Expert Systems with Applications 39 (2012) 10420–10437

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

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

An MDA approach to knowledge engineering

Nicolas Prat [a], Jacky Akoka [b,*], Isabelle Comyn-Wattiau [c]

[a] ESSEC Business School, 1, Av. B. Hirsch, 95000 CERGY, France
[b] CEDRIC-CNAM & Institut Telecom/TEM Research, 2 Rue Conté, 75003 Paris, France
[c] CEDRIC-CNAM & ESSEC Business School, 2 Rue Conté, 75003 Paris, France

ARTICLE INFO

Keywords:
Knowledge engineering
Model-driven approach
Knowledge model
Production rule
CommonKADS
Production rule representation (PRR)

ABSTRACT

This paper proposes an MDA approach to knowledge engineering, centered on the CommonKADS knowledge model. The latter corresponds to the CIM level of MDA whereas PRR, which represents production rules and rulesets, corresponds to the PIM level. The paper explores the mapping between CommonKADS knowledge models and production rules and rulesets based on PRR. Mapping CommonKADS knowledge models into PRR is very useful, due to the fact that the CIM level remains relatively unexplored, despite its key role in MDA. This motivates our choice to focus on the CIM and PIM levels. Furthermore, the mapping between PIM and PSM (i.e. the implementation of production rules in specific rule-based systems) constitutes less of an issue. To map CommonKADS knowledge models into PRR production rules and rulesets, we propose and illustrate a set of transformations. To ease these transformations, we start by grouping elements of the CommonKADS knowledge models into so-called "inference groups". We propose and illustrate an algorithm that defines these inference groups automatically. The definition of transformations between models (CIM to PIM levels) requires a specific metamodel for CommonKADS as well as a dedicated metamodel for PRR. Unlike PRR, there is no published CommonKADS metamodel. This paper proposes a comprehensive CommonKADS knowledge metamodel. We describe and discuss an example, applying the whole approach.

1. Introduction

Fox (2011) defines knowledge engineering as "the engineering discipline that involves integrating knowledge into computer systems in order to solve complex problems normally requiring a high level of human expertise". The activity of knowledge engineering typically results in expert systems, with knowledge represented in the form of production rules. The latter enable to represent business rules in information systems. A fair number of rule engines support production rules, among which Jess, Ilog JRules, and JBoss. Although knowledge engineering is now an established academic discipline, practitioners often regret a lack of methodological support for the development of rule-based systems (Zacharias, 2008). This paper presents a contribution in the domain of knowledge engineering. More specifically, we provide methodological support for the mapping of knowledge models into production rules, for subsequent mapping into expert systems. One originality of our work is to combine the knowledge engineering method CommonKADS (Schreiber et al., 2000) with the model-driven approach (Object Management Group, 2003), which originated in the domain of software engineering.

CommonKADS is a widespread methodology, and the most influential academic method in knowledge engineering to date (Zacharias, 2009). The methodology proposes several models, with a strong focus on knowledge models. These models require further efforts before implementation.

In model-driven approaches, models drive the software development process. The latter distinguishes different abstraction levels: conceptual, logical and physical. Models of lower abstraction levels result from mapping models from upper levels. This distinction between abstraction levels is now a consensus and forms the core of information systems engineering. The Model-Driven Architecture (*MDA*) emerged more recently. It is a standard model-driven approach proposed by the Object Management Group (OMG). It facilitates the definition of guidelines, helping learners in the appropriation of the different design steps. MDA also facilitates tool support in the development of information systems, thus directly affecting productivity and maintainability. MDA presents the advantage of separating design aspects from architecture issues. To this end, it defines three abstraction levels (quite similarly to the traditional distinction between conceptual, logical and physical levels in information systems engineering). The Computation Independent Model (CIM) constitutes the more abstract level. It represents the context and purpose of the information system without any computational consideration. It focuses on the business and the conceptual considerations. The Platform Independent

* Corresponding author.
E-mail addresses: prat@essec.edu (N. Prat), akoka@cnam.fr (J. Akoka), wattiau@cnam.fr (I. Comyn-Wattiau).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2012.02.010

Model (PIM) describes the behavior and structure of the information system, regardless of the implementation choices. Finally, at the third level, the Platform Specific Model (PSM) contains all required information allowing developers to build the code and to execute the resulting application. Applying MDA to the development of knowledge systems requires a mapping between MDA abstraction levels and knowledge models.

Gartner, Inc. (2010) considers that MDA has the potential for providing high payoff to companies choosing this development approach. Among the several benefits that this approach provides, let us mention portability, interoperability and reusability (Zhu, 2006). Although MDA primarily focuses on the development of transactional applications, it also has the potential to apply to the development of other types of applications. For example, MDA, and more generally model-driven approaches, provide major benefits to the development of data warehouses (Mazon & Trujillo, 2008; Prat, Akoka, & Comyn-Wattiau, 2006). Similarly to data warehousing, MDA also has great potential to apply in the development of rule-based systems and, more generally, in knowledge engineering. As Baumeister, Reutelshoefer, and Puppe (2011) point out, knowledge engineers often need to face the flexibility/productivity dilemma. MDA provides a means for solving this dilemma, by distinguishing between the three abstraction levels. Thanks to this distinction, after defining a model at the PIM level for instance, developers may generate several models at the PSM level, depending on the target expert system.

Some previous work has already combined MDA with knowledge engineering and knowledge systems. However it suffers from two major limitations:

1. It often focuses on rule languages (Wagner, Antoniou, Tabet, & Boley, 2004). Knowledge engineering involves the definition of rules, but extends far beyond this activity.
2. When the focus is not just rule languages but knowledge engineering, CommonKADS knowledge models are placed at the PIM level (Abdullah, Benest, Paige, & Kimble, 2007; Cañadas, Palma, & Tùnez, 2005). We argue that the PIM level is not the right level for CommonKADS knowledge models. These models provide all the tools required to analyze tasks in terms of knowledge involved at different granularities. Knowledge analysis aims at studying the tasks at a conceptual level. Thus, we consider CommonKADS knowledge models as the CIM level of MDA.

To provide methodological support for mapping knowledge models into production rules, the originality of our approach lies in combining CommonKADS knowledge models (CIM level of MDA) with production rules representation language (PRR) proposed by the Object Management Group (2009). OMG has defined PRR at the PIM level of MDA. Production rules constitute a very rich representation of a large body of knowledge. Many languages enable the definition of production rules. Admittedly, the model-driven approach forms one of the theoretical bases of CommonKADS. However, we argue that the more abstract levels, and more particularly the knowledge model, are the central contribution of the CommonKADS methodology. Thus, by combining this knowledge model with a standard for production rule representation (PRR), we combine the advantages of CommonKADS with a standard for production rule representation.

The paper contributes to methodological support in knowledge engineering in the following ways:

1. It proposes an MDA approach to knowledge engineering, centered on the CommonKADS knowledge model. The CommonKADS knowledge model corresponds to the CIM level of MDA; PRR, which represents production rules and rulesets, corresponds to the PIM level.
2. The paper explores the mapping between CommonKADS knowledge models (CIM level of MDA) and production rules and rulesets based on PRR (PIM level). There exists a possible convergence between CommonKADS and PRR, since both rely on the unified modeling language (UML: Object Management Group, 2010c). However, the two are not completely related. Therefore, mapping CommonKADS knowledge models into PRR constitutes an open research problem. We argue that such a mapping can be very useful. We concur with Gartner, Inc. (2010) that the CIM level remains relatively unexplored, despite its key role in MDA. This motivates our choice to focus on the CIM and PIM levels. Furthermore, the mapping between PIM and PSM (i.e. the implementation of production rules in specific rule-based systems) constitutes less of an issue (Object Management Group, 2009). To map CommonKADS knowledge models into PRR production rules and rulesets, we propose and illustrate a set of transformations. To ease these transformations, we start by grouping elements of the CommonKADS knowledge models into so-called "inference groups". We propose and illustrate an algorithm that defines these inference groups automatically.
3. The definition of transformations between models (CIM to PIM levels) requires a specific metamodel for CommonKADS as well as a dedicated metamodel for PRR. Unlike PRR, there is no published CommonKADS metamodel. This paper proposes a comprehensive CommonKADS knowledge metamodel. Complying with MDA, it represents the metamodel with UML.

The outline of the paper follows. Section 2 surveys related work. It encompasses reference models at the different levels of knowledge engineering. It synthesizes and compares previous related work referencing CommonKADS as well as MDA and PRR concepts. Section 3 presents the example used throughout the paper to illustrate our approach. Section 4 describes the main concepts of the two metamodels: CommonKADS and PRR metamodels. Section 5 presents our MDA-based knowledge engineering approach. This section situates the approach within the MDA framework and presents the transformations mapping CommonKADS knowledge models into PRR and activity models. Finally, Section 6 discusses the contributions and implications of this research, as well as future research opportunities.

2. Related work

In this section, we first present a brief review of research on rule modeling, especially with the PRR language. The second part is dedicated to knowledge engineering methods, with a particular accent on CommonKADS.

2.1. Rule modeling

Abdullah et al. (2007) propose a UML profile for modeling knowledge-based systems within the context of MDA. They define a mapping of the profile elements to Jess concepts. Thus, their mapping is concentrated on PIM to PSM transformation. Their choice of a specific UML profile is justified by the immaturity of PRR standardization work in 2007. They do not address the CIM viewpoint.

Several authors propose different meta-models to represent production rules. Lukichev and Wagner (2006) extend the UML metamodel with the concept of rule and describe an implementation. Milanovic et al. (2009) describe a bridge for transforming abstract and concrete syntax of Web rule languages based on a rule metamodel.

Wagner et al. (2004) have been among the first authors to attempt a link between MDA viewpoints and rule languages. Their