

Ein Kommunikations-Framework für die Selbstorganisation von Software-Systemen

Thorsten Stein

Labor für Verteilte Systeme
Fachhochschule Wiesbaden
Kurt-Schumacher-Ring 18
65197 Wiesbaden
tstei001@stud.informatik.fh-wiesbaden.de

Abstract: Die immer höher werdenden Anforderungen an die Erbringung von rechnergestützten Dienstleistungen werden in komplexen und hochdynamischen Architekturen im Kontext von Selbstmanagement durch spezielle Softwarekomponenten, sog. Selbstmanagern, überwacht. Im klassischen Fall liegt der Fokus eines solchen Selbstmanagers auf der Optimierung eines Systems nach lokalen Kriterien. Im Rahmen der hier vorgestellten Diplomarbeit wurde ein Framework zur selbstorganisierten Koordination einzelner Selbstmanagerinstanzen entwickelt, das die Optimierung eines aus mehreren Komponenten bestehenden Systems nach übergeordneten Zielen erlaubt. In diesem Rahmen wurden verschiedene Koordinationsstrategien evaluiert und prototypisch in einem Framework umgesetzt.

1 Motivation und Zielsetzung

Unternehmensweite Anwendungen werden immer komplexer und damit ebenfalls die Abläufe innerhalb aktuell für diesen Bereich verwendeter Service-orientierter Architekturen (kurz: *SOA*) [ZDGH05]. Gleichzeitig spielt das Antwortzeitverhalten angebotener Dienstleistungen eine immer größere Rolle für den geschäftlichen Erfolg eines Unternehmens. Die Anforderungen an eine Dienstleistung werden formal in speziellen Vereinbarungen, sog. *Service Level Agreements* (SLAs), beschrieben [Lu03]. Bei der Verletzung eines SLAs drohen dem Anbieter eines Dienstes i.d.R. Konventionalstrafen. Daher ist die Implementierung eines effektiven *Service Level Managements* (SLM) [SMJ00, Le01, HAN99] notwendig. Aufgrund der hohen Komplexität und Dynamik vieler SOA-Umgebungen können Management-Aufgaben, d.h. permanente Überwachung und Optimierung von Abläufen innerhalb der Umgebung, nur noch bedingt von einem menschlichen Operator gewährleistet werden. Daher gilt es, Möglichkeiten zu finden, die in einer SOA-Umgebung existierenden Softwaresysteme in die Lage zu versetzen, möglichst selbstständig für die Einhaltung der in einem SLA als *Service Level Objectives* (SLOs) beschriebenen Ziele Sorge zu tragen [HMG05, KC03, ST05]. Im Rahmen dieser Ziele sollen die einzelnen Softwaresysteme in der Lage sein,

sich mit Hilfe einer Manager-Komponente selbsttätig zu rekonfigurieren. Die dadurch erfolgende Automatisierung von Management-Aufgaben wird durch den Oberbegriff *Selbstmanagement* ausgedrückt. Da in SOA-Umgebungen unterschiedliche Dienste kooperativ in einem Workflow an der Erfüllung einer Aufgabe mitwirken, genügt es nicht, wenn das Selbstmanagement lokal für eine einzige Komponente betrieben wird. Vielmehr ist es nötig, lokale Management-Bemühungen zu koordinieren und somit einen insgesamt akzeptablen Systemzustand zu erreichen. Da das für die Erbringung der Gesamtleistung vereinbarte SLA nicht verletzt werden darf, müssen die in dem SLA enthaltenen SLOs auf alle beteiligten Softwaresysteme heruntergebrochen werden. Der Anteil eines Softwaresystems an einem SLO wird als *individuelle Service Level Objective* (iSLO) bezeichnet. Wenn ein Softwaresystem nicht in der Lage ist, sein iSLO zu erfüllen, sollte das iSLO erweitert werden können. Da die Gesamtheit aller iSLOs das globale SLO nicht überschreiten darf, muss in diesem Fall das iSLO eines anderen Softwaresystems im gleichen Maße reduziert werden. Diese Aufgabe soll von den zuständigen Managern übernommen werden. Im Labor für Verteilte Systeme an der Fachhochschule Wiesbaden wurde ein Selbstmanager zur Durchführung von lokalen Selbstmanagement-Aufgaben entwickelt. Das Ziel dieser Arbeit ist die Erweiterung des Selbstmanagers um Mechanismen zur Koordination mit anderen Selbstmanagern.

2 Architektur und Ergebnisse

Zur Abwicklung der Kommunikation und der Koordination von Selbstmanagern wurde ein Framework auf Basis der P2P-Kommunikationsplattform JXTA¹ implementiert. Das Framework besteht aus Teilsystemen für *Kommunikation und Gruppenmanagement* sowie *Koordination*. Aufgabe von *Kommunikation und Gruppenmanagement* ist es, die JXTA-eigenen Mechanismen zur Nachrichtenübermittlung sowie zur Bildung von Peer-Gruppen abzubilden [CK01] und dem Teilsystem *Koordination* eine Schnittstelle zu dem unterlagerten JXTA-Framework zur Verfügung zu stellen. Dadurch wird es möglich, das JXTA-Framework zu einem späteren Zeitpunkt gegen einen anderen Mechanismus austauschen zu können, ohne dass die übrigen Softwarekomponenten angepasst werden müssen. Das Teilsystem *Koordination* implementiert Mechanismen zur Koordination der Selbstmanager. Diese ermöglichen es einzelnen Managern, ein iSLO zu erweitern bzw. zu reduzieren. Hierzu wurden drei Möglichkeiten betrachtet: *Auktionen*, *Markt-Mechanismen* und *Mechanismen zur sozialen Interaktion*. Das Teilsystem *Koordination* enthält Mechanismen zur Abwicklung von Transaktionen auf einem virtuellen Marktplatz und Mechanismen zur Durchführung von Auktionen [WWWM01]. Von den möglichen Auktionsvarianten *englische Auktion*, *holländische Auktion*, *Ausschreibung* und *Vickrey-Auktion* [Kr02] wurde die englische Auktion implementiert, als Variante wurde noch eine Auktion im Stil des Online-Auktionshauses eBay² vorgesehen. Die Ermittlung des Auktionators wird von einem Wahl-Mechanismus übernommen. Dieser Wahl-Mechanismus besteht aus einer abgewandelten Implementierung des Bully-Algorithmus [Gm82]. Damit das aus diesen beiden Teilsystemen bestehende Framework an den existierenden Selbstmanager angebunden werden konnte, wurden ein spezielles

¹ <http://www.jxta.org>

² <http://www.ebay.de>

Erweiterungsmodul sowie einige anhängende Komponenten entworfen und implementiert. Abbildung 1 gibt einen Überblick über die entworfene und realisierte Gesamtarchitektur. Ein sog. Strategie-Adapter ist direkt an den Selbstmanager angebunden und ist für die Weitergabe der durch den Selbstmanager ermittelten Leistungswerte des überwachten Systems an das Strategie-Modul zuständig. Zusätzlich empfängt das Strategie-Modul noch eine Datenstruktur, welche zur Berechnung des zur Erreichung des aktuellen Management-Zustands benötigten Aufwands verwendet werden kann. Diese Werte werden von dem Strategie-Modul in Form der sog. *Compliance* [Sc07] berechnet. Das Strategie-Modul trifft basierend auf den empfangenen Leistungswerten und der Compliance strategische Entscheidungen und teilt das Ergebnis dem Strategie-Adapter mit, welcher dann für die korrekte Umsetzung innerhalb des Selbstmanagers zuständig ist. Im Rahmen der Strategie-Entscheidungen kann das iSLO bei Bedarf erweitert bzw. reduziert werden. Der hierzu notwendige Mechanismus (der sog. *SalesAgent*) wird über einen Plug-In-Mechanismus an das Strategie-Modul angebunden und kann somit flexibel ausgetauscht werden. Damit je nach Szenario eine dynamische Anpassung auch zur Laufzeit möglich ist, wurde für die hierzu benötigte Logik eine Reihe von externen Tcl³-Skripten vorgesehen.

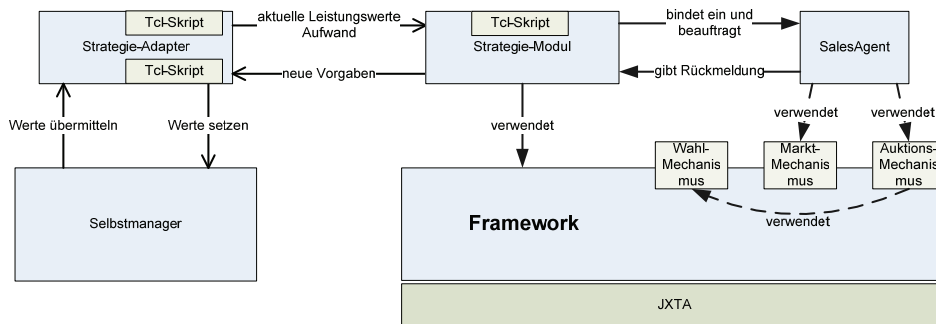


Abbildung 1: Architekturüberblick

Durch die implementierten Koordinations-Mechanismen sind die Selbstmanager in der Lage, selbsttätig über die Erweiterung bzw. Reduktion der iSLOs zu verhandeln.

3 Fallbeispiel

Zur Bewertung des Nutzens des entwickelten Frameworks wurden mehrere Experimente zu konkreten Fallbeispielen ausgewertet. Anhand eines Beispiel-Workflows mit mehreren Diensten wurde die Funktionalität des Ansatzes verifiziert. Die beteiligten Selbstmanager konnten sich mit Hilfe des implementierten Frameworks gegenseitig durch die Abgabe von Teilen ihrer iSLOs unterstützen und somit für eine bedarfsgerechte Verteilung sorgen. In weiteren Tests wurden Reaktionsgeschwindigkeit und Skalierbarkeit des Ansatzes untersucht. Nächste Schritte sehen die Evaluation des Ansatzes anhand einer realen Anwendung vor.

³ <http://www.tcl.tk/>

Literaturverzeichnis

- [ZDGH05] Zimmermann, O., Doubrovski, V., Grundler, J., Hogg, K.: Service-oriented architecture and business process choreography in an order management scenario: rationale, concepts, lessons learned, ACM Press, 2005
- [Lu03] Ludwig, H.: Web services QoS: external SLAs and internal policies or: how do we deliver what we promise?, Web Information Systems Engineering Workshops, 2003.
- [SMJ00] Sturm, R., Morris, W., Jander, M.: Foundations of Service Level Management, SAMS Publishing, 2000.
- [Le01] Lewis, L.: Managing Business and Service Networks, Kluwer Academic Publishers, 2001.
- [HAN99] Hegering, Abeck, Neumair: Integriertes Management vernetzter Systeme, dpunkt-Verlag, 1999
- [HMG05] Herrmann, K., Muehl, G., Geihs, K.: Self-Management: The Solution to Complexity or Just Another Problem?, IEEE Distributed Systems Online 6(1), 2005.
- [KC03] Kephart, J.O., Chess, D.M.: The Vision of Autonomic Computing, Computer 36(1), 2003.
- [ST05] Salehie, M., Tahvildari, L.: Autonomic Computing: Emerging Trends and Open Problems, ACM Press, 2005
- [CK01] Chockler, G.V., Keidar, I., Vitenberg, R.: Group communication specifications: a comprehensive study, ACM Comput. Surv. 33(4), 2001
- [WWW01] Wellman, M., Walsh, W., Wurman, P., Mason, M.J.: Auction protocols for decentralized scheduling, Games and Economic Behavior 35, 2001
- [Kr02] Krishna, V.: Auction Theory, Academic Press, 2002
- [Gm82] Garcia-Molina, H: Elections in Distributed Computer Systems, IEEE Transactions on Computers, C-31, 1982.
- [Sc07] Schmid, M.: Ein Ansatz für das Service Level Management in dynamischen Architekturen. Workshop Selbstorganisierende, Adaptive, Kontextsensitive verteilte Systeme (SAKS), Accepted for Publication, 2007.