A New Approach in Object-Based Knowledge Representation: The AROM System

Michel Page[1, 2], Jérôme Gensel[1, 2], Cécile Capponi[4], Christophe Bruley[1], Philippe Genoud[1, 3], Danielle Ziébelin[1, 3], Daniel Bardou[1, 2], and Véronique Dupierris[1]

[1]Action Romans, INRIA Rhône-Alpes, ZIRST, 655, av. de l'Europe, Montbonnot
38334 Saint Ismier Cedex - France
[2]Université Pierre Mendès-France, BP 47,
38040 Grenoble Cedex 9, France
[3]Université Joseph Fourier, Grenoble, BP 53,
38041 Grenoble Cedex 9, France
[4]Labo. d'Informatique de Marseille, Univ. de Provence, 163 av. de Luminy,
13288 Marseille Cedex 9, France
arom@inrialpes.fr

Abstract. This paper presents AROM, a new object-based knowledge representation system. AROM is original in two ways. Firstly, in addition to classes, knowledge representation in AROM uses associations for describing links between classes instances. Secondly, AROM integrates an algebraic modeling language (AML) for describing operational knowledge (equations, constraints and queries) in a declarative way. AROM comes as a platform for knowledge representation and manipulation. It includes an interactive modeling environment, a Java API, an AML interpreter for processing queries and solving sets of equations, and a tool for consulting and editing knowledge bases on the Web.

1 Introduction

Most object knowledge representation systems fall into two categories: description logics (DL) and object-based knowledge representation systems (OBKRS). DL systems like KL-ONE [1] represent concepts using formal syntax and semantics; emphasize is put on complexity study of both subsumption and concept classification, which is the main inference mechanism of these systems. OBKRS (KRL [2], KEE [3], for instance) are declarative systems for describing, organizing and processing large amounts of knowledge; they offer various inference mechanisms such as classification, procedural attachment, default values, filters, etc.

In this paper, we present AROM (Associating Relations and Objects for Modeling), a new OBKRS which departs from existing systems in two ways. Firstly, in addition to classes which are usually found in existing OBKRS, AROM uses *associations* similar to those found in UML [4] for organizing links between objects having common structure and semantics. Secondly, in addition to classical inference mecha-

L. Monostori, J. Váncza, and M. Ali (Eds.): IEA/AIE 2001, LNAI 2070, pp. 113-118, 2001.

nisms found in other OKBRS, AROM integrates an algebraic modeling language (AML) for expressing operational knowledge in a declarative manner. This language allows one to write constraints, queries, numerical and symbolic equations between the various elements composing a knowledge base (KB).

AROM comes as a platform with a Java API, an interactive graphical environment for designing and manipulating KBs, an AML interpreter for processing queries and solving sets of equations, and tools for consulting and editing KBs on the Web.

The paper is organized as follows. Knowledge representation in AROM is presented in section 2 along with an example. The AROM platform is detailed in section 3. Related work and conclusion are presented is section 4.

2 Knowledge Representation in AROM

Many data and knowledge engineering methods involve two notions for modeling a domain: entity and relation. In object-oriented models like UML, three kinds of relation are considered: specialization/generalization, association and aggregation (a particular case of the latter being composition). An association denotes a group of similar links between instances of two or more, distinct or not, classes.

Most OBKRS handle specialization, and sometimes aggregation (and/or composition). Associations usually have no specific representation; they are implemented either by *link-attributes* or by reification of associations as classes.

A link-attribute is a slot typed by a class and its value references one (or several) instance(s) of this class. Several OBKRS like SMECI [5], implement binary associations with two inverse link-attributes, and integrate mechanisms for maintaining their consistency. This technique has two problems. It works fine only for binary associations [6], and it does not easily take into account attributes of associations (*i.e.* properties attached to links; for instance, the wedding date of two married persons).

Using reification, associations are considered as classes whose attributes correspond either to links between objects or to specific properties. This enables associations with an arity greater than two to be described. However, since no distinction is made between classes and associations, operations or mechanisms specific to either of them cannot be defined.

Neither link-attributes nor reification is thus satisfactory for representing associations. Furthermore, using both techniques within a KB can only lead to inconsistencies and lack of extensibility. This has led us to promote associations as first-class entities for knowledge representation, on equal ground with classes.

2.1 Knowledge Representation in AROM Using Classes and Associations

Classes and objects. A class in AROM describes a set of objects sharing common properties and constraints. Classes are descriptive: they provide a set of necessary yet not sufficient conditions for membership. Each class is characterized by a set of properties called variables and by a set of constraints.