

**Fragen, Antworten und Kommentare zur aktuellen Vorlesung**

Frage: Was sind Java Agents?

Java Agents sind normale Java-Programme, die als Besonderheit den Byte-Code neu geladener Klassen verändern können. Generell liegen Klassen in systemunabhängigen Byte-Code vor und werden bei der ersten Nutzung geladen und passend zur genutzten Maschine umgesetzt. Dies erklärt auch die langen Startzeiten beim Start von Java-Programmen. Mit dem genutzten Class-Loader können Java-Agents zusammenarbeiten und dabei das Verhalten verändern. Ein Ansatz ist z. B., dass am Anfang eines jeden Methodenaufrufs Programmcode ergänzt wird, mit dem die Aufrufinformationen in der Konsole ausgegeben werden. Die Eingriffe können noch viel weitergehen und das gesamte Verhalten beeinflussen. Der Ansatz wird z. B. von JPA-Werkzeugen genutzt, die eine enge teil-automatisierte Anbindung an eine Datenbank ermöglichen. In Richtung C und C++ ist der Ansatz mit der Nutzung von Pre-Compilern verwandt.

Frage: Ist Promela nicht einfach eine weitere Programmiersprache für verteilte Systeme?

Antwort: Nein. Genauer benutzt die Spezifikationssprache klassische Elemente der imperativen Programmierung, unterstützt dazu aber explizit den Nichtdeterminismus. Dadurch beschreibt eine Spezifikation immer eine Menge von Spezifikationsabläufen. Diese Menge wird beim zentralen Ziel der Verifikation betrachtet.

Wichtig ist dabei zu erkennen, dass der Simulator nur ein Entwicklungshilfsmittel ist, um Flüchtigkeitsfehler bei der Entwicklung der Spezifikation zu finden.

Frage: Warum kann ein Model Checker nicht immer ein Gegenbeispiel liefern?

Antwort: Das hängt von der Anforderung ab, ob dies überhaupt möglich ist. Ist die Anforderung „bei allen Ausführungen etwas soll immer gelten“ ist klar, was ein Gegenbeispiel liefert. Bei der Anforderung „es soll eine Ausführung geben, die folgendes erreicht“ kann es kein sinnvolles Gegenbeispiel, wohl aber das Positivbeispiel geben.