

A Classification of Stereotypes for Object-Oriented Modeling Languages¹

Beitrag für den
7-ten Workshop des GI-Arbeitskreises GROOM der GI-Fachgruppe 2.1.9
4.-5. April 2000, Universität Koblenz-Landau

Martin Glinz
Universität Zürich
<http://www.ifi.unizh.ch/~glinz>

Stefan Berner
FJA, Zürich
stefan.berner@fja.com

Stefan Joos
Robert Bosch GmbH, Stuttgart
stefan.joos@de.bosch.com

Introduction

Since 1996, various attempts have been made to unify different object-oriented modeling languages. As a result of this endeavor, two languages have been developed: the Unified Modeling Language UML [4] and the Open Modeling Language OML [2]. Both UML and OML introduce a distinctive new feature: they allow users to extend or even to modify the base language in order to adapt the language to specific situations or needs. The language construct that is used to implement this feature is called a *stereotype*.

The notion of stereotypes was introduced by Rebecca Wirfs-Brock [5]. Her principal idea is to provide a secondary classification for objects: stereotypes classify objects according to their use, independently of the primary classification by classes and class inheritance. UML and OML both generalize Wirfs-Brock's notion from a secondary classification to a concept that allows for general extensions of the base language. A stereotype in UML and OML can add new properties to elements of the underlying language or can modify existing ones.

In our contribution we discuss the UML/OML kind of stereotypes in a general context of object-oriented modeling languages. However, the notion of stereotypes is not limited to object-oriented approaches. Hence, we define a stereotype as follows.

DEFINITION. A *stereotype* in a modeling language is a well-formed mechanism for expressing user-definable extensions, refinements or redefinitions of elements of the language without (directly) modifying the metamodel of the language.

Stereotypes provide language users with limited metamodeling capabilities without giving them (direct) access to the metamodel of the language. This is a very powerful mechanism. However, as is frequently the case with powerful features, stereotypes have both a bright and a dark side. On the bright side, stereotypes can lead to modeling languages which are more flexible and expressive and which are better adaptable to specific problem types and application domains. On the dark side, unsystematic or excessive use of stereotypes can lead to a proliferation of incompatible dialects of a language and can make a language both difficult to handle and to understand. Thus, unconsidered use of stereotypes can do more harm than good.

Designing good stereotypes would be considerably easier if we had a proper design methodology or at least a set of design guidelines. However, neither a methodology nor guidelines presently exist. In order to develop guidelines (and finally arrive at a methodology), a deeper understanding of the nature of stereotypes and of the implications of their use is necessary.

We contribute to the solution of the stereotype design problem. We introduce a classification of stereotypes according to their expressiveness – that means according to their potential to alter the syntax and semantics of the base language. Every class represents a related set of purposes for using stereotypes and has specific stereotype design requirements associated with it. The extent to which stereotypes alter a language ranges from mere notational variations to a complete redefinition of the language. Simple stereotypes typically change the notation (i.e. the concrete syntax and/or visual representation) of a language element and/or introduce new features that serve as a kind of 'structured comment'. Powerful stereotypes, on the other hand, impose semantic restrictions on the added language elements or even redefine the semantics of language elements. This can go up to a complete syntactic and semantic redefinition of the base language. We classify stereotypes according to their expressiveness into four categories (Figure 1). Note that our classification forms an inclusion hierarchy, not a partition. The more powerful categories include all the potential of the less powerful ones.

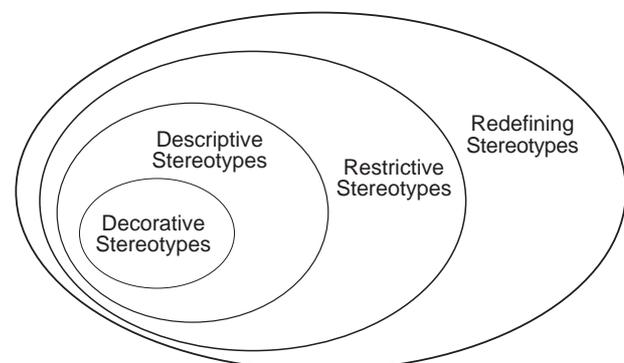


Figure 1. Classification of stereotypes according to their expressive power

¹ This is a condensed version of the paper we presented at the 2nd International Conference on the Unified Modeling Language [1].

Decorative Stereotypes

A *decorative stereotype* modifies the concrete syntax of a language element and nothing else. Decorative stereotypes vary the way in which a language element is visually represented. They do not introduce any essential additional information or new concepts into the base language. The represented model and the essence of the language that expresses the model remain unchanged.

Decorative stereotypes are typically used to adapt the notation of a language or of some of its elements to some given standard or to personal preferences.

Descriptive Stereotypes

A *descriptive stereotype* modifies the abstract syntax of a language element and defines the pragmatics of the newly introduced element. The semantics of the base language remains unchanged. Additionally, a descriptive stereotype may modify the notation (the concrete syntax) of the stereotyped language element.

Descriptive stereotypes are on a pure syntactic level. They do not impose any semantic restrictions on the extended or modified syntax. The persons who use a descriptive stereotype must rely on the description of the stereotype pragmatics in order to use and interpret the stereotype properly. When compared with simple comments, descriptive stereotypes have the advantage of a well-defined syntactic structure, which makes some formal checking and analyses possible.

Secondary classifications (in the sense of Wirfs-Brock's stereotypes [5]) and standardized annotations are typical applications of descriptive stereotypes.

Restrictive Stereotypes

A *restrictive stereotype* is a descriptive stereotype that additionally defines the semantics of the newly introduced element.

Typically, the semantics impose compulsory structural restrictions on the newly introduced language element – hence the name restrictive stereotype. A restrictive stereotype does not change the semantics of the base language – it only extends it. The concept of restrictive stereotypes allows for a fully formal definition of the stereotype. However, in practice the definition will frequently be semi-formal only. Restrictive stereotypes are first-class members in the language they are added to. They have the same expressive power and can be defined with the same degree of rigor as the elements of the base language themselves. Restrictive stereotypes are typically used to add missing features to some elements of a language, to strengthen weak features or to introduce a metalanguage on top of a given language.

Redefining Stereotypes

A *redefining stereotype* redefines a language element, changing its original semantics. Concerning syntax, a redefining stereotype behaves in the same way as a restrictive one.

With decorative, descriptive and restrictive stereotypes, instances of the stereotype remain valid instances of the stereotyped language element. For redefining

stereotypes, this is no longer true. A redefining stereotype can introduce a new language element that is no longer related to the element of the base language that it stereotypes.

Using redefining stereotypes, deep and radical changes can be imposed to a language. New language concepts can be introduced. In its extreme, redefining stereotypes can embed another language in a given base language.

The early versions of UML heavily used redefining stereotypes. For example, Use Case and Actor were stereotypes of Class. In the more recent versions, all predefined redefining stereotypes have become ordinary UML metaclasses [3]. However, it is still possible to create redefining stereotypes in UML.

Strengths and Weaknesses of Stereotypes

The main general advantage of stereotypes is that they make a language *flexible* and *adaptable*. When used properly, they improve a modeling language, making models easier to express and to understand.

On the other hand, there are two general drawbacks and risks.

- Working with stereotypes requires effort for designing and maintaining them, and for training all the users and readers of a stereotyped language how to use and interpret the stereotypes.
- Badly designed stereotypes and the use of an excessive number of stereotypes both turn the potential benefit of stereotypes into its contrary: they harm a language, making it more difficult to use and to understand.

The potential benefit of stereotypes as well as the drawbacks and risks grow with increasing power of the stereotypes. More details are given in [1].

Guidelines for Stereotype Design

As mentioned earlier, designing stereotypes is a demanding task and the potential benefit of stereotypes heavily depends on taking the right design decisions.

From our experience with stereotypes we have assembled a preliminary set of guidelines for stereotype design. We include some general advice here. More details may be found in [1].

- Define a stereotype policy and enforce it: who (identify roles) has the right to define stereotypes of which category (e.g. according to our classification) for which purpose and with which scope (e.g. individual, project, department-wide, and company-wide).
- Make sure that every stereotype is properly defined and documented.
- Have every stereotype definition reviewed prior to using it.
- Avoid the creation of stereotypes when its scope is below the level of a project.
- Whenever you define a new stereotype, make sure that you will be able to maintain it in the scope and for the duration of its use.
- Make the stereotype definitions available to all people who need to know them and train these people how to apply and how to interpret them, respectively.
- Define less stereotypes and apply the existing ones more uniformly and with a wider scope.

Summary and Conclusions

Stereotypes are powerful, but care and experience is required to harness this power. Our classification helps to better understand the nature of stereotypes and to control their application. Every category in this classification represents a typical kind of applications for stereotypes. Using decorative and redefining stereotypes both is highly problematic. Variations of the concrete syntax or the style of representation as well as a fundamental redefinition of the semantics of the base language should be done very restrictively only. Hence, descriptive and restrictive stereotypes are the most important ones in practice. Stereotypes from these two categories are especially useful to

- make models more expressive by augmenting them with additional information in a standardized way,
- compensate for deficits and weaknesses in a given modeling language in order to make it better adapted to some classes of problems or to given domains.

Stereotypes are no silver bullet. Their application does not automatically result in 'better' models. They increase the complexity of the base language and intro-

duce overhead for definition, training and maintenance. So, before introducing language extensions or modifications based on stereotypes, always make sure that these are clearly beneficial.

References

- [1] Berner, S., Glinz, M., Joos, S. (1999). A Classification of Stereotypes for Object-Oriented Modeling Languages. *Proceedings of the Second International Conference on the Unified Modeling Language*, Fort Collins, Oct 1999. Berlin, etc.: Springer. 249-264.
- [2] Firesmith, D., Henderson-Sellers, B. H., Graham, I., Page-Jones, M. (1998). *Open Modeling Language (OML) – Reference Manual*. SIGS reference library series. Cambridge, etc.: Cambridge University Press.
- [3] OMG (1999). *OMG Unified Modeling Language Specification Version 1.3*. OMG document ad/99-06-08. <ftp://ftp.omg.org/pub/docs/ad/99-06-08.pdf>
- [4] Rumbaugh, J., Jacobson, I., Booch, G. (1999). *The Unified Modeling Language Reference Manual*. Reading, Mass., etc.: Addison-Wesley.
- [5] Wirfs-Brock, R., Wilkerson, B., Wiener, L. (1994). Responsibility-Driven Design: Adding To Your Conceptual Toolkit. *ROAD 1*, No. 2; (July-August 1994). 27-34.