Modelling the Package/Access Concept of Java in Isabelle

Norbert Schirmer
Technische Universität München, Instititut für Informatik
`http://www.in.tum.de/~schirmer`

November 23, 2001

Our contribution to Verificard is to build and investigate a type system, a operational semantics and an axiomatic semantics for the JavaCard Source Language. The starting point of our work is the Bali Project, which already captures large parts of the Java source language (e.g. classes, interfaces, arrays, object initialisation). But the package concept and the access modifiers are not part of Bali. This is the content of this talk. The access modifiers determine the access restrictions and visibility of class and interface types and their members. Since safety and security properties of Java are based on the bare language itself the access modifiers are the main means to protect data.

During the effort to model the package/access concept formally in the theorem prover Isabelle some intrinsic problems occurred for various reasons. The Java Language Specification is very imprecise and ambiguous in many points concerning the package/access concept. The Java implementations in the Java SDK don't exactly follow the specification but seem to capture the intended behaviour. Some inconsistencies where detected and reported to SUN. They are accepted as Bugs by SUN. In some cases the competing ideas of information hiding - through to packages - and spreading implementations - through to subclass inheritance - don't fit together very well. Sometimes it is not obvious if we should expect the information to be hidden inside the package or to cross the boundaries of the package due to subclassing. Some of these problems are omitted in JavaCard by putting more constraints on the placing of access modifiers. To illustrate these problems, I want to give some Java examples and show that it is often not intuitive what the intended semantics should be. Since our formal model is closer to the implementation of Java in the SDK then to the Java Language Specification, I want to give an overview of the model and discuss some differences and clarifications to the Java Language Specification. The main points are:

- In the Java Language Specification, the access modifiers are used both to describe the legal access to a member and to determine which members are inherited. We have to distinguish both cases.

- Throughout the Java Language Specification accessibility of members is defined uniformly for all kinds of members (class, interface, field, or method). But due to overriding methods actually have weaker access restrictions.

- The notion of overriding is different in the type system (conform with the Java compiler) and the operational semantics (conform with the JVM).