Chapter II

The Object-Oriented Design Knowledge Ontology

Javier Garzás, Oficina de Cooperación Universitaria (OCU) S.A., Spain

Mario Piattini, University of Castilla - La Mancha, Spain

Abstract

It has been a long time since the object-oriented (OO) paradigm appeared. From that moment, designers have accumulated much knowledge in design and construction of OO systems. Patterns are the most refined OO design knowledge. However, there are many others kinds of knowledge than are not yet classified and formalized. Therefore, we feel it necessary to define ontology in order to structure and unify such knowledge; a good understanding of practical experience is crucial to software engineers. Therefore, this chapter proposes an ontology for object-oriented design knowledge.
**Introduction**

Since Simula 67 up until the present day, knowledge related to the construction of object-oriented (OO) systems has evolved significantly. Nowadays, due to experience acquired during years of investigation and development of OO systems, numerous techniques and methods that facilitate their design are available to us.

By the middle of the 1990s the first catalogue of patterns was published by Gamma, Helm, Johnson, and Vlissides (1995). The application of patterns in OO design was consolidated, among others, by the work of Coad (1992), Gamma et al. (1995), Buschmann, Meunier, Rohnert, Sommerlad, and Stal (1996), Fowler (1996), and Rising (1998).

However, more knowledge exists apart from that related to patterns, and this other knowledge is frequently “hidden.” Moreover, now the exclusive use of patterns is not sufficient to guide a design, and the designer’s experience is necessary to avoid overload, non-application or the wrong use of patterns due to unawareness, or any other problems that may give rise to faulty and counteractive use of the patterns. In summary, when patterns are used, several types of problems may occur (Wendorff, 2001): difficult application, difficult learning, temptation to recast everything as a pattern, pattern overload, deficiencies in catalogues (search and complex application, high dependence of the programming language, and comparatives), and so on.

In this sense, we need others’ chunks of knowledge such as principles, heuristic, patterns, best practices, bad smells, refactorings, and so on. Nevertheless, there is much uncertainty with the previous elements, and these elements have never been studied as a whole. Its compatibility has been studied nor does a method based in this knowledge exist.

In order to improve OO designs, using all OO design knowledge in a more systematic and effective way, we have defined an ontology, which unifies principles, heuristics, best practices, and so on, under the term of “rule”; the ontology show the relationship among these “rules” and patterns and refactorings. We have also defined an improved OOD process, which takes into account this ontology and the OOD knowledge.

Moreover, we present in this chapter an empirical evaluation of this approach. The empirical validation is based on Prechelt, Unger, Philippson, and Tichy (1997); Prechelt, Unger, Tichy, Bössler, and Votta (2001); and Wohlin, Runeson, Höst, Ohlson, Regnell, and Wesslen (2000). This controlled experiment is ascertain if the usage of the ontology for OOD knowledge really improves the OOD process, helping in the detection of defects (rules violated) and solutions (patterns).