Reference Ontologies for Interoperability across Multiple Assembly Systems

Muhammad Imran* and R. I. M. Young

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

Email: M.Imran2@lboro.ac.uk

Reference Ontologies for Interoperability across Multiple Assembly Systems

The role of information and communication technologies (ICTs) is crucial for future manufacturing organizations in order to support effective collaboration and information sharing. However the contemporary ICT based systems lack the required ability to adequately support interoperability across multiple domain systems. The capability of such ICT based systems to interoperate is impeded by the semantic conflicts arising from loosely defined meanings and intents of the participating system concepts. The aim of this paper is to investigate the interoperability of assembly systems at multiple levels of concept specializations using the concept of a formal reference ontology. Formal ontologies are providing a promising way to computationally capture the domain meanings which can subsequently provide a base to support interoperability across multiple systems and in our case multiple assembly systems. This paper takes the example of Manufacturing Bill of Materials (MBOM) concept and three different domain specific interpretations to explore and demonstrate the potential of formal reference ontologies to support interoperability.

Keywords: Information and Communication Technologies (ICTs); assembly systems; interoperability; formal ontologies and Manufacturing Bill of Materials (MBOM)

1. Introduction

The competitiveness of manufacturing organizations depends partly upon their capability to support effective information sharing. Current ICT systems lack the ability to seamlessly exchange information across multiple systems due to loosely defined meanings and intents of the content of the information to be shared. This poses a serious challenge for the current ICT based systems to support effective information sharing. The solution to this challenge lies in addressing the interoperability issues (Ouksel and Sheth, 1999).

The term interoperability is derived from the term "interoperable" where the latter is defined in Oxford dictionary which states that computer systems are interoperable with each other if they are "*able to exchange information and* *make use of information*". This suggests that interoperability is the ability of computer systems to exchange as well as understand the information. There are various other definitions of interoperability found in the literature. For example, Woodley (2005) defines interoperability as "*The ability of different types of computers, networks, operating systems, and applications to work together effectively without prior communication, in order to exchange information in a useful and meaningful manner*". A more relevant definition for this work is provided by Chen and Vernadat (2004) who define interoperability as "*the ability of two or more systems or components to exchange and use shared information*". A similar definition is also given in IEEE standard computer dictionary (1991) where interoperability has been described as "*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*".

However, contemporary ICT based systems lack interoperability and the most common reason for that is the incompatible information structures of participating systems that require interoperation (Brunnermeier and Martin, 2002; Cutting-Decelle *et al*, 2002; Das *et al*, 2007). The incompatibility of information structures is caused by syntactic and semantic incompatibilities of the information to be shared (Das et al., 2007). Syntactic incompatibilities are instigated due to the software systems using different information representation structures whereas semantic incompatibilities are caused due to the lack of clearly defined semantics of the information to be shared (Chen, 2006).

In the literature researchers have mainly emphasized the resolution of semantic interoperability issues (Chungoora, 2010; Chen and Vernadat, 2004; Chungoora et al. 2012) because sufficient efforts have already been made to resolve the syntactic interoperability issues (Rezaei et al., 2014a). Semantic interoperability issues potentially arise either due to the common terms having different meanings or different terms having the same meanings (Ray and Jones, 2003). For example, the term Manufacturing Bill of Materials (MBOM) may have different meanings for different manufacturing systems, which in turn can cause semantic interoperability issues across these systems. Ontologies are playing a vital role to resolve the semantic interoperability issues (Plastiraset al., 2014; Beydoun et al., 2014; Ahmed and Han, 2015).

Traditional approaches to achieve interoperability in Product Lifecycle Management (PLM) systems have been focused on establishing a common schema or product master model which imposes a rigid structure (Hoffman and Joan-Arinyo, 1998). However, this interoperability method becomes problematic when multiple viewpoints of design and/or manufacturing information exist (Raine et al., 2001; Kugathasan and McMahon, 2001) or when a set of domain specific terms are used by engineers (Chungoora, 2010) working across multiple PLM systems.

Standard based interoperability approaches use standards to promote interoperability. Two examples of standards related to this work have been reported in (Panetto and Molina, 2008; Panetto et al., 2012; Tursi et al., 2009; Chungoora et al., 2012) which are: ISO 10303 (also known as STandard for the Exchange of Product Model Data (STEP)) (ISO/TS 10303, 2004) and IEC 62264 (IEC 62264, 2002). STEP provides standard neutral representation of product data in computer understandable form (SCRA, 2006) and largely deals with information such as product specifications, BOM and, other similar manufacturing and assembly related information (Panetto et al., 2012). IEC 62264 provides a reference model between business and manufacturing control applications (Tursi et al., 2009). In a broader context, these standards can support information exchange between various applications such as Enterprise Resource Planning (ERP), Computer Aided Design (CAD), Product Data Management (PDM) and Manufacturing Execution System (MES) (Panetto et al., 2012).

Despite these standardization efforts to support interoperability, researchers have found potential issues in standard based interoperability approaches. For instance, one of the potential barriers to the development of standard based interoperable systems is the resistance from software/hardware vendors who exploit the opportunity of lack of standards (Newman et al., 2008). This argument is further supported by Young et al. (2009) who claim that implementation of such standards requires consensus from users to commit one standard way of information representation which has not been successful over the years due to the lack of flexibility. However, even if the communities agree on a specific standard, interoperability issues will remain because of the different understanding of the meanings of the terms involved in that standard (Ray and Jones, 2006). The underlying reason is the semantic conflicts that exist because of the lack of rigorous definitions of the domain concepts (Young et al., 2007).

In the 2000s, the European Commission (EC) established an expert group to initiate projects in order to deal with the emerging interoperability issues in enterprise software applications (Chen and Doumeingts, 2003). The expert group identified (1) Enterprise Modelling (2) Architecture and Platform and (3) Ontologies as the three major research themes to be addressed (Chen and Doumeingts, 2003). Enterprise modelling specifies interoperability requirements, architecture and platforms provide implementation solutions, and ontologies provide semantics for interoperability (Panetto et al., 2004; The-ATHENA-Consortium, 2004-2006).

The IDEAS (Thematic network Interoperability Development of Enterprise Applications and Software) interoperability framework, which was built on the above three research themes/domains, was the first initiative in Europe under the Fifth Framework Programme (FP5) to address the enterprise and manufacturing interoperability concerns (Chen et al., 2008). The IDEAS interoperability framework aimed at achieving interoperability between two enterprises on different levels such as business, knowledge and ICT levels (Berre et al., 2007). The interoperability at business level is considered as the operational and organizational ability of an enterprise to collaborate with other/external organizations, the interoperability at knowledge level can be achieved if competencies, skills and knowledge assets of an enterprise are compatible with other/external organizations, and the interoperability at ICT level can be achieved when an enterprise's ICT systems are capable of cooperating with those of other/external organization (The-ATHENA-Consortium, 2004-2006). However, it has been reported in (Rezaei et al., 2014b) that the IDEAS framework lacks the ability to address interoperability on advanced levels because it is more focussed on other research areas than interoperability.

Under the Sixth Framework Programme (FP6), two main initiatives relating to interoperability were taken in the form of ATHENA (Advanced

Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications) and INTEROP (Interoperability Research for Networked Enterprises Applications and Software) (Chen et al., 2008). Both ATHENA and INTEROP frameworks emphasize the need to integrate/merge three research themes/domains: enterprise modelling, architecture and platforms, and ontologies to support the development of enterprise applications interoperability (The-ATHENA-Consortium, 2004-2006; Panetto et al., 2004; Kosanke and Zelm, 2005).

Research efforts have been initiated to build Model Driven Interoperability (MDI) architectures within the framework of INTEROP NoE (Chen et al., 2008). MDI is based on the Model Driven Architecture (MDA) which is a framework introduced by the Object Management Group (OMG) (http://www.omg.org/mda/). MDA supports the creation of highly abstract, machine readable models (Kleppe et al., 2003) which can then be transformed into domain specific models (MDA-Guide-Document, 2003; Sanya and Shehab, 2014) to support interoperability. Typically MDA is comprised of (1) Computation Independent Model (CIM) (2) Platform Independent Model (PIM) and (3) Platform Specific Model (PSM) (MDA Guide, 2003). The CIM illustrates system requirements by specifying computer independent models and the PIM defines system's functionality without considering a specific platform (Marcos et al., 2006). A PIM can be transformed into a PSM by adopting a suitable platform (Panetto, 2007; Marcos et al., 2006). In MDA, multiple model transformations can occur within and between the CIM, PIM and PSM abstraction levels (Ou et al., 2006; Cranefield and Pan, 2007). However, Panetto (2007) reports that interoperability issues take place while exchanging models within the same abstraction levels as well as between different abstraction levels. These interoperability issues are caused by the lack of unambiguous specification of domain concepts (Young et al., 2007).

The MDA based approach can be effectively applied to support interoperability with the help of ontologies (Roser and Bauer, 2006). This has been demonstrated in (Chungoora et al., 2013a) where they have successfully applied the MDA approach combined with the ontological engineering approach to support interoperability across product design and manufacture. Potential reasons behind the success of ontologies are their pivotal role in mapping concepts across multiple systems and their ability to resolve semantic conflicts (Vernadat, 2007). In particular, semantic interoperability can be effectively achieved when the meanings of the information to be shared are well understood across these systems (Wache et al., 2001). Ontology based interoperability is an emerging research area (Vernadat, 2007) in general and more specifically, recent efforts from Chungoora (2010), Chungoora et al. (2012), Chungoora et al. (2013a), Usman et al. (2013), Imran and Young (2013) and Bruno et al. (2015) have been focussed on formal ontology based approaches to support interoperability in PLM systems.

The research reported in this paper explores assembly related concepts and is focused on resolving the interoperability issues across multiple assembly systems. This research takes the view that a common knowledge base can be built using formalized assembly reference concepts which are common across multiple assembly systems and thus can provide a route for interoperability across these systems. This work employs the Common Logic (CL) based Knowledge Frame Language (KFL) from Highfleet Inc. to formally define the concepts used in this research. The CL based ontological formalism has more powerful expressive and reasoning capabilities as compared to the Web Ontology Language (OWL).

For example, in contrast to OWL, KFL is based on the Closed World Assumption (CWA) (Chungoora et al., 2013b) whereas the CWA assumes that everything stated or implied is true and everything else is false (Date, 2007). This is potentially required in complex domains such as assembly which are fact driven and require certainty; therefore a CL based approach with CWA best suits for this domain. Other potential advantages of CL based approach (in contrast to OWL) is that it supports ternary and/or higher order relations, binary and/or higher order functions, conjunction, disjunction, and the negation operators (Palmer et al., 2012) which are required for modelling complex domains such as assembly. Research conducted by Chungoora et al. (2013b) stipulates that CL has proved itself more competent than OWL in rigorously defining the semantics which is a key requirement for heavyweight modelling. Therefore CL based approach should be well suited to formalize the assembly domain.

2. Reference Ontologies

Domain or application ontologies comprise of formally defined concepts and relationships intended to represent an application area (Musen, 1998;Jean et al., 2006), and are hardly used outside the particular research environment they are designed for (Navigli and Velardi, 2004). In contrast to domain ontologies, foundation or upper ontologies (FinES-Cluster, 2011) consist of generic, abstract and high level concepts which can be applied to a wide range of domains and provide formally defined concepts to support more specialized ontologies (Sanchez-Alonso and Garcia-Barriocanal, 2006).

We argue that there is a need of an ontology which sits between the very specific domain and the very general foundation ontologies. This type of ontology is called a reference ontology. The terms core ontology or core concept ontology have also been used for this type of ontology (Gangemi and Borgo 2004; Usman et al. 2011) however, in this research the term reference ontology is used to describe such ontology. The term reference ontology was first introduced by Nicola Guarino who described it as the clarification of "*the meanings of terms used in a specific domain*" (Grenon, 2003). Burgun (2006) describes reference ontologies as a way of representing domain knowledge without focussing on specific objectives. Leila (2009) summarises the definitions of reference ontology and described it as an ontology which represents a domain adequately and is validated by majority of the domain experts. He further argues that reference ontologies tend to be broad, satisfy needs of large community of domain, support shared meanings, use axioms, and can be derived from the foundation ontology. Thus a reference ontology can be described as an ontology which adequately and formally represents domain concepts without focussing on specific domain objectives. A reference ontology comprises formally defined reference concepts which can be reused, extended or specialized for multiple applications and consequently provides a base to support interoperability across them. Reference ontologies are a comparatively new development (Brinkley et al., 2006) and are emerging as potential candidates to support interoperability across multiple domains.

Recently a few reference ontologies have been developed in the field of medicine (Burgun, 2006) however they do not have wide spread applications in other domains. Some work has been published in the manufacturing domain by Chungoora et al. (2012) and Usman et al. (2013) where they have exploited reference ontologies to support interoperability in the manufacturing domain. Their main focus was on single piece part machining and they identified the need for reference ontologies in the assembly domain.

Imran and Young (2013) investigated the role of reference ontologies for assembly where multiple perspectives of assembly feature were explored to support knowledge sharing across assembly design and assembly process planning. The work reported in this paper exploits the same approach but applied to interoperability across multiple assembly process planning systems. The reference ontology has been formally defined in CL based KFL. CL is a logic framework aimed for sharing and transmission of information (ISO/IEC-24707, 2007) and is based on first order logic which is a foundation for knowledge representation (Nemuraite et al., 2009). The Integrated Ontology Development Environment (IODE) has been used to test and evaluate the formalized ontology. IODE is an ontology development tool developed by the HighFleet which provides a platform to build knowledge bases, to assert the instances, to delete the assertions, to browse the ontology and to allow queries to be made using the query tool (IODE, 2013).

The value of formal reference ontologies is that they provide computer interpretable semantics of concepts. Therefore if two or more systems exploit the same reference ontology they can share the same semantics and therefore the same understanding. In reality what this means as that systems developers have a decision to make as to how interoperable they want their system to be. If they have concepts that they wish to develop that are not based on the reference ontology then they will clearly have no basis from which they can knowingly be shared with other systems. Likewise if system developers wish to specialise concepts such that they are only partly consistent with the reference ontology then only that part that remains consistent will be knowingly sharable with other systems.

3. The Assembly Reference Ontology

The assembly reference ontology (ARO) is proposed to represent the assembly knowledge and to support interoperability across assembly application specific systems. The ARO is specialized from a foundation ontology provided in our case by the Highfleet software systems. The ARO comprises of a set of reference concepts that sit between foundation and domain specific concepts and are specialized from the most generic level to the most specialized level as shown in figure 1. The specialization levels defined in this research are: generic reference concepts, product lifecycle reference concepts, design and manufacturing reference concepts and assembly specific reference concepts.

The higher level concepts are needed for assembly but are recognised as having applicability across other application areas. For example the product lifecycle reference concepts are applicable to any product lifecycle aspect.

Conceptually assembly is significantly different from the single piece part manufacturing as the former deals with multiple parts rather than a single part. Hence the assembly domain requires additional concepts which can represent the knowledge and can provide reference concepts to support interoperability across the assembly design and assembly process planning application specific systems. A detail investigation of how these concepts can be exploited to represent and interoperate across application specific systems is provided in section 4. It is important to understand that the reference concepts shown in figure 1 do not rigidly follow the level by level specialization. For instance, the assembly specific concepts EBOM and MBOM are specialized from the product lifecycle reference concept BOM bypassing the design and manufacturing reference level.

This article explores BOM concepts in detail to investigate the interoperability across assembly systems and to demonstrate the success of the approach.

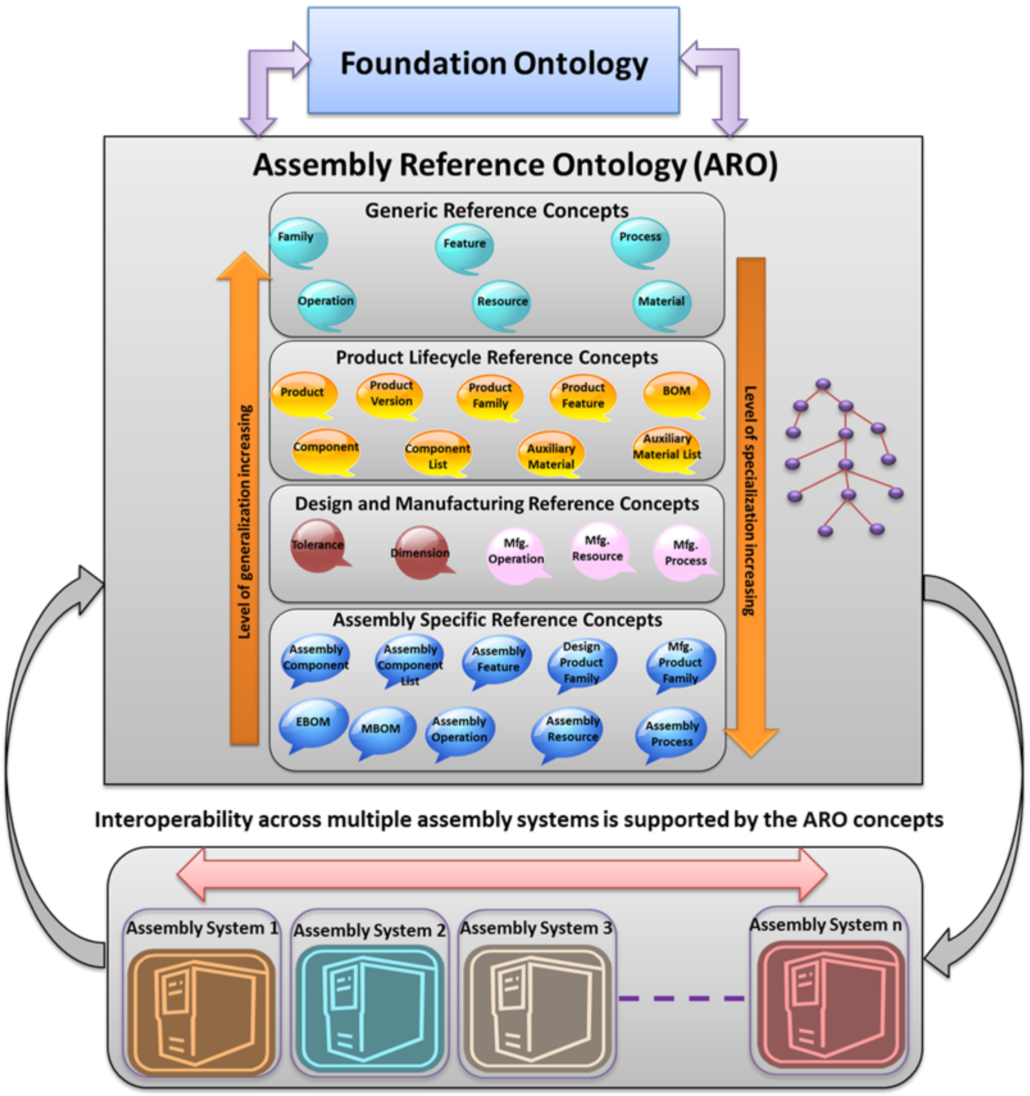


Figure 1. Assembly Reference Ontology (ARO) to support interoperability across multiple assembly systems

4. Exploration of BOM Concepts for Assembly Systems Interoperability

4.1 Bill of Materials (BOM)

Bill of Materials (BOM) is a core component of product lifecycle information management (Zhang, et al., 2010) and is a key concept for the assembly domain. Generally BOM is described as a list of components and raw materials (Chang et al., 1997). BOM is found in different forms and have multiple viewpoints (Chang, et al., 1997; Jiao, et al., 2000). Although there exists various types of BOM in the literature however Engineering Bill of Materials (EBOM) and Manufacturing Bill of Materials (MBOM) are the two most important categories (Vollmann, 1997; Zhang, et al., 2010). EBOM comprises of list of items as described in assembly drawing (Xu, et al., 2008; Tursi, et al., 2009) and is constructed on the basis of product design taking into account the functions of its components (Jiao, et al., 2000; Chang, et al., 1997). However EBOM does not consider the manufacturing aspects hence it should not be used directly in the assembly planning (Lee, et al., 2011; Tursi et al., 2009).

MBOM takes into account the assembly (process planning) aspects and is arranged according to the assembly plan of the product (Tursi et al., 2009). MBOM comprises of list of all the materials along with their quantities required for a product to manufacture (Jones, et al., 2001) and is a different organization of EBOM which can be adapted for manufacturing purpose. As far as the structure of MBOM is concerned, it represents the hierarchical assembly groups based on the way they are assembled on shop floor (Chang, et al., 1997).

This paper explores the MBOM concept and three different specific domain interpretations to show that the ARO supports interoperability across these multiple heterogeneous assembly systems. Therefore in the following sections, MBOM concepts and their formalization process has been described in detail.

4.2 Definition of MBOM and Related Concepts

In this research an MBOM is defined as a list of assembly components. However its subsumptions may have other items as well e.g. auxiliary materials. In this section, first the key ARO concepts which help to define MBOM concepts have been described. Then three different application specific interpretations of MBOM have been explained.

4.2.1 Definition of Key ARO Concepts

4.2.1.1 Assembly Component

The concept assembly component represents those items which are directly used to build a product. It has been specialized from the concept component which is widely found in the literature and has different meanings and interpretations. The majority of the sources explored, describe component as either a single piece part or a subassembly. For example, standard ISO/TC 10303-224 (2003) defines it as: "The component specifies either a Single_piece_part or another Manufactured_assembly used to define an assembly". In the same way Molloy et al. (1998) and Lohse (2006) describe component as either a single piece part or a subassembly used for building a product. Similarly, Siemens NX 7.5 assembly modeller and Teamcenter 8 also identify component as a single piece part or a subassembly. However, Boothroyd's DFA 9.4 software system does not use the term component, rather it uses the terms part and subassembly to build assemblies.

In this research we take the view that a component represents both single piece part and subassemblies and the concept assembly component has been used to represent the assembly related information. Some examples of assembly components are nuts, bolts and base parts as shown in figure 2.

The concept assembly component has been further specialized into As Required (AR) assembly components and As Designed (AD) assembly components. The AR assembly components represent small and standard components which are described as AR items on the assembly drawing. It implies that they are not purchased through the Material Requirement Planning (MRP) process instead they are acquired as bulk. The AD assembly components are those assembly components which are not AR assembly components and are purchased through the MRP process. The examples of AR assembly components are small and standard size components such as nuts and bolts and those of AD assembly components are the large and/or non-standard components such as base parts of assembled products as shown in figure 2.

4.2.1.2 Auxiliary Material

Auxiliary materials are the materials which are indirectly used for the production of a part or an assembled product (Frohlich, 2004). For instance, machine oil, paint, and tape used in the assembly of a product are examples of auxiliary materials as shown in figure 2. In this research, the concept auxiliary material has been introduced to represent the indirect materials used during the assembly of a product. This concept has been further explored to capture the MBOM semantics.

4.2.1.3 Assembly Component List and Auxiliary Materials List

The concepts assembly component list and auxiliary material list represent the list of assembly components and auxiliary materials. The assembly component list has been specialized into two further concepts which are: AR assembly component list and AD assembly component list. The AR assembly component list represents the list of AR assembly components whereas AD assembly component list represents the list of AD assembly components. The list related concepts have been further explained in the formalization section.

Concept Description	Concept Name	Concept Examples
Items directly used in assembly of product.	Assembly Component	Examples of assembly components are nuts, bolts, base parts and other such components which are directly used to build a product.
Items described as A/R (as required) on assembly drawing.	AR Assembly Component	Examples of AR assembly components are the standard components purchased as bulk such as nuts and bolts.
Items other than A/R items.	AD Assembly Component	Examples of AD assembly components are all other assembled components which are not AR assembly components e.g base parts and other such components .
Items indirectly used in assembly of product.	Auxiliary Material	Examples of auxiliary materials are all other items indirectly consumed during the assembly of products such as oil, tape and paint etc.

Figure 2. Description of concepts used in the definitions of MBOM concepts

4.2.2 *Definition of Application Specific MBOM Concepts*

Three different interpretations of MBOM have been analyzed to investigate the interoperability in multiple assembly systems. Two of them are extracted from the existing literature. The third one is based on author's understanding of the MBOM concept. These three MBOM interpretations are discussed in the following sections.

4.2.2.1 *MBOMs*

The first informal definition of MBOM is based on Stark (2011)'s interpretation of MBOM. In this research we describe this interpretation as MBOMs. Stark (2011) describes MBOMs as a list of items in EBOM and other things needed to make a product e.g. machine oil. The additional bit from the assembly point of view are the things which facilitate the making of product assembly and these may include, for example, machine oil for lubrication, paint and tape to mark the floor. The items directly used in the assembly of a product have been described as assembly component and the indirect items as auxiliary materials.

Figure 3a shows the UML based representation of MBOMs where a constraint has also been attached. The constraints states that MBOMs should have auxiliary material list. The other concept linked with MBOMs is the assembly component list as shown in figure 3a.

4.2.2.2MBOMh

The second MBOM (informal) definition is based on the interpretation from Hirata (2009)'s work. Hirata (2009) describes that generally MBOM does not include items such as paint, tape and some small items like bolts, nuts whereas these small items are described as AR items on engineering drawings. Although Hirata (2009) concludes that these items should be part of MBOM however his general description of MBOM excludes these items from MBOM. The interpretation of MBOM based on Hirata (2009) has been termed as MBOMh.

It can be deduced from the above information that MBOMh does not have AR assembly components and auxiliary materials. Hence it can be defined as a list of AD assembly components only. Two constraints have been attached toMBOMh which state that MBOMh should not have the AR assembly component list and the auxiliary materials list as shown in figure 3b. It is interesting to note that MBOMh is considerably different from that of MBOMs.

4.2.2.3 MBOMi

The informal definition of MBOMi is based on our understanding of the MBOM concepts gained from the existing literature (Vollmann, 1997; Zhang, et al., 2010; Xu, et al., 2008; Tursi, et al., 2009; Jiao, et al., 2000; Chang, et al., 1997; Lee, et al., 2011; Jones, et al., 2001). The MBOMi is described as a list of items which are directly used in building a product assembly and does not include the indirect items such as oil, paint and tape. More appropriately, MBOMi is described as list of assembly components without auxiliary materials. This is shown in figure 3c where MBOMi has been linked to the concept assembly component list.

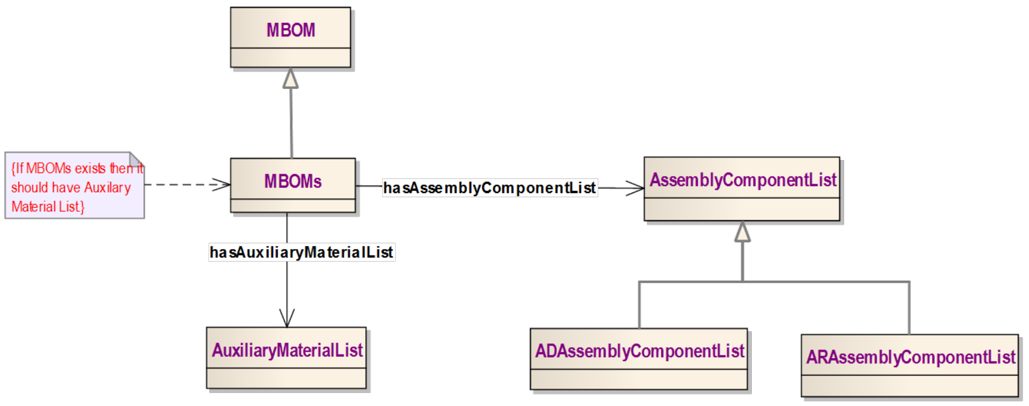


Figure 3a: UML based lightweight representation of MBOMs

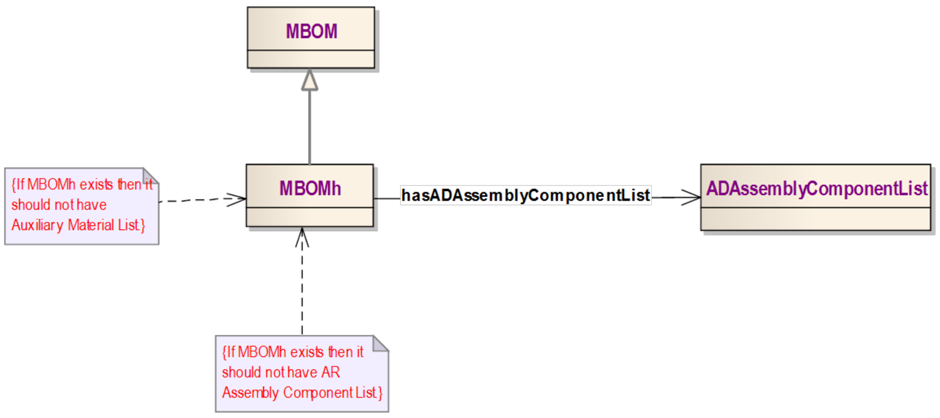

MBOM
{If MBOMh exists then it should not have Auxiliary Material List}
MBOMh
hasADAssemblyComponentList
ADAssemblyComponentList
{If MBOMh exists then it should not have AR Assembly Component List}

Figure 3b: UML based lightweight representation of MBOMh

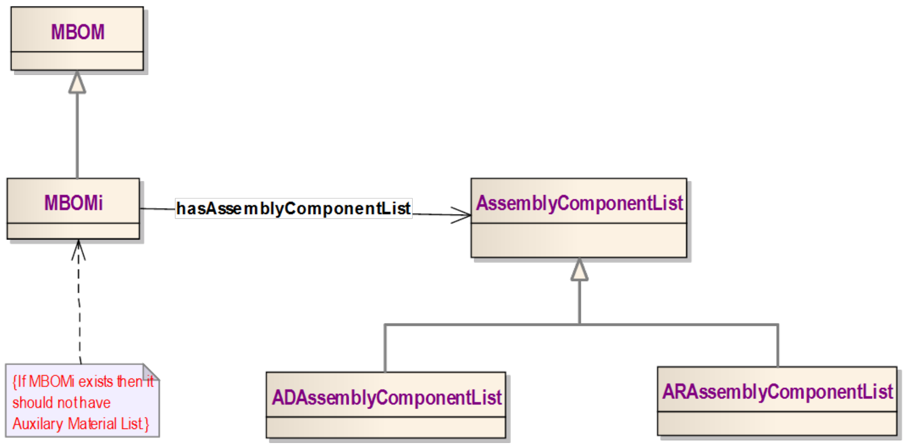


Figure 3c: UML based lightweight representation of MBOMi

Figure 3. UML based lightweight representation of three MBOM concepts.

4.2.3 Combined Representation of MBOM Concepts

Figure 4 shows a combined representation of foundation concepts, ARO concepts and the application specific concepts. The concepts enclosed in the box at the top of figure 4 represents the foundation ontology concepts. The middle box in figure 4 shows ARO concepts which have been specialized from the foundation concepts. These concepts act as reference concepts to application specific systems and support interoperability across these systems. For example, in figure 4 MBOMs, MBOMh and MBOMi are application specific concepts which have been specialized from the ARO concept MBOM. These application specific concepts are also linked with other ARO concepts such as assembly component list and auxiliary material list. The ARO concepts help to formally define the application specific concepts. This leads to the fact that the ARO can be used as reference ontology for multiple heterogeneous assembly systems. The formal definitions of MBOM concepts are explained in the next section.

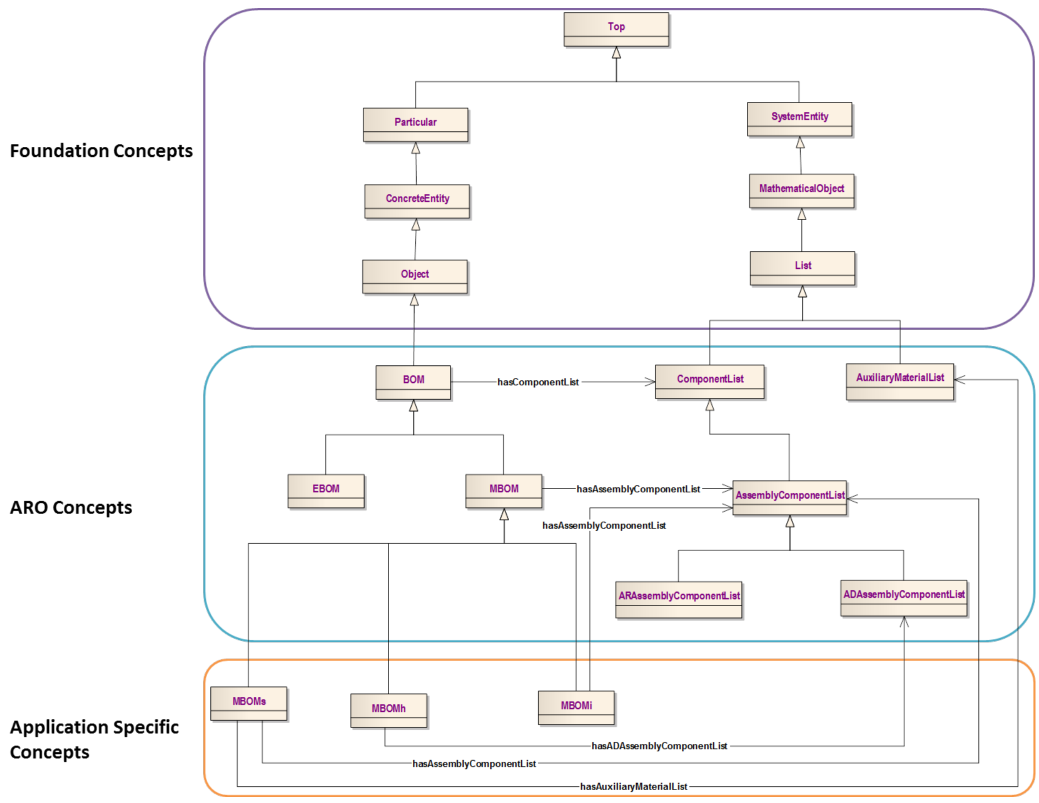


Figure 4.Combined representation of MBOM concepts.

4.3 Formalization of MBOM and Related Concepts

The formalization process comprises of declaration of properties (classes), relationships and axioms. The properties in the KFL are declared as follows.

```
:Prop MBOM
```

```
:Inst Type

:supBOM

:name "Manufacturing Bill of Materials"

:rem "MBOM has assembly component list."
```

The :prop directive introduces the property (MBOM in the above case), :Inst directive states the kind of instantiation and :sup directive states the super property of the declared property. The name and remark directives are optional and can be added to facilitate the modeller.

Similarly the application specific interpretations of MBOM can be declared using the same KFL format. For example, the MBOMs property has been declared in KFL as follows.

```
:Prop MBOMs

:Inst Type

:sup MBOM
```

The other two MBOM concepts: MBOMh and MBOMi have also been declared using the similar format. The properties: assembly component list and auxiliary materials list represent the list of assembly components and auxiliary materials. For example, the property assembly component list has been declared in KFL as follows:

```
:PropAssemblyComponentList

:Inst Type

:supComponentList
```

Similarly the other list related properties: AD assembly component list, AR assembly component list and auxiliary material list have been declared in KFL.

Most of the relations associated with MBOM concepts are instances of binary relation. For example, MBOMs has the “hasAuxiliaryMaterialList” relation with auxiliary material list which is actually a binary relation. The relation can be declared as follows:

```
      :RelhasAuxiliaryMaterialList
      :InstBinaryRel
      :Sig BOM AuxiliaryMaterialList
```

However the relation between all the subsumption of the property "list" has variable arity. By variable arity means that a relation declared in KFL can take any number of arguments e.g. 10, 20 unlike the fixed arity relations such as binary or ternary relations in which only two and three arguments can be taken respectively. This is typically required in case of asserting variable number of assembly components or auxiliary materials. For example, an assembly component list may have any number of assembly components and the assembly components may vary depending upon the product for which the MBOM has been created. The only variable arity relation in KFL is the relation "item" which relates the list subsumptions with the concepts such as assembly component and auxiliary material.

The properties and relations do not fully define the semantics of concepts therefore axioms are applied to constrain and capture the semantics of concepts. For example, the following two axioms have been applied to capture and constrain the semantics of the concept "assembly component list":

```
(=> (AssemblyComponentList ?l)
      (exists (?c)
            (and (AssemblyComponent ?c)
                  (item ?l ?c))))
:IC hard "Every assembly component list consists of at least one
assembly component within the list."


(=> (AssemblyComponentList ?l)
       (not (exists (?other)
                 (and (item ?l ?other)
                  (not (AssemblyComponent ?other))))))
      :IC hard "Every assembly component list should consist
      of exclusively assembly components that make up the
      list."
```

The first axiom dictates that an instance of assembly component list should have at least one instance of assembly component. It implies that the system would not accept any list which does not have at least one assembly component. However there are still possible chances that the system accepts a list which has one or more instances of assembly component as well as instances of other concepts such as auxiliary material. The second axiom averts such attempts and ensures the system accept instances of assembly components only for the property “assembly component list”.
The concepts “AR assembly component list” and “AD assembly component list” are mutually exclusive meaning that no single instance of AR assembly component can be found in AD assembly component list and no single instance of AD assembly component list should be found in AR assembly component list. This can be captured by applying the following axioms.

```
(=> (ARAssemblyComponentList ?l)
       (not (exists (?x)
            (and (ADAssemblyComponent ?x)
                        (item ?l ?x)))))
:IC hard "Every AR assembly component list should not consist of
AD assembly components."


(=> (ADAssemblyComponentList ?l)
       (not (exists (?x)
            (and (ARAssemblyComponent ?x)
                  (item ?l ?x)))))
      :IC hard "Every AD assembly component list should not
      consist of AR assembly components."
```

The first axiom (mentioned above) says that AR assembly component list cannot have any AD assembly component whereas the second axiom dictates that AD assembly component list cannot have AR assembly component list. Similarly semantics related to auxiliary material list has been captured using these kinds of axioms.

So far, constraints have been applied to assembly component lists and auxiliary material list. The constraint attached to the MBOM concept is specified as follows.

```
(=> (MBOM ?mbom)
      (exists (?aclist)
      (and (AssemblyComponentList ?aclist)
      (hasAssemblyComponentList ?mbom ?aclist))))
:IC hard "Every MBOM should have assembly component list."
```

The axiom shown above enforces that every MBOM should have the assembly component list. It suggests that whenever there exists an instance of MBOM there should also exist an instance of assembly component list and the instance of MBOM should have that instance of assembly component list.

Similar axioms have been applied to the application specific MBOM concepts: MBOMs, MBOMh and MBOMi. The constraints shown in the UML diagrams in figure 3 are represented in KFL by using the axioms. A summary of such axioms is shown in figure 5.

5. Case Study Investigation

This paper uses the example of a butterfly valve to demonstrate the potential of the ARO to support interoperability in the assembly domain. Figure 6 shows the butterfly valve assembly, its components and some auxiliary items. Based on the definition of assembly component (see section 4.2.1), the valve assembly components, which directly build up the final product, are termed as assembly components as displayed in figure 6. The bolts and nuts shown in figure 6 are standard assembly components which have been purchased in bulk hence they are termed as AR assembly components. The rest of assembly components e.g. body bracket assembly, valve blade, side cover, plate lever, handle ball assembly and top cover are called AD assembly components. The two AD assembly components: body bracket assembly and handle ball assembly are subassemblies which have been joined before they are assembled with other valve components. The components of these subassemblies such as body and bracket of body bracket assembly, and handle and ball of handle ball assembly are also considered as assembly components.

Concept	Axioms applied to constrain the semantics of MBOM concepts	Explanation
MBOMs	`(=> (MBOMs ?mboms) (exists (?amlist) (and (AuxiliaryMaterialList ?amlist) (hasAuxiliaryMaterialList ?mboms ?amlist))))` `:IC hard "Every MBOMs should have auxiliary material list."`	This axiom describes auxiliary materials as necessary part of MBOMs. Examples of auxiliary materials include oil, tape, paint etc.
MBOMh	`(=> (and (MBOMh ?mbh) (AuxiliaryMaterialList ?aml)) (not (hasAuxiliaryMaterialList ?mbh ?aml)))` `:IC hard "MBOMh should not have Auxiliary Material List."`	This axiom specify that auxiliary materials should not be included in the definition of MBOMh.
MBOMh	`(=> (and (MBOMh ?mbh) (ARAssemblyComponentList ?aacl)) (not (hasARAssemblyComponentList ?mbh ?aacl)))` `:IC hard "MBOMh should not have AR Assembly Component List."`	This axiom specify that AR assembly components should not be included in the definition of MBOMh. Examples of AR assembly components are standard components such as nuts and bolts etc.
MBOMi	`(=> (and (MBOMi ?mbi) (AuxiliaryMaterialList ?aml)) (not (hasAuxiliaryMaterialList ?mbi ?aml)))` `:IC hard "MBOMi should not have Auxiliary Material List."`	This axiom specify that auxiliary materials should not be included in the definition of MBOMi.

Figure 5. Example of four different axioms applied to constrain the meanings of MBOMs, MBOMh and MBOMi.

The valve assembly is carried out in a manufacturing facility which requires some auxiliary items to assist the valve assembly. As described in section 4.2.1, these auxiliary items are called auxiliary materials. In figure 6, auxiliary materials: tape, paint and lubrication oil are displayed which have been used to support the valve assembly. The tape and paint have been used for the purpose of marking the shop floor and the lube oil have been used for lubrication of assembly machines such as press machine to assist the handle and ball assembly.

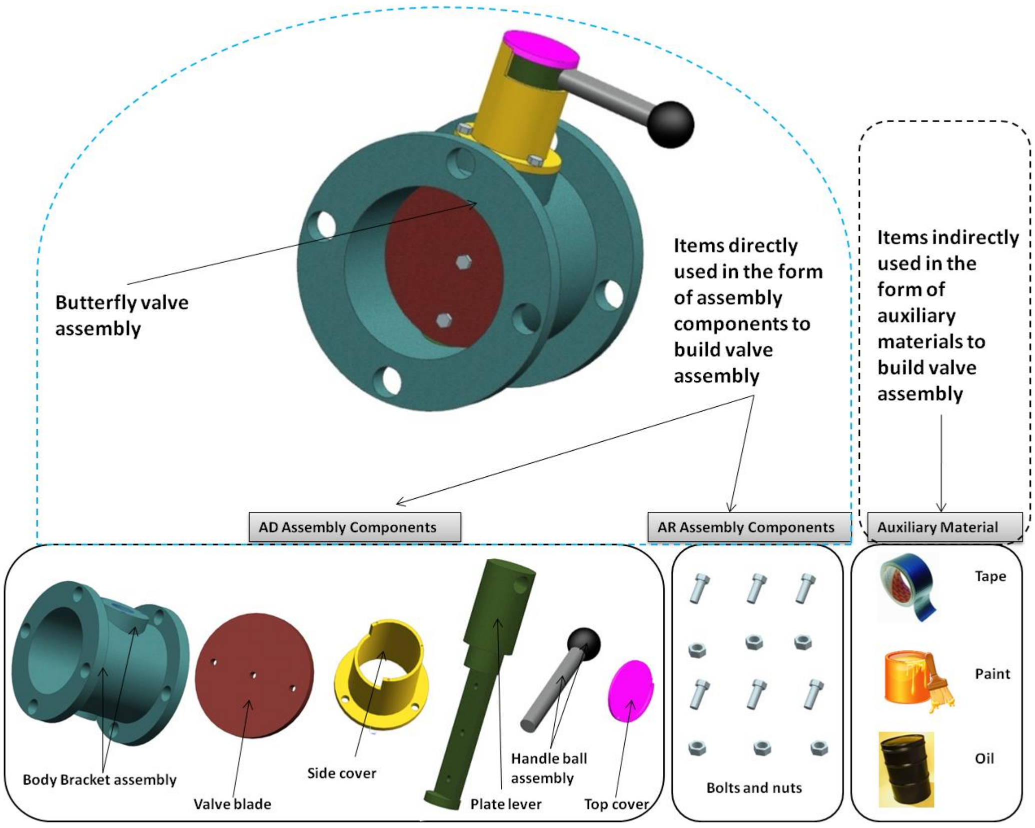


Figure 6. Case study scenario.

With reference to section 4.2.2, MBOMs, MBOMh and MBOMi for valve assembly shown in figure 6 can be easily differentiated. MBOMs for the valve assembly comprises of list of assembly components e.g. bolts, nuts, body bracket assembly, valve blade, side cover, plate lever, handle ball assembly and top cove plus auxiliary materials e.g. tape, paint and oil. MBOMh consists of AD assembly components e.g. body bracket assembly, valve blade, side cover, plate lever, handle ball assembly and top cover. Finally MBOMi comprises of AD assembly components e.g. body bracket assembly, valve blade, side cover, plate lever, handle ball assembly and top cover and/or AR assembly components e.g. bolts and nuts as shown in figure 6.

Based on the valve assembly example, various experiments have been carried out to evaluate the semantics of the concepts formalized in the previous section. These experiments will demonstrate that the semantics defined in the previous section, actually work when they are tested in the experimental tool IODE. For example, the definition of MBOMi says that it should only have assembly components. The verification of its semantics will demonstrate that any attempt to assert auxiliary material e.g. any instance of tape, paint or oil (shown in figure 6) will prompt a warning message and the system will not accept such wrong assertions. This will also be explained later in this section. In summary, the experimental investigation will show that the resulting knowledge base

- Allows the fact assertions when they satisfy the formal definition of the concepts.
- Does not allow the fact assertions when they do not satisfy the formal definition of the concepts.
- Reports the reasons of fact assertions which do not satisfy the formal definition of the concepts.
- Shows that the violation of formal definition is applicable to the related specialized concepts.
- Demonstrates that a route can be established to share information across the participating systems using the common concept/s.

The facts related to the valve assembly (shown in figure 6) have been asserted to evaluate the semantics of MBOM. Table 1 shows a summary of these facts. The ARAssemblyComponent and ADAssemblyComponent are specialized classes of assembly components as explained in section 4.2. The auxiliary materials are the materials indirectly used to support the assembly of components. The AR assembly component list, AD assembly component list and auxiliary materials list have been instantiated with the help of "listof" function as shown in the table 1.
Because the facts displayed in table 1 do not violate any constraint (please refer to section 4.2 for respective constraints) therefore these facts have been successfully asserted in the database. Once these facts have been asserted they can be used to test the semantics of MBOM and its specialized classes e.g. MBOMs, MBOMh, and MBOMi.
The constraint attached with MBOM states that it should have an assembly component list. This constraint should also work for the specialized classes of MBOM (e.g. on MBOMs, MBOMh, and MBOMi). For example if an instance of MBOMs is asserted in the database without asserting AD or AR

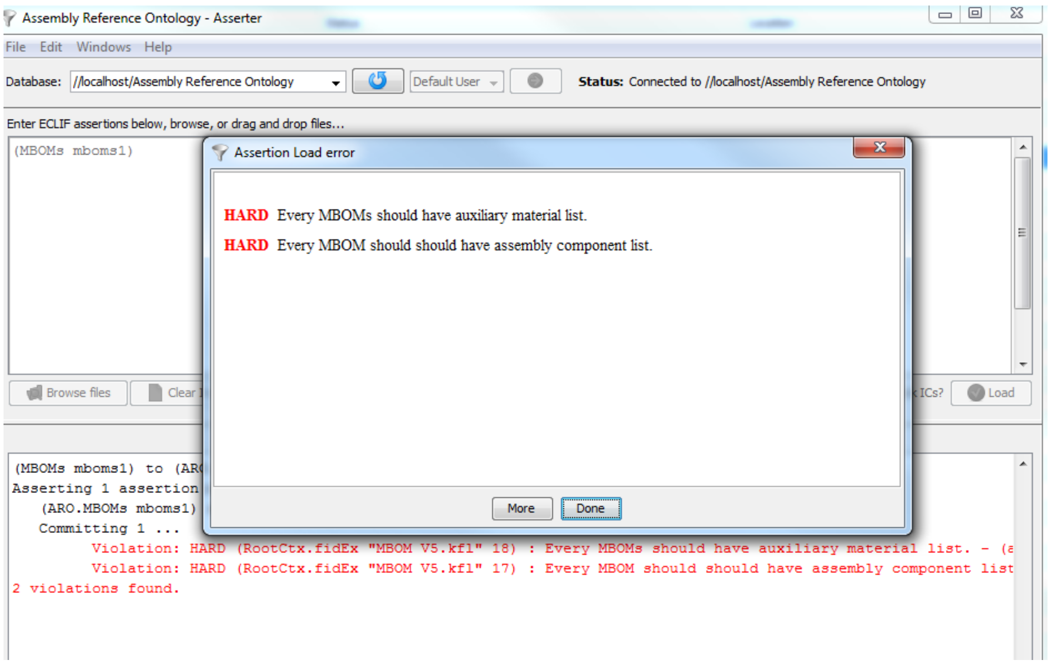
assembly component lists, the system will display an error message reporting that every MBOM should have assembly component list as shown in figure 7.

Table 1.Facts of assembly component, auxiliary materials and their corresponding lists

Classes	Instances
ARAssemblyComponent	Nut01 Nut02 Nut03 Nut04 Nut05 Nut06 Bolt01 Bolt02 Bolt03 Bolt04 Bolt05 Bolt06
ARAssemblyComponentList	(listof Nut01 Nut02 Nut03 Nut04 Nut05 Nut06 Bolt01 Bolt02 Bolt03 Bolt04 Bolt05 Bolt06)
ADAssemblyComponent	Bracket01 MainBody01 BodyBrassy01 Platelever01 Cover01 Handle01 Ball01 HandleBallassy01 Blade01 TopCover01
ADAssemblyComponentList	(listof Bracket01 MainBody01 BodyBrassy01 Platelever01 Handle01 Ball01 HandleBallassy01 Blade01 TopCover01)
AuxiliaryMaterial	PaintABC01 TapeXYZ01 OilShell01
AuxiliaryMaterialList	(listof PaintABC01 TapeXYZ01 OilShell01)

The IC violation caused due to the absence of an assembly component list in figure 7 suggests that any instance of a specialized level of a concept which does not satisfy the formal definition of its parent class will also violate the constraints applied on the parent class. However, as the parent classes are more generic as compared to their child classes, therefore, more constraints can be applied on the child classes which can then be used for specific applications.

Figure 7 also shows the IC violated due to the lack of an auxiliary material list. This is because there is an IC attached to MBOMs (MBOMs is specialized from MBOM) as well which states that whenever an MBOMs exists, it should also have auxiliary material list. Therefore when MBOMs is asserted with the assembly component lists and auxiliary materials lists the system will accept the MBOMs facts assertions.


Assembly Reference Ontology - Asserter
File Edit Windows Help
Database: //localhost/Assembly Reference Ontology
Default User
Status: Connected to //localhost/Assembly Reference Ontology
Enter ECLIF assertions below, browse, or drag and drop files...
Assertion Load error
Browse files
Clear
ICs?
Load
More
Done

```
(MBOMs mboms1)
```

HARD Every MBOMs should have auxiliary material list.
HARD Every MBOM should should have assembly component list.

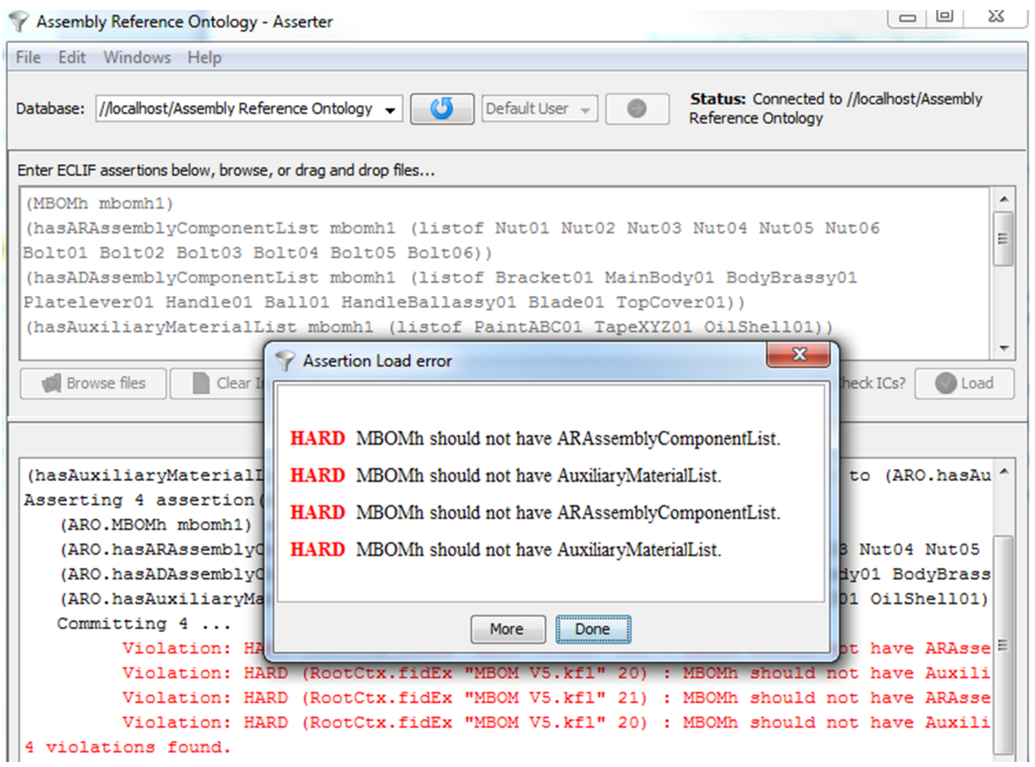
```
(MBOMs mboms1) to (AR
Asserting 1 assertion
   (ARO.MBOMs mboms1)
   Committing 1 ...
         Violation: HARD (RootCtx.fidEx "MBOM V5.kfl" 18) : Every MBOMs should have auxiliary material list. - (a
         Violation: HARD (RootCtx.fidEx "MBOM V5.kfl" 17) : Every MBOM should should have assembly component list
2 violations found.
```

Figure 7. MBOMs fact assertion without assembly component list and auxiliary material list

It is to be noted that MBOMs asserted without an assembly component list violates IC due to its parent class MBOM. Whereas MBOMs asserted without an auxiliary material list is due to the IC applied on the concept itself. However MBOMs can be instantiated successfully when asserted with AR assembly component list or AD assembly component list or both (as both of these concepts are specialized from the assembly component list) and the auxiliary material.

Similarly, whenever instances of MBOMh and MBOMi are asserted without the assembly component list, the system displays an error message suggesting to include the assembly component list with the instances of MBOMh and MBOMi.

In the next step, AD, AR and auxiliary material lists have been asserted for MBOMh, however the system displays an error as shown in figure 8. This is because of the axioms applied on MBOMh to constrain its semantics (please refer back to figure 5). These constraints actually avert any attempt made to assert the AR assembly component list and the auxiliary material list as evident from figure 8. This shows that the system understands the definition of MBOMh formalized in the previous section.


Assembly Reference Ontology - Asserter
File Edit Windows Help
Database: //localhost/Assembly Reference Ontology
Default User
Status: Connected to //localhost/Assembly Reference Ontology
Enter ECLIF assertions below, browse, or drag and drop files...
Browse files
Clear
Check ICs?
Load
Assertion Load error
More
Done

```
(MBOMh mbomh1)
(hasARAssemblyComponentList mbomh1 (listof Nut01 Nut02 Nut03 Nut04 Nut05 Nut06
Bolt01 Bolt02 Bolt03 Bolt04 Bolt05 Bolt06))
(hasADAssemblyComponentList mbomh1 (listof Bracket01 MainBody01 BodyBrassy01
Platelever01 Handle01 Ball01 HandleBallassy01 Blade01 TopCover01))
(hasAuxiliaryMaterialList mbomh1 (listof PaintABC01 TapeXYZ01 OilShell01))
```

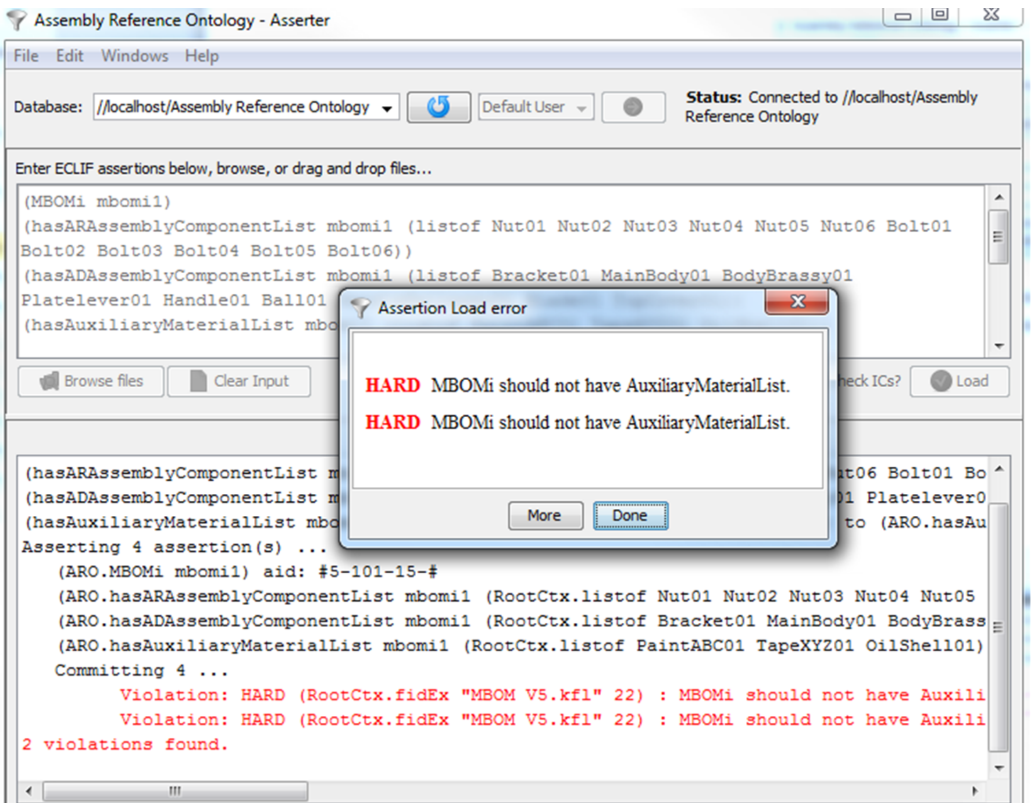
HARD MBOMh should not have ARAssemblyComponentList.
HARD MBOMh should not have AuxiliaryMaterialList.
HARD MBOMh should not have ARAssemblyComponentList.
HARD MBOMh should not have AuxiliaryMaterialList.

```
(hasAuxiliaryMaterialL                                      to (ARO.hasAu
Asserting 4 assertion
    (ARO.MBOMh mbomh1)
    (ARO.hasARAssemblyC                                     Nut04 Nut05
    (ARO.hasADAssemblyC                                     dy01 BodyBrass
    (ARO.hasAuxiliaryMa                                     1 OilShell01)
   Committing 4 ...
          Violation: HA                                     t have ARAsse
          Violation: HARD (RootCtx.fidEx "MBOM V5.kfl" 20) : MBOMh should not have Auxili
          Violation: HARD (RootCtx.fidEx "MBOM V5.kfl" 21) : MBOMh should not have ARAsse
          Violation: HARD (RootCtx.fidEx "MBOM V5.kfl" 20) : MBOMh should not have Auxili
4 violations found.
```

Figure 8. IC violations caused due to AR assembly component list and auxiliary material list when asserting the facts for MBOMh

Finally when the AD, AR and auxiliary material lists were asserted for MBOMi, the system displayed an error message. The IC violation in this case has been observed because auxiliary material list was also asserted for MBOMi. The MBOMi definition states that it should not have auxiliary material list, therefore the system has returned the expected results by not allowing the assertion of auxiliary material list for the MBOMi as shown in figure 9.

Now if the auxiliary material list is removed and the AR assembly component list and AD assembly component list are asserted, the system will accept the assertion. It is evident from these assertions that the system only accepts those facts which comply with the formal definitions of the concepts. Once the facts have been successfully asserted, queries can be made to further validate the semantics of MBOM concepts.



Status: Connected to //localhost/Assembly Reference Ontology

Enter ECLIF assertions below, browse, or drag and drop files...

```
(MBOMi mbomi1)
(hasARAssemblyComponentList mbomi1 (listof Nut01 Nut02 Nut03 Nut04 Nut05 Nut06 Bolt01
Bolt02 Bolt03 Bolt04 Bolt05 Bolt06))
(hasADAssemblyComponentList mbomi1 (listof Bracket01 MainBody01 BodyBrassy01
Platelever01 Handle01 Ball01
(hasAuxiliaryMaterialList mbo
```

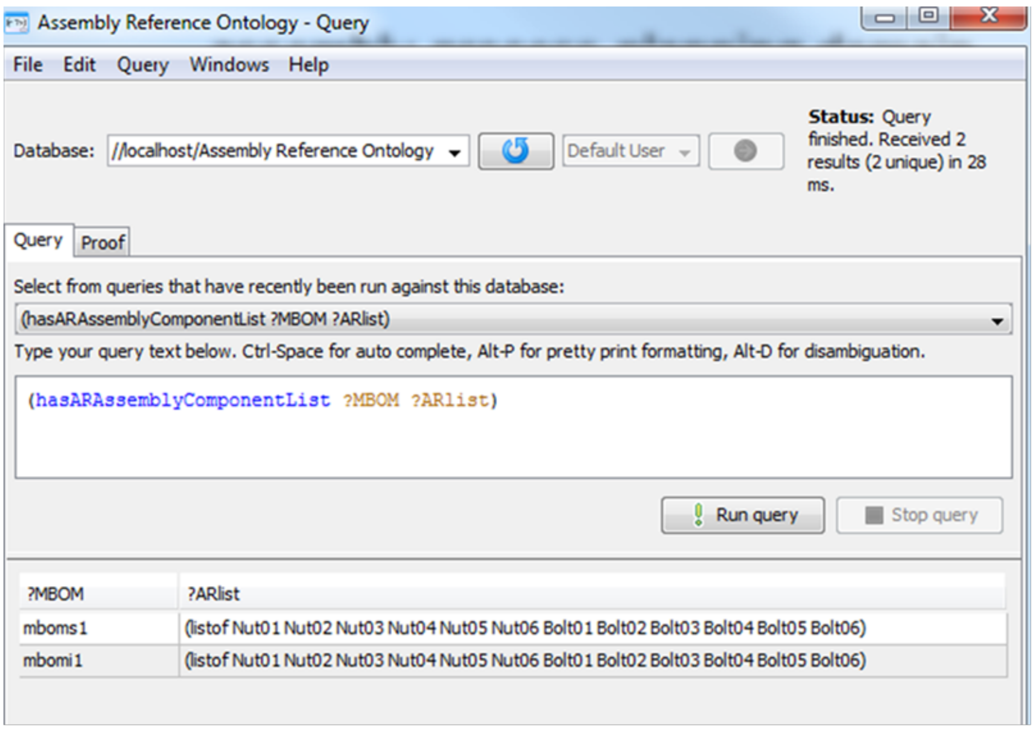
HARD MBOMi should not have AuxiliaryMaterialList.
HARD MBOMi should not have AuxiliaryMaterialList.

```
(hasARAssemblyComponentList m                                      t06 Bolt01 Bo
(hasADAssemblyComponentList m                                      01 Platelever0
(hasAuxiliaryMaterialList mbo                                      to (ARO.hasAu
Asserting 4 assertion(s) ...
   (ARO.MBOMi mbomi1) aid: #5-101-15-#
   (ARO.hasARAssemblyComponentList mbomi1 (RootCtx.listof Nut01 Nut02 Nut03 Nut04 Nut05
   (ARO.hasADAssemblyComponentList mbomi1 (RootCtx.listof Bracket01 MainBody01 BodyBrass
   (ARO.hasAuxiliaryMaterialList mbomi1 (RootCtx.listof PaintABC01 TapeXYZ01 OilShell01)
   Committing 4 ...
        Violation: HARD (RootCtx.fidEx "MBOM V5.kfl" 22) : MBOMi should not have Auxili
        Violation: HARD (RootCtx.fidEx "MBOM V5.kfl" 22) : MBOMi should not have Auxili
2 violations found.
```

Figure 9. IC violated due to assertion of auxiliary material list for MBOMi

For instance, if a query is made to find out the MBOM having AR assembly component list, it will return the results for MBOMs and MBOMi as shown in figure 10 (a). The query does not show MBOMh because the AR assembly component list was not allowed to be asserted. The next query asks to find out instances of MBOM which have AD assembly component list. The system returns instances of MBOMs, MBOMh and MBOMi along with the instance of AD assembly component list as shown in figure 10 (b). This is because all MBOM specialized classes have AD assembly component list. The last query shown in figure 10 (c) is made to find out an instance of MBOM which have auxiliary material list. The system returns an instance of MBOMs along with an instance of auxiliary material list. This shows that only MBOMs has auxiliary material list.

The queries results suggest that the only common list found between MBOMs, MBOMh and MBOMi is the AD assembly component list. This implies that AD assembly component list can provide a link between all three MBOM classes which can subsequently provide a route to support information sharing across these application specific systems.



Database: //localhost/Assembly Reference Ontology

Status: Query finished. Received 2 results (2 unique) in 28 ms.

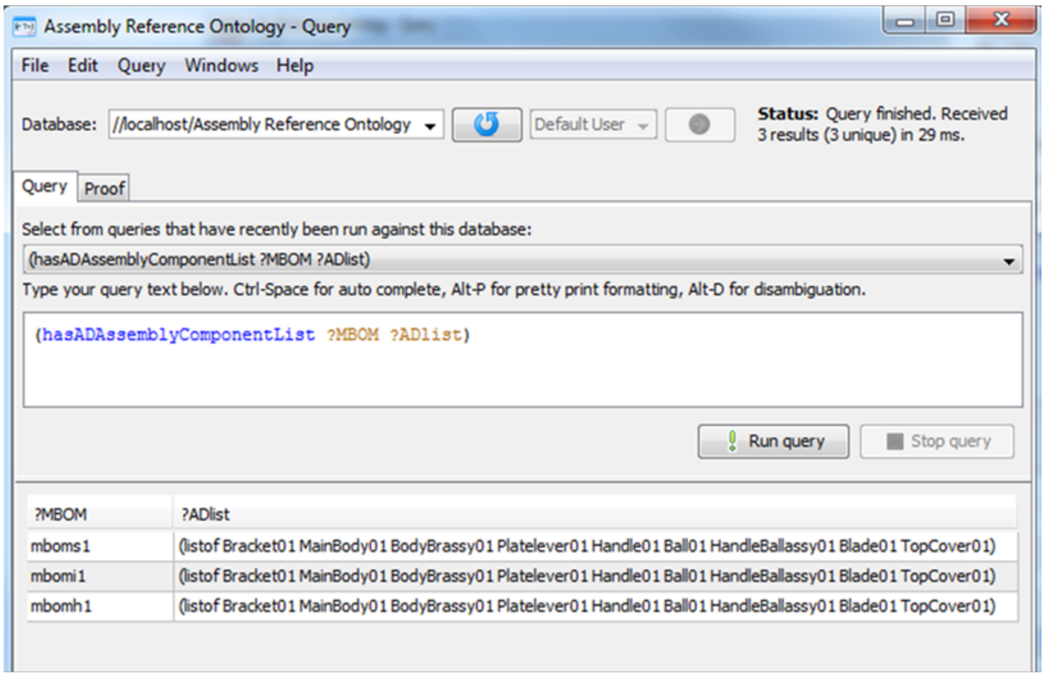
Select from queries that have recently been run against this database:
(hasARAssemblyComponentList ?MBOM ?ARlist)

Type your query text below. Ctrl-Space for auto complete, Alt-P for pretty print formatting, Alt-D for disambiguation.

```
(hasARAssemblyComponentList ?MBOM ?ARlist)
```

?MBOM	?ARlist
mboms1	(listof Nut01 Nut02 Nut03 Nut04 Nut05 Nut06 Bolt01 Bolt02 Bolt03 Bolt04 Bolt05 Bolt06)
mbomi1	(listof Nut01 Nut02 Nut03 Nut04 Nut05 Nut06 Bolt01 Bolt02 Bolt03 Bolt04 Bolt05 Bolt06)

(a) Query for AR assembly component list



Database: //localhost/Assembly Reference Ontology

Status: Query finished. Received 3 results (3 unique) in 29 ms.

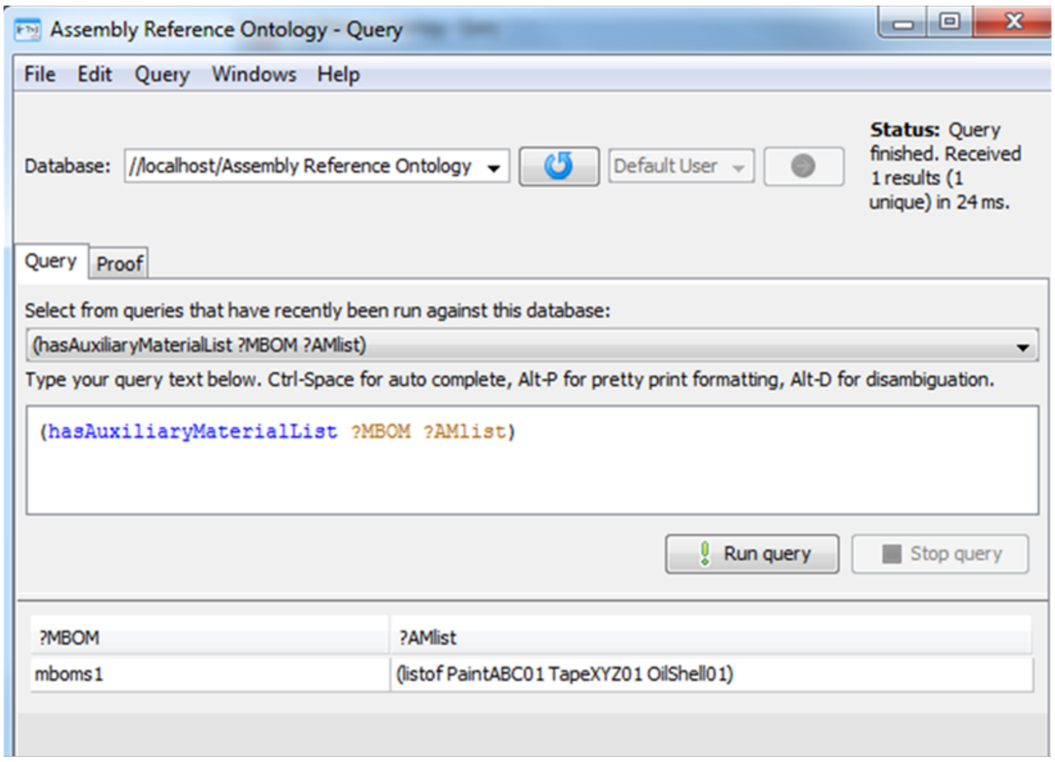
Select from queries that have recently been run against this database:
(hasADAssemblyComponentList ?MBOM ?ADlist)

Type your query text below. Ctrl-Space for auto complete, Alt-P for pretty print formatting, Alt-D for disambiguation.

```
(hasADAssemblyComponentList ?MBOM ?ADlist)
```

?MBOM	?ADlist
mboms1	(listof Bracket01 MainBody01 BodyBrassy01 Platelever01 Handle01 Ball01 HandleBallassy01 Blade01 TopCover01)
mbomi1	(listof Bracket01 MainBody01 BodyBrassy01 Platelever01 Handle01 Ball01 HandleBallassy01 Blade01 TopCover01)
mbomh1	(listof Bracket01 MainBody01 BodyBrassy01 Platelever01 Handle01 Ball01 HandleBallassy01 Blade01 TopCover01)

(b) Query for AD assembly component list


Assembly Reference Ontology - Query
File Edit Query Windows Help
Default User
Query
Proof
Run query
Stop query

Database: //localhost/Assembly Reference Ontology

Status: Query finished. Received 1 results (1 unique) in 24 ms.

Select from queries that have recently been run against this database:

(hasAuxiliaryMaterialList ?MBOM ?AMlist)

Type your query text below. Ctrl-Space for auto complete, Alt-P for pretty print formatting, Alt-D for disambiguation.

```
(hasAuxiliaryMaterialList ?MBOM ?AMlist)
```

?MBOM	?AMlist
mboms1	(listof PaintABC01 TapeXYZ01 OilShell01)

(c) Query for auxiliary material list

Figure 10. Queries made to find out AR assembly component list, AD assembly component list and auxiliary material list.

It is evident from the results of the experiments (based on the case study) that the knowledgebase system is capable of understanding the semantics of MBOM concepts and therefore does not allow assertions which do not follow the formal definitions of the MBOM concepts. Furthermore, the results have also shown that the formal definition of the concepts are inherited by the specialized classes and that the specialized concepts e.g. MBOMs, MBOMh, MBOMi cannot violate the definitions of their parent classes e.g. MBOM. The results have also revealed that more constraints can be applied to the specialized concepts to exploit them for specific application. This also suggests that the ARO concepts can be specialized into different application specific concepts by applying axioms to control their semantics.

The results of the experiments have also shown that the systems having different interpretations of MBOM can be partly interoperable through the identification of common concept. Thus a potential route to enable interoperability can be established across the heterogeneous systems.

6. Conclusions and Further Work

The research work reported in this paper has demonstrated the potential of formal reference ontologies to support interoperability in the assembly domain. The Common Logic (CL) based Knowledge Frame Language (KFL) has been used as formal ontological approach to capture the semantics of assembly concepts. HighFleet's Integrated Ontology Development Environment (IODE) has been used to implement and evaluate the ontology.

This research has proposed an interoperability framework called Assembly Reference Ontology (ARO) to support interoperability across multiple assembly systems. The ARO has multiple layers of reference concepts which have been specialized from the most generic level to the most specialized level to capture the meanings of concepts at various levels of specializations. It has been demonstrated with the example of MBOM concepts that the ARO can be used as a reference ontology to support the capture and sharing of application specific assembly concepts. Three different application specific interpretations of MBOM were considered. First they were informally defined and represented in UML diagrams. Subsequently their semantics were formally captured using the ARO concepts. Finally, with the help of a case study, the ontology was experimentally evaluated by asserting facts and building queries. It was found that the system understands the formal definitions of these concepts at various levels of specializations and that a route to enable interoperability can be identified across the participating systems by using a common concept.

It has also been shown that the CL based KFL has more expressive and reasoning capabilities as compared to other formalisms such as Web Ontology Language (OWL) and Semantic Web Rule Language (SWRL). For example, in contrast to OWL and SWRL, KFL fully supports ternary and/or higher order relations, binary and/or higher order functions, negation operator and Integrity Constraints (ICs) in axioms (Palmer, 2012). In section 4.3, it was exposed that the relation "item" had variable arity and could support relations having any number of arguments. Similarly the function "listof" was used to represent higher order functions (upto 12 arguments in this work) as displayed in figures 8 and 9. The higher order relations and functions have enabled the capture of the semantics of complex assembly concepts and relationships. Similarly, in section 4.3, the negation operator "not" and ICs have been used in axioms to define the semantics of assembly components lists, auxiliary material list and three application specific MBOM interpretations.

A next issue for this area of work is to explore the definition of specific assembly systems that are partly based on the reference ontology. In that case there would be partial semantic interoperability based on the level of compliance that the 2 systems have with the reference ontology. The suitability of the 2 systems in terms of information sharing would then depend on which concepts within the 2 ontologies were semantically consistent.

The research work reported in this paper has focused on interoperability issues within the assembly process planning perspective however future research can be extended to explore the interoperability issues across the assembly design and assembly process planning domains. The ARO concepts: BOM and product family can be explored to support interoperability across these domains.

The ARO concepts can be exploited for other related domains where products are disassembled and re-assembled as part of the domain activity. Two such domains are: (1) repair, and (2) remanufacturing. These domains require disassembly and re-assembly of products; therefore the ARO can be investigated to support interoperability in these domains.

Acknowledgements

This research has been supported by the Loughborough University and the University of Engineering & Technology Lahore. We are thankful to these organizations.

References

Ahmed, F., & Han, S. (2015). Interoperability of product and manufacturing information using ontology. Concurrent Engineering: Research and Applications , 1-14.

Berre, A.-J., Elvesæter, B., Figay, N., Guglielmina, C., Johnsen, S. G., Karlsen, D., et al. (2007). The ATHENA Interoperability Framework.In Enterprise Interoperability II (pp. 569-580). London: Springer.

Beydoun, G., Low, G., García-Sánchez, F., Valencia-García, R., & Martínez-Béjar, R. (2014). Identification of ontologies to support information systems development. Information Systems, 45-60.

Brinkley, J. F., Suciu, D., Detwiler, L. T., Gennari, J. H., &Rosse, C. (2006).A framework for using reference ontologies as a foundation for the semantic web.AMIA 2006 Symposium Proceedings Page.

Brunnermeier, S., & Martin, S. (2002). Interoperability costs in us automotive supply chain. Supply Chain Management : An International Journal. (2), 71-82.

Bruno, G., Antonelli, D., & Villa, A. (2015). A reference ontology to support product lifecycle management. 9th CIRP Conference on Intelligent Computation in Manufacturing Engineering - CIRP ICME '14 (pp. 41-46). Elsevier B.V.

Burgun, A. (2006). Desiderata for domain reference ontologies in biomedicine. Journal of Biomedical Informatics, 39, 307-313.

Chang, S.-H., Lee, W.-L., & Li, R.-K. (1997). Manufacturing bill-of-material planning. PRODUCTION PLANNING & CONTROL, 8(5), 437-450.

Chen, D. (2006). Framework for Enterprise Interoperability.Proc.of IFAC Workshop EI2N.

Chen, D., &Doumeingts, G. (2003). European initiatives to develop interoperability of enterprise applications—basic concepts, framework and roadmap. Annual Reviews in Control , 27, 153-162.

Chen, D., Doumeingts, G., &Vernadat, F. (2008). Architectures for enterprise integration and interoperability: Past, present and future. Computers in Industry , 59, 647-659.

Chen, D., &Vernadat, F. (2004). Standards on enterprise integration and engineering - state of the art. INT. J. COMPUTER INTEGRATED MANUFACTURING, 178(3), 235-253.

Chungoora, N. (2010).A Framework to Support Semantic Interoperability in Product Design and Manufacture. Loughborough: NitishalChungoora.

Chungoora, N., Cutting-Decelle, A., Young, R., Gunendran, G., Usman, Z., Harding, J. A., et al. (2013b). Towards the ontology-based consolidation of production-centric standards. International Journal of Production Research , 51 (2), 327-345.

Chungoora, N., Gunendran, G. A., Young, R. I., Usman, Z., Anjum, N. A., Palmer, C., . . . Cutting-Decelle, A.-F. (2012). Extending product lifecycle management for manufacturing knowledge sharing. J Engineering Manufacture, 226(12), 2047-2063.

Chungoora, N., Young, R. I., Gunendran, G., Palmer, C., Usman, Z., Anjum, N. A., et al. (2013a). A model-driven ontology approach for manufacturing system interoperability and knowledge sharing. Computers in Industry , 64, 392-401.

Cranefield, S.,& Pan, J. (2007). Bridging the Gap Between the Model-Driven Architecture and Ontology Engineering. International Journal of Human-Computer Studies , 65 (7), 595-609.

Cutting-Decelle, A., Dubois, A., & Dubois, J. (2002). An information system for the building industries: a communication approach based on industrial components. Data Science Journal , 257-270.

Das, B., Cutting-Decelle, A. F., Young, R., Case, K., Rahimifard, S., Anumba, C. J., &Bouchlaghem, N. (2007).Towards the understanding of the requirements of a communication language to support process interoperation in cross-disciplinary supply chains. International Journal of Computer Integrated Manufacturing, 20(4), 396-410.

Date, C. J. (2007). Logic and Databases The Roots of Relational Theory. Trafford Publishing.

FinES-Cluster.(2011). 2012 Fines Standardization TF. Retrieved 08 26, 2013, from http://www.fines-cluster.eu/fines/wp/standardizationtf/2012/03/27/7-4-1-foundation-and-domain-ontologies/#7

Frohlich, J. (2004). Fabrikokologie/Entsorgungslogistik. TU Dresden. TU Dresden.

Gangemi, A., &Borgo, S. (2004). Core Ontologies in Ontology Engineering Successful cases and best practices for ontology engineering: reusing wellfounded. Proceedings of the EKAW*04 Workshop on Core Ontologies in Ontology Engineering. Northamptonshire (UK).

Hoffmann, C., & Joan-Arinyo, R. (1998). CAD and the product master model. Computer-Aided Design , 30 (11), 905-918.

Grenon, P. (2003). [DL] CFP: Reference Ontology and Applications Ontology: Workshop in Hamburg, 15-18 September 2003. Retrieved 08 26, 2013, from http://dl.kr.org/pipermail/dl/2003/001594.html

Hirata, T. T. (2009).Customer Satisfaction Planning.Taylor & Francis Group.

IEC-62264. (2002). Enterprise-control system integration, Part 1. Models and terminology, Part 2: Model object attributes. Geneva: ISO/IEC.

IEEE-Std-Computer-Dictionary.(1991). IEEE Standard Computer Dictionary A Compilation of IEEE Standard Computer Glossaries. In IEEE Std 610.

Imran, M., & Young, B. (2013). The application of common logic based formal ontologies to assembly knowledge sharing. Journal of Intelligent Manufacturing. doi:DOI 10.1007/s10845-013-0768-4

IODE. (2013). Highfleet Tools User Manual.Highfleet.

ISO/IEC-24707. (2007). Information technology — Common Logic (CL): a framework for a family of logic based. Retrieved 08 14, 2011, from Common Logic Standard: http://standards.iso.org/ittf/licence.html

ISO/TC 10303-224. (2003). ISO/DIS 10303-224:2003(E) Document. ISO.

ISO/TS-10303. (2004). STEP modules related to Product Data Management.Industrial automation systems and integration—Product data representation and exchange. Geneva.

Jean, S., Pierra, G., & Ait-Ameur, Y. (2006). DOMAIN ONTOLOGIES : A DATABASE-ORIENTED ANALYSIS. Proc. of Web Information Systems and Technologies. Setubal, Portugal.

Jiao, J., Tseng, M. M., Ma, Q., &Zou, Y. (2000).Generic Bill-of-Materials-and-Operations for High-Variety Production Management. Concurrent Engineering: Research and Applications, 8(4), 297-321.

Jones, A., Yih, Y., & Wallace, E. (2001).Monitoring and Controlling Operations. In G. Salvendy (Ed.), Handbook of Industrial Engineering (pp. 1768-1790).John Wiley & Sons, Inc.

Kleppe, A., Warmer, J., &Bast, W. (2003). MDA EXPLAINED The Model Driven Architecture: Practice and Promise. PAPERBACK.

Kosanke, K., & Zelm, M. (2005). Interoperability, the INTEROP Project and Standardisation.Proceedings of the 11th International conference on concurrent enterprising-Integrated engineering of products, services and organisations, (pp. 49-51). Nottingham.

Kugathasan, P., & McMahon, C. (2001). Multiple viewpoint design models for automotive body-in-white design. International Journal of Production Research , 39 (8), 1689-1705.

Lee, C., Leem, C. S., & Hwang, I. (2011).PDM and ERP integration methodology using digital manufacturing to support global manufacturing. Int J Adv Manuf Technol, 399-409.

Lohse, N. (2006).Towards an Ontology Framework For the Integrated Design of Modular Assembly Systems.University of Nottingham. Nottingham: University of Nottingham.

Leila, Z.-G. (2009, November-December 29-04). Reference Ontology Presentation. Retrieved 08 26, 2013, from http://www.slideshare.net/ontoini/sitis-kare-09-reference-ontology-presentation

Marcos, E., Acuña, C. J., & Cuesta, C. E. (2006). Integrating Software Architecture into a MDA Framework. In Software Architecture: Lecture Notes in Computer Science (pp. 127-143). Berlin Heidelberg: Springer-Verlag.

MDA-Guide-Document. (2003). MDA Guide Version 1.0. (J. Miller, & J. Mukerji, Eds.) OMG.

Molloy, O., Tilley, S., &Warman, E. A. (1998). Design For Manufacturing and Assembly. Chapman & Hall.

Musen, M. A. (1998). Domain Ontologies in Software Engineering: Use of Protégé with the EON Architecture. Stanford, California.

Navigli, R., &Velardi, P. (2004). Learning Domain Ontologies from Document Warehouses and Dedicated Web Sites. Computational Linguistics, 30(2), 152-179.

Nemuraite, L., Ceponiene, L., &Vedricka, G. (2009). Representation of Business Rules in UML & OCL Models for Developing Information

Models. In A. P. Janis Stirna, The Practice of Enterprise Modeling: First IFIP WG 8.1 Working Conference, PoEM 2008. Laxenburg: IFIP International Federation For Information Processing.

Newman, S., Nassehi, A., Xu, X., Rosso, R., Wang, L., Yusof, Y., et al. (2008).Strategic Advantages of Interoperability for Global Manufacturing Using CNC Technology. Robotics and Computer Integrated Manufacturing , 24, 699-708.

Ou, S., Georgalas, N., Azmoodeh, M., Yang, K., & Sun, X. (2006).A Model Driven Integration Architecture for Ontology-Based Context Modelling and Context-Aware Application Development. In Model Driven Architecture – Foundations and Applications (pp. 188-197). Berlin: Springer-Verlag.

Ouksel, A. M., &Sheth, S. (1999). Semantic interoperability in global information systems. SIGMOD Rec., 28(1), 5-12.

Sanchez-Alonso, S., & Garcia-Barriocanal, E. (2006).Making use of upper ontologies to foster interoperability between SKOS concept schemes. Online Information Review, 30(3), 253-277.

Palmer, C., Chungoora, N., Young, R., Gunendran, A. G., Usman, Z., Case, K., et al. (2012). Exploiting unified modelling language (UML) as a preliminary design tool for Common Logic-based ontologies in manufacturing. International Journal of Computer Integrated Manufacturing , 1-17.

Panetto, H. (2007). Towards a classification framework for interoperability of enterprise applications. International Journal of Computer Integrated Manufacturing , 20 (8), 727-740.

Panetto, H., & Molina, A. (2008). Enterprise Integration and Interoperability in Manufacturing Systems: trends and issues. Computers in Industry , 59 (7), 641-646.

Panetto, H., Dassisti, M., & Tursi, A. (2012). ONTO-PDM: Product-driven ONTOlogy for Product Data Management interoperability within manufacturing process environment. Advanced Engineering Informatics , 26, 334-348.

Panetto, H., Scannapieco, M., & Zelm, M. (2004). INTEROP NoE: Interoperability Research for Networked Enterprises Applications and

Software. In On the Move to Meaningful Internet Systems 2004: OTM 2004Workshops (pp. 866-882). Berlin Heidelberg: Springer-Verlag.

Plastiras, P., Sullivan, D. O., & Weller, P. (2014). An Ontology-Driven Information Model for Interoperability of Personal and Electronic Health Records.eTELEMED 2014 : The Sixth International Conference on eHealth, Telemedicine, and Social Medicine (pp. 130-133). IARIA.

Raine, J., Pons, D., & Whybrew, K. (2001). The design process and a methodology for system integrity. IMechE, Part B: J Engineering Manufacture , 215 (4), 569-576.

Ray, S., & Jones, A. (2003). Manufacturing interoperability. Proceedings of the 10th ISPE International Conference, (pp. 535-540). Madeira Island, Portugal.

Ray, S. R., & Jones, A. T. (2006). Manufacturing Interoperability. Journal of Intelligent Manufacturing , 17, 681-688.

Rezaei, R., Chiew, T. K., & Lee, S. P. (2014b). A review on E-business Interoperability Frameworks. The Journal of Systems and Software , 93, 199-216.

Rezaei, R., Chiew, T. K., Lee, S. P., & Aliee, Z. S. (2014a). A semantic interoperability framework for software as a service systems in cloud computing environments. Expert Systems with Applications , 5751-5770.

Roser, S., & Bauer, B. (2006).Ontology-Based Model Transformation.In Satellite Events at the MoDELS 2005 Conference (pp. 355-356). Berlin: Springer.

Sanya, I., & Shehab, E. (2015). A framework for developing engineering design ontologies within the aerospace industry. International Journal of Production Research , 2383-2409.

SCRA. (2006). STEP APPLICATION HANDBOOK ISO 10303 VERSION 3. SCRA.

Stark, J. (2011). Product Lifecyle Management: 21st Century Paradigm for Product Realisation .Springer-Verlag.

The-ATHENA-Consortium. (2004-2006). ATHENA Interoperability Framework (AIF). Retrieved April 28, 2015, from ATHENA: http://athena.modelbased.net/motivation.html

Tursi, A., Panetto, H., Morel, G., &Dassisti, M. (2009).Ontological approach for products-centric information system interoperability in networked manufacturing enterprises. Annual Reviews in Control, 33(2), 238-245.

Usman, Z., Young, R., Chungoora, N., Palmer, C., Case, K., & Harding, J. (2011). A Manufacturing Core Concepts Ontology for Product Life Cycle Interoperability. International Ifip Working conference on Enterprise Interoperability. Stockholm, Sweden: Springer.

Usman, Z., Young, R., Chungoora, N., Palmer, C., Case, K., & Harding, J. (2013).Towards a formal manufacturing reference ontology. International Journal of Production Research. doi: http://dx.doi.org/10.1080/00207543.2013.801570

Vernadat, F. (2007). Interoperable enterprise systems: Principles, concepts, and methods. Annual Reviews in Control , 31, 137-145.

Vollmann, T. E. (1997). Manufacturing Planning and Control Systems (4th ed.).

Wache, H., Vogele, T., Visser, U., Stuckenschmidt, H., Schuster, G., Neumann, H., et al. (2001). Ontology-Based Integration of Information-A Survey of Existing Approaches. IJCAI--01 Workshop: Ontologies and Information Sharing .

Woodley, M. S. (2005, 11 07). DCMI Gloassary. (The Dublin Core® Metadata Initiative) Retrieved 02 28, 2011, from http://dublincore.org/documents/usageguide/glossary.shtml

Xu, H., Xu, X., & He, T. (2008).Research on Transformation Engineering BOM into Manufacturing BOM Based on BOP. Applied Mechanics and Materials, 10-12, 99-103.

Young, R., Gunendran, G., Chungoora, N., Harding, J., & Case, K. (2009). Enabling interoperable manufacturing knowledge sharing in PLM. International conference on Product Lifecycle Management. Inderscience Enterprises Ltd.

Young, R., Gunendran, A. G., Cutting-Decelle, A. F., &Gruninger, M. (2007). Manufacturing knowledge sharing in PLM: a progression towards the use of heavy weight ontologies. International Journal of Production Research, 45(1), 1505-1519.

Zhang, Y., Mo, R., Shi, Y., & Chang, Z. (2010).A Product Manufacturability Reverse Mapping Method Based on BOM Transformation. International Conference on Computer Application and System Modeling. IEEE.