

## **Erhard Weinell: Ein Rahmenwerk für operationale Spezifikationsprachen**

**Promotion:** RWTH Aachen, Fakultät für Mathematik, Informatik und Naturwissenschaften

**Erstgutachter:** Prof. Dr.-Ing. M. Nagl, RWTH Aachen

**Zweitgutachter:** Prof. Dr.rer.pol. M. Jarke, RWTH Aachen

**Datum der Prüfung:** March 10, 2011

**Veröffentlichung:** Shaker Verlag, Reihe AIB-SE 8, 300pp, 2011

### **Kurzfassung:**

Die modellgetriebene Softwareentwicklung lebt von der Grundidee, komplexe softwaretechnische Sachverhalte kompakt aber dennoch präzise zu beschreiben. Dabei erstellt der Entwickler zunächst eine abstrakte Spezifikation eines Softwaresystems, die sich an fachlichen Konzepten der jeweiligen Anwendungsdomäne orientiert. Details zur Umsetzung dieser Spezifikation in Quellcode, wie etwa die zu verwendende Ausführungsplattform, werden erst in späteren Entwicklungsphasen festgelegt. Auf dieser Basis kann die spätere Umsetzung in ein lauffähiges System in wesentlichen Teilen automatisiert erfolgen. Für den Entwickler ergibt sich aus dem hohen Abstraktionsgrad der Spezifikation ein echter Mehrwert, der sich insbesondere in einer kürzeren Entwicklungszeit sowie einer leichteren Wartbarkeit des spezifizierten Systems niederschlägt.

Zur Durchsetzung der modellgetriebenen Softwareentwicklung sind zwei Faktoren entscheidend, die jeweils durch geeignete Entwicklungswerkzeuge und Infrastrukturlösungen abgedeckt werden müssen: Eine geeignete Beschreibungsform wird benötigt, die sowohl dem Kompaktheits- als auch dem Präzisionsanspruch gerecht wird. Nur auf diese Weise kann ein Entwickler eine verdichtete und für weitere Projektbeteiligte leicht verständliche Spezifikation erstellen. Ferner ist die weitere Verarbeitung der erstellten Spezifikationen zu berücksichtigen, die entweder generativ durch Ableitung von Quellcode, oder interpretativ als Konfiguration einer Programmbibliothek realisiert werden kann. Auf diese Weise lassen sich sowohl statische Anteile einer Spezifikation wie etwa dem Datenschema, als auch dynamische Anteile wie die Analyse und Veränderung entsprechender Instanzdaten verarbeiten.

Abseits spezieller Anwendungsgebiete und Forschungslösungen konnte sich bislang allerdings nur die Modellierung statischer Anteile in der Praxis etablieren. Dies liegt zum Teil an der Notwendigkeit fallspezifische Ausdrucksmittel zur Verhaltensspezifikation definieren zu müssen, um den erwünschten Abstraktionsgrad zu wahren. Die Abbildung dieser speziellen Ausdrucksmittel, entweder auf Quellcode einer Programmiersprache oder einem Interpreter, gestaltet sich jedoch häufig

komplexer als im statischen Fall. Der dadurch entstehende Realisierungsaufwand lässt die Spezifikation von Systemverhalten als zu kostspielig und somit nicht rentabel erscheinen, weshalb hierzu häufig auf herkömmliche Programmierung zurückgegriffen wird.

Im Rahmen der vorliegenden Arbeit wird ein Ansatz vorgestellt, um die Realisierung verhaltensorientierter Spezifikationsprachen zu unterstützen. Zentraler Aspekt ist hierbei eine sogenannte Kernsprache, die als kompakte Beschreibungsform über dem Abstraktionsniveau klassischer Programmiersprachen bereitgestellt wird. Konzeptionell basiert diese Sprache auf den im softwaretechnischen Umfeld langjährig erprobten Graphgrammatiken. Da jedoch unmöglich Anforderungen aller denkbaren Anwendungsszenarien bei der Ausgestaltung dieser Kernsprache berücksichtigt werden können, wird insbesondere eine leichte Anpassbarkeit und Erweiterbarkeit sichergestellt. So sind die angebotenen Sprachkonstrukte strikt im Hinblick auf ihre Zuständigkeiten getrennt, so dass Erweiterungen nach Möglichkeit lokal, das heißt mit möglichst wenigen Ergänzungen der ursprünglichen Definition umgesetzt werden können. Zudem wird ein breites Spektrum an möglichen Datenstrukturen unterstützt, die Graphgrammatiken auf komplexen Graphmodellen nutzbar machen.

Die der Kernsprache zugeordnete Ausführungsmaschine lässt sich leicht an fallspezifische Gegebenheiten, wie etwa der zugrundeliegenden Datenzugriffsschicht, anpassen. Als Abstraktionsebene kommt hierbei ein Nicht-Standard-Datenbanksystem für graphbasierte Daten zum Einsatz, das sich zum Adaptieren bestehender Plattformtechnologien nutzen lässt. Auch wird hierbei die technische Erweiterbarkeit der Kernsprache direkt berücksichtigt: So lassen sich Spracherweiterungen auf die Kernsprache zurückführen, um die bestehende technische Infrastruktur nutzen zu können. Alternativ kann auch die bestehende Implementierung der Ausführungsmaschine leicht erweitert werden.

Um die Werkzeugunterstützung von Spezifikationsprachen schneller entwickeln zu können, ist eine leichte Nutzbarkeit der Kernsprache zu gewährleisten. Hierzu wird eine Möglichkeit angeboten, durch automatisierte Transformationen eine verhaltensorientierte Spezifikationsprache in die bereitgestellte Kernsprache zu übersetzen. Dieser Übersetzungsformalismus berücksichtigt Spezifika der Kernsprache, um seine kompakte und intuitive Nutzung zu fördern. Dabei folgt die formulierte Übersetzung zwar einem bestimmten Standardverhalten, das jedoch fallweise durch den Entwickler mit eigenen Festlegungen überladen werden kann.

Die vorliegende Arbeit begegnet somit der Notwendigkeit, verhaltensorientierte Spezifikationen durch spezifische Ausdrucksmittel rasch und sei es zunächst für eine prototypische Erprobung nutzbar

zu machen. Hierfür bietet sich die vorgeschlagene Kernsprache an, da sie auf vergleichbarem Abstraktionsniveau angesiedelt ist, wie die hiermit zu realisierenden Spezifikationsprachen. Prototypische Entwicklungs- und Ausführungswerkzeuge erleichtern zusätzlich ihre Nutzung. Als Ergebnis entsteht somit ein Werkzeugsatz, der die Realisierung und Erprobung komplexer operationaler Spezifikationsprachen wirksam unterstützt.