Constraint Language (OCL), 2006. OMG Available Specification Version 2.0. Available from: <http://www.omg.org/technology/documents/formal/ocl.htm> [Accessed March 2010]
- Patil, L., Dutta, D. and Sriram, R., 2005. Ontology-based exchange of product data semantics. *IEEE Transactions on Automation Science and Engineering*, 2(3), 213-225.
- Protégé - Ontology Editor and Knowledge Acquisition System. Available from: <http://www.protege.stanford.edu> [Accessed November 2009]
- Ray, S.R., 2004. NIST's semantic approach to standards and interoperability. PowerPoint Presentation. Available from: [http://ontolog.cim3.net/file/resource/presentation/NIST\\_Semantics--SteveRay\\_20040212a.ppt](http://ontolog.cim3.net/file/resource/presentation/NIST_Semantics--SteveRay_20040212a.ppt) [Accessed November 2009]
- Tam, S., Lee, W.B., Chung, W.W.C. and Lau, H.C.W., 2000. An object-based process planning and scheduling model in a product design environment. *Logistics Information Management*, 13(4), 191-200.
- Unified Modelling Language (UML). Available from: <http://www.uml.org/> [Accessed November 2009]
- Wang, H.H., Noy, N., Rector, A., Musen, M., Redmond, T., Rubin, D., Tu, S., Tudorache, T., Drummond, N., Horridge, M. and Seidenberg, J., 2006. Frames and OWL side by side. *Proceedings of the 9<sup>th</sup> International Protégé Conference*, Stanford, California.

Young, R.I.M, Gunendran, G., Chungoora, N., Harding, J.A. and Case, K., 2009. Enabling interoperable manufacturing knowledge sharing in PLM. *Proceedings of the 6<sup>th</sup> International Product Lifecycle Management Conference*, University of Bath, Bath, UK.

### Figure Captions

Figure 1. Examples of lightweight and heavyweight ontological approaches

Figure 2. Methodology for multi-domain knowledge representation and sharing

Figure 3. Case study scenario

Figure 4. Using UML class and activity diagrams in the lightweight model

Figure 5. Heavyweight ontological structures

Figure 6. Capturing the semantics of the mouldability domain

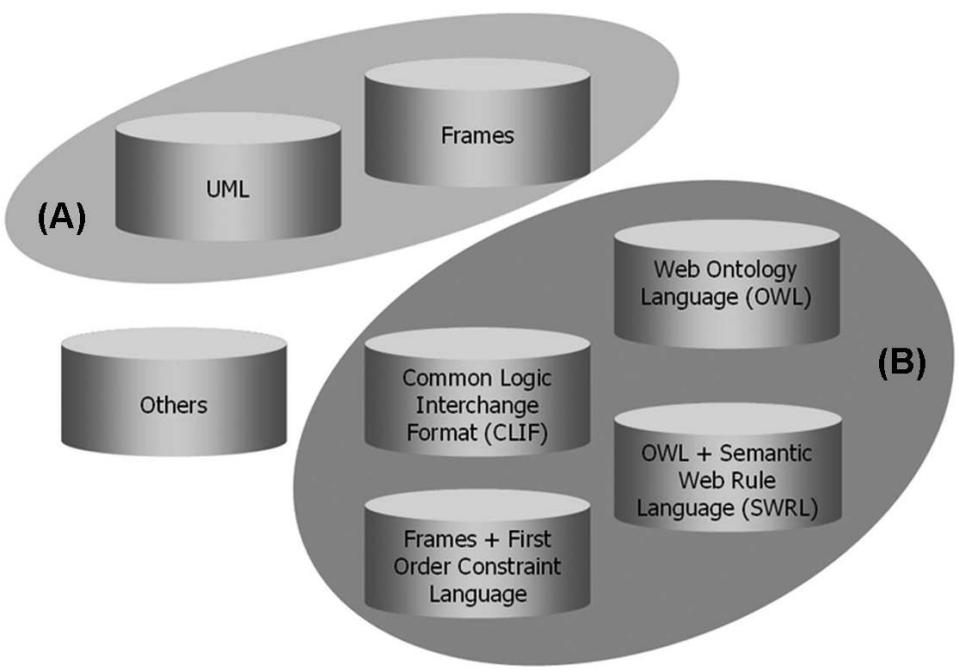
Figure 7. Reporting an integrity constraint violation

Figure 8. Example of a complex integrity constraint

Figure 9. Example of a mapping rule for sharing between the mouldability and rotational core design domains

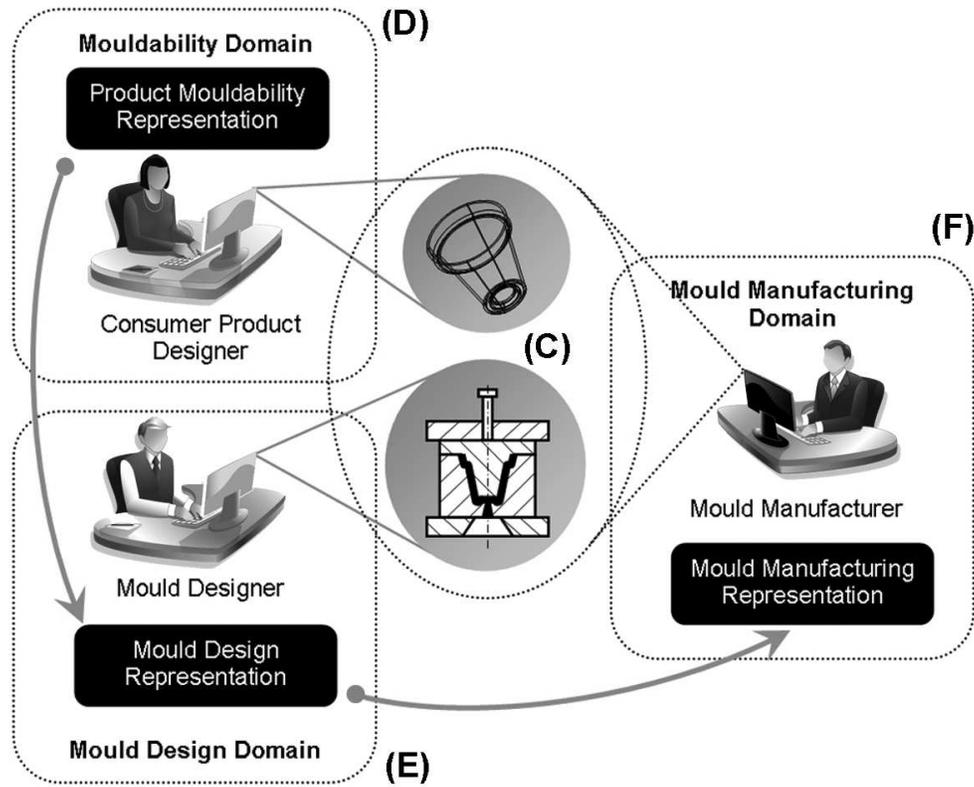
Figure 10. Example of a mapping rule for sharing between the rotational core design and manufacturing domains

Figure 11. Samples of processed mapping rules for multi-domain knowledge sharing

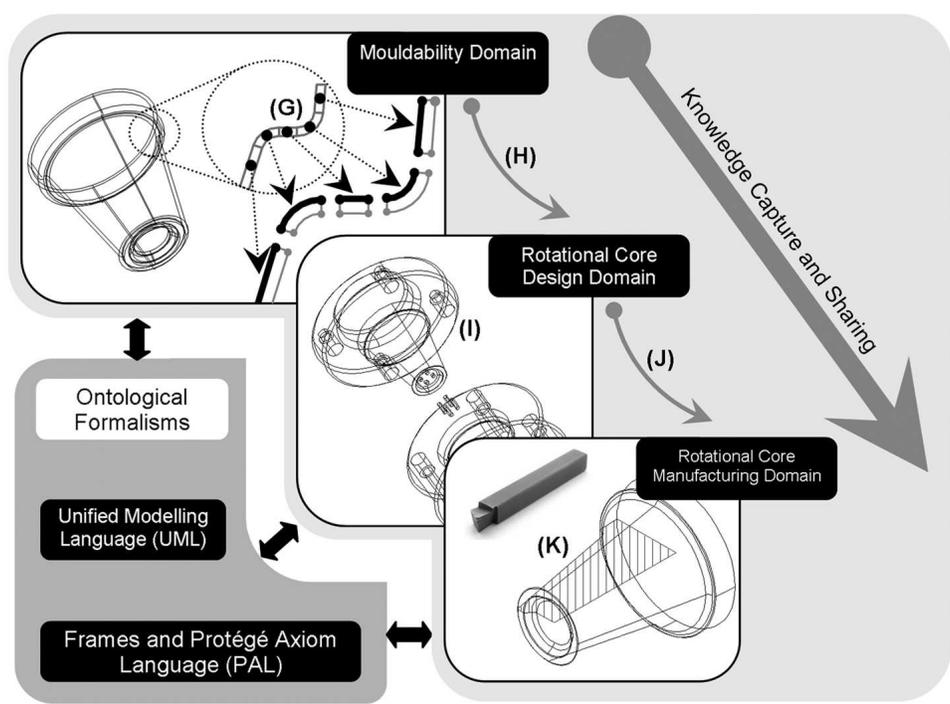


Examples of lightweight and heavyweight ontological approaches  
49x33mm (600 x 600 DPI)

Review Only

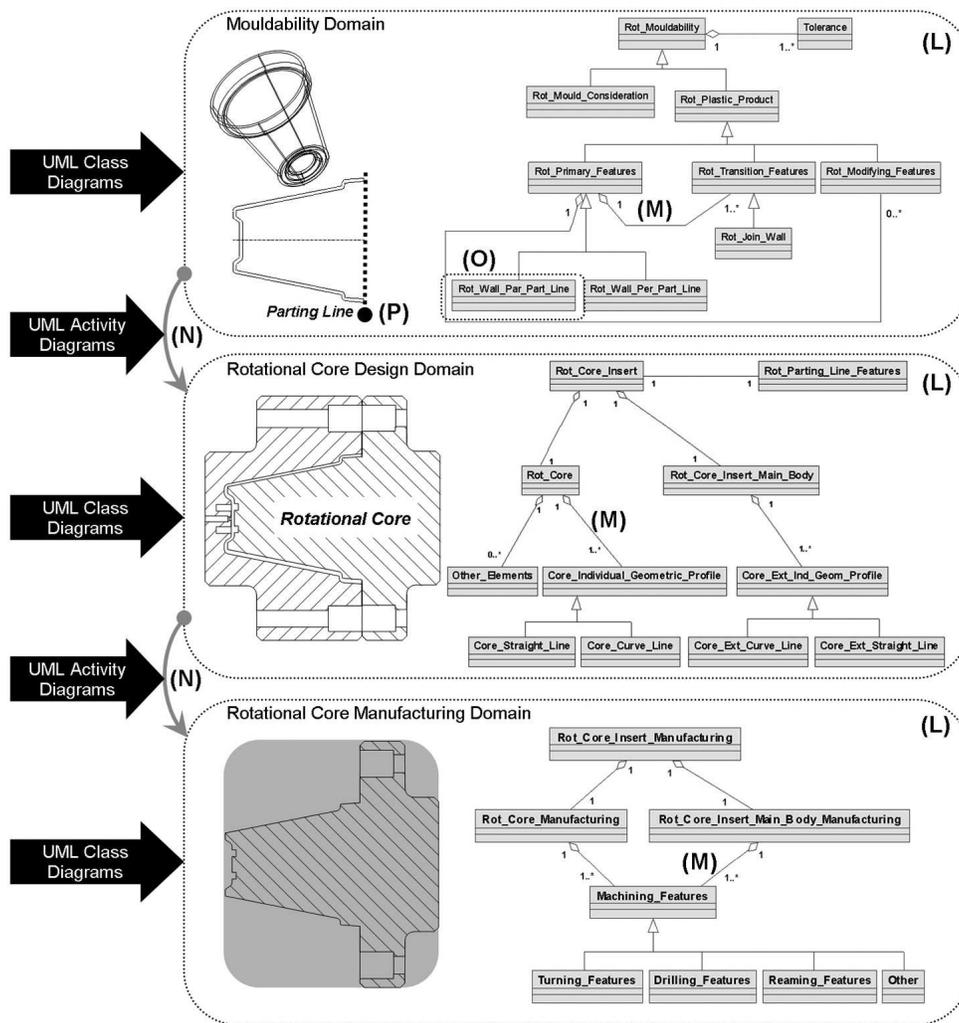


Methodology for multi-domain knowledge representation and sharing  
50x41mm (600 x 600 DPI)



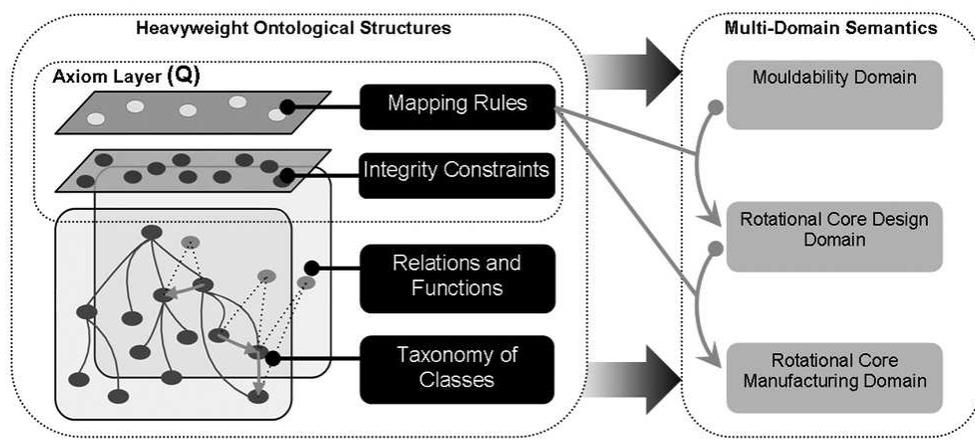
Case study scenario.  
63x46mm (600 x 600 DPI)

View Only



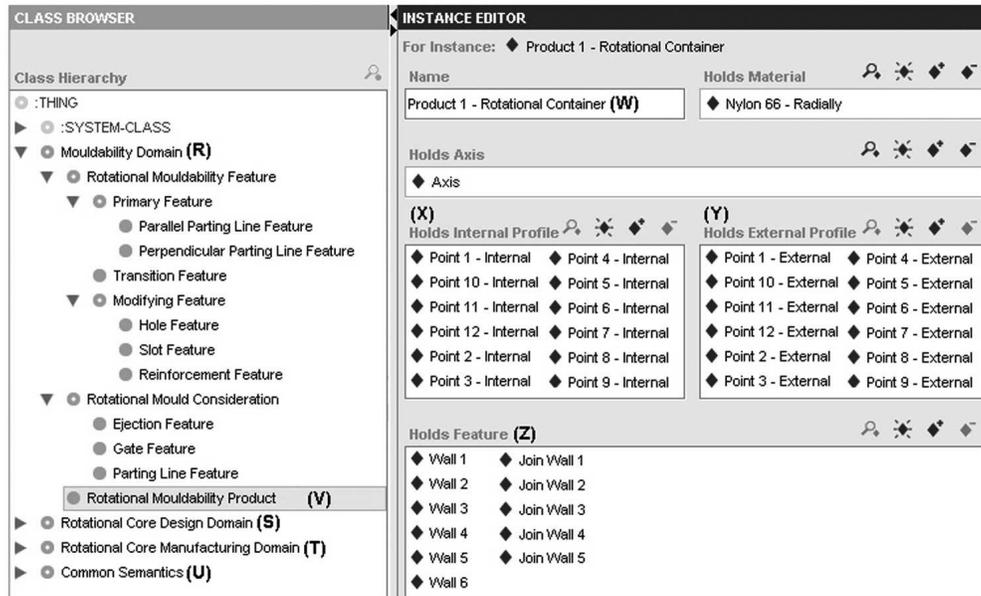
Using UML class and activity diagrams in the lightweight model  
63x66mm (600 x 600 DPI)



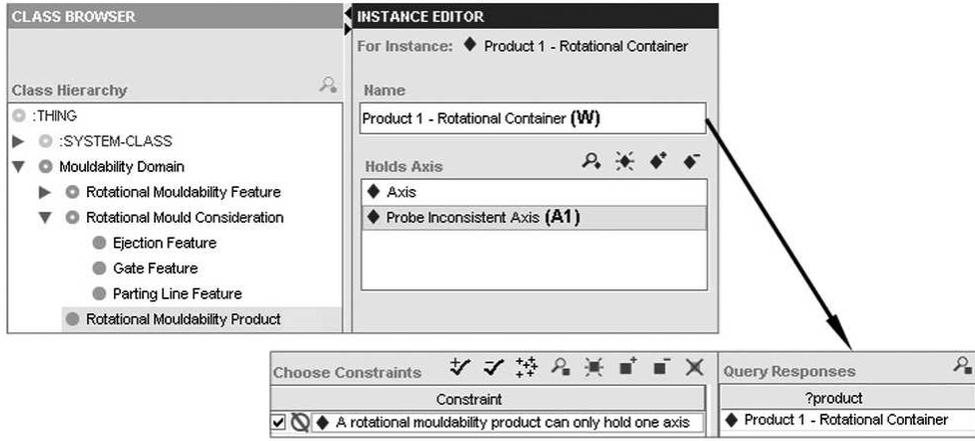


Heavyweight ontological structures  
42x19mm (600 x 600 DPI)

Review Only

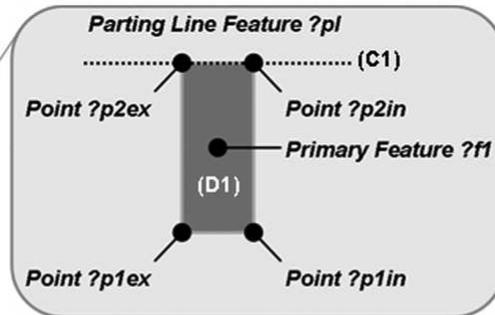


Capturing the semantics of the mouldability domain  
63x38mm (600 x 600 DPI)



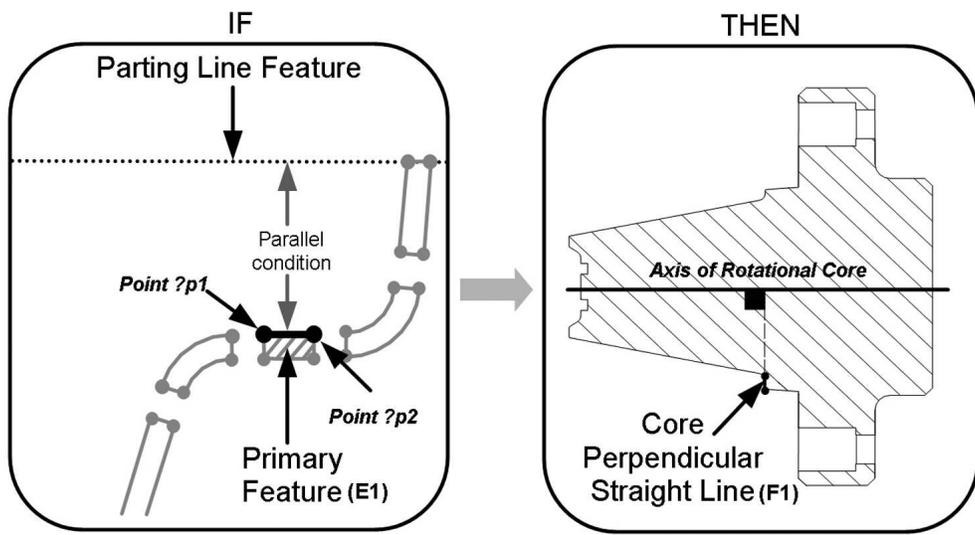
Reporting an integrity constraint violation  
42x19mm (600 x 600 DPI)

Review Only



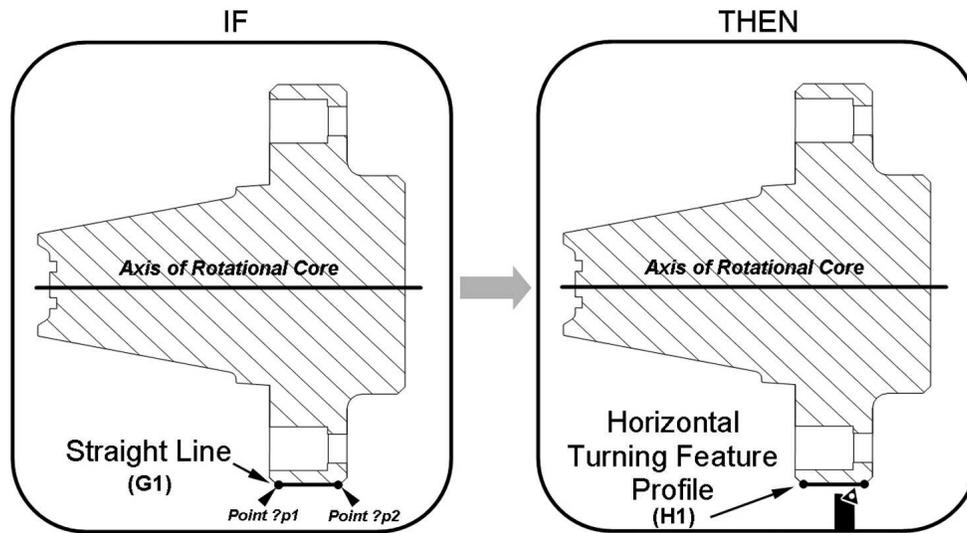
Name	Description
A parting line definition should conform to Parting Line Condition A	If a primary feature is related to a parting line feature where neither the primary feature nor the parting line feature relate to another primary feature, then the parting line feature meets the two topmost points of the profile of the primary feature.
<b>Statement (B1)</b> <pre>(forall ?f1 (forall ?pl (forall ?p1in (forall ?p1ex (forall ?p2in (forall ?p2ex (=&gt; (and (relates_to ?f1 ?pl) (not (exists ?f2 (and (instance-of ?f2 'Primary Feature') (not (= ?f1 ?f2)) (relates_to ?pl ?f2) (relates_to ?f1 ?f2)))))) (holds_profile_point ?f1 ?p1in) (holds_profile_point ?f1 ?p1ex) (holds_profile_point ?f1 ?p2in) (holds_profile_point ?f1 ?p2ex) (= (holds_X_coordinate ?p1in) (holds_X_coordinate ?p1ex)) (&lt; (holds_Z_coordinate ?p1in) (holds_Z_coordinate ?p1ex)) (= (holds_X_coordinate ?p2in) (holds_X_coordinate ?p2ex)) (&lt; (holds_Z_coordinate ?p2in) (holds_Z_coordinate ?p2ex)) (&gt; (holds_X_coordinate ?p2in) (holds_X_coordinate ?p1in)) (&gt; (holds_X_coordinate ?p2ex) (holds_X_coordinate ?p1ex))) (and (meets ?pl ?p2in) (meets ?pl ?p2ex))))))))))</pre>	<b>Range</b> <pre>(defrange ?f1 :FRAME 'Primary Feature') (defrange ?f2 :FRAME 'Primary Feature') (defrange ?p1in :FRAME 'Point') (defrange ?p1ex :FRAME 'Point') (defrange ?p2in :FRAME 'Point') (defrange ?p2ex :FRAME 'Point') (defrange ?pl :FRAME 'Parting Line Feature')</pre>

Example of a complex integrity constraint  
42x42mm (600 x 600 DPI)



Example of a mapping rule for sharing between the mouldability and rotational core design domains  
63x35mm (600 x 600 DPI)

Review Only



Example of a mapping rule for sharing between the rotational core design and manufacturing domains  
63x35mm (600 x 600 DPI)

Query Responses  ◆ Rotational core hole(s) can potentially be specified	
?hole	
◆ Hole 1	
◆ Hole 2	
◆ Hole 3	
◆ Hole 4	<b>(I1)</b>

Query Responses  ◆ Rotational core perpendicular straight line(s) must be specified	
?p1	?p2
◆ Point 1 - Internal	◆ Point 2 - Internal
◆ Point 5 - Internal	◆ Point 6 - Internal
◆ Point 9 - Internal	◆ Point 10 - Internal
	<b>(J1)</b>

Query Responses  ◆ Horizontal turning feature(s) must be specified	
?p1	?p2
◆ Point D	◆ Point C
◆ Point H	◆ Point G
	<b>(K1)</b>

Query Responses  ◆ Facing turning feature(s) must be specified	
?p1	?p2
◆ Point 1 - Internal	◆ Point 2 - Internal
◆ Point 5 - Internal	◆ Point 6 - Internal
◆ Point 9 - Internal	◆ Point 10 - Internal
◆ Point 12 - Internal	◆ Point I
◆ Point A	◆ Point B
◆ Point E	◆ Point F
	<b>(K1)</b>

Samples of processed mapping rules for multi-domain knowledge sharing  
63x56mm (600 x 600 DPI)

Only