

# DYNAMISCHE ASPEKTORIENTIERUNG IN BETRIEBSSYSTEMKERNEN

MICHAEL ENGEL, BERND FREISLEBEN

Fachbereich Mathematik und Informatik,  
Philipps-Universität Marburg  
Hans-Meerwein-Str., D-35032 Marburg,  
E-Mail: {engel,freisleb}@informatik.uni-marburg.de

Aktuelle Betriebssystemkerne zählen mit Millionen Zeilen Code und tausenden Quellcode-Dateien zu den komplexeren Projekten des Software-Engineering. Viele dieser Systeme basieren auf oft Jahrzehnte altem Code, der in einer prozeduralen Programmiersprache entwickelt wurde und den Einsatz modernerer Softwareentwicklungs-Paradigmen behindert. Damit gestaltet sich die Erweiterung von Funktionalität wie auch Wartung und Modifikationen in einem solchen System sehr aufwendig.

Eines der Hauptprobleme dabei ist, dass gerade in Betriebssystemkernen zu implementierende Funktionalität nicht in die bereits existierende Modularisierungsstruktur zu integrieren ist und Code, der dieses Problem löst, an vielen unterschiedlichen Stellen in der Code-Basis manuell eingefügt werden muss. Entsprechende Problemstellungen werden "crosscutting concerns" genannt. Dabei ist der entsprechende Code mit anderem Code, der eine komplett unterschiedliche Funktion ausführt, verschlungen (tangled) und schwieriger zu identifizieren und zu warten.

Aspektorientierte Programmierung (AOP) ist ein neues Paradigma für Softwareentwicklung, das es ermöglicht, Code, der ein crosscutting concern implementiert, separat von der restlichen Code-Basis als sog. "Advice" zu modularisieren, die Stellen im Code, an denen der neue Code Funktionen ausführen soll, durch sog. Pointcuts (Mengen von Joinpoints, die die entsprechende Stelle im Code definieren) zu identifizieren und den Advice-Code automatisch an den entsprechenden Stellen einzufügen. AOP ist ein genereller Ansatz, der auch in prozeduralen Programmiersprachen einsetzbar ist. Man unterscheidet zwischen statischer Aspektorientierung - der Advice-Code wird zur Compile-Zeit in den restlichen Quelltext eingefügt - und dynamischer Aspektorientierung, bei der Advice-Module dynamisch zur Laufzeit in Code eingefügt und wieder entfernt werden können. Beispiele für die Anwendung von AOP sind die Implementierung von Logging, Sicherheits-Mechanismen oder Energie sparenden Strategien.

TOSKANA ist ein Toolkit, das es ermöglicht, ohne Änderung des zugrunde liegenden Quelltextes dynamische Aspektorientierung in einem traditionellen, in C geschriebenen Betriebssystemkern (NetBSD) zu verwenden. Als dynamisch ladbare Kernelmodule implementierter Advice Code kann vor, nach, oder um einen beliebigen Funktionsaufruf im Original-Code herum in den Codefluss eingefügt und dynamisch wieder entfernt werden.

Dieser Beitrag gibt einen Überblick über aspektorientierte Technologien und ihren Einsatz in Betriebssystemkernen, stellt Struktur und Implementierungsdetails des TOSKANA-Toolkits sowie mögliche Anwendungen vor. Weiterhin werden spezielle Probleme Aspektorientierter Programmierung in Binärcode angespro-

chen und Lösungsansätze diskutiert. Den Abschluß bildet ein Ausblick auf weitere Einsatzmöglichkeiten für dynamische Aspektorientierung mit TOSKANA in microkernel-basierten Multi-Server Betriebssystemumgebungen.