

Aspektororientierung und Betriebssysteme

- Tutorium -

Olaf Spinczyk und Andreas Gal
(Olaf.Spinczyk,Andreas.Gal)@informatik.uni-erlangen.de
Friedrich-Alexander-Universität Erlangen-Nürnberg

Zeit und Ort

Datum: 7. November 2002
Zeit: 9:30 bis etwa 13 Uhr
(Kaffeepause von 11:00 bis 11:30)
Adresse: Technische Universität Berlin
Raum CR 102
Jebensstr. 1 / Eingang: Hertzallee 41
10787 Berlin

Motivation

Unter dem Begriff Aspekteorientierte Programmierung (AOP) werden verschiedene Techniken zusammengefasst, mit denen im Sinne des Prinzips *Separation of Concerns* die modulare Implementierung sogenannter *Crosscutting Concerns* möglich ist. Das sind Teilaufgaben eines Softwaresystems, die bei Verwendung herkömmlicher Programmiersprachen nicht in Module gekapselt werden können und so zu Code führen, der über weite Teile des Systems verstreut vorliegt. Ein Beispiel dafür ist die Synchronisation nebenläufiger Aktivitäten in einem Betriebssystem, bei der sich die verfolgte Strategie, etwa grob- oder feingranulares Sperren, durch die Punkte im Programmcode ausdrückt, von denen aus die Synchronisationsprimitiven aufgerufen werden. Mit AOP ist die modulare und getrennte Implementierung solcher Strategien möglich.

Seit 1997 finden regelmäßig Workshops zum Thema Aspekteorientierte Programmierung bei namenhaften, programmiersprachlich ausgerichteten Konferenzen wie der ECOOP und der OOPSLA statt. In diesem Jahr wurde dazu sogar eine eigene Konferenzserie namens AOSD ins Leben gerufen. Immer mehr Wissenschaftler und Unternehmen setzen sich mit diesem neuen Konzept auseinander und versuchen, es gewinnbringend in den verschiedensten Anwendungsbereichen einzusetzen. Erste Ansätze dazu gibt es auch im Betriebssystemsektor.

Dieses Tutorium soll vermitteln, warum AOP gerade für den Betriebssystembau eine interessante Technologie ist und welche Ergebnisse bisher erzielt wurden.

Inhalt

Das Tutorium gliedert sich in zwei etwa 90-minütige Vortragsblöcke mit abschließender Diskussion. Dabei wird folgender Inhalt abgedeckt:

- Einführung in die aspektorientierte Programmierung: Die Idee und frühe Arbeiten
- Erste Anwendungsbeispiele für AOP im Betriebssystemsektor, z. B. a-kernel (University of British Columbia - Kiczales/Coady) und AOSA (Illinois Institute of Technology - Elrad/Netinant)
- Einordnung von AOP, verwandte Konzepte: Metaobjektprotokolle, Reflektion, Subjektorientiertes Programmieren, Rewriting, Programmspezialisierung, ...
- Sprachliche Unterstützung für AOP im Betriebssystemsektor: AspectC++ und Alternativen
- Aspekte für die Entwicklungsphase: Kontrollflussverfolgung, Profiling, Zusicherungen und Kontrakte
- Produktionsaspekte: Unterbrechungssynchronisation, Fadensynchronisation
- Bewertung: Vorteile und Risiken von AOP im Betriebssystembau
- Ausblick
- Fragen & Antworten

Adressaten

Ansprechen möchten wir jeden, der sich für die Strukturierung von Betriebssystemen interessiert, oder AOP einmal anhand konkreter Beispiel erläutert bekommen möchte. Spezielle Vorkenntnisse zum Thema AOP werden nicht vorausgesetzt. Grundlegende Kenntnisse über Aufgaben und Struktur von Betriebssystemen sowie ein grobes Verständnis der Programmiersprache C++ sind dagegen wünschenswert.

Vortragende

Olaf Spinczyk

befasst sich seit 1998 mit dem Thema AOP im Hinblick auf die Anwendung im Betriebssystemsektor. Er hat an der Otto-von-Guericke-Universität Magdeburg Teile der Betriebssystemfamilie PURE aspektorientiert entworfen und implementiert. Nach der Fertigstellung seiner Dissertation mit dem Titel "Aspektorientierung und Programmfamilien im Betriebssystembau" arbeitet er heute am Lehrstuhl für Informatik 4 (Betriebssysteme und Verteilte Systeme, Prof. Schröder-Preikschat) der Universität Erlangen-Nürnberg.

Andreas Gal

war als wissenschaftlicher Mitarbeiter und Doktorand an der Otto-von-Guericke-Universität Magdeburg, der University of California (Irvine) und ist heute ebenfalls an der Universität Erlangen-Nürnberg tätig. Zusammen mit Olaf Spinczyk ist er maßgeblich am Entwurf und an der Implementierung der aspektorientierten C++ Spracherweiterung AspectC++ beteiligt.

Olaf Spinczyk 2002-09-23