

## Programmadaptation durch Abstraktion statt Konfiguration

Ein Software-Unternehmen hat viele Kunden mit unterschiedlichen Wünschen und Ansprüchen. Obwohl der Zweck der Software für einen Kundenkreis identisch ist, unterscheiden sich die verschiedenen Programme in der Funktion so stark, dass man höchstens von einer minimalen Verwandtschaft sprechen kann. Das heisst, es müssen nicht nur verschiedene Versionen der Software geschrieben werden, es müssen auch alle einzeln gewartet werden, obwohl im Grunde genommen alle denselben Verwendungszweck haben. Daraus stellt sich die Frage: Wie kann dieses grosse Handicap in der Arbeitswelt eliminiert werden und welche Möglichkeiten gibt es, intelligente Software zu erstellen?

Eine mögliche Variante für den Entwickler ist eine Softwareadaptation durch Abstraktion. Voraussetzung ist eine funktionsfähige Software, die problemlos auf alle möglichen Kundenwünsche angepasst werden kann. Eine solche Basisversion kann der Entwickler adaptieren und für einen bestimmten Kunden zusätzlich erweitern, ohne die bereits geschriebene Software in irgendeiner Weise verändern zu müssen. Dadurch resultiert eine grosse Flexibilität im Umgang mit den zum Teil sehr individuellen Kundenwünschen.

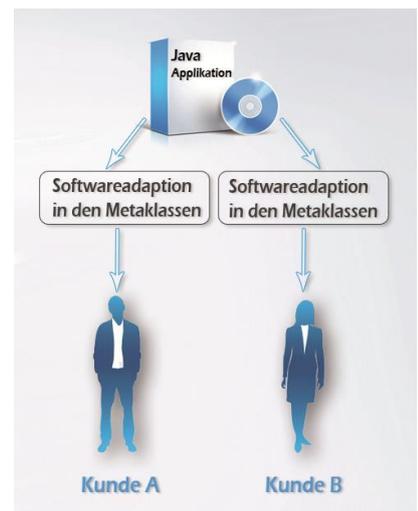
Ein wesentliches Hilfsmittel auf dem Weg der Adaption ist Groovy. Groovy ist eine dynamische Skriptsprache, die auf einer Java Virtual Maschine (JVM) läuft und problemlos mit Klassen, die in Java geschrieben wurden, umgehen kann. Im Mittelpunkt der Adaption steht die Dynamik. Während der Laufzeit einer Applikation sollen neue Methoden und Attribute hinzugefügt werden können. Der Hauptgrund, wieso sich besonders Groovy dafür eignet, hat einen Namen: Meta Object Protocol (MOP). Mit Hilfe dieses Protokolls ist es möglich, Klassen zu erweitern, ohne die Basisklasse zu verändern.

Ein weiteres grosses Thema ist die persistente Speicherung der neu hinzugefügten Attribute in einer Datenbank. Mit Hilfe des Java Persistence API (JPA) werden die Objekte in einer SQL-Datenbank abgespeichert. Die grosse Herausforderung besteht darin, bei einer Adaption die Basisklassen nicht zu verändern, weil JPA gerne mit Annotation arbeitet. Die Lösung dafür findet sich im Deployment Deskriptor "orm.xml", wo die Annotationen in einem zentralen Dokument beschrieben werden können. Dabei waren die wichtigsten Elemente "embedded" und "embeddable". Mit diesen Elementen kann man eine existierende Klasse in eine einbetten und somit adaptieren.



Diplomierende  
Philipp Lombriser  
Adrian Müller

Dozent  
Ewald Mund



Ein Szenario, wie es heute in der Software-Entwicklung immer wieder vorkommt: Kunde A beauftragt ein Unternehmen, eine neues individuelles Programm zu schreiben. Mit dem Erfolg der Software wird auch Kunde B hellhörig und bestellt ebenfalls eine auf ihn abgestimmte Software. Anstatt zwei verschiedene Produkte zu entwickeln und auch zu warten, kann der Entwickler ein einziges funktionstüchtiges Programm schreiben und dieses durch Abstraktion adaptieren. D.h. die Basis-Software wird nicht mehr verändert, sondern mittels dem Meta Object Protocol von Groovy auf individuelle Kundenwünsche angepasst.