

Heinrich Niemann: Regelbasierte Replikationsstrategie für heterogene, autonome Informationssysteme

1. Gutachter: Prof. Dr. Wilhelm Hasselbring (Universität Kiel)

2. Gutachter: Prof. Dr. Stefan Conrad (Universität Düsseldorf)

Datum der Prüfung: 1.7.2009

Zusammenfassung:

Die Replikation von Daten stellt in heterogenen, autonomen Informationssystemen hohe Ansprüche an eine geeignete Replikationsstrategie. So müssen Schreib- und Lesezugriffe auf die Replikate derart koordiniert werden, dass ein optimaler Kompromiss hinsichtlich der konkurrierenden Replikationsziele Verfügbarkeit, Performance und Konsistenz erreicht wird. Dieses Abwägen hinsichtlich der Replikationsziele wird dadurch erschwert, dass die beteiligten Informationssysteme ihre Systemzustände ändern und dass auf diese Veränderungen reagiert werden muss. Für diese Anwendungsbereiche wurden adaptive Replikationsstrategien entwickelt, die sich zur Laufzeit den veränderten Systemzuständen anpassen.

In dieser Dissertation wird die regelbasierte Replikationsstrategie RegRess sowie die Regelsprache RRML vorgestellt, die die Formulierung von Replikationsregeln für RegRess ermöglicht. Bei RegRess erfolgt die Koordination für Schreib- und Lesezugriffe auf Basis dieser Regeln, indem vor jedem Zugriff eine Inferenz der Regeln durchgeführt wird, wodurch die von dem Zugriff betroffenen Replikate ermittelt werden. Durch diese Vorgehensweise wird unterschiedlichstes Konsistenzverhalten von RegRess realisiert, insbesondere werden temporäre Inkonsistenzen toleriert. Eine Regelmenge mit für den Anwendungsfall spezifizierten Regeln bildet die Konfiguration von RegRess. Weil in den Regeln Systemzustände berücksichtigt werden können, kann zur Laufzeit das Verhalten angepasst werden. Somit handelt es sich bei RegRess um eine konfigurierbare, adaptive Replikationsstrategie.

Mit der Regelsprache RRML können so genannte Reaktionsregeln formuliert werden. Bei einer Regel der RRML wird auf Zugriffe auf Replikationseinheiten, die Teilmengen aller logischen Objekte bilden, reagiert, indem die Bedingung der Regel geprüft wird. Die Bedingung einer Replikationsregel beinhaltet neben Gültigkeitszeiträumen vor allem fachliche und technische Konsistenzbedingungen, die z.B. eine Reaktion auf zeitlichen Verzug der Aktualisierungen oder Nicht-Verfügbarkeit eines Rechners erlauben. Wenn die Bedingung einer Replikationsregel erfüllt ist, dann

wird im Aktionsteil der Regel die Zugriffsart auf die Replikate festgelegt. Weil die Replikationsregeln widersprüchliche Aktionen auslösen können, beinhaltet die RRML eine Widerspruchsbehandlung. Zur Realisierung der Replikationsstrategie RegRess dient der Replikationsmanager KARMA, der neben den Protokollen für die Schreib- und Lesezugriffe einen Regelinterpreter für die Replikationsregeln der RRML beinhaltet. Für den KARMA wird eine Softwarearchitektur konzipiert, wobei eine Spezifikation der einzelnen Komponenten des KARMA vorgenommen wird. Ein wichtiger Aspekt bei den Zugriffen auf die Replikate ist die transaktionale Anbindung der beteiligten Systeme. Daher werden Transaktionskonzepte spezifiziert, die bei der Umsetzung der Protokolle benötigt werden.

Der Replikationsmanager KARMA ist mittels Plugin-Mechanismus in den Simulator F4SR integriert, der im Rahmen dieser Dissertation entstanden ist. Mit dem F4SR können Replikationsstrategien oder unterschiedliche Konfigurationen einer Replikationsstrategie verglichen werden. So kann beispielsweise das Verhalten hinsichtlich der Replikationsziele Verfügbarkeit, Performance und Konsistenz für verschiedene Regelmengen untersucht werden. Der F4SR bietet für die Analyse verschiedene Diagramme, die die Ergebnisse eines Simulationslaufs illustrieren.

Veröffentlicht als: Heinrich Niemann: Regelbasierte Replikationsstrategie für heterogene, autonome Informationssysteme, EIDiss Uni Kiel, 2009.