

Modellbasierte Konfiguration von Transaktionsdiensten

Sten Löcher
Technische Universität Dresden
Sten.Loecher@inf.tu-dresden.de

Abstract: Die Integration von anwendungsspezifischer Logik und Infrastrukturdiensten ist ein essentieller Bestandteil des Softwareentwicklungsprozesses in modernen Komponententechnologien. Der Kurzbeitrag diskutiert den aktuellen Stand der Technik in bezug auf Transaktionsdienste und stellt anschließend einen Konzeptrahmen für deren modellbasierte Konfiguration vor. Der Konzeptrahmen dient als Ausgangspunkt für die Entwicklung von Mechanismen für die deklarative Konfiguration von Transaktionsdiensten. Die technische Umsetzung des vorgestellten Ansatzes basiert auf Metamodellierung. Aufgrund der benutzten Konzepte und Technologien gliedert sich die vorgestellte Arbeit in die Model Driven Architecture der OMG ein.

1 Einführung

Aktuelle komponentenbasierte Technologien für die Entwicklung serverseitiger Geschäftsanwendungen, wie z.B. Enterprise JavaBeans (EJB)[DmYK01], basieren auf zwei wesentlichen Konzepten der Softwaretechnik. Einerseits werden Komponenten benutzt, um die Fachlogik in kohärente Einheiten zu strukturieren. Andererseits benutzt man die Trennung von Aspekten, um anwendungsneutrale Infrastrukturdienste, wie z.B. Transaktionsverwaltung, von der Fachlogik zu trennen. Die Integration der Fach- und der Transaktionslogik kann deklarativ realisiert werden. Das bedeutet, vordefinierte Konfigurationsattribute werden mit den Komponenten assoziiert und von der Laufzeitumgebung der Komponenten interpretiert. Die deklarative Konfiguration ermöglicht eine klare Aspekttrennung und dadurch die Möglichkeit der Bearbeitung der Aspekte durch Domänenexperten. Sie ist außerdem einfach und dennoch präzise und ermöglicht die explizite Beschreibung der benötigten Transaktionslogik einer Komponente.

2 Problemstellung

Trotz der positiven Eigenschaften des deklarativen Ansatzes für die Konfiguration von Transaktionsdiensten liefert eine Analyse der Situation in aktuellen Komponententechnologien verschiedene Schwachpunkte. So ist die verfügbare *Attributmenge* für die Konfiguration klein und nicht erweiterbar. Sie schränkt dadurch die möglichen Konfigurationen stark ein. Eine Ursache dafür ist das durch aktuelle komponentenbasierte Technologien

fast ausschließlich unterstützte flache Transaktionsmodell, welches hinsichtlich Funktionalität und Performanz Einschränkungen besitzt. Während auf dem Gebiet der Datenbanktechnologie erweiterte Transaktionsmodelle entwickelt und angewendet wurden[E192], ist dies für komponentenbasierte Technologien noch nicht der Fall. Außerdem ist die schlechte Transparenz in bezug auf die in den Attributen enthaltene Information ein Nachteil bei deren Benutzung[Pr02]. Ein weiteres Problem ist die fehlende genaue Klärung der *Verantwortlichkeiten* für die Benutzung der Konfigurationsattribute. Dies ist aus methodologischer Sicht nicht befriedigend und kann durch zugeschnittene Konfigurationsmechanismen gelöst werden. Außerdem ist der *Konfigurationszeitpunkt* zu spät. Komponenten werden nach dem Zusammenbau der Anwendung konfiguriert und getestet. Um Fehler im Transaktionsdesign frühestmöglich zu erkennen, ist ein modellbasierter Konfigurationsansatz von Vorteil. Ein allgemeines Problem aktueller Komponententechnologien ist auch die unvollständige *explizite Deklaration benötigter Dienste*. Dies gilt auch für die Transaktionslogik, da Anforderungen an die Transaktionslogik benachbarter Komponenten nicht deklariert werden können.

3 Lösungsansatz

Ein modellbasierter Ansatz für die Konfiguration von Transaktionsdiensten soll die diskutierten Probleme lösen. Der entwickelte Konzeptrahmen besteht aus drei miteinander korrelierten Metamodellen: einem Metamodell für die Modellierung der relevanten Aspekte der Fachlogik, mehreren Metamodellen für die Beschreibung der benötigten Funktionalität der Transaktionsdienste und schließlich einem integrierten Metamodell, welches die Basis für die Analyse und Anpassung der Konfigurationen entsprechend funktionaler und nicht-funktionaler Eigenschaften dient. Um den verschiedenen am Softwareentwicklungsprozess beteiligten Rollen, wie z.B. Komponentenhersteller, Anwendungsentwickler und Deployer, geeignete Konfigurationsmechanismen zur Verfügung zu stellen und spezifische Transaktionsmodelle zu berücksichtigen, ist der Konzeptrahmen offen für verschiedenste Konfigurationsmetamodelle. Die entsprechenden Konfigurationsmodelle werden für die Analyse und Anpassung transformiert und mit dem Fachlogik-Modell integriert. Die so entstehenden Konfigurationen sind in Bezug auf die Zieltechnologie plattform-unabhängig und müssen auf plattform-spezifische Versionen, z.B. für EJB, abgebildet werden.

Literatur

- [DmYK01] DeMichiel, L. G., Ümit Yalcinalp, L., und Krishnan, S. (Hrsg.): *Enterprise JavaBeans Specification, Version 2.0*. Sun Microsystems. 2001.
- [E192] Elmagarmid, A. K. (Hrsg.): *Database Transaction Models for Advanced Applications*. Morgan Kaufmann Publishers. 1992.
- [Pr02] Procházka, M.: *Advanced Transactions in Component-Based Software Architectures*. PhD thesis. Charles University, Faculty of Mathematics and Physics, Department of Software Engineering, Prague. 2002.