Formal Foundations for the Unified Modeling Language

A. Funes, A. Dasso, D. Riesco, G. Montejano, R. Uzal

SEG
http://sel.unsl.edu.ar/
Universidad Nacional de San Luis
Ejército de los Andes 950
D5700HHW San Luis
Argentina
{afunes, arisdas, driesco, gmonte, ruzal}@unsl.edu.ar
Tel.: +54 (0) 2652 42 4027 ext. 251
Fax: +54 (0) 2652 43 0224

Abstract

We present in this work an outline of an ongoing research line in the framework of the Software Engineering Group (SEG) at the National University of San Luis. We describe here the previous work carried out by the group in formalizing UML using RSL, as well as the current and future work in the matter.

Introduction

The Unified Modeling Language (UML) [1][21][25] is a graphical language for modeling and specifying software systems and it consists of a set of constructs common to most object-oriented languages. However, although UML notations are easily communicated, their semantics are informal and –consequently– they can be ambiguous. There are an important number of theoretical works that deal with the integration of graphical notations and mathematically precise formalisms (see [3][13][17][26][23][15][5]). A good classification of the different proposals to carry out this integration can be found in [22]. Several efforts have been conducted to formalize the semantics and the syntax of different parts of UML (see [19][16][14][8][9][6][2][7]). Only a few use the RAISE Specification Language (RSL) as formal basis (see [24][18][4][10]).
RSL is a formal specification language [11], which receives its name from the RAISE method. The RAISE (Rigorous Approach to Industrial Software Engineering) [12] consists of a number of techniques and strategies for doing formal development and proofs. Its language, RSL, is a wide spectrum specification language. It allows the use of different styles of specifications: applicative or imperative; sequential or concurrent; direct (explicit) or axiomatic (implicit); algebraic (with abstract data types) or model-oriented (with concrete data types).

Past, Current and Future Work

In [24] we presented a first proposal for the semantics of a class using RSL. In this work only the basic syntactic elements were considered. Also binary associations were treated. All of them given through particular examples.

In [18] we gave a semantics in RSL for association class, aggregation and generalization through concrete examples.

In [4] we presented a proposal for using the translation of a class diagram applying the semantics given in the two previous works ([18] and [24]) as an initial applicative specification for the RAISE method.

In [10] an exhaustive and generic treatment of classes, n-ary associations, dependency, aggregation and composition, generalization, template classes, abstract classes and other syntactic elements were considered. Their semantics were given taking into account the integration with the other syntactic elements in the context of an integrated frame. The integration of the specifications was given by the semantics of the class diagram. Note that the semantics for associations, generalizations, association classes and aggregation were not given using the same RSL constructions than in [18] and [24]. Furthermore in [10], the semantics were given for the general cases. From this analysis a set of templates could be abstracted. These templates were used as the basis for the implementation of an automatic tool, which translate a class diagram into a RSL
specification. The templates as well as the tool are also reported in [10]. Another important point in this work is that the syntax for the RSL-translatable class diagrams was formally specified using also RSL.

Current and future work includes the study of the integration between an specification obtained by following [10] and dynamics diagrams in UML. Currently, we are working specifically in an extension using state machines in UML to model the dynamic aspects of a class. This allows to model the life of an object, that is, the different states in which an object of the class can be and the transitions among these states. An interpretation in RSL for these states machines given with the class diagram will be embedded in the entire specification obtained for the class diagram.

References


