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

Daniel Schmidt

25. Januar 2015

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

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

Daniel Schmidt

25. Januar 2015

#### Inhalt

- 1 Einleitung
- 2 Anpassungsfähigkeit
- 3 Metriken
  - AAS und RAS
  - MAAS und MRAS
  - LSA
- 4 Adapt und Adapt +
- 5 Beispiel
- 6 Analyse des Ansatzes
- 7 Beschränkungen

#### Daniel Schmidt

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

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

Inhalt

Anpassungsfähigkeit

Metriken

AAS und RAS

MAAS und MRAS

LSA

Adapt und Adapt

Analyse des Ansatzes
Beschränkungen

# Einleitung

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

Einleitung

—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

#### Definition

An adaptable software system can tolerate changes in its environment without external intervention.

Lawrence Chung Nary Subramanian. Metrics for Software Adaptability. URL: https://www.utdallas.edu/~chung/ftp/sqm.pdf

### -Anpassungsfähigkeit

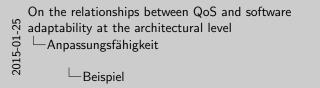
adaptability at the architectural level

-Anpassungsfähigkeit

On the relationships between QoS and software

- 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
- Uber 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 Egyed 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

# Beispiel



Beispiel

#### Metriken

#### Definition $(UC_i)$

Komponenten die den Service i bereitstellen

#### Definition $(C_i)$

Komponenten die den Service i bereitstellen können

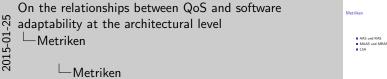
On the relationships between QoS and software adaptability at the architectural level Metriken

└─ Metriken



#### 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

#### Definition (Absolute adaptability of a service)

$$AAS_i = |UC_i|$$

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

Metriken

AAS und RAS

AAS

Children (Abeliate substability of a service)

A445 — [DC]

**AAS** misst die Anzahl der benutzten Komponenten, welche gewisse Dienste bereitstellen.

AAS und RAS

RAS

#### Definition (Relative adaptability of a service)

$$RAS_i = \frac{|UC_i|}{|C_i|}$$

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

AAS und RAS

RAS



**RAS** 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

#### Definition (Mean of absolute adaptability of service)

$$MAAS = \frac{\sum_{i=1}^{n} AAS_i}{n}$$

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

MAAS und MRAS

MAAS

whoton (Man of absolute adaptability of service)

AAS = \frac{\text{Dist}\_{int} ASS}{\text{dist}}

MAAS

**MAAS** misst die durchnittliche Nummer der genutzten Komponenten pro Dienstleistung.

MAAS und MRAS

# MRAS

#### Definition (Mean of relative adaptability of service)

$$MAAS = \frac{\sum_{i=1}^{n} RAS_i}{n}$$

On the relationships between QoS and software
adaptability at the architectural level
Metriken
MAAS und MRAS
MRAS

Definition (Mass of relative adaptability of service)  $\frac{N}{2} = \frac{N}{2} \frac{645}{\pi}$ 

MRAS

**MRAS** misst den Durchschnitt des RAS (Relative Adaptability of a service).

LS/

LSA

#### Definition (Level of system adaptability)

$$LSA = \frac{\sum_{i=1}^{n} AAS_i}{\sum_{i=1}^{n} |C|}$$

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

LSA

LSA



**LSA** 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 +

TODO: definiere  $A_i$ 

#### Definition (Adapt<sup>-</sup>)

Das niedrigste  $A_i$  für welches man eine Architektur finden kann, welche die Anforderungen erfüllt.

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

Adapt - und Adapt +

Adapt - und Adapt +

Adapt - und Adapt +

TOOL definiere A

Estimator (Adapt)

Dan mininger A. für wirklas man eine Anchinetur finder kann, mit he de Anchinetur finder kann, mit he de Anchineturgen erfolt.

# Adapt - und Adapt +

#### Definition (Adapt+)

Das niedrigste  $A_i$  für dessen Grenzen  $Q_{A_iU}$  und  $Q_{A_iL}$  die Anforderungen erfüllen.

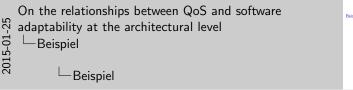
On the relationships between QoS and software adaptability at the architectural level —Adapt - und Adapt +

-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  $Q_{A_i,I_i}$  und  $Q_{A_i,I_i}$  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

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

Analyse des Ansatzes

└─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.

Analyse des Ansatze

- 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 Komponente: Wenn die
    Komponente in der aktuellen Architektur nicht verwendet wird

# Beschränkungen

- Weicher Erfüllungsgrad kann mit dem aktuellen Ansatz nicht vereint werden, da Adapt+ und Adapt− in einem durchgehenden Erfüllbarkeitsschema nicht existieren würden
- Keine Gewichtung von Komponenten & Services
- Fehlendes Wissen über die tatsächliche Umgebung und die Schwierigkeit bei der Definition architektureller Parameter

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

#### Weicher Erfollungsgrad kann mit dem aktuellen Ansatz nich

Beschränkunger

wesint werden, da Adapet und Adapet in einem durchgehenden Erfollbarkeitsschema nicht existieren wurder IK eine Gewichtung von Komponenten & Services IF ehlendes Wissen über die tatschlichte Umgebung und die Schwinzipkeit bei der Definition architektureller Parameter

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 environment and consequently the difficulty in defining architecture parameters)

#### Literatur



Lawrence Chung Nary Subramanian. *Metrics for Software Adaptability*. URL:

https://www.utdallas.edu/~chung/ftp/sqm.pdf.

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

-Literatur

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).

Lawrence Chung Nary Subramanian. Metrics for Software Adaptability. URL: https://www.utdallas.edu/~chung/ftp/sqs.pdf.