Chapter 10
Refactoring and its Application to Ontologies

Jordi Conesa
Universitat Oberta de Catalunya, Spain

Antoni Olive
Universitat Politècnica de Catalunya, Spain

Santi Caballé
Universitat Oberta de Catalunya, Spain

ABSTRACT

Over the last years, a great deal of ontologies of many different kinds and describing different domains has been created, and new methods and prototypes have been developed to search them easily. These ontologies may be reused for several tasks, such as for increasing semantic interoperability, improving searching, supporting Information Systems or the creation of their conceptual schemas. Searching an ontology that is relevant to the users’ purpose is a big challenge, and when the user is able to find it a new challenge arises: how to adapt the ontology in order to be applied effectively into the problem domain. It is nearly impossible to find an ontology that can be applied as is to a particular problem. Is in that context where ontology refactoring takes special interest. This chapter tries to clarify what ontology refactoring is and presents a possible catalog of ontology refactoring operations.

INTRODUCTION

Lately, lots of ontologies of many different kinds and describing different domains have been created and new methods and tools have been developed to search them easily. Maybe the most prominent of these tools is Swoogle\(^1\), a web-based ontology search engine that searches over ten thousand available ontologies. Even though the facilities these new search technologies provide, finding an ontology that is relevant to a given purpose is a big challenge. And even, when we are able to find an ontology that is good enough to be reused, a new challenge arises: how to adapt the ontology in order to be applied effectively into the problem domain. It is nearly impossible to find an ontology that can be applied as is to a particular problem. In fact, once found an ontology

DOI: 10.4018/978-1-61520-921-7.ch010
Refactoring and its Application to Ontologies

it needs to be restructured to make it more usable (by deleting irrelevant terms, adding additional constraints…) and to adapt it to the conceptualization the user has in mind (by restructuring it to represent the same semantics but with a better structure) (Conesa J, 2008).

Evolution operations have played an important role in the software engineering field. Some of these operations aim to modify a program or a conceptual schema and can be used to add, modify or delete the functionalities of a system or to improve the quality of the system without modifying its semantics. These operations are called refactoring operations when they are applied to object-oriented artifacts or restructuring operations when applied to other kind of artifacts. Over the last decade, several researchers have attempted to use refactoring operations not only in software but at higher degrees of abstraction (Sunye, Pollet et al., 2001): databases, Unified Modeling Language (UML) models (OMG, 2003), Object Constraint Language (OCL) rules (OMG, 2003), and more shyly in ontologies. But not only refactoring operations allow for modifying the structure of something while keeping its semantics, other technologies propose similar evolution operations that can be considered as refactoring operations, such as some schema evolution operations of, program slicing operations…

Ontology refactoring is the process of modifying the structure of ontologies but preserving their semantics (Conesa J, 2008). Refactoring has successfully applied to several domains (Mens and Tourwe, 2004), but has not applied effectively to ontologies yet. The two main lacks of the application of refactoring to ontologies is the lack of an unambiguous and agreed definition of ontology refactoring and a catalog of the refactoring operations that makes sense for refactoring ontologies. Such agreed catalogs are present within other fields such as software refactoring and database refactoring.

We believe that the refactoring work of other fields, in combination with other techniques such as program restructuring (Griswold and Notkin, 1993) or schema transformation (Batini, Ceri et al., 1992), may be used to successfully define ontology refactoring and identify a catalog of its potential operations within the context of ontologies. Therefore, the goals of this chapter are: 1) to define clearly what a refactoring is, 2) to analyze various techniques for restructuring programs and schemas in order to identify in what cases these techniques or their operations can be useful for ontology refactoring, and 3) to present an exhaustive catalog of ontology refactoring operations that includes, after being adapted, operations from refactoring in other fields (software, databases and UML models mostly) and from other schema transformation techniques.

In the next section, the reader will find a brief history of existing restructuring techniques and how they have evolved to become the refactoring techniques we have today. Section 3 defines ontology refactoring taking into account the definitions of refactoring in other fields. Later, a catalog of ontology refactoring operations is presented. Such a catalog has been created by adapting refactoring operations from other techniques or from the refactoring of other fields. Finally, section 5 concludes the chapter presenting our conclusions and lessons learnt.

FROM PROGRAM REESTRUCTURING TO REFACTORING PASSING THROUGH SCHEMA TRANSFORMATION

In the last decades, several operations with common purposes but different uses have been defined using different names: refactoring operations (Oppyke, 1992; Moore, 1996; Roberts, Brant et al., 1997; Fowler, 1999; Ambler, 2003), restructuring operations (Eick, 1991; Batini, Ceri et al, 1992; Assenova and Johannesson, 1996; Halpin, 2001), program transformation operations (Griswold and Notkin, 1993; Griswold, Chen et al., 1998),