Computers in Industry 64 (2013) 392–401

Contents lists available at SciVerse ScienceDirect

Computers in Industry

journal homepage: www.elsevier.com/locate/compind

A model-driven ontology approach for manufacturing system interoperability and knowledge sharing

Nitishal Chungoora [a,*], Robert I. Young [a], George Gunendran [b], Claire Palmer [a], Zahid Usman [a], Najam A. Anjum [a], Anne-Françoise Cutting-Decelle [c,d,e], Jennifer A. Harding [a], Keith Case [a]

[a] Wolfson School of Mechanical and Manufacturing Engineering, Loughborough University, Loughborough, Leicestershire LE11 3TU, UK
[b] Control Techniques Ltd., The Gro, Newtown, Powys SY16 3BE, UK
[c] CODATA France, F-75016 Paris, France
[d] Université Lille Nord de France, F-59000 Lille, France
[e] LM2O, Ecole Centrale de Lille, Cité Scientifique, BP 48, 59651 Villeneuve d'Ascq, France

ARTICLE INFO

Article history:
Received 13 December 2011
Received in revised form 9 October 2012
Accepted 11 January 2013
Available online 6 March 2013

Keywords:
Ontologies
Model Driven Architecture
Product Lifecycle Management
Knowledge sharing

ABSTRACT

The requirements for the interoperability of semantics and knowledge have become increasingly important in Product Lifecycle Management (PLM), in the drive towards knowledge-driven decision support in the manufacturing industry. This article presents a novel concept, based on the Model Driven Architecture (MDA). The concept has been implemented under the Interoperable Manufacturing Knowledge Systems (IMKS) project in order to understand the extent to which manufacturing system interoperability can be supported using radically new methods of knowledge sharing. The concept exploits the capabilities of semantically well-defined core concepts formalised in a Common Logic-based ontology language. The core semantics can be specialised to configure multiple application-specific knowledge bases, as well as product and manufacturing information platforms. Furthermore, the utilisation of the expressive ontology language and the generic nature of core concepts help support the specification of system mechanisms to enable the verification of knowledge across multiple platforms. An experimental demonstration, using a test case based on the design and manufacture of an aerospace part, has been realised. This has led to the identification of several benefits of the approach, its current limitations as well as the areas to be considered for further work.

1. Introduction

The traditional approach towards information sharing across the product lifecycle has been based on the utilisation of a single common schema, or product master model [1], which prescribes a rigid information structure for use across several engineering functions. However, it has been recognised that this approach remains problematic as engineers are interested in different lifecycle viewpoints, terms, definitions and representations [2]. These factors inevitably have repercussions in the ability to achieve system interoperability and knowledge sharing. It has been shown, e.g., that the lack of interoperability is very costly to globally distributed industries in the manufacturing sector [3]. This issue is also prevalent across other fields such as services, healthcare and business-to-business e-commerce amongst others and, therefore, the issue of knowledge sharing still remains to be addressed.

Several resourceful efforts have been fostered to progress towards improved system configurations and interoperability following the Model Driven Architecture (MDA) [4], Model Driven Interoperability (MDI) and ontology development methods [5–11]. Related works such as Djuric et al. [12] also indicate that there is a tendency to exploit the combined use of MDA with Semantic Web ontology languages. However, while Semantic Web ontology languages like the Web Ontology Language (OWL) [13] can prove suitable for certain model-driven systems [9,12,14], these ontology languages fall short of meeting the rich and verifiable semantics and structures that are required by product lifecycle systems. It is also evident that current work does not entirely address the practical deployment and testing side of MDA, MDI and ontology-driven system development applied to industrial test cases across the product lifecycle.

This article identifies a concept based on MDA, MDI and ontology-driven specifications as contribution towards the understanding of novel model-driven methods for achieving manufacturing system interoperability and knowledge sharing across multiple platforms in the product lifecycle. More

* Corresponding author.
E-mail address: tishal82@gmail.com (N. Chungoora).

0166-3615/$ – see front matter.
http://dx.doi.org/10.1016/j.compind.2013.01.003

specifically, this work targets the combined exploitation of model-driven and ontology-driven system development using the expressive power of the Extended Common Logic Interchange Format (ECLIF) as ontology language [26]. The model-driven concept has been explored and tested as part of the Interoperable Manufacturing Knowledge Systems (IMKS) research project [15]. The concept builds on the ideas of extensible generic or core ontologies of manufacturing [16–20] as well as knowledge verification methods [21–23]. The understanding and results from the completed IMKS project are being presented in the context of MDA and MDI.

The problem that is specifically addressed in the paper is how to relate the manufacturing engineering understanding of a product to that of a design engineer such that a manufacturing system can identify and provide the appropriate manufacturing consequences of design decisions back to a designer through an underlying understanding of design and manufacturing viewpoints. This paper argues that the novel combination of the MDA approach with formal ontology driven specifications based on core and specialised domain ontologies can provide a route to improved knowledge sharing across product design and manufacture. Other aspects of the product lifecycle are not addressed in this paper although the principles that are used should be equally applicable.

The methodology applied in this work has comprised of an industrial use case exploration based on the design and manufacture of specific aerospace parts. This has been used to explore the application of MDA in combination with ontology engineering approaches to the definition and construction of a manufacturing knowledge-sharing environment. This has aided the specification of the key requirements and concepts that are relevant to supporting multiple system development and knowledge verification across the design and manufacture phases of the product lifecycle. A Manufacturing Core Ontology has been proposed to provide an extensible set of formally-defined core concepts from which application-specific systems can be developed [24]. Essential mechanisms for enabling knowledge verification across systems have also been researched. The experimental environment has been tested with a number of case study examples to demonstrate that relevant product manufacture knowledge can be provided back to reflect a change being made from a product design perspective.

This article is structured as follows: Section 2 identifies the proposed ontology-based model-driven concept, leading to a description of the key elements in the development of the approach in Section 3. Section 4 documents an industrial test case which exploits the configuration provided by the IMKS experimental system to validate the applicability of the model-driven concept. A discussion of the findings from this work is elaborated along with the relevant conclusions in Section 5.

2. IMKS model-driven concept

2.1. Overview

Fig. 1 depicts the IMKS model-driven concept in relationship to the hierarchy of models within MDA. MDA is an approach to IT system specification that divides up the specification of system functionalities from the intended technology-specific implementation platform [6]. The architecture defines three viewpoints [6–8] and their corresponding models notably: the Computation Independent Model (CIM), the Platform Independent Model (PIM) and the Platform Specific Model (PSM), whose interactions involve transformations for converting one model to another on the same system. The implications of the different levels within MDA and their respective models applied to IMKS are next explained.

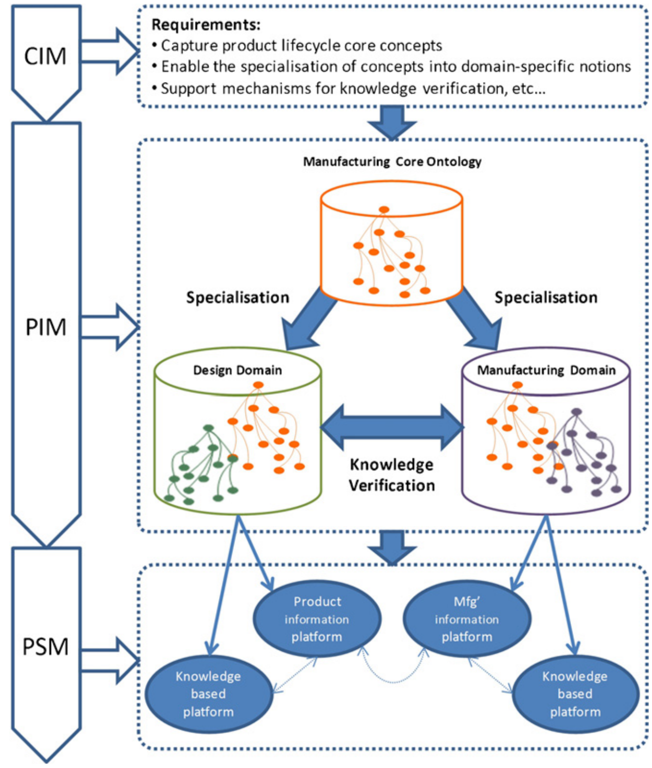

Fig. 1. The IMKS model-driven concept.

2.2. Specification models and their transformations

2.2.1. Computation independent models

The CIM specifies the high-level requirements that should be fulfilled by the models to be developed at the PIM level. Some examples of generic requirements, identified in Fig. 1, for enabling the specification of the Manufacturing Core Ontology consist of (1) the ability to capture product lifecycle core concepts, (2) the requirement for accommodating domain concepts via specialisation mechanisms and (3) the need to support mechanisms for knowledge verification amongst others. A more detailed view of requirements defined at the CIM level is presented in Section 3.

2.2.2. Platform independent models

The PIM, on the other hand, defines a model at a relatively high level of abstraction, where the model is used to describe the software solution using a technology independent view [25]. In Fig. 1, at the heart of the PIM level is the Manufacturing Core Ontology. The methodology for specifying this ontology comprises the initial capture of a lightweight, platform-independent, UML class model. This lightweight representation is then formalised as heavyweight ontology encoded in the Extended Common Logic Interchange Format (ECLIF) [26]. The latter is an extended version of the Common Logic Interchange Format which, as a recommendation from ISO/IEC 24707 [27], supports a logic framework for the exchange and transmission of information.

The Manufacturing Core Ontology can be specialised into multiple domain-specific PIMs, in order to address the semantic requirements of different domains in the product lifecycle such as design and manufacture, as shown in Fig. 1. These domain-specific PIMs are also formalised in ECLIF and are, therefore, platform-independent as far as the semantics of their structures are concerned.