

Ausschreibung Abschlussarbeit (Bachelor/Master Informatik)

Programmierabstraktion (High-Level API) für Nachrichtenprotokolle in ereignisbasierten Systemen

Motivation. Verschiedene nachrichtenbasierte Protokollspezifikationen resultieren in unterschiedliche Implementierungen. Hierfür sind in der eigenen Anwendung die jeweiligen Programmbibliotheken einzubinden, Konfigurationen zu setzen und Methodenaufrufe durchzuführen. Ein Austausch von Protokoll oder Implementierung bedarf meist manueller Anpassungen des eigenen Quellcodes.

Grundlage. Neben der Umsetzung klassischer Request/Reply-Kommunikation erlauben die zu betrachtenden Nachrichtenprotokolle auch die Umsetzung ereignisbasierter Kommunikation. In der ereignisorientierten Kommunikation werden push-basierte Kommunikationsmodelle wie Point-to-Point oder Publish/Subscribe unterstützt. Des weiteren können je nach Spezifikation weitere Aspekte wie Nachrichtenpriorisierung, Zustellgarantien, Bestätigungen oder Dienstgüte angegeben werden. Prominente Spezifikationen und eine Auswahl derer Implementierungen sind zum Beispiel: Advanced Message Queuing Protocol (AMQP) → Apache Apollo, RabbitMQ; Data Distribution Service (DDS) → OpenDDS; Java Messaging Service → Open MQ; OpenMAMA, OpenWire → Apache ActiveMQ; Message Queue Telemetry Transport (MQTT) → Mosquito, Mosquitto; Streaming Text Oriented Messaging Protocol (STOMP) → Apache ActiveMQ, RabbitMQ; Web Services Eventing → Apache Airavata.

Ziel der Arbeit. Der Quellcode eigener Anwendungen soll für Entwickler möglichst wartungsarm und einfach konfigurierbar gehalten werden. Um dies auch für die Einbindung von Nachrichtenprotokollen zu erreichen, die ereignisbasierte Kommunikation ermöglichen, soll eine High-Level API konzipiert werden. Diese API abstrahiert protokoll- und implementierungsspezifische Methoden mit generalisierten Vertretern. Dieser Ansatz ist auch vom Logging in Java bekannt: Hier stellt die *Simple Logging Facade for Java (SLF4J)* einheitliche Methoden gegenüber der Anwendung bereit. Die nachgeordnet verwendeten Frameworks, z.B. *java.util.logging*, *logback* und *log4j*, können jedoch ohne Änderung des eigenen Quellcodes ausgetauscht werden.

Die eingehende Recherche stellt Gemeinsamkeiten und Unterschiede der jeweiligen Spezifikationen gegenüber und zeigt verfügbare Implementierungen auf. Eine anschließende Analyse behandelt, welche Kommunikationsaspekte generalisiert werden können sowie welche anderen erforderlichen oder optionalen Konfigurationsparameter der Spezifikationen berücksichtigt werden müssen oder können. Nach Definition einer API setzt eine prototypische Implementierung das Konzept mit den generalisierten Methoden um und zeigt exemplarisch die Adaption auf eine Teilmenge von Spezifikationen auf. Die anschließende Evaluierung prüft die korrekte Funktionsweise und Austauschbarkeit der Implementierungen.

Die vorgenannten Ausführungen stellen lediglich eine Diskussionsgrundlage dar. Den genauen Inhalt einer Abschlussarbeit erarbeiten wir in einem persönlichen Gespräch.

Matthias Prellwitz, M.Sc.

Fon +49 381 498-0
+49 381 498-7638
Fax +49 381 498-7622

Mail matthias.prellwitz@uni-rostock.de
wwwava.informatik.uni-rostock.de/prellwitz.html

www.informatik.uni-rostock.de

Architektur von
Anwendungssystemen (AVA)

Chair:
Prof. Dr.-Ing. habil Gero Mühl

Sitz Albert-Einstein-Straße 22
Konrad-Zuse-Haus
18059 Rostock
Fon +49 381 498-7630
Fax +49 381 498-7622

Mail ava.office@uni-rostock.de
wwwava.informatik.uni-rostock.de