Objects and Classification
A Natural Convergence

Marianne Huchard\textsuperscript{1}, Robert Godin\textsuperscript{2}, and Amedeo Napoli\textsuperscript{3}

\textsuperscript{1} LIRMM, 161 rue Ada, 34392 Montpellier cedex 5, France
marianne.huchard@lirmm.fr

\textsuperscript{2} Université du Québec à Montréal, Département d’informatique C.P.8888, Succ.CV, Montréal (Québec), Canada H3C 3P8
godin.robert@uqam.ca

\textsuperscript{3} LORIA – UMR 7503 (CNRS – INRIA – Universités de Nancy), B.P. 101, 615 rue du jardin botanique, 54602 Villers-lès-Nancy Cedex, France
amedeo.napoli@loria.fr

\textbf{Abstract.} Classification is a central concept in object-oriented approaches such as object-oriented programming, object-oriented knowledge representation systems (including description logics), object-oriented databases, software engineering and information retrieval. Nevertheless, research works on classification have often been carried out separately within these different approaches, and they have not always been precisely confronted and connected. The goal of the workshop was to confront these complementary viewpoints on classification, to exhibit and discuss commonalities and differences within these approaches.

1 Introduction

Object-oriented approaches are mainly based on a proximity between real world entities and their computer representation. This leads primarily to the well-known “classes”\textsuperscript{1}, “instances”, and specialization relationships implemented by inheritance. More generally, a common trend in object-orientation is to reify all the considered items, and classify them. Unfortunately, research works that deal with classification in object-oriented approaches are spread over different fields such as object-oriented programming, object-oriented knowledge representation systems (including description logics\textsuperscript{[10, 21]}), object-oriented databases, software engineering and information retrieval\textsuperscript{[15, 6]}. Furthermore, they are concerned by various entities like use cases, classes, prototypes, methods, patterns, and they have different goals, like designing, reasoning, indexing and programming. Due to these differences, many research works are carried out separately, although they have a common concern.

The workshop intended to put together and to confront all these complementary viewpoints on classification, in order to bring on the foreground common questions, and to provide a basis to discuss and share classification techniques.

\textsuperscript{1} When ambiguous, denoted by \textit{oo-classes} for “classes in object-oriented systems”.

© Springer-Verlag Berlin Heidelberg 2000
This document reports and synthetizes the contributions (position papers and discussions) with regard to the following points:

- main trends of the workshop,
- definitions of classification (as “classification” is polymorphic),
- objectives followed in the classification process,
- classified entities,
- context of the classification (relatively to the life-cycle of the software),
- structure of the classifications produced (partitions, trees, lattices, etc.),
- systems, tools, techniques in general.

2 An Overview of the Contributions

The workshop was appropriately introduced by [w10], which recalled the Aristotle’s ideas on classification and how they concern object-oriented approaches: either because they are already present although hidden, or because they could be beneficial.

Apart this first contribution, two main trends were represented.

The first, or “software classification”, is concerned with classification problems applied to software engineering. It not only aims at managing classes and hierarchies, but it also extends the notion of classification to all software artifacts: oo-class hierarchy evolution [w9], reuse of persistent instances [w3], tagging software [w5], logic predicates for reasoning on the software [w7] or generating code [w12], use case management [w14].

The second trend is closer to knowledge representation concerns. It focuses more on the notion of oo-class hierarchy, develops specific models (classes described by disjunctions [w8], prototypes [w1], a UML-like model [w4], graphs [w2, w6]), and techniques as instance classification [w8, w4], conceptual clustering [w2, w13, w6], or genetic algorithms [w11].

3 The Meaning of “Classification”

The term “classification” usually covers both the process and the result of:

- class discovery
  - either by dividing a set of entities; classes obtained in this way can be called “extensional classes”; pure extensional classes (sets) are rather uncommon in object-oriented approach;
  - or by associating a description to a set of entities; these classes are “intensional classes”; in this case, the intension can be given either a priori, for example by a human actor from his knowledge of the domain, or a posteriori, when it is deduced from the analysis of a set of objects. Such intensional classes also have an extensional counterpart: the objects created from or covered by the intension.
- class organization (“class classification”) into specialization structures (trees, lattices, etc.),
- associating a class to an entity (“entity/instance classification”).