



The Model OFL

Pierre Crescenzo

► To cite this version:

Pierre Crescenzo. The Model OFL. 12th Workshop for Ph.D. Students in Object-Oriented Systems lors de la conférence ECOOP 2002 (16th European Conference on Object-Oriented Programming), Jun 2002, Malaga, Spain. hal-01308798

HAL Id: hal-01308798

<https://hal.science/hal-01308798>

Submitted on 28 Apr 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THE MODEL OFL

Pierre Crescenzo

Address: Laboratoire I3S (UNSA/CNRS)
Projet OCL
2000, route des lucioles
Les Algorithmes, Bâtiment Euclide B
BP 121
F-06903 Sophia-Antipolis CEDEX (France)
E-Mail: Pierre.Crescenzo@unice.fr
Web: <http://www.crescenzo.nom.fr/>

Keywords: Customised Relationships, Meta-Programming, Hyper-Generic Parameters

The model *OFL* (Open Flexible Languages) [Cre01] aims to describe the main object-oriented programming languages (such as *Java* [GJSB00], *C++* [Str97], *Eiffel* [Mey92], ...) to allow their evolution and their adaptation to specific programmer's needs. To reach this goal, *OFL* reifies all elements of an object-oriented programming language in a set of components. Thus classes, methods, expressions, messages, and so on are the *OFL*-components and are integrated in a specific MOP (Meta-Object Protocol) which allows to extend the set of entities needed for the reification of both languages and user applications.

The meta-programmer creates a language by selecting adequate *OFL*-components in pre-defined libraries. (S)he can also specialise a given *OFL*-components in order to generate one dedicated to some specific uses. To separate the default *OFL*-components of the *OFL*-components created for a specific language, we call *OFL*-Atom the default one.¹

Classes are reified by *OFL*-components. Take the example of *Java*. We have *ComponentJavaClass*, *ComponentJavaInterface*, *ComponentJavaArray*, ... An originality of *OFL* is that relationships are also reified. So, we have for *Java*: *ComponentJavaExtendsBetweenClasses*, *ComponentJavaExtendsBetweenInterfaces*, *ComponentJavaImplements*, ... A more complete list of *OFL*-components for *Java* is given in [CCL02].

To facilitate the creation of an *OFL*-component, *OFL* provides some meta-components, called *OFL*-concepts. So, we have a *ConceptRelationship* and a *ConceptDescription*². Thus, *ConceptDescription* is equivalent to a meta-meta-class. In each concept, a set of parameters gives the meta-programmer powerful possibilities to create or adapt an *OFL*-component.

References

- [CCL02] A. Capouillez, P. Crescenzo, and P. Lahire. Le modèle OFL au service du méta-programmeur - Application à Java. In *LMO'2002 (Langages et Modèles à Objets)*. Hermes Science Publications, *L'objet : logiciels, bases de données, réseaux*, volume 8, numéro 1-2/2002, January 2002. also Research Report I3S/RR-2001-04-FR (Laboratoire d'Informatique, Signaux et Systèmes de Sophia-Antipolis), <http://www.crescenzo.nom.fr/>.
- [Cre01] P. Crescenzo. *OFL : un modèle pour paramétrer la sémantique opérationnelle des langages à objets - Application aux relations inter-classes*. PhD. Thesis, Université de Nice-Sophia Antipolis, December 2001. <http://www.crescenzo.nom.fr/>.
- [GJSB00] J. Gosling, B. Joy, G. Steele, and G. Bracha. *The Java Language Specification*. The Sun Microsystems Press Java Series. Sun Microsystems, June 2000. <http://java.sun.com/docs/books/jls/>.
- [Mey92] B. Meyer. *Eiffel: The Language*. Object-Oriented Series. Prentice Hall, 1992. <http://www.eiffel.com/doc/>.
- [Str97] B. Stroustrup. *The C++ Programming Language*. Addison-Wesley Publishing Co., 3rd edition, 1997. <http://www.research.att.com/~bs/3rd.html>.

¹In other words, *OFL*-Atoms are supplied by the model, other *OFL*-components, created for a specific language, are not.

²The word *description* has been chosen to represent classes and all entities which look like classes, such as interfaces.