

## **Zusammenfassung**

Durch einen sorgfältigen Architekturentwurf kann die Komplexität großer, verteilter System auf einem hohen Abstraktionsniveau erfasst werden. Dies ist besonders für rekonfigurierbare Systeme von Bedeutung, deren Konfiguration ständigen Veränderungen zur Laufzeit unterliegt. Beim Entwurf derartig dynamischer Architekturen muss ein Softwarearchitekt die funktionalen Anforderungen an das System mit den Kommunikations- und Rekonfigurationsmechanismen in Einklang bringen, die von der Zielplattform zur Verfügung gestellt werden.

Aufgrund der Schwierigkeit, diese nicht selten divergierenden Anforderungen in einem Architekturmodell zu integrieren, schlagen wir ein schrittweises Vorgehen vor, das dem MDA Ansatz ähnelt. Zunächst wird mit einem plattformunabhängigen Modell begonnen, das nur die funktionalen Anforderungen berücksichtigt. In einem nachfolgenden Schritt werden dann plattformspezifische Details hinzugefügt. So ergeben sich verschiedene Abstraktionsebenen bezüglich der zugrunde liegenden Plattform, deren Charakteristika jeweils durch einen eigenen Architekturstil beschrieben werden. Die Konformität eines Architekturmodells zu einem solchen Stil impliziert damit die Konsistenz des Modells mit der assoziierten Zielplattform.

Neben den Laufzeitkonfigurationen aus Komponenten und deren Verbindungen umfassen die Architekturmodelle auch Prozessbeschreibungen zur Kontrolle des Kommunikations- und Rekonfigurationsverhaltens. Um sie mit einer operationalen Semantik auszustatten, werden die Modelle formal als Graphen und die Architekturstile als Graphtransformationssysteme dargestellt. Zur konkreten Notation durch den Architekten kann die Modellierungssprache UML benutzt werden, nachdem sie durch UML Profile an bestimmte Architekturstile angepasst worden ist.

Wegen des schrittweisen Vorgehens muss die wechselseitige Konsistenz zwischen den Architekturmodellen auf den verschiedenen Abstraktionsstufen sichergestellt werden. Zu diesem Zweck definieren wir formale Prüfkriterien, die sowohl die Bewahrung von Struktur- als auch von Verhaltenseigenschaften auf dem jeweils tieferen Abstraktionsniveau erfordern. In diesem Zusammenhang dient eine Verfeinerungsbeziehung zwischen abstrakten und plattformspezifischen Architekturstilen als Eingabe für einen Algorithmus, der überprüft, ob alle abstrakten Geschäftsprozesse im plattformspezifischen Modell erhalten bleiben ohne dass dort irgendein neues Verhalten hinzukommt. Diese Verfeinerungstechnik ermöglicht einen schrittweisen und plattformkonsistenten Entwicklungsprozess für dynamische Softwarearchitekturen.