Abstract

Aspect Interfaces: Towards Separate Type-checking of Aspect-oriented Programs with Inter-type Declarations

Manabu Toyama[1,a] Tomoyuki Aotani[2,b] Eric Bodden[3,c]
Hidehiko Masuhara[2,d] Éric Tanter[4,e]

Presented: June 3, 2013

This presentation discusses about separate type-checking of aspect-oriented programs with inter-type declarations. Inter-type declarations is one of the mechanisms that achieve aspect-oriented programming and supported by AspectJ. It enables aspects to introduce methods and fields to classes and interfaces. Separate type-checking is a method that, given a set of modules and their interfaces, ensures the program consists of the modules is type safe if each module is type safe with respect to the interfaces. Although inter-type declarations is useful, it makes aspects and classes hard to be type-checked separately. In other words, if a class depends on the introduced methods and/or fields (e.g., invoking or accessing them), its type safety cannot be assured without aspects because the signatures of the introduced methods or fields are absent. In this presentation, we propose to achieve separate type-checking of classes and aspects with inter-type declarations using Aspect Interfaces, which are contracts between aspects and classes. Aspect interfaces describe the signatures of the introduced methods. Introduced classes, which are the target classes of the inter-type declarations, specifies aspect interfaces so that the classes provide the signatures of introduced methods along with the methods within the classes. Then each calls to the introduced methods can be then type checked normally. Introducing aspects, which provide the bodies of the introduced methods, specifies aspect interfaces that contains the signatures of the introduced methods so that it is easy to check every introduced method in the aspect interfaces is actually provided by the introducing aspect. When composing aspects and classes (i.e., weaving aspects), it is enough to ensure type safety to check whether or not at least one introducing aspect exists for each aspect interface along with the usual linking-time type-checking.

[1] The University of Tokyo, Meguro, Tokyo 153–8902, Japan
[2] Tokyo Institute of Technology, Meguro, Tokyo 152–8552, Japan
[3] EC SPRIDE, Technische Universität Darmstadt, Darmstadt, Germany
[4] Universidad de Chile, Santiago, Chile
[a] touyama@graco.c.u-tokyo.ac.jp
[b] aotani@is.titech.ac.jp
[c] bodden@acm.org
[d] masuhara@acm.org
[e] etanter@dcc.uchile.cl