On the relationships between QoS and software adaptability at the architectural level

Daniel Schmidt

31. Dezember 2014

Inhalt

Einleitung

- Garantierte Anpassungsfähigkeit von Software kann andere Qualitätsattribute wie Geschwindigkeit, Verlässlichkeit und Wartbarkeit beeinflussen. - Ansatz ist bei einem wechselnden Kontext nützlich, er wird benutzt um zu testen ob die ausgewählten Komponenten die Voraussetzungen des Systems erfüllen.

Anpassungsfähigkeit

- Quantifizierung des Grads der Anpassungsfähigkeit wichtig - Korrelation mit Ansatz von "Yang et al. (2009)", welche eine Abwägungsanalyse zwischen Qualitätsattributen und Anpassungsfähigkeit darstellt. Dabei nimmt dieser Ansatz Änderungen des Kontextes mit auf und die Entscheidung eine Anpassungsstrategie zur Laufzeit auszuführen wenn das System den aktuellen Kontext kennt. - Über heuristische Verfahren kann eine automatische Anpassung der Architektur erfolgen, hin zu einer Architektur, welche die Qualitätsmerkmale erfüllt oder nah dran ist. - Die Grundsätze des hier gewählten Ansatzes sind ähnlich derer in Ëgyed and Wile (2006)", obwohl das Ziel divergiert. Im Gegensatz zu diesen Verfahren wird die Menge der möglichen Architekturen reduziert, indem Ansätze, welche eine Einschränkung nicht erfüllen gelöscht werden. - Die Ziele des Papers sind: - Eine

Beispiel

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ne tum quidem te respicies et cogitabis sibi quemque natum esse et suis voluptatibus? Dicam, inquam, et quidem discendi causa magis, quam quo te aut Epicurum reprehensum velim. Cum id fugiunt, re eadem defendunt, quae Peripatetici, verba. Egone non intellego, quid sit don Graece, Latine voluptas? Aliter enim nosmet ipsos nosse non possumus. Quis, quaeso, illum negat et bonum virum et comem et humanum fuisse? Duo Reges: constructio interrete.

Metriken

- AAS und RAS
- MAAS und MRAS
- LSA

- Der Ansatz basiert auf einer Component-and-Connector Ansicht, da sie allgemein verwendet wird um über die Qualitätswerte zur Laufzeit zu reden.

AAS und RAS

- **AAS** (Absolute adaptability of a service): misst die Anzahl der benutzten Komponenten, welche gewisse Dienste bereitstellen.
- **RAS** (Relative Adaptability of a service): misst die Anzahl der verwendeten Komponenten, welche einen gegebenen Service bereitstellen in hinsicht auf die Anzahl der Komponenten, die tatsächlich solchen Service anbieten.

MAAS und MRAS

- **MAAS** (Mean of absolute adaptability of service): misst die durchnittliche Nummer der genutzten Komponenten pro Dienstleistung. - **MRAS** (Mean of relative adaptability of service): misst den Durchschnitt des RAS (Relative Adaptability of a service).

LSA

- **LSA** (Level of system adaptability): misst die Anzahl der Komponenten die benutzt werden um das System abzubilen im Verhältnis zu der Anzahl der Komponenten die die anpassungsfähigste Architektur nutzen würde

Adapt $^+$ und Adapt $^+$

- Um die Bedeutung von Adapt+ und Adapt- zu erklären wird sich auf die Abbildung Fig.4 bezogen: In (a) und (d) ist Adapt- das niedrigestes A_i für welches man eine Architektur finden kann, welche die Anforderungen erfüllt. Adapt+ ist das niedrigste A_i, dessen Grenzen Q_{A_iU} und Q_{A_iL} die Anforderungen erfüllt. - Die Werte zeigen, dass die Erfüllung der Anforderungen eine Anpassungsfähigkeit von Adapt- voraussetzen und, dass jede Architektur die mindestens Adapt+ hat die Anforderungen auch erfüllt. Für Anpassungsfähigkeit dazwischen gibt es Architekturen, die die Anforderungen erfüllen und solche die es nicht tun.

Beispiel

- Es lassen sich bei Nutzung der gleichen Metrik zwei QoS in einen Graphen einzeichnen. Hierbei wird eine Fläche eingezeichnet, die die Werte bei allen möglichen Architekturen anzeigt. Es lassen sich Adapt+ und Adapt- für beide Qualitätsattribute einzeichnen, so entstehen (vielleicht) Bereiche in denen beide Anforderungen erfüllt sind, nur einer erfüllt ist oder keiner erfüllt ist.

Analyse des Ansatzes

- Ziel der Analyse ist es zu zeigen, dass es eine Reihe von Möglichkeiten gibt ein System durch die Anwendung des Ansatzes zu entwerfen, welches die Anforderungen erfüllt und manchmal auch die gesamte Qualität und / oder Anpassbarkeit verbessert. - Der Ansatz dauert länger als andere Ansätze, die sich auf das Finden einer Architektur mit den höchsten Nutzen für die konkreten Systemvoraussetzungen. Allerdings sind die Erkenntnisse aus den anderen Ansätzen nutzlos sobald sich die Anforderungen ändern und die Analysen müssen wiederholt werden. Bei dem hier gewählten Ansatz muss lediglich die Asymptote der Anforderungen neu gezeichnet werden und dann die neuen Komponenten entsprechent ausgewählt werden. - Die Umgebung stellt eine neue Komponente bereit: Der Ansatz kann angewendet werden, da es neue Möglichkeiten gibt - Die Umgebung zerstört eine

Beschränkungen

- SOLAR (SOftware quaLities and Adaptability Relationships) ist ein Programm, welches den Ansatz umsetzt. Es hat jedoch performance probleme (bei 30 komponenten bis zu 20 minuten) - Es wird für den Ansatz generell nur eine binäre Erfüllung der Anforderungen genutzt (erfüllt, nicht erfüllt). Eine weichere Form kann mit dem aktuellen Ansatz nicht vereint werden, da Adapt+ und Adapt- in einem durchgehenderen Erfüllbarkeitsschema nicht existieren würden. - Bisher gibt es keine Gewichtung in der einige Komponenten, bzw Services wichtiger sein können als andere (WIP). - Normale Probleme (lack of knowledge about the real world execution enviroment and consequently the difficulty in defining architecture parameters)

Literatur

 José Merseguer Diego Perez-Palacin Raffaela Mirandola. "On the relationships between QoS and software adpatability at the architectural level". In: *The Journal of Systems and Software* (2013).

TODOs

- Notizen sind unter `https://github.com/DanielMSchmidt/software-architecture-presentation` zu finden
- Pro Thema entscheiden was an text auf die Folie soll und wie viele Folien zu dem Thema gehören
- Graphiken rendern / finden
- Beispiel übernehmen oder selbst überlegen
- Aufhänger überlegen