Tagungsbericht

„Aspektorientierung“

Workshop der Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung der Gesellschaft für Informatik e.V. (GI) Paderborn, 3.-4. Mai 2001

Pascal Costanza, Universität Bonn
Günter Kniesel, Universität Bonn
Katharina Mehner, Universität Paderborn
Elke Pulvermüller, Universität Karlsruhe

Am 3. und 4. Mai fand in der Universität Paderborn der erste Workshop "Aspektorientierung" der Gesellschaft für Informatik statt. Der Workshop wurde von Katharina Mehner (Universität Paderborn), Prof. Mira Mezini (Universität Darmstadt), Elke Pulvermüller (Universität Karlsruhe) und Dr. Andreas Speck (Intershop Research) organisiert. Die Zielsetzung dieses Workshops war es, ein Forum für Gruppen im deutschsprachigen Raum zu schaffen, die auf dem Gebiet der aspektorientierten Programmierung (AOP), der aspektorientierten Modellierung und verwandten Gebieten arbeiten. Der Workshop hatte 21 Teilnehmer. Das Programm umfasste 10 Vorträge zu den folgenden Themenbereichen.

Modellierung und Rollen Der Beitrag *Separation of Concern mit Rollen, Subjekten und Aspekten* (Ralph Depke, Katharina Mehner, Universität Paderborn) untersucht, inwieweit diese Techniken additive und invasive Integration von Teilmodellen unterstützen, und stellt einen eigenen Ansatz vor, der beide Integrationsweisen erlaubt. Der ausgeklügelte Vorschlag zur *Rollenmodellierung* von Peter Werner (T-Nova, Deutsche Telekom) fasst Rollen als zusammengehörige Methoden und Variablen von Objekten auf und Aspekte als thematisch zusammengehörige Mengen von Rollen verschiedener Objekte. Der Beitrag *Separation of Concerns and Roles in the Object Coordination Nets Approach* (Holger Giese, Universität Paderborn) diskutiert die besonderen Probleme, die sich in einer offenen, verteilten Umgebung bei der Komposition verschiedener Dienste stellen, und schlägt eine explizite Komposition auf der Design-Ebene vor.

AOP und vordefinierte Kompositionsmechanismen Der Beitrag *Object-Oriented Composition is Tangled* (Klaus Ostermann, Siemens AG; Mira Mezini, Universität Darmstadt) zeigt, dass es wünschenswert wäre, einzelne Merkmale gängiger Kompositionsmechanismen in objektorientierten Sprachen unabhängig voneinander einsetzen zu können. *Concerning AOP and Inheritance* (Stefan Hanenberg, Rainer Unland, Universität Essen) diskutiert die Unterschiede zwischen den Konzepten der Vererbung und der aspektorientierten Erweiterung und schlägt einen Vererbungsmechanismus für Aspekte vor.

Integration und Konsistenz von Softwareeinheiten Der Beitrag *Unabhängige Erweiterbarkeit für Aspekt-Orientierte Systeme* (Michael Austermann, SCOOP GmbH; Pascal Costanza, Günter Kniesel, Universität Bonn) diskutiert die Gründe, warum Aspekte nicht ohne weiteres als Komponenten aufgefasst werden können, und schlägt eine erste Lösung für dieses Problem vor. *Concern-Integration mit JADE* (Mira Mezini, Michael Haupt, Universität Darmstadt) beschreibt den Mangel an Sprachkonstrukten für die Integration von Komponenten in objektorientierten Programmiersprachen und schlägt Erweiterungen in dieser Richtung für Java vor. *Combining Composition Styles in the Evolvable Language LAC* (Stephan Herrmann, Universität Berlin; Mira Mezini, Universität Darmstadt) ist ein weiterer Beitrag, der verschiedene Kompositionstechniken für Komponenten auf Grundlage der Skriptsprache Lua integriert.

Semantik In dem Beitrag *Semantics of Method Call Interception* (Ralf Lämmel, Universität Amsterdam) wird ein Sprachkonstrukt für das Abfangen von Methodenaufrufen als Grundlage für AOP vorgestellt, das die Eigenschaften der Typsicherheit und der unabhängigen Übersetzbarkeit der (objektorientierten) Basissprache erhält.

Anwendungsgebiete Der Beitrag *Separating Concerns in Product Lines Engineering* (Joachim Bayer, IESE Kaiserslautern) demonstriert, wie die Komplexität von Produktlinien reduziert werden kann, indem zum einen die im Domänenmodell unterschiedenen Anforderungen in den nachfolgenden Phasen als Design-Artefakte in Form von Views erhalten werden. Zum anderen werden die gemeinsam genutzten generischen Anteile einer Produktlinie explizit durch Entscheidungsmodelle instantiiert.

Im Anschluss an die Vorträge haben unabhängige Diskussionsrunden zu folgenden Themen stattgefunden.

Begriffsbildung und Klassifikation Eine Diskussionsgruppe beschäftigte sich mit begrifflichen Bestimmungen (z.B. von Rollen und deren Beziehung zu Aspekten und Concerns) sowie der Frage nach Kriterien für eine Klassifikation der verschiedenen Mechanismen für Separation of Concerns. Eine Klassifikation könnte helfen, die unterschiedlichsten Ansätze miteinander zu vergleichen und ein gemeinsames Verständnis zu entwickeln. Eine Reihe offener Fragen, wie z.B. mit Hilfe von AOP widersprüchliche oder unvollständige Anforderungen behandelt werden können, wurden für weiterführende Diskussionen über den Workshop hinaus aufgeworfen.

Theoretische Grundlagen Trotz einer Reihe regelmässiger Workshops zum Thema AOP (begleitend zu Konferenzen wie z.B. ECOOP, OOPSLA oder ICSE), hat sich eine theoretische Fundierung bisher nicht etabliert. Ziel einer Diskussionsgruppe war es, die ersten Ansätze einer theoretischen Aufarbeitung der Konzepte zu sammeln, zu untersuchen sowie fortzuführen. Aspektsprachen sollten wie jede Programmiersprache u.a. eine Definition der statischen und dynamischen Semantik aufweisen. Zu klären ist auch, welche grundlegenden Sprachkonstrukte in einer Aspektsprache enthalten sein müssen.

Aspekte und Frameworks In einer weiteren Diskussionsgruppe wurden Kompositionsmechanismen für objektorientierte und aspektorientierte Frameworks näher betrachtet und verglichen. Bei objektorientierten Frameworks müssen nicht nur Bedingungen bei der Verwendung einzelner Klassen zur Instantiierung oder Vererbung eingehalten werden (wie z.B. Aufrufreihenfolgen und Vor-/Nachbedingungen), sondern die Verwendung einer Klasse kann auch die Verwendung anderer Klassen erfordern oder verbieten. In der Regel fehlen Techniken zur automatischen Überprüfung solcher Konsistenzbedingungen. Auch Aspekte können sich gegenseitig erfordern oder ausschließen. Zusätzlich kann eine Reihenfolge vorgeschrieben sein, in der die Aspekte angewendet bzw. eingewebt werden müssen. Auch hier ist eine automatische Konsistenzprüfung wünschenswert. Als Alternative wurde vorgeschlagen, zusammen mit Frameworks Testsuites zu definieren, die bestimmte Konsistenzen prüfen.

AOP/ASOC(Advanced Separation of Concerns), ein Paradigmenwechsel? Oft wird AOP nur als Ergänzung des objektorientierten Paradigmas angesehen. In einer Diskussionsgruppe zu dieser Fragestellung wurde das Konzept der *Control Tables* aus dem Ansatz des *Table-Oriented Programming* (http://www.geocities.com/SiliconValley/Lab/6888/top.htm) herangezogen, die eine Art explizite, zweidimensionale Darstellung einer Methodentabelle für alle Klassen eines Programms sind. Vererbung bedeutet in diesem Modell die gleichzeitige Belegung mehrerer Einträge einer solchen Tabelle durch eine Methodendefinition, wenn die entsprechenden Klassen voneinander erben. In AOP können über solche "einfache" Vererbung hinaus beliebige Pfade definiert oder modifiziert werden. Insofern stellt AOP gegenüber OOP mindestens ebenso ein neues Paradigma dar, wie OOP gegenüber imperativer Programmierung.

Alles in allem herrschte bei den Vorträgen und Diskussionen eine sehr produktive Atmosphäre, so dass beschlossen wurde, über den Workshop hinaus in Kontakt zu bleiben. Zu diesem Zweck wurden eine Mailing-Liste und ein Wiki-Server eingerichtet. Näheres ist über die WWW-Seite des Workshops zu erfahren (http://i44w3.info.uni-karlsruhe.de/~pulvermu/workshops/aop2001/). Hier sind auch Folien mit Übersichten der Diskussionsthemen und den Ergebnisse verfügbar. Der Tagungsband ist erschienen als Technischer Bericht tr-ri-01-233 der Universität Paderborn. Die Tagungsbeiträge sind auch in elektronischer Form erhältlich auf der WWW-Seite des Workshops.

Der nächste Workshop zum Thema Aspektorientierung soll am 21. und 22. Februar 2002 an der Universität Bonn stattfinden. Ansprechpartner ist Pascal Costanza (costanza@cs.uni-bonn.de).