Institut für Systemsoftware

O.Univ.Prof. Dr. Hanspeter Mössenböck

JOHANNES KEPLER
UNIVERSITÄT LINZ
Netzwerk für Forschung, Lehre und Praxis

Dynamic Mixins for Java

Master thesis for Kerstin Breiteneder
Matr.-Nr.: 0755946
Email: kerstin.breiteneder@aon.at

A *mixin* is a class whose methods and fields can be added to a *target class* without declaring a super type relationship. Dynamic mixins allow programmers to add and remove mixins of a target class at run time. A prototype implementation of dynamic mixins for Java is part of the Dynamic Code Evolution VM (http://ssw.jku.at/dcevm/mixins/). The goal of this thesis is to extend and improve the existing prototype in order to increase the usability of dynamic mixins for Java.

In particular, the improvements should include:

- **Reflection:** The meta-information of a class can be retrieved via a java.lang.Class object. For classes with mixins, however, the information about a target class may be confusing because of artificial renaming or newly introduced methods when rewriting the bytecodes of a target class. Also, the current API does not allow users to get information about the mixins of a target class. The thesis should define a framework for reflectively accessing the mixin information of a target class.
- **Life Cycle:** Initialization and disposal of mixins are important if mixins carry state. Therefore, mixin constructors and destructors should be implemented that are invoked when a mixin is added or removed. The constructor of a mixin must be executed for every existing instance of the target class as well as for newly created target class instances. The destructor must be executed when an instance of the target class is garbage collected, or when the mixin is removed from the target class.
- **Dynamic AOP:** The thesis should discuss possible applications of dynamic mixins for dynamic aspect-oriented programming. The implementation should be enriched by providing the possibility to let mixin methods replace methods of the target class based on name patterns. Also, additional annotations for specifying the mixin semantics (e.g. that the mixin method should be executed before or after a shadowed target method) should be introduced.

The work progress should be discussed with the supervisor every two weeks. Please note the guidelines of the Institute for System Software when preparing the written thesis.

Supervisor: Dipl.-Ing. Thomas Würthinger
Start: January 2011