IDEA GROUP PUBLISHING **ITB13067**

701 E. Chocolate Avenue, Suite 200, Hershey PA 17033-1240, USA
Tel: 717/533-8845; Fax 717/533-8661; URL-http://www.idea-group.com

This paper appears in the publication, *Advanced Topics in Database Research, vol. 5*
edited by Keng Siau © 2006, Idea Group Inc.

Chapter VIII

A Template-Based Analysis of GRL

Patrick Heymans, University of Namur, Belgium
Germain Saval, University of Namur, Belgium
Gautier Dallons, DECIS SA/NV, Belgium
Isabelle Pollet, SmalS-MvM/Egov, Belgium

ABSTRACT

This chapter applies the template proposed by Opdahl and Henderson-Sellers to the Goal-oriented Requirements Engineering Language (GRL). It proposes a metamodel of GRL that identifies the constructs of the language and the links between them. Each construct is then described through the template in order to extract and formalise detailed syntactic and semantic information. The latter takes the form of a mapping between a construct and its meaning*, defined in terms of the Bunge-Wand-Weber ontology. Evaluations of both GRL and the template are provided as well as suggestions for improvements. The purpose of our work is to improve the quality of goal modelling. Indeed, despite the increasing popularity of the goal-oriented paradigm, especially in requirements engineering and enterprise modelling, the central notion of goal remains one of the most controversial. A possible cause might be that researchers have devoted too little attention to studying the ontological foundations of goal-oriented languages. This chapter addresses such issues for GRL.*

INTRODUCTION

In 2003, the UEML thematic network (IST Project 2001-34229) started to develop the so-called Unified Enterprise Modelling Language (UEML), a conceptual modelling language designed to be a common ground for representing the various aspects of the enterprise and facilitating the exchange of enterprise models. Reaching these objectives was deemed of utmost importance to improve the development, interoperability, and integration of enterprise information systems.

Indeed, an increasing number of new technologies that strongly rely on enterprise models and ontologies are being introduced into the enterprise: enterprise application integration, domain-specific languages, content and knowledge management systems, semantically enriched agents and Web services, virtual enterprises, and the like. As more and more partial models of the enterprise are created, in different tools and different languages, the risk is that knowledge becomes dispersed and inconsistent. UEML was developed as part of a solution to these problems.

In 2004, UEML 1.0 was delivered. Due to the nature of the UEML project,[1] UEML 1.0 allows mainly the modelling of process aspects but leaves out other aspects (such as static information, functional and non-functional requirements, resources, and goals) and covers only a small set of the identified requirements. It was defined by integrating subsets of three existing enterprise modelling languages (EMLs) — namely, global returnable asset identifier (GRAI) (Doumeingts, 1984), EEML (Jorgensen & Carlsen, 1999) and IEM (Mertins & Jochem, 1999) — by following a methodology inspired by database integration (Petit, 2003). Development of the UEML has since been taken over by the InterOP Network of Excellence (InterOP Project Web site, 2004). It is currently an on-going activity carried out by a consortium made up of the leading practitioners and researchers in the domain of enterprise modelling (EM). The adopted language development approach reconciles scientific rigour and pragmatism. First, it is a requirements document under continuous elaboration that drives the language development process. Second, commonly used EMLs are analysed, each in turn, according to a quality evaluation framework inspired by Krogstie and Sølvberg (2000) in order to guarantee that not only the most used but also the most appropriate and sound constructs are incorporated into the UEML definition. Third, the integration of these constructs into the UEML is done in such a way that syntactic and semantic problems (widespread in other unified languages, such as UML) do not arise. Examples are synonymous, homonymous, underdefined, ill-defined, overly complex or poorly integrated constructs. Finally, evaluations of the successive versions of the language are performed to provide continuous feedback to the language development requirements and process.

The work reported in this chapter describes some contributions of the authors towards the development of UEML 2.0. We focus on the analysis of existing EMLs and, more specifically, on the analysis of GRL, the goal-modelling language standardized by the International Telecommunication Union (ITU, 2003a). GRL is presented in section "GRL in a Nutshell." The future integration of a goal-modelling language into the UEML is expected to improve it by adding the goal dimension that is currently missing. GRL is one of the candidates. In contrast with the previous version of the UEML, for which the semantic aspects of the EMLs were not taken into account in a systematic manner, this time, for UEML 2.0, we decided to adopt the template-based approach (or template, for short) defined by Opdahl and Henderson-Sellers (2004). The template was designed as