Enhancing UML Extensions with Operational Semantics
Behaviored Profiles with Templates

Arnaud Cuccuru, Chokri Mraidha, Francois Terrier, and Sebastien Grard
CEA, LIST, Gif-sur-Yvette, F-91191, France
{arnaud.cuccuru, chokri.mraidha, francois.terrier, sebastien.gerard}@cea.fr

Abstract. The objective of the ongoing OMG standard about a foundational UML subset semantics (fUML) is twofold: providing operational semantics for a UML subset, and ease unambiguous and automatic model exploitations. Its impact could however be limited if usual UML profiling practices do not evolve. Profiles are the traditional way to specialize UML semantics and handle semantic variation points. However, they are usually defined in a way that only informally addresses the semantic issue, potentially limiting the benefits that fUML could bring in UML based methodologies. UML profiling practices must evolve: we propose to explicitly encapsulate operational semantics into stereotype operations, and provide a way to intuitively handle semantic variation points through template parameters. We illustrate the usage of these mechanisms and demonstrate their potential benefits. We also show that no UML metamodel modifications are required to support them, so that their implementation in L3-compliant UML tools is straightforward.

1 Introduction

Since the earliest days of the Unified Modeling Language and right up to its current version (2.1.1), the most fundamental criticism made by the ”models” community has been its lack of semantics. The informal nature of UML’s semantic description (i.e. natural language) inevitably generates ambiguities, often making it difficult to agree on the meaning of a given model. These ambiguities involuntarily leave the door open to multiple and potentially contradictory interpretations of a same model. In [6], Steve Cook thus talks about the ”cognitive semantics” of UML, as opposed to the ”objectivist semantics” of a well-defined programming or modeling language. This drawback limits the impact that ”pure” UML could have on model-driven engineering, where (semi-)automatic model exploitations

---

1 This work has been performed in the context of the Usine Logicielle project (www.usine-logicielle.org) of the System@tic Paris Region Cluster. This project is partially funded by the ”Direction Gnrale des Entreprises of the French administration”, the ”Conseil Rgional d’le de France”, the ”Conseil Grral des Yvelines”, the ”Conseil Grral de l’Essonne” and the ”Conseil Grral des Hauts de Seine”.

© Springer-Verlag Berlin Heidelberg 2007
(and therefore automatic and unambiguous interpretations of the underlying semantics) are required (the most obvious example of this is code generation).

UML profiles are usually used to overcome these undesirable side-effects. A profile basically provides users with a specialization of the UML 2 metamodel, through stereotypes and constraints definitions, thus enabling to clarify and/or specialize its semantics. Often associated with dedicated methodologies, profiles have proven useful for many modeling domains. The Accord/UML methodology [13] [14] we proposed in previous works is a good example of UML profiling efficiency, namely for real-time systems engineering. More generally, for a given profiled model, a given stereotyped model element can be "recognized" by the tool chain associated with the methodology, and interpreted in a proper way. OCL constraints associated with the definition of profile elements are typically used to guarantee that the underlying model repository will be well-formed and interpretable for the tool chain. A similar usage of UML profiles is of course also recommended to fix the explicitly identified semantic points for which potential interpretations may vary (i.e. fix the "Semantic Variation Points").

More pragmatically, UML profiling mechanisms have achieved their primary objective: providing UML users with an intuitive means for customizing UML, from both syntactic and semantic terms. However, as in the case of the UML 2 metamodel, semantic descriptions associated with a profile remain "cognitive" (i.e. expressed in natural language), and are not an explicit part of the profile definition: "Automatic semantic interpretation" (as mentioned above) must therefore be "hard-coded" in one way or another in the tool chain. At a time when the OMG officially recognizes a need for formalizing UML 2 semantics (see its RFP of August 2005 [9], on the Semantics of a Foundational Subset for Executable UML Models), we are convinced that the role of profiles in UML-based methodologies should be enhanced. The offshoot of this RFP (which is further described in section 2.3) should be an executable semantics description (in operational style) for a UML subset. UML profiles (traditionally used to define specialized UML semantics) would thus directly and explicitly influence operational semantics: UML profiles should explicitly encapsulate such operational semantic description and provide users with an explicit means for varying semantics where needed (i.e. for identifying and/or fixing semantic variation points).

In section 2 we first clarify the meaning of "semantics", and provide an overview of existing works on formalizing UML semantics. This focuses specifically on the OMG standard being developed for UML semantics and motivates our own "operational profiles" approach. To better explain how traditional use of profiling mechanisms can be enhanced, section 3 gives a recap of UML profiling basics, by defining a simple profile for specializing UML state machines. In the rest of the paper, we use this basic example to explain and illustrate how the role of UML profiles could be enhanced. In section 4 we elaborate on the basic profile by first showing how operational semantics can be encapsulated in stereotype definitions, then how template parameters help to explicitly handle semantic variation points. We also show why and how these original mechanisms