Data & Knowledge Engineering 99 (2015) 72–104

Contents lists available at ScienceDirect

Data & Knowledge Engineering

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

Editorial

Ontological anti-patterns: empirically uncovered error-prone structures in ontology-driven conceptual models

Tiago Prince Sales, Giancarlo Guizzardi

Ontology and Conceptual Modeling Research Group (NEMO), Computer Science Department, Federal University of Espírito Santo (UFES), Vitória, ES, Brazil

ARTICLE INFO

Article history:
Received 28 March 2015
Accepted 9 June 2015
Available online 25 June 2015

Keywords:
Ontology-driven conceptual modeling
Ontological anti-patterns
OntoUML
UFO

ABSTRACT

The construction of large-scale reference conceptual models is a complex engineering activity. To develop high-quality models, a modeler must have the support of expressive engineering tools such as theoretically well-founded modeling languages and methodologies, patterns and anti-patterns and automated supporting environments. This paper proposes a set of Ontological Anti-Patterns for Ontology-Driven Conceptual Modeling. These anti-patterns capture error-prone modeling decisions that can result in the creation of models that fail to exclude unintended model instances (representing unintended state of affairs) or forbid intended ones (representing intended states of affairs). The anti-patterns presented here have been empirically elicited through an approach of conceptual models validation via visual simulation. The paper also presents a series of refactoring plans for rectifying the models in which these anti-patterns occur. In addition, we present here a computational tool that is able to: automatically identify these anti-patterns in user's models, guide users in assessing their consequences, and generate corrections to these models by the automatic inclusion of OCL constraints implementing the proposed refactoring plans. Finally, the paper also presents an empirical study for assessing the harmfulness of each of the uncovered anti-patterns (i.e., the likelihood that its occurrence in a model entails unintended consequences) as well as the effectiveness of the proposed refactoring plans.

1. Introduction

Information is the foundation of all rational decision-making. Without suitable *Information Systems*, individuals, organizations, communities and governments can neither systematically take optimal decisions nor understand the full effect of their actions. We use the term *Information System* here in a broader sense that includes also *Socio-technical Systems*. Moreover, we subscribe here to the so-called *Representation View* of information systems [1]. Following this view, an information system is a representation of a certain *conceptualization* of reality. To be more precise, an information system contains information structures that represent *abstractions* over certain portions of reality, capturing aspects that are relevant for a class of problems at hand. In this view, the quality of an information system directly depends on how truthful are its information structures to the aspects of reality it purports to represent.

In his ACM Turing Award Lecture entitled "*The Humble Programmer*" [2], E. W. Dijkstra discusses the sheer complexity one has to deal with when programming large computer systems. His article represented an open call for an acknowledgement of the complexity at hand and for the need of more sophisticated techniques to master this complexity. Dijkstra's advice is timely and even more insightful in our current scenario, in which semantic interoperability becomes a pervasive force driving and constraining the process of creating information systems in increasingly complex combinations of domains. More and more, information systems are created either by combining existing independently developed subsystem, or are created to eventually serve as components in multiple larger yet-

E-mail addresses: tpsales@menthor.net (T.P. Sales), gguizzardi@inf.ufes.br (G. Guizzardi).

http://dx.doi.org/10.1016/j.datak.2015.06.004

to-be-conceived systems. In this scenario, information systems engineering, in particular, and rational governance, in general, cannot succeed without the support of a particular type of discipline. A discipline devoted to establishing well-founded theories, principles, as well as methodological and computational tools for supporting us in the tasks of understanding, elaborating and precisely representing the nature of conceptualizations of reality, as well as in tasks of negotiating and safely establishing the correct relations between different conceptualizations of reality. On one hand, this discipline should help us in producing representations of these conceptualizations that are *ontologically consistent*, i.e., that represent a worldview that aggregates a number of abstractions that are consistent with each other. On the other hand, it should help us to make explicit our *ontological commitments*, i.e., to make explicit what exactly is the worldview to which we are committing. In summary, this discipline should help to produce concrete representation artifacts (models) of conceptualizations of reality that achieve the goals of *intra-worldview consistency* and *inter-worldview interoperability*.

The discipline to address the aforementioned challenges is the discipline of *Conceptual Modeling*. However, in order to do that, conceptual modeling languages, methodologies and tools must be informed by another discipline, namely, the discipline of *Ontology*, in philosophy. *Formal Ontology* has exactly the objective of developing domain-independent theories and systems of categories and their ties that could then be used to articulate conceptualizations in different domains in reality. More recently, the discipline of *Applied Ontology* has developed systematic and repeatable techniques for applying these theories in solving problems in concrete domains [3]. Given this essential role played by Ontology in this view of the discipline of Conceptual Modeling, we term it *Ontology-Driven Conceptual Modeling*.

The importance of Ontology as a foundation for Conceptual Modeling is not new in this discipline. There is an established tradition and a growing interest in using ontological theories for analyzing conceptual modeling languages as well as for proposing methodological guidelines for using these languages in the production of ontologically consistent models [1,4–6]. Ontology has been used not only as an analysis tool but also in the development of engineering tools such as conceptual modeling languages with explicitly defined and properly axiomatized metamodels [7], as well as computational environments supporting automated model verification, validation and transformation [8,9]. These are complexity management tools that are fundamental for addressing the challenge highlighted by Dijkstra's advice.

In the invited paper [10] companion to his keynote talk delivered at the 2014 edition of the International Conference on Conceptual Modeling (ER), the second author of the present paper makes the case for a particular set of complexity management tools needed for Ontology-Driven Conceptual Modeling, namely *Ontological Conceptual Patterns*, *Ontological Anti-Patterns*, and *Ontology Pattern Languages*. In the present paper, we focus on one of these conceptual tools, namely, *Ontological Anti-Patterns*.

An anti-pattern is a recurrent error-prone modeling decision [11]. In this paper, we are interested in one specific sort of anti-patterns, namely, model structures that, albeit producing syntactically valid conceptual models, are prone to result in *unintended domain representations*. In other words, we are interested in configurations that when used in a model will typically cause the set of valid (possible) instances of that model to differ from the set of instances representing *intended state of affairs* in that domain [12]. We name these configurations *Ontological Anti-Patterns*.

In this paper, we focus on the study of Ontological Anti-Patterns in a particular conceptual modeling language named OntoUML [7]. OntoUML is an example of a conceptual modeling language whose meta-model has been designed to comply with the ontological distinctions and axiomatization of a theoretically well-grounded foundational ontology named UFO (Unified Foundational Ontology) [13]. UFO is an axiomatic formal theory based on theories from Formal Ontology in Philosophy, Philosophical Logics, Cognitive Pyschology and Linguistics. OntoUML has been successfully employed in a number of industrial projects in several different domains, such as petroleum and gas, digital journalism, complex digital media management, off-shore software engineering, telecommunications, retail product recommendation, and government. Besides the modeling language itself, the OntoUML approach also offers a model-based environment for model construction, verbalization, code generation, formal verification and validation [8,9]. In particular, the validation strategy employed there makes use of an approach based on visual model simulation [9]. In this paper, we make use of this approach for eliciting *Ontological Anti-Patterns* in OntoUML.

This paper can also be seen as an extension of another publication presented at the same edition of the ER conference in which [10] was presented, namely [14]. In comparison to [10], this paper is much narrower in scope but much deeper in its investigation regarding ontological anti-patterns; in comparison to [14], we have expanded the paper in the following manner. Firstly, we have used an enlarged model repository with two added conceptual models. We here also present a much more detailed characterization of the repository and a more detailed analysis of an empirical study for uncovering the Ontological Anti-Patterns. Secondly, we present here a very detailed definition for the set of Anti-Patterns uncovered by this study, precisely defining their characteristics and structures and providing examples of their occurrences in the models in the repository. More importantly, we define here formal refactoring plans that have been implemented in a computational tool for automatically rectifying the potential modeling mistakes correlated with the presence of these anti-patterns. In this paper, we also present a novel industrial empirical study, which analyzes in depth the largest conceptual modeling in our repository with the goal of establishing: (i) the likelihood that an anti-pattern entails unintended consequences in the model; (ii) the effectiveness of our proposed refactoring plans. The paper also presents a newer implementation of the tool (in comparison to the one presented in [10] and [14]) that incorporates a model wizard for helping the modelers in applying the proposed refactoring plans.

The contributions of this paper are three-fold. Firstly, we contribute to the identification of Ontological Anti-Patterns in Conceptual Modeling. We do that by carrying out an empirical qualitative approach over a model benchmark of 54 OntoUML models. In particular, we employ the visual simulation capabilities embedded in OntoUML editor [8]. Secondly, after precisely characterizing these anti-patterns, we propose a set of refactoring plans that can be applied to the models in order to eliminate the possible unintended consequences induced by the presence of each of these anti-patterns. Finally, we present an extension to the OntoUML editor with a