

Evaluation von Synchronisations- und Konfliktlösungsverfahren im Web-Umfeld

Martin Eigenmann

Zürcher Hochschule für Angewandte Wissenschaften (26. Juni 2015)

Das Abgleichen von Daten zwischen Server- und Client-Applikation (Datensynchronisation) stellt eine spezielle Herausforderung dar, gerade dann, wenn mehrere Clients zeitlich versetzt, dieselben Daten abzugleichen versuchen. Das gleichzeitige Mutieren von Daten kann zu Inkonsistenzen und damit zu Synchronisationskonflikten führen. Speziell mobile Applikationen für Smartphones oder Tablets dürfen nicht von einem ständigen Kommunikationskanal mit dem Server ausgehen und müssen deshalb mit potentiellen Konflikten umgehen können. Diese Arbeit handelt von der Analyse und Erstellung von Konzepten und Lösungsansätzen zur Abschwächung dieses schweren Datensynchronisationsproblems im Web-Umfeld. Bekannte Synchronisationsverfahren verlangen entweder eine andauernde Verbindung zwischen dem Server und allen Clients, schränken den Funktionsumfang (Client kann nur lesend auf die Daten zugreifen) ein oder garantieren keine zeitliche Aktualität der clientseitigen Daten.

Anhand einer umfassenden Analyse der Beschaffenheit und Struktur der Daten unter Zuhilfenahme von Fallbeispielen, welche den allgemeinen Anwendungsfall repräsentativ wiederzugeben versuchen, werden Konzepte zur Synchronisation, Datenhaltung, Konfliktvermeidung und Konfliktauflösung erarbeitet, bewertet und überprüft. Basierend auf den erarbeiteten Lösungsansätzen ist eine Überprüfung der Umsetzbarkeit jedes einzelnen Konzepts durchgeführt, um so allfällige Schwächen und bis dahin unentdeckte Probleme festzustellen.

Aufgrund der gewonnenen Erkenntnissen wird ein Leitfaden zur Abschwächung des Synchronisationsproblems erarbeitet, welcher die wichtigsten zu beachtenden Punkte wiedergibt. Eine Anforderungsanalyse zu dem Fallbeispiel "Synchronisation einer Kontaktdatenbank" mit Use Cases und funktionalen, sowie nicht funktionalen Anforderungen setzt die funktionsbezogenen Leitplanken für den Prototyp. Unter Betrachtung der Vor- und Nachteile jedes einzelnen Konzepts und unter Berücksichtigung des Leitfadens werden für die Entwicklung des Prototyps die geeignetsten Konzepte ausgewählt, konzipiert und in Form eines Proof of Concept implementiert. Sowohl das Konzept der Zusammenführung (Synchronisation auf Ebene von Attributen und nicht ganzer Datensätze) sowie das Konzept der kontextbezogenen Zusammenführung (Beachtung der Abhängigkeiten zwischen Attributen eines Datensatzes) erlauben zusammen mit der unterschiedsbasierten Synchronisation (Übermittlung nur des Deltas der Änderung) und dem expliziten vorsehen von Synchronisationskonflikten eine deutliche Reduktion der Komplexität. Über die gesamte Erstellungsphase hinweg, stehen dabei Techniken zur Erstellung guter Software, wie Test Driven Development und Domain Driven Architecture, im Mittelpunkt. Neben der nachrichtenbasierten Kommunikation zwischen Client und Server, kommunizieren auch die einzelnen Layer und Komponenten, client- wie auch server-seitig über Nachrichten und somit vollständig asynchron. Die so erreichte Entkoppelung und implizite Schnittstellendefinition der Software gegen innen und aussen erlaubt darüber hinaus eine verständliche und nachvollziehbare Aufstellung von Softwaretests.

Mit dem Prototyp des implementierten Fallbeispiels kann gezeigt werden, dass nachrichtenorientierte Kommunikation zusammen mit der Einführung des Konzepts einer kontextuellen Beziehung zwischen den einzelnen Attributen innerhalb eines Datensatzes sowie die Miteinbeziehung von Techniken zur Konfliktvermeidung und Konfliktauflösung, die Komplexität des Datensynchronisationsproblems reduziert.