

Ontological Rules for UML-Based Conceptual Modeling: Design Considerations and a Prototype Implementation

Shan Lu, Around America Aviation Group, Canada

Jeffrey Parsons, Memorial University of Newfoundland, Canada

ABSTRACT

UML is used as a language for object-oriented software design, and as a language for conceptual modeling of applications domains. Given the differences between these purposes, UML's origins in software engineering might limit its appropriateness for conceptual modeling. In this context, Evermann and Wand have proposed a set of well-defined ontological rules to constrain the construction of UML diagrams to reflect underlying ontological assumptions about the real world. The authors extend their work using a design research approach that examines these rules by studying the consequences of integrating them into a UML CASE tool. The paper demonstrates how design insights from incorporating theory-based modeling rules in a software artifact can be used to shed light on the rules themselves. In particular, the authors distinguish four categories of rules for implementation purposes, reflecting the relative importance of different rules and the degree of flexibility available in enforcing them. They propose distinct implementation strategies that correspond to these four rule categories and identify some redundant rules as well as some rules that cannot be implemented without changing the UML specification. The rules are implemented in an open-source UML CASE tool.

Keywords: CASE Tools, Ontological Foundations of Conceptual Modeling, Rule Categories, Rules, Unified Modeling Language (UML)

INTRODUCTION

Recently, a significant stream of information systems research has focused on developing and evaluating guidelines for conceptual modeling in information systems development. Much of that work has focused on justifying and evaluating prescriptions derived from ontology (e.g.,

Burton-Jones & Meso, 2006; Burton-Jones & Weber, 1999; Evermann & Wand, 2006; Gemino & Wand, 2005; Parsons & Wand, 2008). The primary research methodology used to evaluate ontologically derived modeling guidelines has been laboratory experimentation. Most prior work has evaluated the *usefulness* of modeling rules or guidelines by comparing the differences between following versus not following the rules, on measures of recall, comprehen-

DOI: 10.4018/jismd.2011010102

sion, and problem-solving among readers of diagrams (Gemino & Wand, 2004). In this paper, we propose a complementary approach for assessing modeling rules. Specifically, we propose to examine the *feasibility* of automating the enforcement of modeling rules in software and to highlight design decisions that need to be made in the course of incorporating rules into a modeling tool. This kind of evaluation can also provide insights on the consistency and potential usefulness of such rules.

We evaluate a set of ontologically grounded modeling rules proposed by Evermann and Wand (2001a, 2001b) and Evermann (2003) to constrain the use of the Unified Modeling Language (UML) for conceptual modeling. Although the origins and initial focus of the UML lie in the area of object-oriented software design, it has evolved into a language for *conceptual modeling* – representing aspects of a real-world domain for which a system is required (Evermann, 2003; Burton-Jones & Meso, 2006). In this context, the UML is used as a language for communicating between clients and developers in understanding and eliciting requirements, and also for documenting the outcome of analysis (Dobing & Parsons, 2006, 2008).

The UML was developed as a language for software design (Booch et al., 2005). As a result, extending UML for use in conceptual modeling has raised questions about its appropriateness for that purpose. More generally, the applicability of object-oriented modeling in the early development phases is controversial (Kobryn, 2002). In addition, the UML in its entirety is very complex (Siau & Cao, 2001). In speaking of earlier versions of UML, Kobryn noted that “although UML 1.x has enjoyed widespread acceptance, its shortcomings include: excessive size and gratuitous complexity” (Kobryn, 2002). UML 2.0 and later versions have increased this level of complexity.

Against this backdrop, several researchers have proposed approaches to constraining the use of UML to conform to certain underlying principles. Such work is grounded in the view that, since conceptual modeling involves representing aspects of the real world, *ontology*, the branch of

philosophy dealing with the nature and structure of the real world, is an appropriate foundation to guide UML-based conceptual modeling. Accordingly, researchers have proposed ontological rules that would constrain the construction of UML diagrams to ensure they properly reflect underlying ontological assumptions. If followed, these constraints effectively impose a method for using the UML for conceptual modeling that reduces the degree of freedom available to an analyst in creating a model. The outcome of following this method should be a model that is sound with respect to the ontological foundation on which the rules are based.

Two such approaches, based on alternative ontological foundations, are noteworthy. First, the OntoUML language (Guizzardi, 2005) proposes an approach to improving ontological fidelity. OntoUML is based on foundational ontological concepts based on an ontological theory originating in the General Formalized Ontology (GFO) that underlies the General Ontological Language (Degen et al., 2001), and which integrates what later became known as the Unified Foundational Ontology (UFO) (Guizzardi & Wagner, 2010). A complementary tool has been developed to supporting the underlying ontological foundation (Benevides & Guizzardi, 2009). Second, using a foundational ontology based on Bunge (1977, 1979), Evermann and Wand (2001a, 2001b) and Evermann (2003) provide ontology-based guidance for using the UML in conceptual modeling, manifested as a set of rules and corollaries. However, they do not provide a tool to enforce these rules in constructing models.

In this paper, we focus on the approach proposed by Evermann and Wand. We believe that manually enforcing the set of complex rules they propose is impractical. Simply remembering and applying the rules would be a challenge in a real systems development project. Moreover, Evermann and Wand do not analyze their rules in the following senses: (1) How can the rules be implemented in a UML tool to support conceptual modeling? (2) Are there redundancies in the rules? (3) Are any of the rules already implied by the UML specification? (4) Are there

19 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/article/ontological-rules-uml-based-conceptual/51577

Related Content

Fostering Organizational Collective Intelligence From Human-Generative AI Interaction

Shoma Hino, Takuya Nakata, Sinan Chen, Sachio Saiki and Masahide Nakamura (2026). *International Journal of Software Innovation* (pp. 1-26).

www.irma-international.org/article/fostering-organizational-collective-intelligence-from-human-generative-ai-interaction/411903

"Multiple Sightseeing Scheduling System" Enabling Tourist Guidance Specialized for Time Performance

Kazuya Murata and Takayuki Fujimoto (2019). *International Journal of Software Innovation* (pp. 81-101).

www.irma-international.org/article/multiple-sightseeing-scheduling-system-enabling-tourist-guidance-specialized-for-time-performance/230925

Modeling Autonomic Systems: Review, Classification, and Research Challenges

Marwa Hachicha, Riadh Ben Halima and Ahmed Hadj Kacem (2022). *International Journal of Software Innovation* (pp. 1-22).

www.irma-international.org/article/modeling-autonomic-systems/303585

Contributions and Future Directions of Software Reviews

Yuk Kuen Wong (2006). *Modern Software Review: Techniques and Technologies* (pp. 281-290).

www.irma-international.org/chapter/contributions-future-directions-software-reviews/26909

Increasing the Accuracy of Software Fault Prediction using Majority Ranking Fuzzy Clustering

Golnoush Abaei and Ali Selamat (2014). *International Journal of Software Innovation* (pp. 60-71).

www.irma-international.org/article/increasing-the-accuracy-of-software-fault-prediction-using-majority-ranking-fuzzy-clustering/120519