Integrating Natural Language Requirements Models with MDA

María Carmen Leonardi
Universidad Nacional del Centro de la Pcia. de Buenos Aires, Argentina

María Virginia Mauco
Universidad Nacional del Centro de la Pcia. de Buenos Aires, Argentina

INTRODUCTION

The model driven architecture (MDA) is a framework for software development defined by the OMG (Object Management Group, 2006). The MDA initiative shifts the focus of software development from writing code to building models, separating the specification of functionality from the specification of the specific implementation of that functionality (Miller & Mukerji, 2003). Key to MDA is the importance of models and transformations between them in the software development process.

The first model of MDA is the computation independent model (CIM), which describes the business model independently of the software system to be implemented. The CIM is described with a vocabulary that is familiar to stakeholders. As it captures the domain without reference to a particular system implementation or technology, the CIM would remain the same even if the system were implemented mechanically, rather than in computer software (Meservy & Fenstermacher, 2005). CIM reduces the gap between stakeholders and software engineers (Miller et al., 2003).

Recently, some proposals related to business modeling and MDA have appeared, for example, the use of activity diagrams (Mersevy et al., 2005), BPMN (White, 2004), or goal-oriented strategies (Birol, 2006). Several authors agree in the importance of using the "language of the business experts" (Francis, 2006) during the first stages of development, enhancing communication between the domain experts domain and software engineers. From the requirements engineering area, we have been working with natural language requirements models to describe the universe of discourse (Leite, Hadad, Doorn, & Kaplan, 2000). But, to fit in MDA framework, we have to map them into object-oriented models, defining a CIM that will be the basis for a MDA software development. To do this, we have proposed a transformation strategy and developed its associated tool, CIMTool, thus allowing the integration of the natural language models into the MDA framework. In this article, which is an integration of Leonardi (2003), Leonardi (2005), Leonardi and Mauco (2004), and Leonardi, Mauco, and Leoni (2005), we present the overall strategy defining OCL based transformation rules to derive a CIM from the language extended lexicon (LEL) and the scenario model (Leite et al., 2000).

BACKGROUND

MDA is an approach that makes modeling the primary focus of the software development (Miller et al., 2003). It is based on modeling different aspects and levels of abstraction of a system, and exploiting interrelationships between these models. The MDA starts with the idea of separating the specification of the operation of the system from the details of its implementations. In MDA, all artifacts such as requirements specification, architecture descriptions, design descriptions, and code are regarded as models. One of the key features of this framework is the notion of automatic transformations that are used to modify one model in order to obtain another one. MDA defines how models expressed in one language can be transformed into models in other languages. The MDA standard proposes UML as the specification language, and is divided into the following main steps:

- Construct a model describing the business system: Computation independent model (CIM)
- Construct a model with a high level of abstraction: Platform independent model (PIM)
- Transform the PIM into one or more platform specific models (PSMs)
- Transform the PSMs to code

There are some works that propose UML extensions to represent business models (Johnston, 2006; Vasconcelos et al., 2001). These extensions, though not conceived in MDA context, allow the construction of UML models that can be considered as CIMs as they model the business knowledge without considering any software system. However, stakeholders think in terms of processes, goals, resources, actors, among others rather than objects. Object orientation is an abstraction not easily understandable by stakeholders because objects encapsulate data and behavior in the same level (Jackson, 1995). Then, the introduction of techniques

and models closer to stakeholders' way of thinking would be really useful.

During the early stages of development, when the interaction with the stakeholders is crucial, the use of natural language oriented requirements models seems necessary in order to enhance communication. Everyone can read natural language, so it is still used to define the requirements documents (Sommerville, 2005). However, natural language can be ambiguous, surprisingly opaque, and is often misunderstood. This kind of requirements models has to be reinterpreted by software engineers into a more formal design on the way to a complete implementation. In particular, in MDA context it is necessary to transform them to UML models representing the CIM. We can mention some strategies that, though not developed in MDA context, derive UML models from different business models. For example, Díaz, Pastor, Moreno, and Matteo (2004) obtain sequence diagrams from use cases and in Alencar, Pedroza, Castro, Silva, and Ramos (2006), a strategy is proposed to derive class diagrams starting from i* model. In order to improve CIM construction, we describe in the next section a transformation process to automatically derive a class diagram representing a CIM, starting from two natural language requirements models.

A STRATEGY TO DERIVE A CIM FROM REQUIREMENTS MODELS

In this section, we describe the transformation strategy to map the natural language models into a CIM. The section is organized in three subsections: one to present the requirements models, the other presents the derivation strategy, and finally CIMTool, the tool implementing the strategy.

Natural Language-Oriented Requirements Models

The models presented in this section are well known, used, and accepted by the requirements engineering community (Leite et al., 2000). The models are: language extended lexicon model and scenario model.

- **Language extended lexicon:** The language extended lexicon (LEL) is a structure that allows the representation of significant terms of the Universe of Discourse. The purpose of the lexicon is to help to understand the vocabulary and its semantics. It unifies the language allowing communication with the stakeholder. LEL is composed by a set of symbols with the following structure: symbol name: word or phrase and set of synonyms, notions defining the denotation of the symbol and behavioral responses describing the symbol connotation. In the description of the symbols, two rules must be followed simultaneously: the "closure principle" that encourages the use of LEL symbols in other LEL symbols, and the "minimum vocabulary principle" where the use of symbols external to the application language is minimized. LEL terms define objects, subjects, verbal phrase, and states. Figure 1 shows the heuristics to define each type of symbol.
- **Scenario model:** A scenario describes situations of universe of discourse. A scenario uses natural language as its basic representation and it is connected to LEL. Figure 2 describes the components. In Leite et al. (2000) the scenario construction process is described.

Figure 1. Heuristics to represent LEL terms

Subject	**Notions:** Who the subject is.
	Behavioral responses: Register actions executed by the subject.
Object	**Notions:** Define the object and identify relationship with other objects.
	Behavioral responses: Describe the actions that may be applied to this object.
Verb	**Notions:** Describe who executes the action, when it happens, and procedures involved in it.
	Behavioral responses: Describe the constraints on the happening of an action, which are the actions triggered in the environment and new situations that appear as consequence.
State	**Notions:** What it means and the actions, which triggered the state.
	Behavioral responses: Describe situations and actions related to it.