The essence of reflection

Hassan Aït-Kaci

ILOG, S.A.*
Research & Development

hak@ilog.fr

December 29, 2002

Abstract

In programming languages, the difficulty usually associated with *reflection* (also known as *introspection*) is the cyclic nature of the services it procures, and the sempiternal potential inability to manage completeness without falling into the trap of paradox. However, reflection is simply the ability to describe and/or interpret a given structure in itself. In other words, reflection is the, or rather some, ability *to go "meta"*—to self-examine oneself and, possibly, affect one's behavior by directing the so-examined part of one's structure into taking live action contingent upon one's own self-analysis.

Formalizing this phenomenon has occupied the minds of several generations of mathematicians, philosophers, and other logicians, and, though more recently, of a few computer scientists.[1]

The modern proliferation of information submerges us. This information must now be processed by software. Such software's efficiency and reliability in sorting out this information is greatly enhanced when data is self-described. In other words, with all this data, we desperately need *meta*data—data about data. XML lends itself easily to representing graph-like data on the web. Recently, a formalism—the RESOURCE DESCRIPTION FORMAT (RDF)—has been proposed to provide an XML vocabulary to describe XML resources (and thus RDF is representable in RDF). This *reflective* power of RDF poses the question of whether well-foundedness and correctness of processing RDF descriptions can be at all studied, let alone garanteed—how can one be formally convinced of the *processability* and *correctness* of an RDF specification? However, all this technology is being developed mainly haphazardly today as markets dictate pressure to have *anything* that *kinda* works, *a-s-a-p*—please.

Be that as it may, there has indeed been, not so far from the rumors of web frenzy, a beautiful body of mathematics that provides a wonderful set of tools and techniques to explain and effectively use such mixing of levels of discourses: *viz.*, *Category Theory*. It has been put to great uses to formalize higher-order logic simply and effectively. It should suffice to take on the formalization of reflective web structures, *methinks*.

I propose to formalize this concept of structural reflection using Category Theory.[2] The object of this document is to introduce our idea and reflect upon its potential.[3]

*B.P. 85, 9, rue de Verdun, 94253 Gentilly Cedex, France.

[1] If granted gracefully be that this lot possessed such objects as *er... minds*—or rather wishing that at least so did a category among them (who might then be called "sharp arrows" for actually using those objects they do possess!)... ;-)

[2] Seriously! And the idea is so simple, that it is surprising that it has not been proposed (more loudly) before by anyone—it *must* have!

[3] ... and whether it may point to effective ways of providing autonomous web agents the power to enable intelligent protocols of comuunication based on self-described structure and semantics.

Introduction

The reader will find in [1], at the end of Section 2. of Part 0, Exercises 2 and 3 referring to Definition 1.2 in the book's previous Section (1). Encapsulated in those, lies *the essence of reflection*.

These exercises are recalled next for the reader's appreciation. The words "source" and "target" are as defined by Definition 1.2; namely:

> **Definition 1.2** A *graph* (ususally called a *directed graph*) consists of two classes: the *arrows* or (*oriented edges*) and the class of *objects* (usually called *nodes* or *vertices*) and two mappings from the class of arrows to the class of objects, called *source* and *target* (also often called *domain* and *codomain*).

source

$$\boxed{\textit{Arrows}} \rightrightarrows \boxed{\textit{Objects}}$$

target

> One writes '$f : A \to B$' (or better yet, '$A \xrightarrow{f} B$'), for "the *source* of f is A and the *target* of f is B" (or, using a deliberately "object-oriented" syntax, for "$f.source = A$ and $f.target = B$").

And now, here are the exercises:

> **Exercise 2** If $\mathbf{2}$ is the category $\cdot \to \cdot$ (with identity arrows not shown), show that the objects of $\mathcal{A}^{\mathbf{2}}$ are essentially the arrows of $\mathcal{A}$ and that "source" and "target" may be viewed as functors $\delta, \delta' : \mathcal{A}^{\mathbf{2}} \rightrightarrows \mathcal{A}$.

> **Exercise 3** If $F, G : \mathcal{A} \rightrightarrows \mathcal{B}$, show that a natural transformation $t : F \to G$ is essentially the same as a functor $t : \mathcal{A} \to \mathcal{A}^{\mathbf{2}}$ such that $\delta t = F$ and $\delta' t = G$.

The reader not familiar with (naive) Category Theory is referred to Section 1 for a few basic notions and terminology necessary to parse the otherwise straightforward exercises.

1 Naive Category Theory

What is a *category?* There are two easy ways to coinceive of one. One fairly accurate approximation—the *syntactic way*—sees a category essentially as the *reflexive transitive closure of a graph*. Another, as accurate, approximation—the *semantic way*—sees a category as a *formal system of computation*. Formal tools such as these are precious in Computer Science where *everything* is a graph, and *every process* is a computation.

DEFINITION 1.1 (DEDUCTIVE SYSTEM) *A* deductive system *is a graph such that all its objects A, B and arrows f, g verify the conditions of Figure 1.*

DEFINITION 1.2 (CATEGORY) *A* category *is a deductive system such that...*

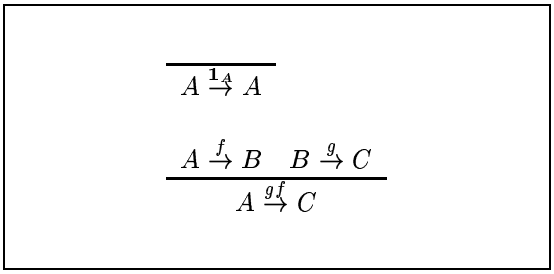
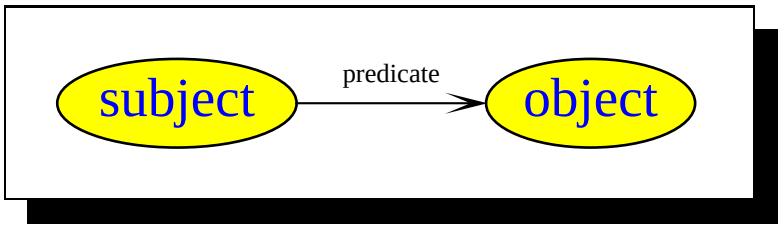


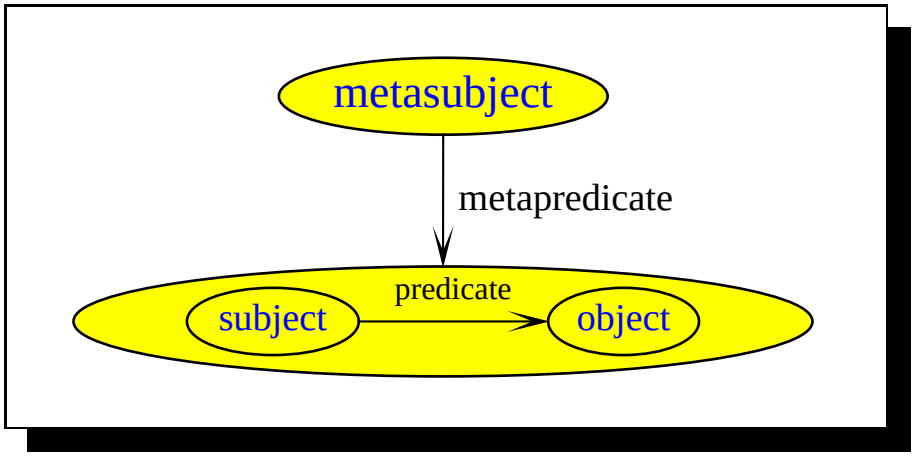
Figure 1: Conditions for deductive system

2 RDF

RDF is a notation for meta-description of data (metadata) using (edge- and node-) *labelled graphs*. The basic building block is a "triple" labelled by "resources"—*i.e.*, URI's. A triple consists of a resource (the subject), linked through a resource (the predicate) to another resource (the object). A triple states that the subject has a property, denoted by the predicate, whose value is the object:



The information carried by a triple is called a "statement." RDF statements can be reified and be denoted as resources—hence, RDF's metalinguistic nature:



RDF uses XML for its serialized syntax. RDF enables the definition of vocabularies which can be shared over the Web thanks to XML namespaces (*e.g.*, Dublin Core). RDF Schema (RDFS) is a meta-description of RDF in RDF; it defines a vocabulary for RDF.

References

[1] Joachim Lambek and Phil Scott. *Introduction to Higher-Order Categorical Logic*. Cambridge University Press, 1986.