

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.



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

# **Towards the ontology-based consolidation of production-centric standards**

N. Chungoora<sup>1</sup>, A.-F. Cutting-Decelle<sup>2</sup>, R.I.M. Young<sup>1</sup>, G. Gunendran<sup>1</sup>, Z. Usman<sup>1</sup>, J. Harding<sup>1</sup>, K. Case<sup>1</sup>

<sup>1</sup>Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire, LE11 3TU, UK.

<sup>2</sup>CODATA France, 5, rue A. Vacquerie, F-75016 Paris, France and Univ - Lille Nord de France, F-59000 Lille, France, LM2O, Ecole Centrale de Lille, Cité Scientifique -- BP 48 -- 59651 Villeneuve d'Ascq.

## **Abstract**

Production-centric international standards are intended to serve as an important route towards information sharing across manufacturing decision support systems. As a consequence of textual-based definitions of concepts acknowledged within these standards, their inability to fully interoperate becomes an issue especially since a multitude of standards are required to cover the needs of extensive domains such as manufacturing industries. To help reinforce the current understanding to support the consolidation of production-centric standards for improved information sharing, this article explores the specification of well-defined core concepts which can be used as a basis for capturing tailored semantic definitions. The potentials of two heavyweight ontological approaches, notably Common Logic (CL) and the Web Ontology Language (OWL) as candidates for the task, are also exposed. An important finding regarding these two methods is that while an OWL-based approach shows capabilities towards applications which may require flexible hierarchies of concepts, a CL-based method represents a favoured contender for scoped and facts-driven manufacturing applications.

**Keywords:** ISO standards, ontologies, Common Logic, OWL, core concepts, consolidation

# 1 Introduction

The ISO (International Standardisation Organisation) community is responsible for the development and dissemination of shared best practices related to a wide range of subject areas. Of particular relevance to the field of information and knowledge sharing in product design and manufacture is ISO Technical Committee 184 (ISO TC184). The standards developed within their Sub-Committees (SCs) contribute to the establishment of a framework for exchanging product lifecycle information, with the additional benefit of promoting interoperability across manufacturing applications.

However, in several instances, concept terms and definitions acknowledged in standards remain largely textual in nature (Michel, 2005) and are, therefore, liable to subjective interpretation. Furthermore, it is not always obvious how certain concepts, supposed to be coherent across standards, are consistent with one another (Usman *et al.*, 2010). These issues imply that, although standards can be of considerable advantage in relatively confined domains where their common interpretation is guaranteed, they nevertheless fall short of meeting seamless information exchanges across larger manufacturing domains. This is because larger domains require a spectrum of standards for satisfying all relevant manufacturing-driven activities. Hence, there is a need for consolidating the meaning of concept terms and definitions across production-centric standards.

The methodology used to leverage improved consolidation of concept semantics involves the ontology-based definition of flexible, foundation or core concepts which can be specialised according to a number of viewpoints (Chungoora and Young, 2010; Usman *et al.*, 2010). Core concepts define the generic semantics of domain terms and these semantics can be tailored according to the definitional requirements of specific standards. This results in the timely capture of meaning in computational form and, thus, better potential for information sharing.

To illustrate the applicability of core concepts and their specialisations, the notion of 'Resource' has been selected as exemplar in this article. In the investigation, two expressive (i.e., heavyweight) ontological formalisms have been exploited, namely (1) the Knowledge Frame Language (KFL) (KFL Reference, 2010) which is derived from Common Logic (ISO/IEC 24707, 2007) and (2) the Web Ontology Language Description Logics (OWL DL) (W3C Website, 2006). These two formalisms have been specifically chosen since our awareness of heavyweight ontologies coupled with ongoing work in the consolidation of standards, such as the 'Future SC4 Architecture' (Leal *et al.*, 2009), stipulate that these formalisms offer clear potentials to meet the level of rigour required for capturing expressive semantics.

The structure of this article involves, in section 2, a description of standardisation activities in the field of production engineering. Section 3 elaborates on the requirements for consolidating standards-based concepts and identifies the need for formalising semantics using ontological approaches. Section 4 identifies and describes the chosen methodology which subsequently leads to a verification of the approach in section 5 using relevant software tools. Finally section 6 provides the concluding remarks to this work and identifies areas of issues for further work.

## 2 Standardisation activities in the domain of production engineering

Standardisation activities in the domain of production engineering, at the international level, are done within the framework of the ISO (ISO Website, 2011), and perhaps the most important activity in this domain is achieved by the ISO TC184 committee, whose scope is '*standardisation in the field of industrial automation and integration concerning discrete part manufacturing and encompassing the applications of multiple technologies, i.e. information systems, machines and equipments and telecommunications*'.

This means that the standards developed are applicable to manufacturing and process industries, relevant to all sizes of businesses, and vital to extending exchanges across the globe through e-business. These standards are developed within the different ISO TC184 sub-committees, namely SC1 (physical device control), SC2 (robots and robotic devices), SC4 (industrial data), and SC5 (interoperability, integration, and architectures for enterprise systems and automation applications). They cover various domains related to industrial automation and integration such as: enterprise modelling, enterprise architecture, communications and processes, integration of industrial data for exchange, access and sharing, life cycle data for process plants, manufacturing management, mechanical interfaces and programming methods, part libraries, physical device control (Cutting-Decelle *et al.*, 2009). The standards mentioned in this article such as ISO 10303 (STEP), ISO 15531 (MANDATE), ISO 18629 (PSL) and ISO 13584 (PLIB), are being developed within the SC4 through a joint committee between SC4 and SC5, known as JWG8. The ISO 13399 cutting-tools standard has been developed by the ISO TC29/WG34 (cutting tool data representation and exchange) committee.

The activity of the SC4 committee deals with standardisation of the industrial data related to products including, but not limited to geometric design and tolerance data, material and functional specifications, product differentiation and configuration, process design data, production data (including cost), product support and logistics, life cycle data, quality data and disposal planning data (ISO TC184/SC4 Website, 2011). It also includes organisational data provided by relationships between enterprises or between components of a single enterprise for the purposes of supplier identification. It includes personnel data to the extent of identification of approvals, including capacities and capabilities. Specifically excluded are business-planning data such as profit projections, cash flow, and any other personnel data or organisational data. The goal of SC4 is the creation and maintenance of standards that enable the capture of information comprising a computerised product model in a neutral form without loss of completeness and integrity throughout the lifecycle of the product.

To date, in terms of information representation and management, standardisation committees concentrate on the semantics of the messages, considered as a fundamental need for sharing, storing and exchanging data, information and knowledge. Besides, product data engineering is now mature and enterprises are more and more interested in the standardisation of their whole information systems and architectures. For this reason, multi-standard based approaches are of considerable benefit to companies, since they facilitate the integration of the whole information system, from the basic technical processes through to the high level business processes.

However, although numerous standards are available and applicable at different levels of production management systems, their joint use highlights various problems among which the lack of compatibility of the information models and the vocabulary used, which have not been defined in the same way even though the terms used are the same. Ontology-based approaches provide an important step towards the investigation of the common 'essence' of the information being handled. Furthermore, ontologies can be integrated with software interfaces, making it easier to convey a higher level of semantics for the exchanges.

Following a resolution proposed during the Parksville SC4 meeting in May 2009, the creation of a Provisional Work Item (PWI) on 'Future SC4 Architecture' has been realised, in order to explore the range of options for exploiting new technologies to support interoperability between SC4 standards and the work of other committees and operation across a range of platforms. Other objectives of the 'Future SC4 Architecture' are related to the recommendation of an overall architecture for future use by SC4 standards, the definition of migration paths for existing SC4 capabilities, the identification of specific licensing, commercial or other administrative barriers and the illustration and validation of any conclusions against a representative example. Elements of the earlier understanding from the research which underlies this paper has already been provided into this group and we also perceive the work presented in this article as a further potential contribution, as it is based on a real example taken from the domain of production-centric information.

### **3 Needs and requirements for consolidated production information**

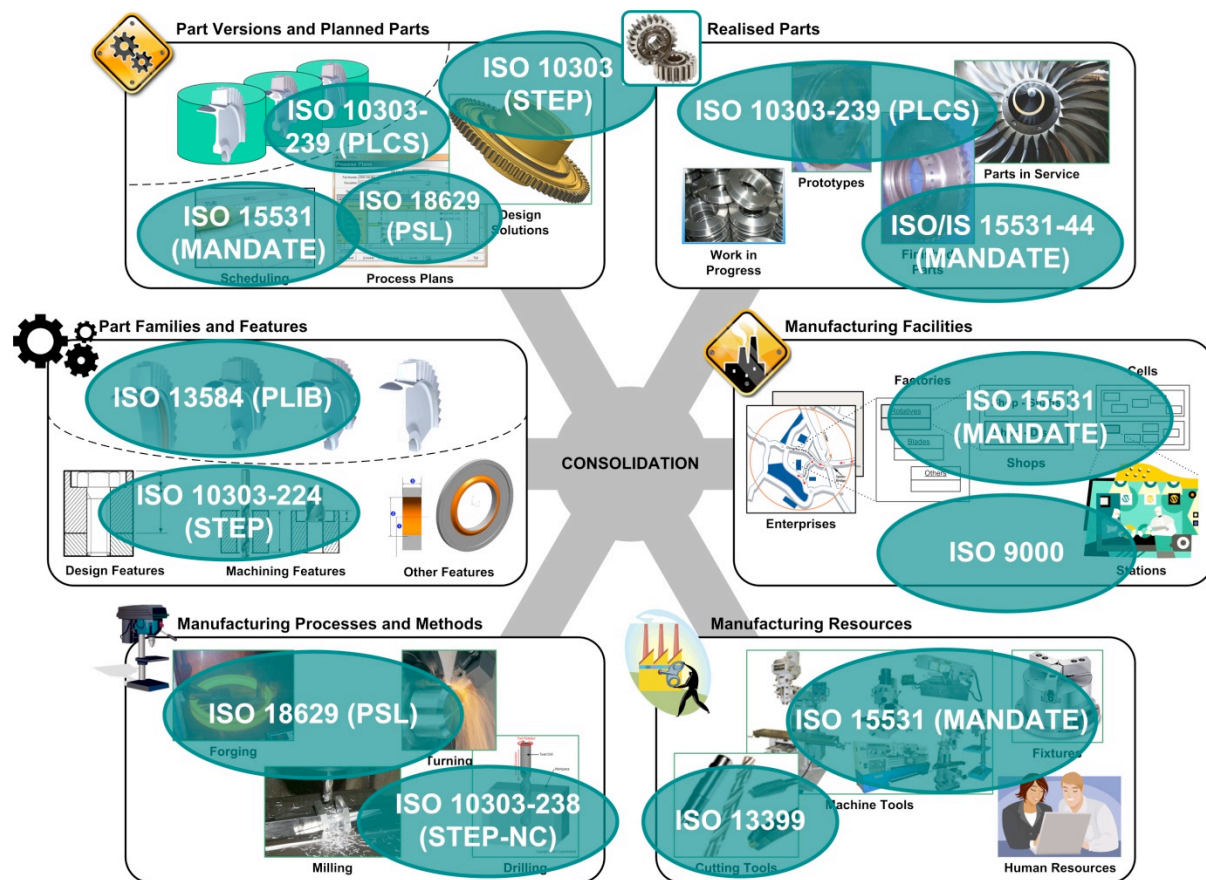
This section analyses the needs and requirements for achieving the consolidation of standards-based concepts. The term 'Resource' is taken as an example, to show that there exist variations in the definition of terms across standards. The ontology-based formalisation of semantics is then identified as a key direction for consolidating meaning in computational form.

#### **3.1 A manufacturing information organisation perspective**

As a consequence of a multitude of standards which have been established to support production engineering, it becomes important to visualise how they fit within today's best practice manufacturing information organisation activities. Figure 1 identifies how examples of production-centric standards are interspersed in relationship to pertinent categories of information, where relevant silos of information and knowledge are attributed to 'part versions and planned parts', 'realised parts', 'part families and features', 'manufacturing facilities', 'manufacturing processes and methods' and 'manufacturing resources'.

Consider the two segments of Figure 1 pertaining to 'part versions and planned parts' and 'part families and features'. It has been recognised that the meaningful classification of parts and features and the knowledge about their interactions enable the improved configuration of design and manufacturing planning solutions (Wang *et al.*, 2003; Gunendran and Young, 2008). From a standards perspective, this is analogous to applying the concepts present in ISO 13584 (PLIB) and ISO 10303-224 (STEP) for structuring libraries of parts and to capture machining feature definitions respectively. Moreover, concepts acknowledged in ISO 10303-239 (PLCS), ISO 18629 (PSL) and ISO

15531 (MANDATE) provide a basis for information representation related to product versions, process planning, scheduling and the like. A similar understanding also applies to the other segments shown in Figure 1.



**Figure 1: A manufacturing information organisation perspective for consolidated product information**

It is clearly discernible from the diagram that there exists overlapping information content defined across various standards which are vital to manufacturing information organisation activities. Consequently, this implies that to progress towards the reuse of standards in a consolidated way across multiple computer systems, production-centric standards need to align together so that they benefit from an appropriate level of compatibility of the defined concepts. For this to happen, it becomes essential to understand the implications of the semantics of the concepts defined across standards.

### 3.2 Semantics and their implications

The formalisation of concepts coming from standards constitutes an important step for enabling the utilisation of these concepts across computer-based applications to support decision-making procedures. To be able to improve the sharing of information across such applications, in a way that the meaning associated to the concepts can be fully interpreted, demands the formal and explicit representation of the semantics associated to these concepts. Currently, the meaning of important terms identified in ISO standards has followed textual definitions which can be varied (Michel, 2005), thereby implying that the definitions can be subjectively interpreted by human beings. Furthermore, the result of an underlying lack of formal semantics leads to computer-based applications that

poorly interoperate since the commonalities and/or differences in the terms and their definitions remain ambiguous.

As an example Figure 2 takes one term, 'Resource', acknowledged in four different standards together with the corresponding informal (i.e., natural language) definitions of the term. It is obvious that the textual definitions are varied, although they are supposed to be coherent based on the underlying requirement for information sharing. Consider the textual definition shown in Figure 2 (1), where it is mentioned that 'this definition includes ISO 10303-49 definition'. It is not directly clear how the two definitions in (1) and (2) can be mapped by simply reading through the informal descriptions. It is also noticeable that the definition of 'Resource' in ISO 15704 adopts a somewhat different viewpoint when viewed against the ISO 15531-1, ISO 18629-1 and ISO 10303-49 definitions.

<p><b>1. Resource (ISO 15531-1; ISO 18629-1)</b> Any device, tool and means, except raw material and final product components, at the disposal of the enterprise to produce goods or services.</p> <p>This definition includes ISO 10303-49 definition.</p>
<p><b>2. Resource (ISO 10303-49)</b> Something that may be described in terms of a behaviour, a capability, or a performance measure that is pertinent to the process.</p>
<p><b>3. Resource (ISO 15704)</b> An enterprise entity that provides some or all of the capabilities required by the execution of an enterprise activity and/or business process.</p>

**Figure 2: Multiple definitions associated with the term 'Resource'**

### 3.3 Ontology-based formalisation of semantics

The need for capturing the semantics of concepts (in computational form) defined across ISO standards can be satisfied by exploiting ontology-based approaches, primarily since ontologies support the formal representation of some shared domain of interest. The level of rigorousness in the representation of formal semantics constitutes an important factor and is dependent on the ontological approach pursued, which can be either 'lightweight' or 'heavyweight' in nature. Lightweight ontologies are simple representations that involve taxonomies of concepts and relations and assume the readily-understandable meanings associated with concept terms (Gómez-Pérez *et al.*, 2004). Heavyweight ontologies, on the other hand, on top of having the lightweight structures also benefit from well-defined axioms that constrain and clarify the interpretation of the terms, (Gómez-Pérez *et al.*, 2004) and are, therefore, preferred for ensuring greater confidence behind the formal meaning of terms.

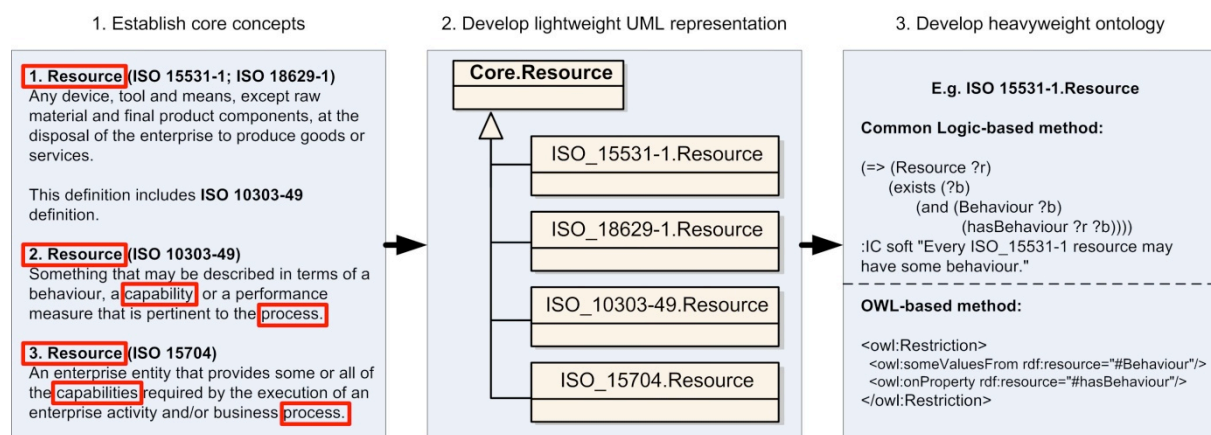
This article, therefore, specifies a heavyweight ontology-based methodology for the rigorous definition of core concepts, which can be extended to address more specific domain semantics. It is important to note that the identified methodology considers the 'Resource' concepts illustrated in Figure 2, because the term 'Resource' is a vital concept that has been acknowledged across various standards in the domain of industrial automation. Several other multiple concept definitions are currently also under consideration within the Interoperable Manufacturing Knowledge Systems (IMKS) project (Usman *et al.*, 2010).

## 4 Methodology for developing concepts and their specialisations

This section identifies the ontology-based methodology which has been investigated in the effort towards the consolidation of standards in the domain of industrial automation. The methodology consists of three main stages, exploiting two ontological formalisms namely KFL (Knowledge Frame Language: a Common Logic-based formalism) and OWL DL (Web Ontology Language Description Logics: a Description Logic-based formalism).

### 4.1 Overview

A generic breakdown of the stages in the methodology is identified in Figure 3. The first two stages lead to the capture of an agreed lightweight UML representation, which in a third stage is formalised in (1) KFL and (2) OWL DL. The choice of these two formalisms is based on their suitability to model heavyweight ontologies and also because they represent examples of ontology languages with two fundamentally different assumptions namely the Closed World Assumption (CWA) and Open World Assumption (OWA) respectively, where each is perfectly coherent with the framework in which it is assumed (Orsi and Tanca, 2010).



**Figure 3: Methodology for exploring KFL and OWL DL ontological formalisms to consolidate semantics**

An important difference between the two formalisms is that in CWA a failure to prove a statement implies that the latter is false, which follows the Negation As Failure (NAF) inference rule, while in OWA, a statement cannot be inferred to be false on the basis of a failure to prove it (Sirin *et al.*, 2008). On the other hand, the decision towards the choice of formalisms is that Common Logic is itself an international standard, intended as a logic framework for information exchange and transmission (ISO/IEC 24707, 2007). Its potentials for consolidating production-centric standards remained to be addressed and, therefore, this article explores the potential application of KFL for this purpose.

Moreover, OWL has become a leading widespread heavyweight Description Logic-based language. Its use has been acknowledged in a number of contributions towards the ISO standards community, such as SemanticSTEP under the S-TEN project (S-TEN Webpage, 2011) and the integration of the International Electrotechnical Commission (IEC) TC 57 standards (Uslar, 2008). While other



heavyweight approaches, e.g., Protégé Frames with Protégé Axiom Language (Protégé Website, 2010) could have been utilised, these fall outside our current scope because our choice of KFL and OWL already catches two broad groups of heavyweight logic languages.

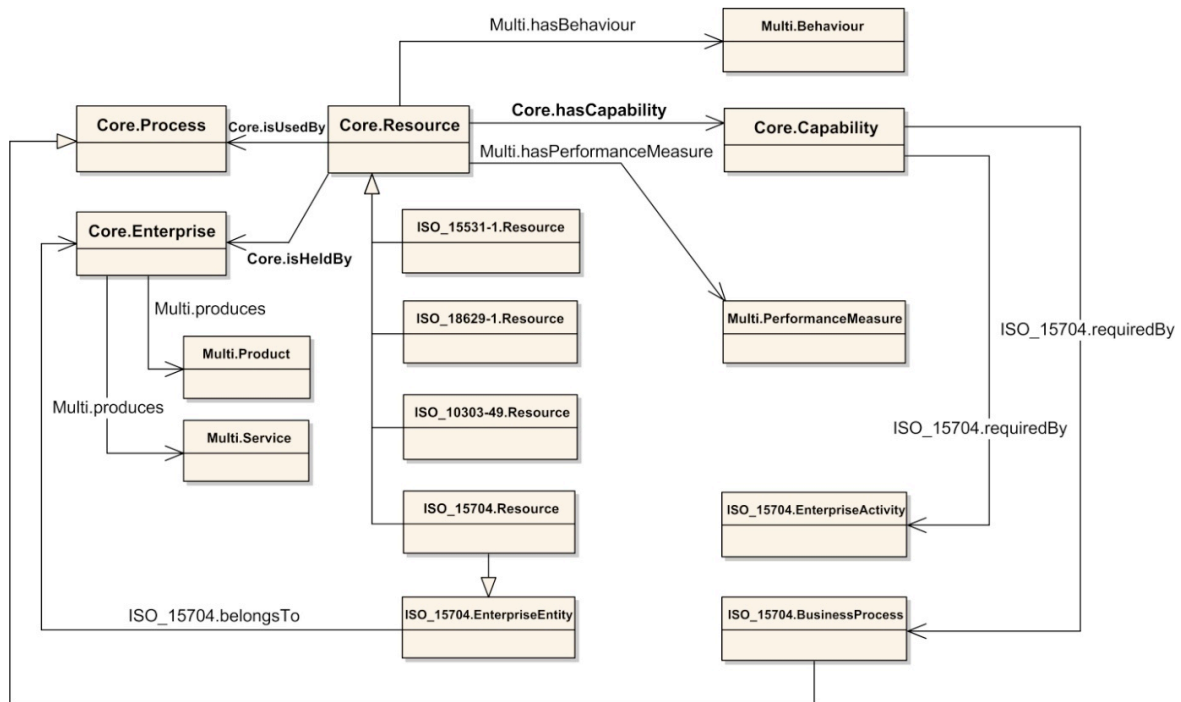
## **4.2 Establish core concepts**

The first step in the methodology requires the analysis of the natural language descriptions leading to the capture of core concepts. This important activity, requiring the input from domain experts and knowledge engineers, helps to derive a first-pass characterisation as well as the identification of other important concepts referenced across multiple definitions. In the case of the 'Resource' concepts, the latter recursively appears across a number of standards. This supports the need for a single generic 'Resource' core concept whose specialisations then result in the tailored definitions for each standard notably ISO 15531-1, ISO 18629-1, ISO 10303-49 and ISO 15704. Other gathered core concepts from this example are, for instance, the notions of 'Capability' and 'Process' which are central to the description of the multiple 'Resource' definitions.

## **4.3 Develop lightweight UML representation**

The second stage starts with the development of an initial lightweight representation which captures entities in the form of proto-classes and proto-relations. The prefix 'proto' suggests that the entities are tentative and subject to further inquiry before final change of status (IDEF5 Method Report, 1994). Some examples of proto-classes are 'Resource', 'Capability', 'Process', 'Enterprise' and 'Behaviour'. Examples of binary proto-relations are 'hasCapability' to associate resources to their capabilities and 'requiredBy' to associate, e.g., capabilities to processes. The initial representation is then gradually refined into an agreed lightweight model.

Figure 4 captures a UML class model of the final lightweight representation for the core 'Resource' concept and its specialisations to meet the needs of distinct standards. A UML class representation has been chosen as it is considered good practice to support the process of ontology development using appropriate ontology design schematics, which serve as platform-independent representation of fundamental ontological structures prior to their implementation (Chungoora *et al.*, 2010). On the other hand, prefixes such as 'Core', 'ISO\_15531-1' and 'ISO\_15704' accompany ontological entities in order to identify the specific domains in which these entities are defined.



**Figure 4: Lightweight UML representation to consolidate multiple 'Resource' concepts**

### 4.3.2 Classes of concepts

The class 'Core.Resource' is a concept defined at the most generic level in order to provide a first-level semantic ground for consolidating standards-specific interpretations. Specialised classes, such as 'ISO\_15531-1.Resource' and 'ISO\_18629-1.Resource', are made sub-classes of 'Core.Resource', which implies that via the exploitation of subsumption, the behaviour exhibited by 'Core.Resource' is inherited by its sub-classes. Similarly, the class 'ISO\_15704.BusinessProcess' identified in Figure 4 should inherit the behaviour displayed by its parent class 'Core.Process'. The specification of class hierarchies enables specialised concepts to follow consistently the core concepts from which they derive. This aspect is particularly important at the heavyweight ontology development stage.

### 4.3.3 Relations between classes

The lightweight representation portrayed in Figure 4 also captures links between classes. These relationships between classes constitute additional entities required to build the vocabulary of the ontology. At the generic 'Core' level three relations exist, namely 'Core.isUsedBy', 'Core.isHeldBy' and 'Core.hasCapability'. The latter is a binary relation that holds between the classes 'Core.Resource' and 'Core.Capability' as well as their specialisations if any. The class 'Core.Capability' and the relation 'Core.hasCapability' are present within the 'Core' identifier itself, as 'Core.Capability' is a term referenced across all four standards under consideration.

On the other hand the relation 'produces', which holds between 'Core.Enterprise' and the union of 'Multi.Product' and 'Multi.Service', is situated in the 'Multi' identifier which holds the relation as it is simultaneously common to the three identifiers 'ISO\_15531-1', 'ISO\_18629-1' and 'ISO\_10303-49'. A similar understanding is required in order to construct the lightweight representation such that it corresponds as closely as possible with the natural language descriptions. After the development of

the agreed lightweight UML representation, the latter is then used as a basis for constructing the corresponding heavyweight ontologies in KFL and OWL DL.

## 4.4 Develop heavyweight ontology in KFL

The Knowledge Frame Language (KFL), exploited in this task, is a CL-based formalism developed by Highfleet (2010) and is described as the combination of a convenience layer of syntactic sugar that sits on top of a base layer of logical syntax called ECLIF (Extended Common Logic Interchange Format) (KFL Reference, 2010). The KFL formalism holds a rigorously-defined meta-model which imparts a rich set of semantics for the development of heavyweight ontologies which typically consist of user-defined contexts (identifiers), classes, relations and integrity constraints.

### 4.4.1 Declaration of contexts

Contexts, i.e., KFL identifiers, are the first elements of the ontology that need to be formalised. In KFL the declaration of contexts such as 'Core' and 'ISO\_15704' is shown next:

```
(UserContext Core)
(supCtx Core MLO)
(name Core "Core")
(rem Core "The 'Core' context defines the identifier for all ontological entities
declared at the core-concept generic level.")

(UserContext ISO_15704)
(supCtx ISO_15704 Core)
(name ISO_15704 "ISO_15704")
(rem ISO_15704 "The 'ISO_15704' context defines the identifier for all ontological
entities declared at ISO 15704 level.")
```

The 'Core' context is created as an instance of 'UserContext'. All contexts require some super-context and in the case of the 'Core' context, the system-specific 'MLO' is stated. A suitable name can be attributed to the context using the 'name' relation and informal remarks added using the 'rem' relation. Also notice how 'ISO\_15704' is defined as having the 'Core' context as its super-context. In the approach, similar hierarchies of contexts can be built to ensure that specialisations are clearly referenced, e.g., 'Core' is a generic context for consolidation which also has 'Multi' as a sub-context which itself is decomposed into the three contexts for the specialised 'ISO\_1551-1', 'ISO\_18629-1' and 'ISO\_10303-49' since the 'Resource' concepts coming from these three standards share obvious commonalities based on their natural language descriptions.

### 4.4.2 Declaration of classes and relations

Using similar KFL assertions, taxonomies of classes and relations can be created. Quoted next are the required assertions for building the specialised 'ISO\_15704.Resource' class.

```
(Property ISO_15704.Resource)
(sup ISO_15704.Resource Core.Resource)
(sup ISO_15704.Resource ISO_15704.EnterpriseEntity)
```

The class 'ISO\_15704.Resource' is created as an instance of the meta-class 'Property'. Using the basis of Figure 4, 'ISO\_15704.Resource' is made a sub-class of 'Core.Resource' through the 'sup' binary relation. The class is also made a sub-class of 'ISO\_15704.EnterpriseEntity'.

The following set of KFL lines illustrate how relations can be created. The definition of the relation 'ISO\_15704.belongsTo' consists of:

```
(BinaryRel ISO_15704.belongsTo)
(argProp ISO_15704.belongsTo 1 ISO_15704.EnterpriseEntity)
(argProp ISO_15704.belongsTo 2 Core.Enterprise)
```

'ISO\_15704.belongsTo' is specified as a binary relation by instantiating 'BinaryRel'. The signature of the relation is captured by using the ternary relation 'argProp' where 'ISO\_15704.EnterpriseEntity' is asserted as the domain (i.e., class in the first argument position) of the relation, and 'Core.Enterprise' as its range (i.e., class in the second argument position).

#### 4.4.3 Declaration of integrity constraints

Integrity constraints (ICs) are axioms written as part of KFL ontologies and act as a knowledge verification method for enabling the consistency checking of the semantics of members of classes (i.e., individuals). In other words, integrity constraints restrict the ways in which KFL-constructed Knowledge Bases can be populated. Consider the 'Core' context where three integrity constraints have been generated as shown next:

```
(=> (Core.Resource ?r)
      (exists (?c)
        (and (Core.Capability ?c)
              (Core.hasCapability ?r ?c))))
:IC soft "Every resource may have some capability."

(=> (Core.Resource ?r)
      (exists (?e)
        (and (Core.Enterprise ?e)
              (Core.isHeldBy ?r ?e))))
:IC soft "Every resource may be held by some enterprise."

(=> (Core.Resource ?r)
      (exists (?p)
        (and (Core.Process ?p)
              (Core.isUsedBy ?r ?p))))
:IC soft "Every resource may be used by some process."
```

The three integrity constraints involve existential quantification in the definition of 'Core.Resource' individuals. For example, if a 'Core.Resource' individual were asserted in the Knowledge Base then it may be required, though not compulsory, to specify a 'Core.Capability' individual that relates to the 'Core.Resource' individual via the 'Core.hasCapability' relation. Because these integrity constraints are meant to allow flexibility over the assertion of individuals in the Knowledge Base, they are appended with the directive ':IC soft'. The message after each ':IC soft' directive would warn the user of a violation of the constraint. It is important to notice that the generic 'Core' context does not

attempt to enforce any compulsory semantics on its specialisations, hence the justified use of soft integrity constraints.

Hard integrity constraints are of value to the specialised contexts for 'ISO\_15531-1', 'ISO\_18629-1', 'ISO\_10303-49' and 'ISO\_15704'. Unlike soft constraints, hard integrity constraints specify compulsory conditions that the instances of classes should meet and are, therefore, useful for formalising instance definitions. One example of a hard integrity constraint applicable to the 'ISO\_15704' context is next identified. The integrity constraint captures a more complex signature, since the relation 'ISO\_15704.requiredBy' has 'Core.Capability' as its domain and a range consisting of either 'ISO\_15704.EnterpriseActivity' only or 'ISO\_15704.BusinessProcess' only or both classes 'ISO\_15704.EnterpriseActivity' and 'ISO\_15704.BusinessProcess'.

```
(=> (ISO_15704.requiredBy ?x ?y)
    (and (Core.Capability ?x)
         (or (ISO_15704.EnterpriseActivity ?y)
             (ISO_15704.BusinessProcess ?y))))
:IC hard "The relation ISO_15704.requiredBy only holds between capabilities and
enterprise activities and/or business processes."
```

Integrity constraints can be of a more intricate nature compared to the ones explored so far. One more complicated axiom is identified next and involves the 'ISO\_15704.Resource' concept. The constraint formally captures the pertinent understanding from the natural language description of the 'ISO 15704.Resource' concept, as can be depicted in the integrity constraint message.

```
(=> (ISO_15704.Resource ?r)
    (or (exists (?c ?a)
         (and (Core.Capability ?c)
              (ISO_15704.EnterpriseActivity ?a)
              (Core.hasCapability ?r ?c)
              (ISO_15704.requiredBy ?c ?a)))
        (exists (?c ?a)
         (and (Core.Capability ?c)
              (ISO_15704.BusinessProcess ?a)
              (Core.hasCapability ?r ?c)
              (ISO_15704.requiredBy ?c ?a))))))
:IC hard "Every ISO 15704 resource has some capability required by some enterprise
activity and/or business process."
```

## 4.5 Develop heavyweight ontology in OWL DL

The third stage in the methodology also demands using the lightweight UML representation as a basis for developing a heavyweight version of the ontology expressed in OWL DL, the latter based on the OWL 1.1 specification (Motik *et al.*, 2006). The various components of the Description Logic-based ontology include the declaration of namespaces (identifiers), classes, relations (known as 'properties' in OWL) and restrictions.

#### 4.5.1 Declaration of namespaces

In the OWL DL approach, namespaces offer the facility to accommodate identifiers for referencing ontological entities. The statements below provide an example of the necessary content for declaring the 'Core' identifier. These statements are present as attributes and embedded in the root element of the ontology for the generic 'Core.Resource' concept.

```
xmlns="http://www.owl-ontologies.com/Core.owl#"
xml:base="http://www.owl-ontologies.com/Core.owl"
```

#### 4.5.2 Declaration of classes and relations

Taxonomies of classes as well as binary relations (commonly referred to as properties in OWL) can readily be specified in OWL DL. The following example considers the 'Resource' concept present in the 'ISO\_15704' namespace. The ontology containing 'ISO\_15704.Resource' imports the core concept ontology since 'ISO\_15704' is a specialisation of 'Core' where 'ISO\_15704.Resource' needs to be made a sub-class of 'Core.Resource'. The import action is declared as thus:

```
<owl:Ontology rdf:about="">
  <owl:imports rdf:resource="http://www.owl-ontologies.com/Core.owl"/>
</owl:Ontology>
```

The declaration of the class 'ISO\_15704.Resource' can then be captured, where the 'rdfs:subClassOf' directive is used to assert that 'ISO\_15704.Resource' is a sub-class of 'Core.Resource'. Note that the statement below assumes a predefined namespace for ISO 15704, which means that the statement *rdf:ID="Resource"* is in fact implying the 'Resource' term within the 'ISO\_15704' namespace.

```
<owl:Class rdf:ID="Resource">
  <rdfs:subClassOf rdf:resource="http://www.owl-ontologies.com/Core.owl#Resource"/>
</owl:Class>
```

For the declaration of a binary relation such as 'ISO\_15704.belongsTo', the 'owl:ObjectProperty' directive is employed. The signature and directionality of the relation can be captured by stating the domain and range of the relation using 'rdfs:domain' and 'rdfs:range' respectively.

```
<owl:ObjectProperty rdf:about="#belongsTo">
  <rdfs:domain rdf:resource="#EnterpriseEntity"/>
  <rdfs:range rdf:resource="http://www.owl-ontologies.com/Core.owl#Enterprise"/>
</owl:ObjectProperty>
```

#### 4.5.3 Declaration of restrictions

Restrictions in OWL DL are used to specify axioms for (1) the consistency checking of individuals populated in OWL-constructed Knowledge Bases and (2) computing inferred taxonomies of classes based on the restrictions they carry, in such a way that equivalence and subsumption amongst classes can be identified. All types of restrictions describe an unnamed set that could contain some individuals, where the set can be thought of as an anonymous class (Horridge *et al.*, 2004).

A ‘necessary condition’ in OWL is a restriction where an anonymous super-class of a named class is specified to support the creation of a primitive class. The expression identified next, portrays a necessary condition placed over the class ‘Core.Resource’, which facilitates (1) the consistency checking of the semantics of instances of ‘Core.Resource’ that should be related to instances of ‘Core.Capability’ via the ‘Core.hasCapability’ relation and (2) the inference of ‘Core.Resource’ as a sub-class of some other class (if any) defined with the condition containing ‘Core.hasCapability’ and ‘Core.Capability’.

```
<owl:Class rdf:ID="Resource">
  <rdfs:subClassOf rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
  <rdfs:subClassOf>
    <owl:Restriction>
      <owl:onProperty>
        <owl:ObjectProperty rdf:ID="hasCapability"/>
      </owl:onProperty>
      <owl:someValuesFrom rdf:resource="#Capability"/>
    </owl:Restriction>
  </rdfs:subClassOf>
</owl:Class>
```

Another form of restriction in OWL is a ‘necessary and sufficient condition’ where an anonymous equivalent class of a named class is specified to support the creation of a defined class. An example of such a restriction has been utilised in the definition of ‘ISO\_15704.EnterpriseEntity’ and this restriction implies (1) the consistency checking of instances of ‘ISO\_15704.EnterpriseEntity’ that should be related to instances of ‘Core.Enterprise’ via the ‘ISO\_15704.belongsTo’ relation, (2) that the inference of some anonymous individual (if any) related to some instance of ‘Core.Enterprise’ through the relation ‘ISO\_15704.belongsTo’ is an instance of class ‘ISO\_15704.EnterpriseEntity’ and (3) the inference of some other class (if any) that shares the same restriction as being an equivalent class.

```
<owl:Class rdf:ID="EnterpriseEntity">
  <owl:equivalentClass>
    <owl:Restriction>
      <owl:someValuesFrom rdf:resource="http://www.owl-
        ontologies.com/Core.owl#Enterprise"/>
    <owl:onProperty>
      <owl:ObjectProperty rdf:about="#belongsTo"/>
    </owl:onProperty>
    </owl:Restriction>
  </owl:equivalentClass>
</owl:Class>
```

## 5 Verification of the approach using KFL and OWL DL platforms

In this section, the heavyweight ontologies have been deployed using platforms capable of handling ontologies expressed in KFL and OWL DL. Relevant tests such as consistency checking and the computation of inferred taxonomies have been performed in order to illustrate the applicability of both formalisms.

## 5.1 KFL-based consolidation of semantics

In this task, the Highfleet Integrated Ontology Development Environment (IODE) 4.0 (Highfleet, 2010) has been utilised as ontological platform for the deployment of the developed KFL ontologies. Figure 5 illustrates elements of a 'Resource Consolidation' ontology featuring the integration of 'Resource' concepts and their semantics for each specialised ISO standard under consideration. The figure highlights the 'Core.Resource' class present in the taxonomy as well all the relations that involve this class in their signature. The integrity constraints previously identified in section 4.4.3, where 'Core.Resource' has been referenced, are also shown. It is also worth noting how the IODE system uses constructs defined in the KFL meta-model such as 'integrityRule' and 'fidEx' to enable the manipulation of integrity constraints (ICs).

Figure 6 then depicts the relevance of the consistency checking mechanism (i.e., knowledge verification) aided through the specification of integrity constraints, as a method for ensuring that facts populated in the Knowledge Base of the ontology conform to their set semantics. In the example in Figure 6 (1), an instance of the class 'ISO\_15704.Resource' denoted as 'ISO\_15704.LboroCuttingTool' is first loaded to the Knowledge Base with missing semantics.

As a consequence of this, all participating violated constraints are fired by the system to prompt the user towards the infringements. In this case, the integrity constraints include those inherited from the core concept 'Core.Resource' as well as other constraints specific to the 'ISO\_15704' context. The presence of hard integrity constraints prevents the incorrect or incomplete assertion of the instance 'ISO\_15704.LboroCuttingTool'. To ascertain that this instance is accurately stored in the Knowledge Base, the user asserts all the required semantics as shown in Figure 6 (2).

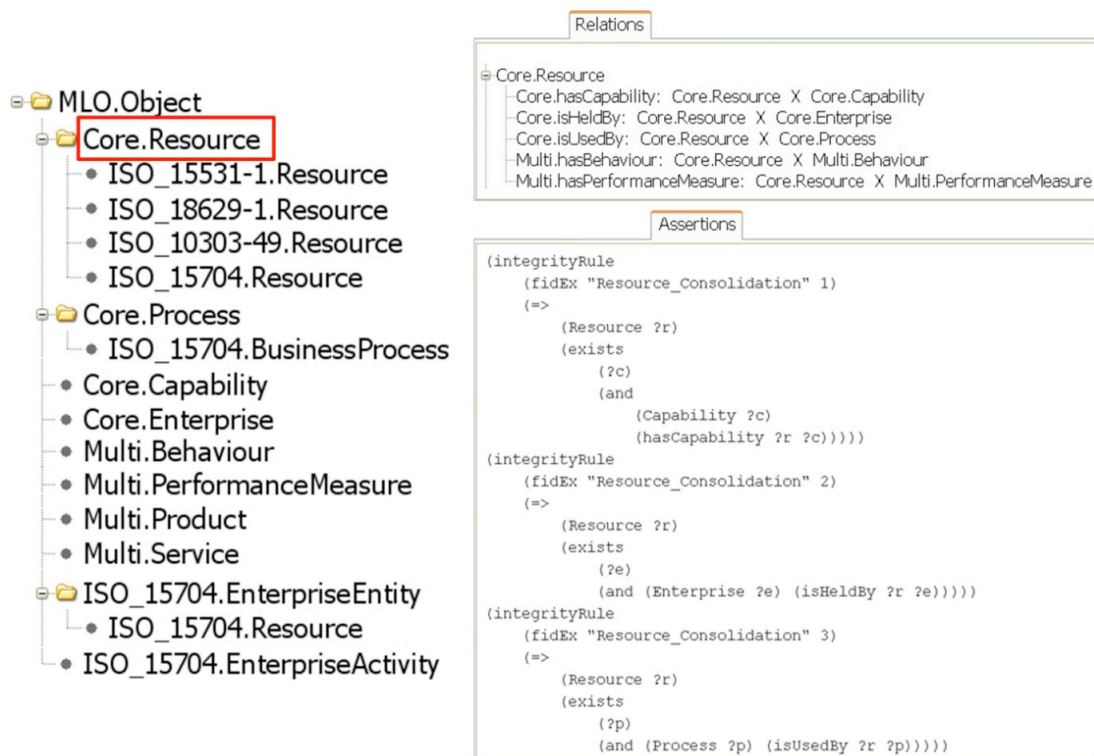
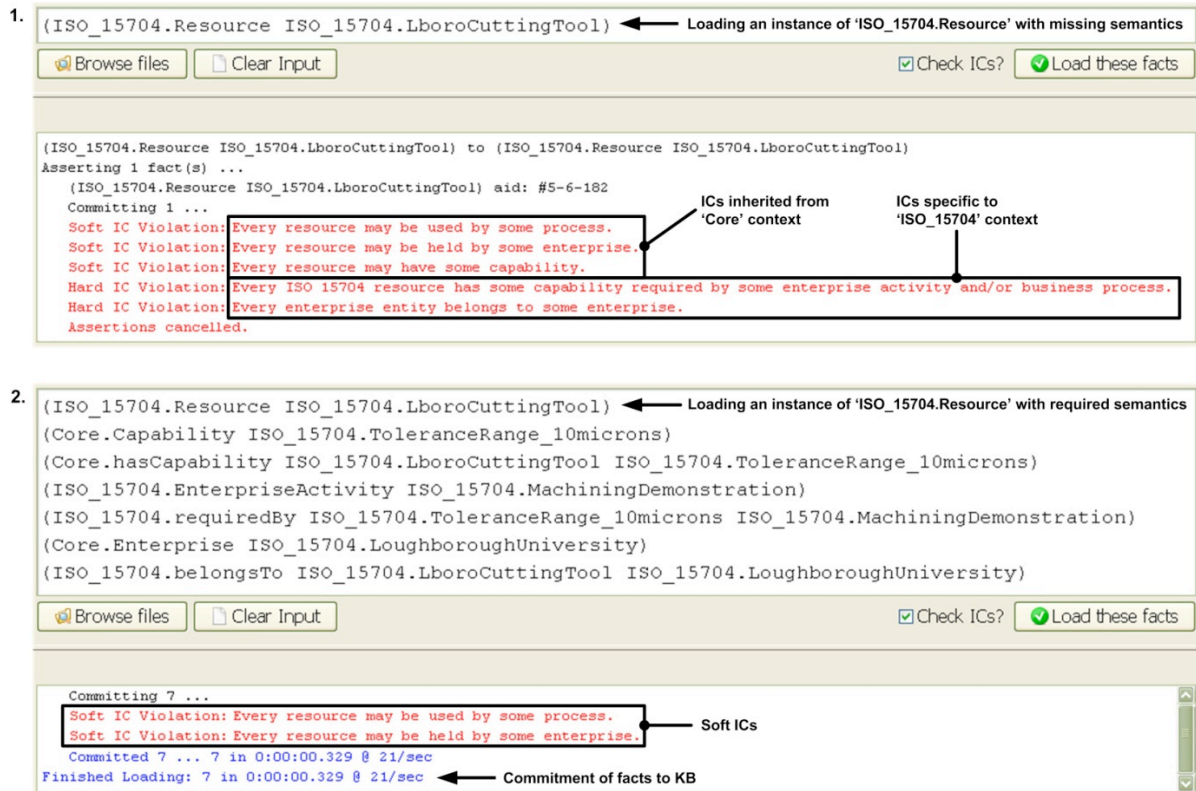


Figure 5: Example of classes, relations and integrity constraints





**Figure 6: Consistency checking of an instance of 'ISO\_15704.Resource'**

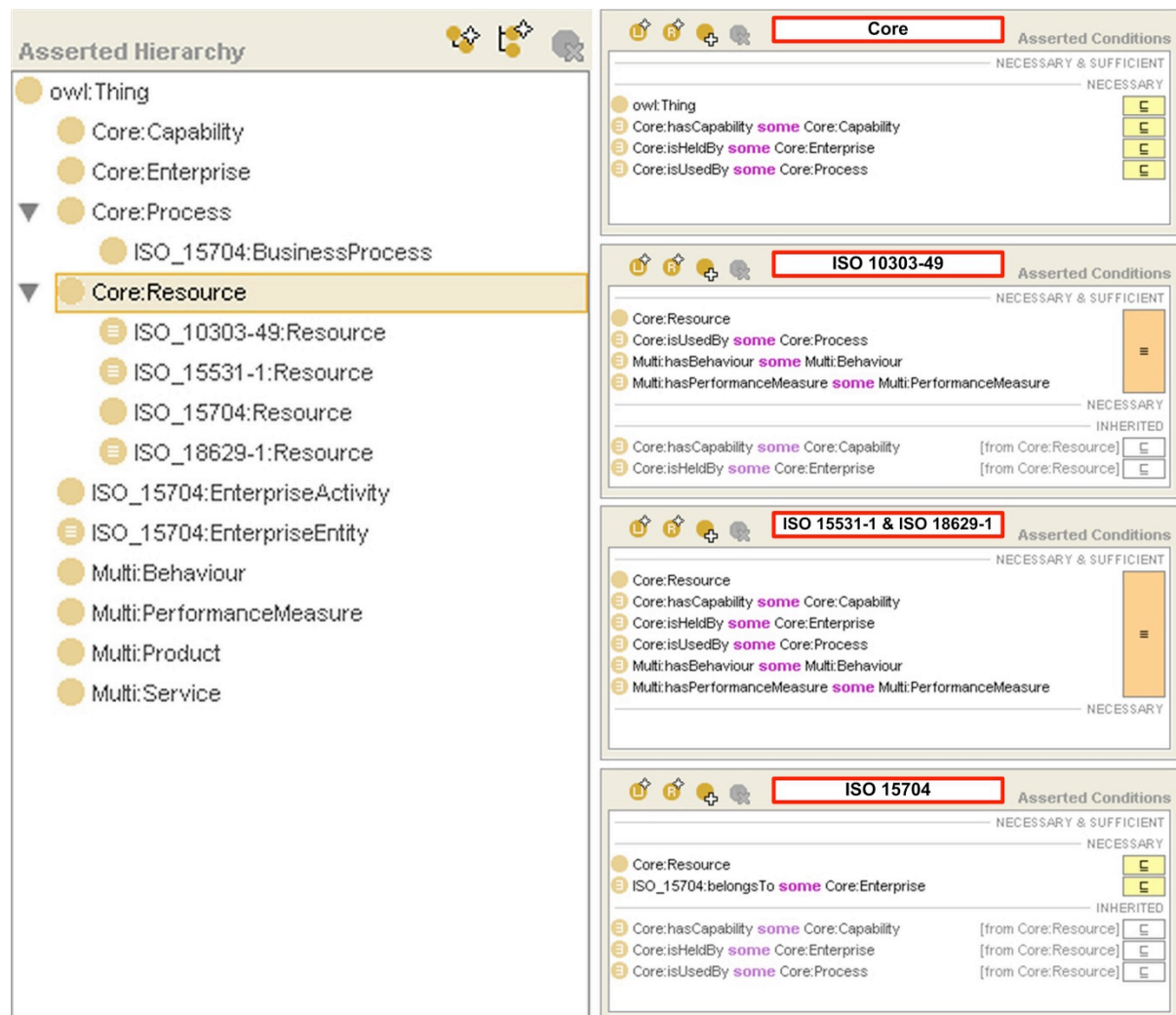
The facts are committed to the Knowledge Base since there are no hard integrity constraint violations. It is important to draw attention towards the persistence of two soft constraints being flagged even if the instance 'ISO\_15704.LboroCuttingTool' has been successfully loaded to the Knowledge Base. This is because, as explained in section 4.4.3, soft integrity constraints are meant to support the flexibility for specialising semantics and are also useful for warning purposes during the assertion of information. A similar understanding applies when loading instances of 'ISO\_15531-1.Resource', 'ISO\_18629-1.Resource' and 'ISO\_10303-49.Resource'.

## 5.2 OWL DL-based consolidation of semantics

This activity uses the Protégé-OWL editor of Protégé 3.4 rc1 (Protégé Website, 2010), as suitable platform for constructing an OWL DL 'Resource Consolidation' ontology. Figure 7 identifies the integrated taxonomy, in which all specialised 'Resource' classes have been imported. The diagram also shows the restrictions for describing 'Core.Resource' as well as its specialisations.

In Figure 7 it can be seen that necessary conditions have been captured for 'Core.Resource'. These semantics are then reused and in conjunction with specific semantics pertaining to each specialisation domain, refined descriptions of 'Resource' concepts are obtained. It is to be noted that the various OWL restrictions shown in Figure 7, comprising 'necessary' and 'necessary and sufficient' conditions, deliver a facility for aiding the consistency checking of instantiated information. An example of this is illustrated in Figure 8 (1), where an instance of 'ISO\_15704.Resource' has been

created with missing semantics. The non-conforming fields are highlighted by the system, which prompts the user to assert the required facts.



**Figure 7: Asserted taxonomy and restrictions**

In addition to exploiting OWL restrictions as a consistency checking mechanism, it has also been possible to use these restrictions as a basis for making inferences about classes. Using the Pellet 1.5.2 reasoner the asserted hierarchy can be classified resulting in an inferred hierarchy as illustrated in Figure 8 (2). The inferred hierarchy enables useful deductions to be made regarding the commonalities between the various 'Resource' concepts in terms of their equivalence and subsumption. The reasoner establishes that, based on the semantics of each 'Resource' class captured via OWL restrictions, the classes 'ISO\_15531-1.Resource', 'ISO\_18629-1.Resource' and 'ISO\_10303-49.Resource' are recursively equivalent to and subsumed under each other, thereby implying the tight interoperability between these concepts. On the other hand, the conditions carried by 'ISO\_15704.Resource' are used to infer that the latter is a sub-class of 'ISO\_15704.EnterpriseEntity', which confirms that the 'Resource' concept defined in ISO 15704 is indeed a type of enterprise entity, as set in its natural language description.

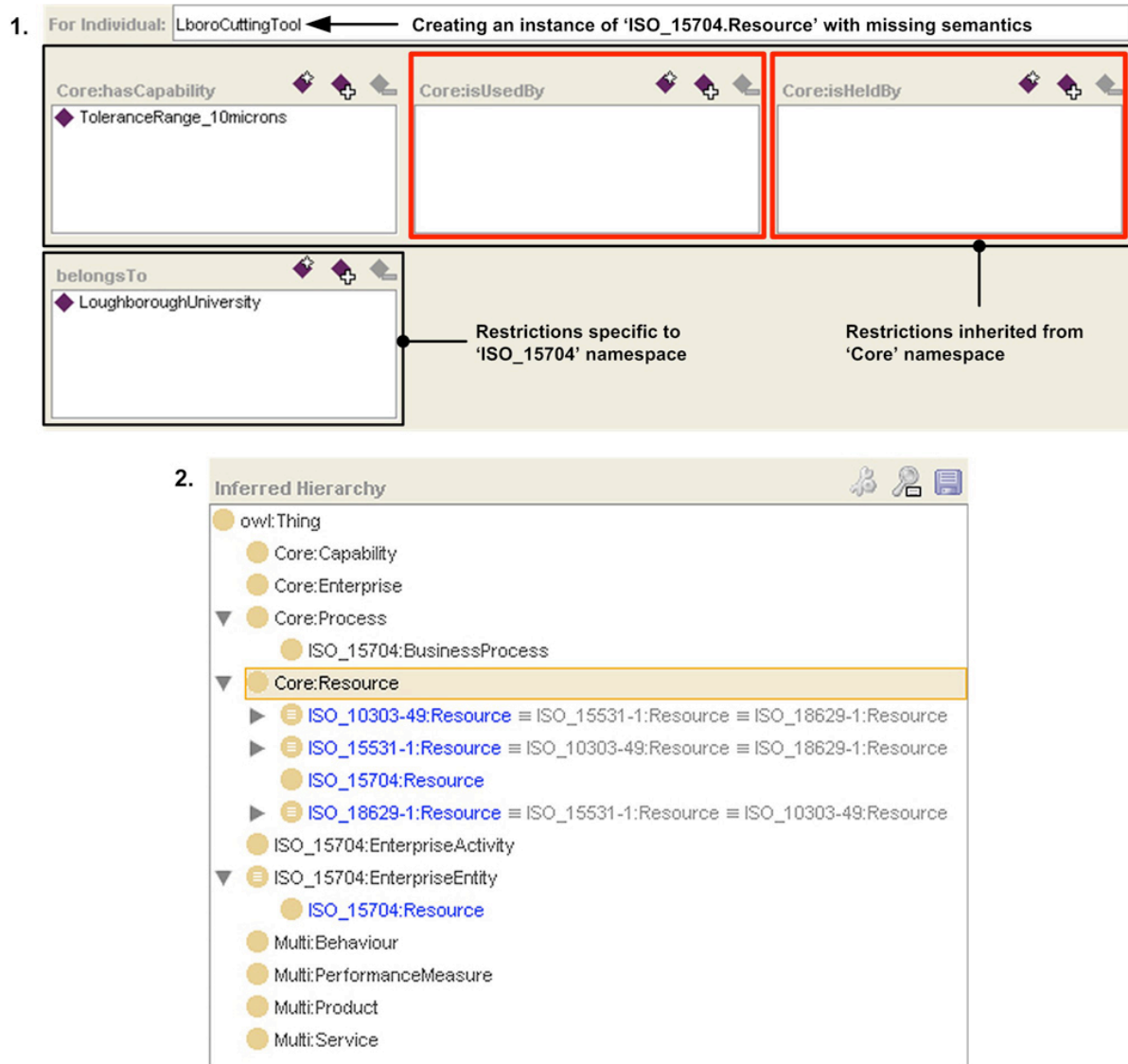


Figure 8: (1) Consistency checking of an instance of 'ISO\_15704.Resource' (2) Inferred hierarchy

## 6 Conclusions and further work

The study of the heavyweight ontology-based methodology has demonstrated the significance of rigorously defining a number of core concepts, from which specialisations can be formalised in an integrity-driven manner. In areas where a spectrum of overlapping terms and definitions exist, well-defined core concepts are imperative to ensuring an adequate level of consensus amongst communities of practice and also across other domains which reuse and exploit these concepts.

Therefore, new ways of developing standards in the field of industrial automation should target the heavyweight logic-based definition of a reference ontology of core concepts. Specific standards can then be specialised on top of the reference ontology. Following this route implies that manufacturing industries which exploit ontology-based standards will achieve greater confidence in the computational meaning associated with information being shared.

From the point of view of consolidating the multiple definitions previously shown in Figure 2, the following has also been achieved:

- CL-based method: validate and ascertain, through the error reporting of integrity constraint messages, that the semantics of instantiated information conforms to rigorously-encoded domain descriptions.
- OWL-based method: ensure that the semantics of instantiated information conforms to less rigorously-encoded domain descriptions. It has also been possible to infer equivalence and subsumption relationships amongst specialised 'Resource' classes using a suitable reasoner.

From this, we stipulate that in order to achieve improved information sharing capability within scoped domains employing ontology-based standards, it is of primary importance to be able to expressively model the information on the basis of what is known to be true of the domain in question. This implies the suitability of a CL-based approach with the Closed World Assumption (CWA) over an OWL-based approach with the Open World Assumption (OWA). A CL-based approach with CWA proves to be more competent in capturing expressive structures for rigorously defining the semantics of instantiated information, which is a vital prerequisite for information sharing.

However, a number of issues remain. As far as the CL-based method is concerned, significant value can be gained from the study of techniques for mapping integrity constraint axioms to help consolidation at the concept definition level. On the other hand, the OWL-based approach would require more expressive axioms for fully constraining instantiated information. The computer science community has already started researching techniques involving the integration of OWL and integrity constraints (Motik *et al.*, 2007; Sirin *et al.*, 2008; Tao, 2010; ExODA project, 2010).

Furthermore, future work should aim at extending the heavyweight ontology-based methodology to encompass other concepts so as to illustrate its applicability to a full range of terms across production-centric standards. Another area where further effort is required is the investigation of methods for structuring levels of increasingly-complex specialisations as well as means of prescribing the management of domain constraints versus desired level of flexibility.

## Acknowledgements

We wish to thank 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 (IMCRC).

## References

Chungoora, N., Canciglieri, O.J. and Young, R.I.M., 2010. Towards expressive ontology-based approaches to manufacturing knowledge representation and sharing. *International Journal of Computer Integrated Manufacturing*. 23(12), pp. 1059-1070.

Chungoora, N. and Young, R.I.M., 2010. The configuration of design and manufacture knowledge models from a heavyweight ontological foundation. *International Journal of Production Research*. DOI: 10.1080/00207543.2010.504754.

Cutting-Decelle, A.F., Barraud, J.L., Young, R.I.M., Michel, J.J., Bigand, M., 2009. Information modeling in production management: the ISO 15531 MANDATE standard. In: Sherif, M.H., ed. *Handbook of enterprise integration*. 3<sup>rd</sup> edition, Taylor & Francis.

ExODA: Integrating Description Logics and Database Technologies for Expressive Ontology-Based Data Access. Research project at the University of Oxford computing laboratory, Oxford, UK. [Online] Retrieved from: <http://www.comlab.ox.ac.uk/projects/ExODA/>

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*. Springer-Verlag London Ltd., London, UK.

Gunendran, A.G. and Young, R.I.M., 2008. Methods for the capture and reuse of manufacturing best practice in Product Lifecycle Management. In: *Proceedings of the 5<sup>th</sup> International Conference on Product Lifecycle Management*. Seoul, Korea.

Highfleet, 2010. [Online] Retrieved from <http://www.highfleet.com/>

Horridge, M., Knublauch, H., Rector, A., Stevens, R. and Wroe, C., 2004. *A practical guide to building OWL ontologies using the Protégé-OWL plugin and CO-ODE tools*. 1<sup>st</sup> ed. University of Manchester, Manchester, UK.

ISO, 2011. [Online] Retrieved from <http://www.iso.ch/>

ISO 10303-224, 2006. *Industrial automation systems and integration – Product data representation and exchange – Part 224: Application Protocol: Mechanical product definition for process planning using machining features*. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 10303-238, 2007. *Industrial automation systems and integration – Product data representation and exchange – Part 238: Application Protocol: Application interpreted model for computerized numerical controllers*. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 10303-239, 2005. *Industrial automation systems and integration – Product data representation and exchange – Part 239: Application Protocol: Product life cycle support*. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 10303-49, 1998. *Industrial automation systems and integration – Product data representation and exchange – Part 49: Integrated generic resources: Process structure and properties*. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 13399, 2006. *Cutting tool data representation and exchange*. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 13584, 2001. *Industrial automation systems and integration – Parts library*. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 15531-1, 2004. Industrial automation systems and integration – Industrial manufacturing management data – Part 1: General overview. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 15531-44, 2010. Industrial automation systems and integration – Industrial manufacturing management data – Part 44: Information modelling for shop floor data acquisition. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 15704, 2000. Industrial automation systems – Requirements for enterprise-reference architectures and methodologies. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 18629-1, 2004. Industrial Automation Systems and Integration – Process Specification Language – Part 1: Overview and basic principles. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO 9000, 2005. Quality management systems. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO/IEC 24707, 2007. Information technology – Common Logic (CL): a framework for a family of logic-based languages. Geneva, Switzerland: International Organization for Standardization (ISO).

ISO TC 184/SC4, 2011. [Online] Retrieved from <http://ng.tc184-sc4.org/>

Knowledge Based Systems Inc., 1994. Information Integration for Concurrent Engineering (IICE): IDEF5 method report. Texas, USA. [Online] Retrieved from: <http://www.idef.com/pdf/Id5.pdf>

Knowledge Frame Language (KFL) Reference, 2010. Document supplied with the installation of Highfleet Integrated Ontology Development Environment (IODE).

Leal, D., Price, D., Barnard Feeney, A. and Bock, C., 2009. Future SC4 architecture PWI overview and plan. Approved resolution by ISO TC184 SC4. In: 57<sup>th</sup> ISO TC 184 SC4 Plenary Meeting. Parksville, Canada.

Michel, J.J., 2005. Terminology extracted from some manufacturing and modelling related standards. CEN/TC 310 N1119R2.

Motik, B., Patel-Schneider, P.F., Horrocks, I., 2006. OWL 1.1 Web Ontology Language: Structural specification and functional-style syntax. W3C Member Submission 19 December 2006. [Online] Retrieved from [http://www.w3.org/Submission/owl11-owl\\_specification/](http://www.w3.org/Submission/owl11-owl_specification/)

Motik, B., Horrocks, I. and Sattler, U., 2007. Adding integrity constraints to OWL. In: 3<sup>rd</sup> international workshop on OWL Experiences and Directions (OWLED 2007). June 6-7, Innsbruck, Austria.

Orsi, G. and Tanca, L., 2010. Introduction to the TPLP special issue, logic programming in databases: From DATALOG to semantic-web rules. TLP 10(3). Cambridge University Press, Cambridge, UK.

Protégé Ontology Editor and Knowledge Acquisition System, 2010. [Online] Retrieved from <http://protege.stanford.edu/>

Sirin, E., Smith, M. and Wallace, E., 2008. Opening, closing worlds – on integrity constraints. In: 5<sup>th</sup> international workshop on OWL Experiences and Directions (OWLED 2008). October 26-27, Karlsruhe, Germany.

S-TEN Project: Intelligent Self-describing Technical and Environmental Networks SemanticSTEP. [Online] Retrieved from <http://www.s-ten.net/>

Tao, J., 2010. Adding integrity constraints to the semantic web for instance data evaluation. In: 9<sup>th</sup> International Semantic Web Conference (ISWC 2010). November 7-11, Shanghai, China.

Uslar, M., 2008. Ontology based integration of IEC TC 57 standards. In: STASIS/BREIN workshop held during the 4<sup>th</sup> international conference on Interoperability for Enterprise Software and Applications (I-ESA '08). March 25, Berlin, Germany.

Usman, Z., Young, R.I.M., Case, K. and Harding, J., 2010. A manufacturing foundation ontology for product lifecycle interoperability. In: Popplewell, K., Harding, J., Poler, R. and Chalmers, R., eds. Enterprise interoperability IV – Making the internet of the future for the future of enterprise. pp. 147-155. Springer-Verlag London Ltd., London, UK.

Wang, F., Fennes, S.J., Sudarsan, R. and Sriram, R., 2003. Towards modelling the evolution of product families. In: Proceedings of the ASME Computers and Information in Engineering Conference. September 2-3, Chicago, Illinois, USA.

Web Ontology Language (OWL), 2009. OWL Web Ontology Language overview. [Online] Retrieved from: <http://www.w3.org/TR/owl-features/>

Web Ontology Language Description Logics (OWL DL), 2009. OWL Web Ontology Language guide. [Online] Retrieved from <http://www.w3.org/TR/owl-guide/>