

## REBECA – eine autonome Publish/Subscribe Middleware



### Helge Parzyjegl

Lehrstuhl Architektur von  
Anwendungssystemen  
Universität Rostock  
Albert-Einstein-Straße 21  
18059 Rostock  
Deutschland  
helge.parzyjegl@uni-rostock.de

*Helge Parzyjegl* ist wissenschaftlicher Mitarbeiter am Lehrstuhl Architektur für Anwendungssysteme an der Universität Rostock. Er erhielt sein Diplom in Informatik von der Technischen Universität Berlin, wo er anschließend im Fachgebiet Betriebs- und Kommunikationssysteme arbeitete. Sein Forschungsinteresse gilt ereignisgetriebenen Anwendungen sowie Publish/Subscribe-Infrastrukturen.



### Arnd Schröter

Fachgebiet Kommunikations- und  
Betriebssysteme  
Technische Universität Berlin  
Einsteinufer 17  
10587 Berlin  
Deutschland  
aschroet@cs.tu-berlin.de

*Arnd Schröter* erhielt 2005 sein Diplom in Ingenieurinformatik an der TU Ilmenau. Zur Zeit ist er wissenschaftlicher Mitarbeiter am Fachgebiet Kommunikations- und Betriebssysteme der TU Berlin. In seinem Promotionsvorhaben beschäftigt er sich mit der Modellierung und Optimierung von verteilten Kommunikationssystemen.



### Anselm Busse

Fachgebiet Kommunikations- und  
Betriebssysteme  
Technische Universität Berlin  
Einsteinufer 17  
10587 Berlin  
Deutschland  
abusse@cs.tu-berlin.de

*Anselm Busse* erhielt sein Diplom im Fach Technischer Informatik von der Technischen Universität Berlin im Jahr 2010. Er arbeitet momentan als wissenschaftlicher Mitarbeiter im Fachgebiet für Kommunikations- und Betriebssysteme an der Technischen Universität Berlin und erstellt seine Promotionsarbeit. Seine Forschungsinteressen liegen im Bereich der Betriebssysteme mit Fokus auf eingebetteten und many-core Systemen.



### Daniel Graff

Fachgebiet Kommunikations- und  
Betriebssysteme  
Technische Universität Berlin  
Einsteinufer 17  
10587 Berlin  
Deutschland  
dgraff@cs.tu-berlin.de

*Daniel Graff* erhielt 2008 seinen Master of Science in Informatik von der Freien Universität Berlin. Er arbeitet momentan als wissenschaftlicher Mitarbeiter im Fachgebiet Kommunikations- und Betriebssysteme an der Technischen Universität Berlin und arbeitet an seiner Promotionsarbeit. Seine Forschungsinteressen liegen im Bereich verteilter, mobiler Systeme.



### Alexej Schepeljanski

Professur Betriebssysteme  
Technische Universität Chemnitz  
Straße der Nationen 62  
09111 Chemnitz  
Deutschland  
alexej.schepeljanski@informatik.tu-chemnitz.de

*Alexej Schepeljanski* studierte von 2002 bis 2008 Informatik an der TU Chemnitz mit der Vertiefung „Parallele Programmierung“. Seine Diplomarbeit schrieb er zum Thema „Untersuchung zur Effizienz von Algorithmen der Linearen Algebra auf Multicore Prozessoren“. Seit 2009 ist er wissenschaftlicher Mitarbeiter an der Professur Betriebssysteme der TU Chemnitz. Sein Forschungsinteresse gilt den Algorithmen für verteilte und eingebettete Systeme.



### Jan Richling

Fachgebiet Kommunikations- und  
Betriebssysteme  
Technische Universität Berlin  
Einsteinufer 17  
10587 Berlin  
Deutschland  
richling@cs.tu-berlin.de

*Jan Richling* hat an der Humboldt Universität in Berlin Informatik studiert und dort 2006 zum Dr.-Ing. promoviert. Er arbeitete als Wissenschaftlicher Mitarbeiter an der Humboldt Universität Berlin, als Wissenschaftlicher Assistent an der Technischen Universität Berlin sowie als Vertretungsprofessor an der Ernst Moritz Arndt Universität Greifswald. Derzeit ist er an der Technischen Universität Berlin mit seiner Habilitation auf dem Gebiet der nicht-funktionalen Eigenschaften (z. B. Echtzeit, Autonomie, Zuverlässigkeit, Energieverbrauch) mit speziellem Fokus auf Fragen der Komponierbarkeit beschäftigt.



**Matthias Werner**

Professur Betriebssysteme  
Technische Universität Chemnitz  
Straße der Nationen 62  
09111 Chemnitz  
Deutschland  
matthias.werner@informatik.tu-chemnitz.de

Matthias Werner studierte Elektrotechnik an der Humboldt-Universität zu Berlin und promovierte und habilitierte sich auf dem Gebiet der Informatik. Er arbeitete an der HU Berlin, der TU Berlin, der University of Texas at Austin, bei der Daimler-Forschung Berlin und bei Microsoft Research in Cambridge. Gegenwärtig hält er die Professur für Betriebssysteme an der TU Chemnitz. Matthias Werner forscht auf dem Gebiet der nichtfunktionalen Eigenschaften verteilter und eingebetteter Systeme und der cyber-physikalischen Systeme.



**Gero Mühl**

Lehrstuhl Architektur von  
Anwendungssystemen  
Universität Rostock  
Albert-Einstein-Straße 21  
18059 Rostock  
Deutschland  
gero.muehl@uni-rostock.de

Gero Mühl erhielt 1998 Diplome in Informatik und Elektrotechnik von der FernUniversität in Hagen. Im Anschluss an sein Studium wechselte er an die Technische Universität Darmstadt und promovierte dort 2002 mit Auszeichnung zum Dr.-Ing. Nach seiner Promotion war er Wissenschaftlicher Assistent und Heisenberg-Stipendiat an der Technischen Universität Berlin. Seit 2009 ist er Professor für „Architektur von Anwendungssystemen“ an der Universität Rostock. Sein aktueller Forschungsschwerpunkt ist Selbstorganisation in verteilten Systemen.

## 1 Überblick

Die wachsende Durchdringung unseres Alltags mit Computer- und Kommunikationstechnologien, aktiven Sensoren und Aktuatoren sowie deren Vernetzung macht immer mehr Objekte des täglichen Lebens und zunehmend auch ihre Nutzer zu Informationsquellen, die Unmengen an digitalen Daten vorhalten oder produzieren. Dem gegenüber steht jedoch das Bedürfnis der Nutzer, möglichst nur die für sie relevanten Informationen zeitnah zu erhalten. Eine Antwort auf diese Herausforderung sucht das Forschungsprojekt REBECA. Ziel ist die Entwicklung einer leistungsstarken Middleware für verteilte, ereignisbasierte Infrastrukturen und informationsgetriebene Applikationen. Ein wesentlicher Aspekt ist hierbei eine weitgehende Autonomie der

Middleware, um einerseits den administrativen Aufwand für ungeübte Nutzer zu reduzieren und andererseits eine hohe Dienstgüte für Applikationen durch fortlaufende Optimierung der Infrastruktur zu gewährleisten. Hierbei ist es möglich, einzelne Vorgänge in der Middleware detailliert zu beobachten und gezielt zu beeinflussen.

## 2 Entwickler und Anwender

Die Entwicklung von REBECA zu einer autonomen Middleware wurde durch das *Bundesamt für Sicherheit in der Informationstechnik (BSI)* im Projekt *ASA\_2-Option* gefördert. An der Entwicklung waren das Fachgebiet *Kommunikations- und Betriebssysteme* der TU Berlin, die Professur *Betriebssysteme* der TU Chemnitz sowie der Lehrstuhl *Architektur von Anwendungssystemen* der Universität Rostock beteiligt. Für das Jahr 2011 ist eine Veröffentlichung der Software unter einer Open-Source-Lizenz (<http://rebeca-middleware.org>) geplant, was die breite Nutzung und Weiterentwicklung durch Dritte ermöglicht. REBECA bietet eine geeignete Plattform zur Erforschung, Entwicklung und Integration neuartiger Autonomiealgorithmen und Publish/Subscribe-Konzepte. Einen detaillierten Überblick über die REBECA-Middleware bietet der Artikel [1]. Zudem macht das entwickelte Monitoring-Werkzeug (Abbildung 1) die Middleware auch für Nutzerkreise interessant, die die Funktionsweise von Publish/Subscribe sowie die Auswirkungen von Autonomie erlernen bzw. veranschaulichen möchten. Damit bietet sich auch ein Einsatz innerhalb von Lehrveranstaltungen an Hochschulen und Universitäten an.

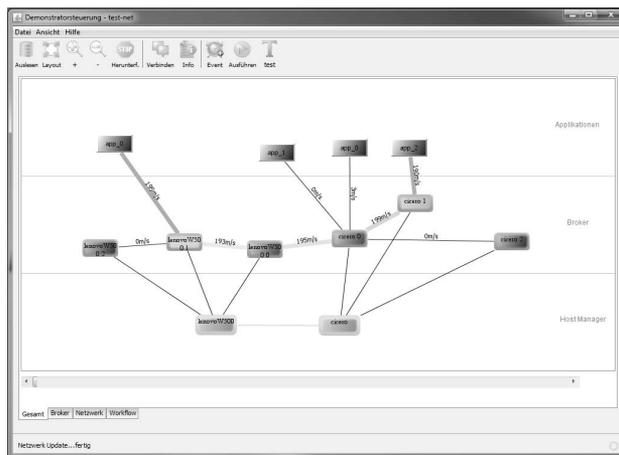


Abbildung 1 REBECA Monitoring-Werkzeug.

## 3 Features und Architektur

REBECA ist eine auf dem Publish/Subscribe-Paradigma basierende, verteilte Middleware, die um Autonomieprinzipien erweitert wurde. Dies umfasst einerseits Fehlertoleranzmechanismen, die selbst bei Ausfall von Komponenten die

vollständige Auslieferung veröffentlichter Nachrichten gewährleisten, und andererseits Algorithmen, die die Publish/Subscribe-Infrastruktur durch fortlaufende Adaptierung in einem möglichst guten Betriebszustand halten. Um die Komposition verschiedener, autonomer Algorithmen zu ermöglichen, wurde eine Plugin-Pipeline-Architektur entwickelt, die das flexible Hinzufügen und Entfernen von Funktionalität erlaubt. Beobachtbarkeit und Steuerbarkeit der Middleware werden durch extern zugängliche Schnittstellen ermöglicht, über die sich interne Statusinformationen abrufen und gezielt das Verhalten einzelner Komponenten beeinflussen lassen. Darüber hinaus bietet REBECA neben einer inhaltsbasierten Publish/Subscribe-Schnittstelle eine traditionelle Request/Reply-Schnittstelle, um Dienstaufrufe entgegenzunehmen und inhaltsbasiert an geeignete Dienstanbieter zu verteilen. Damit bietet sich der Einsatz von Rebeca auch im Rahmen einer *Service-Oriented Architecture* an.

Da autonomes Verhalten „von Außen“ nicht ohne Weiteres beobachtbar ist, wurde ein Monitoring- und Admi-

nistrations-Werkzeug entwickelt, das durch umfangreiche Visualisierungs- und Interaktionsmöglichkeiten sowohl Experten als auch Laien detaillierte Systeminformationen und Eingriffsmöglichkeiten bietet. Neben der Netzwerkvisualisierung können zeitliche Verläufe interner Größen der Middleware sowie globale Systemereignisse dargestellt werden. Ferner lassen sich komplexe Szenarien und Abläufe definieren, die dann automatisch ausgeführt werden.

## Literatur

- [1] H. Parzyjgla, D. Graff, A. Schröter, J. Richling und G. Mühl. Design and implementation of the rebeca publish/subscribe middleware. In K. Sachs, I. Petrov und P. Guerrero (Hrsg.), *From Active Data Management to Event-Based Systems and More, Lecture Notes in Computer Science*, Bd. 6462, S. 124–140. Springer Berlin/Heidelberg, 2010. 10.1007/978-3-642-17226-7\_8.