Object Morphology – A Protean Generalization of Object-Oriented Paradigm

Zbyněk Šlajchrt

Department of Information Technologies,  
Faculty of Informatics and Statistics,  
University of Economics, Prague, Czech Republic

**Abstract.** The abstract should summarize the contents of the paper and should contain at least 70 and at most 150 words. It should be set in 9-point font size and should be inset 1.0 cm from the right and left margins. There should be two blank (10-point) lines before and after the abstract. This document is in the required format.

**Keywords:** We would like to encourage you to list your keywords in this section.

1 Introduction

Philosophical Background

Goals

Literature Review

Methods

Chosen methods reflect the character of the problem, which lies on the border between basic and applied science:

Case studies (one structural, one behavioral)

Evaluation of three OOP languages (Java, Scala, Groovy)

Analysis of problematic aspects identified in the case studies

Generalization of the existing paradigms and methods (OOP, UML, Liskov principle)

Verification through development of P-o-C and its application on a set of problematic scenarios (discussed further)

Theoretical Foundations of OM

This section presents the theoretical foundations of Object Morphology. With respect to its limited scope, this paper introduces the concepts in a rather brief manner with emphasis on intuition. Wherever it is possible, the explanations are accompanied by simple illustrations.

Morph Models

When modeling a protean phenomenon in the outer world, the modeler focuses on the description of the features exhibited by individual instances of the phenomenon. The individual instances may or may not share a set of common features, and even a single instance may not exhibit the same set of features during its existence. A typical example of such a phenomenon is a butterfly and its development; there is little or nothing in common between individual developmental stages, except the butterfly’s identity.

In Object Morphology, the concept of a *fragment* is used to describe a feature of a modeled phenomenon. A fragment can be likened to a potentially complex attribute with possible behavior. In contrast to fragments, attributes hold data only and do not define any behavior, nor do they possess identity. On the other hand, a fragment should not be mistaken for a part of a phenomenon. The reason is that a fragment, in contrast to a part, may constitute or contribute to the identity of its phenomenon, while a part possesses its own identity distinct from its owner’s identity.

A collection of fragments describing a given phenomenon at certain moment is called *alternative*. The individual fragments in the alternative may be *passive* or *active*. A passive fragment’s behavior cannot be influenced by the presence of other fragments, while the behavior of an active fragment can be overridden by other fragments (it is an analogy to final/non-final classes). An instance of an alternative is called a *morph*, which is a representation of a modeled phenomena instance (an analogy to an object).

The collection of all alternatives describing a given phenomenon is called a *morph model*. A morph model is in fact an expression of the concept of the phenomenon. A morph model may describe a phenomenon whose instances are either *immutable* or *mutable*. The immutable instances exhibit the same features until they cease to exist. The individual immutable instances differ in the composition of the fixed feature set, while the diversity of this set among the instances is governed by the morph model. On the other hand, the mutable instances may undergo many mutations during their lifetime, while the only possible mutations are dictated by the morph model.

There may exist one or more disjoint groups of alternatives sharing a certain collection of fragments. These common fragments are called *prototypes* of the morph model and the group sharing a given prototype is called the prototype’s *attractor*. It should be remarked that there may exist more ways to partition a morph model to attractors. Then, it is on the modeler to select the best partitioning.

Examples

The rest of this section presents a couple of examples illustrating the-above mentioned concepts. The models in the examples are expressed using the R-Algebra, a mathematical formalism developed in the thesis, used to construct morph model expressions. This formalism introduces two operations – join (.) and union (|) – and defines rules for using them.

The model in formula (**1**) models 6 basic human emotions. The vertical bar is the union operator delimiting mutually exclusive emotions (in terms of this model). The individual emotions are fragment

|  |  |
| --- | --- |
| Person = Person | (**1**) |

|  |  |
| --- | --- |
| p = Person(“Peter”) | (**2**) |

|  |  |
| --- | --- |
| Emotion = Joy | Surprise | Fear | Sadness | Disgust | Anger | (**1**) |

|  |  |
| --- | --- |
| e = Joy(0.7) | (**2**) |

|  |  |
| --- | --- |
| Emotion = (Joy | Surprise | Fear | Sadness | Disgust | Anger)*2* = Joy | Surprise | Fear | Sadness | Disgust | Anger | Joy.Surprise | Joy.Fear | Joy.Sadness | Joy.Disgust | Joy.Anger | Surprise.Fear | Surprise.Sadness | Surprise.Disgust | Surprise.Anger | Fear.Sadness | Fear.Disgust | Fear.Anger | Sadness.Disgust | Sadness.Anger | Disgust.Anger | (**3**) |

|  |  |
| --- | --- |
| e = Joy(0.6).Surprise(0.2) | (**4**) |

|  |  |
| --- | --- |
| Trip = Plan.(1 | Move | Left | Right).(1 | Trip) | (**3**) |

Plan is a non-stackable => Plan. Plan = Plan

The morphing strategy will allow only transitions from n to n+1alts with the same sub-alternative:

|  |  |
| --- | --- |
| t = Plan(“Prague”,”Prčice”) t = Plan(“Prague”,”Prčice”).Move(1,0) t = Plan(“Prague”,”Prčice”).Move(1,0).Left(10)  t = Plan(“Prague”,”Prčice”).Move(1,0).Left(10).Move(2,0)  t = Plan(“Prague”,”Prčice”).Move(1,0).Left(10).Move(2,0).Right(0.5) | (**6**) |

All fragments have immutable state

|  |  |
| --- | --- |
| Compass(N) = (N.~S | W.~E | S.~N | E.~W)*N* | (**7**) |

The N coefficient determines the resolution of the compass.

|  |  |
| --- | --- |
| c = N.N.E | (**8**) |

Stateless

The model of a traveller’s mind:

The compass points at the destination. Its presence influences the path. The emotion reflects various factors such as: the distance from the destination, the weather condition (outer condition example) etc. Its presence may also influence the choice (development) of the path.

|  |  |
| --- | --- |
| Traveller = Person.Emotion.Path.Compass(2) | (**9**) |

Fragment, are not parts, fragment instances – stubs, individual emotions as fragments, unit fragment

Alternative = a list of fragments, might be a prototype or its attractor (deviations, mixed cases)

Model = a list of alternatives

Morph = an instance of an alternative

R-Algebra, EmoModel^2, the single-fragment alternatives are prototypes, the others (mixed emotions) are attractors

Wrappers, stackable fragments, the left-right model

Inverse fragment, the cardinal directions model

Morphing Strategies

Recognizer

Re-morphing

Morph References

LSP Generalization

Prototypical Analysis

Reference Implementation

Applications

Onto UML

DCI

Square-Rectangle Problem

Modeling Facial Emotions

Further Work

Theoretical aspects: identity, R-Algebra for wrappers, recursive models

Practical aspects: a type system based on the generalized LSP, quantitative aspects (i.e. not only what-it-is, but also how-much-it-is)

Discussion