

Appendix D: Anhang gemäß Promotionsordnung

Zusammenfassung

Enterprise-Applikationen sind transaktionale, verteilte Mehrbenutzer-Anwendungen, die von Organisationen zur Steuerung, Unterstützung und Ausführung ihrer Geschäftsprozesse eingesetzt werden. Datenintensive Enterprise-Applikationen wurden traditionell auf Basis von zentralisierten Transaktionsmonitoren entwickelt. Heutzutage werden solche Transaktionsmonitore durch objektorientierte Mehrschicht-Architekturen abgelöst. In dieser Arbeit geht es primär um zwei Aspekte objektorientierter, datenintensiver Enterprise-Applikationen: Verteilung und Datenverwaltung.

Zu Beginn wird ein Architekturstil für mehrschichtige Enterprise-Applikationen definiert. Der Stil stellt nicht nur auf die Schichtenstruktur einer Enterprise-Applikation ab, sondern auch auf ihre *Prozessstopologie*, welche aus verteilten (Betriebsystem-)Prozessen, Datenspeichern und Client/Server Kommunikationsbeziehungen besteht. Der Entwurf einer adäquaten Prozessstopologie für eine konkrete Enterprise-Applikation ist in der Regel Aufgabe eines Softwarearchitekten. Typischerweise hat ein solcher Entwurf einer ganzen Reihe von Anforderungen zu berücksichtigen wie z.B. Skalierbarkeit, Performanz, Verfügbarkeit, Sicherheit, die Integration existierender Subsysteme sowie organisatorische und juristische Rahmenbedingungen. Als Hilfe für den Entwurf von Prozessstopologien wird in der Arbeit ein Katalog von Mustern (so genannte *Topologie-Muster*) vorgestellt, der eine Muster-Sprache für den zuvor definierten Architekturstil darstellt.

Traditionell wird die Verteilung von Enterprise-Applikationen als Architekturangelegenheit höchster Ebene betrachtet. Entscheidungen bezüglich Verteilung werden möglichst im frühen Entwurfsstadium eines Entwicklungsprojekts getroffen und können später oft nur mit hohem Aufwand geändert werden. Darüber hinaus wird von existierender Middleware für Enterprise-Applikationen nur eine kleine Menge von Standard-Topologien gut unterstützt. In dieser Arbeit wird hingegen argumentiert, dass anspruchsvolle Enterprise-Applikationen sowohl *anwendungsspezifische* als auch *anpassbare* Prozessstopologien benötigen:

- *Anwendungsspezifische* Topologien werden benötigt, um die individuellen Anforderungen von Enterprise-Applikationen zu adressieren, z.B. bezüglich Replikation, Datenverteilung, Caching, oder der Integration existierender Subsysteme. Eine Beschränkung nur auf einfacher Standard-Topologien bedeutet eine wesentliche Einschränkung für anspruchsvolle Enterprise-Applikationen.
- Es ist wahrscheinlich, dass sich die Anforderungen an eine Enterprise-Applikation im Laufe ihres Lebenszyklus ändern. Beispielsweise kann erforderlich sein, dass eine Applikation eine stark steigende Anzahl gleichzeitig arbeitender Nutzer bedient oder einen höheren Grad an Fehlertoleranz bietet. Eine Prozessstopologie sollte deshalb *anpassbar* sein, d.h. einfach nachträglich zu ändern, ohne zugleich auch wesentliche Teile der Applikation neu entwerfen zu müssen.

Leider lassen sich anwendungsspezifische und anpassbare Topologien nur schwer mit existierender Middleware realisieren. In der Arbeit werden diesbezügliche Probleme analysiert und dann sechs grundlegende Anforderungen an Middleware für effiziente Unterstützung von anwendungsspezifischen und anpassbaren Prozessstopologien identifiziert. Unter anderem ist es essentiell, Topologie, Anwendungscode und Datenverteilungsschema paarweise zu entkoppeln. Im nächsten Schritt wird eine Middleware-Architektur, die *Flexible Process Topology* (FPT) Architektur vorgeschlagen, welche alle zuvor identifizierten Anforderungen umsetzt. Die Architektur definiert Grundzüge eines Middleware-Frameworks, welches sowohl anwendungsspezifische als auch anpassbare Prozessstopologien ermöglicht. Sie basiert auf einem Netzwerk von Objektmanager-Komponenten, die kooperativ Datenverwaltungsdienste für Anwendungscode erbringen.

In der Arbeit wird exemplarisch die Implementierung eines Middleware-Frameworks für Enterprise-Applikationen vorgestellt, welches auf den Konzepten der FPT-Architektur basiert. Der Prototyp demonstriert, dass in typischen Fällen die verteilte Struktur einer Enterprise-Applikation allein durch (Re)Konfiguration definiert und angepasst werden kann – ohne dass dabei die Implementierung geändert werden muss. Diese Eigenschaft gibt Entwicklern mehr Flexibilität beim Entwickeln von Enterprise-Applikationen und Kunden mehr Flexibilität beim Einsatz derselben. Das vorgestellte Framework skaliert von einfachen Zweischicht-Architekturen bis hin zu großen verteilten Strukturen mit beliebig vielen Schichten, verteilter Datenhaltung und Replikation von Anwendungsprozessen. Zum Schluss wird die Anpassbarkeit und Performanz des Prototyps anhand mehrerer Szenarien evaluiert.

Lebenslauf

Geboren im April 1974 in Detmold.

1980 – 1984 Grundschule Hiddesen

1984 – 1991 Christian-Dietrich-Grabbe-Gymnasium Detmold

1991 – 1993 Gymnasium Leopoldinum Detmold, Abitur

1993 – 1996 Informatik Grundstudium an der Universität Ulm, Vordiplom

1996 – 1999 Informatik Hauptstudium an der Technischen Universität Berlin, Diplom

1997 – 1998 Studentischer Mitarbeiter am Fraunhofer Institut für Software und Systemtechnik, Berlin

1999 – 2000 Freiberuflicher Berater Software Engineering

ab 2000 Promotionsstudent an der Freien Universität Berlin und im Berlin-Brandenburger
Graduiertenkolleg “Verteilte Informationssysteme”

Erklärung

Ich versichere, dass ich die vorliegende Dissertation auf Grundlage der in der Arbeit angegebenen Hilfsmittel und Hilfen selbständig verfasst habe.

Berlin, 14. November 2003

(Christoph Hartwich)