OBJECT MORPHOLOGY

Mgr. Zbyněk Šlajchrt

Motivation: Stiffness of OOP

- Inability to model real-world protean phenomena using OOP

Goal: Rethink OOP

- Develop theoretical foundations of OM
- Formulate basic tenets of OO analysis w.r.t OM
- Implement a proof-of-concept OM application platform

Introduction

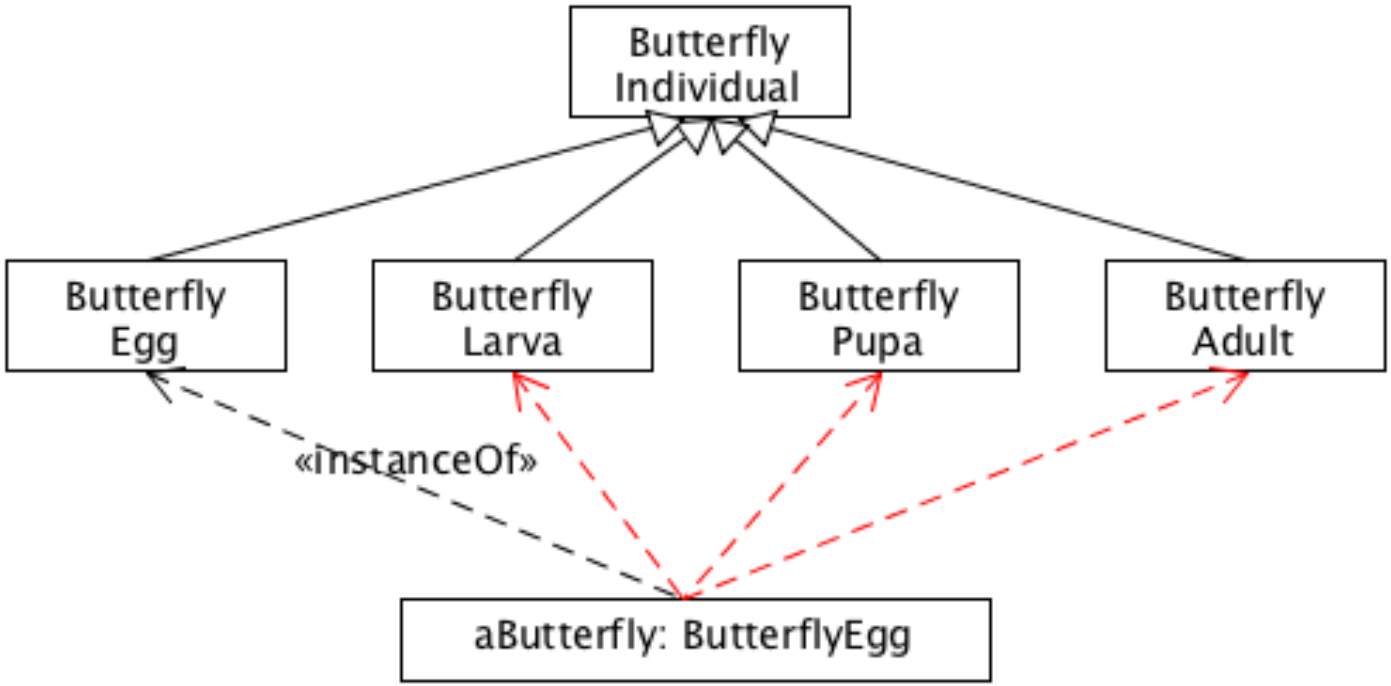
Real-world phenomena examples and philosophical background

Butterfly in Real World

- One individual, 4 forms (egg, larva, pupa, adult)

https://www.pinterest.com/pin/185562447118575246/

Butterfly in OOP



- One class for each form
- But an object cannot change its class! (The class is inherent to the object)

A Class Is Not A Form

- Classes cannot model forms
- Class ButterflyLarva

  represents all butterfly larvae,

  not a specific stage in

  the butterfly's life-cycle

http://www.dkfindout.com/us/animals-and-nature/insects/butterfly-life-cycle

Game Categorization

- Card games, board games, ball games, children games such as Ring-a-Ring-O'Roses etc.
- Each game has something in common with some other games
- It is however impossible to find one common attribute
- Family Resemblance (L. Wittgenstein)

Aristoteles vs. Wittgenstein

- Aristoteles:
  - All instances of a concept share the same set of attributes
  - Concept = Set
  - Aristotelian (set) logic, rules
- Wittgenstein:
  - One instance of a concept resembles another one in one or more attributes
  - Concept = A network of overlapping phenomena
  - Family Resemblance

Prototype Theory

- Eleanor Rosch (*1938)
- Experiment: She asked students to rate various items as a good example of furniture
- Graded categorization: Some items are “more” furniture than others
- The grade: the distance of the item to the imaginary **“furniture prototype”**
- prototype = attractor
- Departs from Aristotelian conceptual framework
- A concept may have more disconnected prototypes (games, colors perception of some tribes)

Solution

Overview of Object Morphology

Morph Models

- Replacing Classes By **Morph Models**
- A morph model all possible forms of the phenomena
- Constructed through **prototypical analysis** of the phenomena
- Class is a special case of a morph model with one alternative

Theoretical Foundations

- The concept of **Abstract Recognizer**
  - A device "perceiving" a phenomena and classifying it using the built-in morph model
- R-Algebra
  - A mathematical formalism used to define morph models
- Generalized Liskov Substitution Principle
  - Analyses when one morph model may substitute another
- UML Extension
  - New elements and stereotypes introduced

Example: Circle/Ellipse

Model = Ellipse * (1 | Circle)

Alternatives:

1. Ellipse
2. Ellipse * Circle

Prototypical Analysis

- Understanding Phenomena
- Identifying Prototypes
- Property Analysis
- Morph Model Construction
- Binding Properties To Context
- Morphing Strategy Construction
- Creating Recognizer

Morpheus: Proof-of-Concept

- Written as an extension of Scala compiler
- Available on GitHub:
  https://github.com/zslajchrt/morpheus
- Used to model facial expressions on humans

Conclusion

- Contributions
  - Case studies
  - **R-Algebra**
  - UML extension
  - **LSP Generalization**
  - Prototypical analysis
  - **Morpheus**
- To Do:
  - To unify the terminology
  - Formal modifications, final clean-up

The End

Thanks for your attention!