

2015-02-02

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

17.02.2015

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

Daniel Schmidt

17.02.2015

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

2015-02-02

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

└─ Inhalt

Inhalt

- Einleitung
- Anpassungsfähigkeit
- Metriken
 - AAS und RAS
 - MAAS und MRAS
 - LSA
- Adapt⁻ und Adapt⁺
- Beispiel
- Analyse des Ansatzes
- Beschränkungen

Einleitung

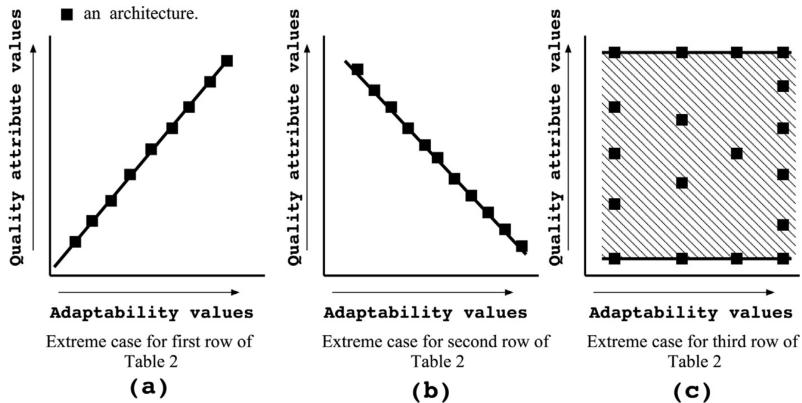


Abbildung : Zusammenhang zwischen Anpassungsfähigkeit und QoS

2015-02-02

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

└ Einleitung

└ Einleitung

Einleitung

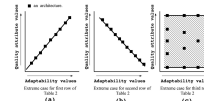


Abbildung : Zusammenhang zwischen Anpassungsfähigkeit und QoS

- 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 (Anpassungsfähiges Software System)

Ein anpassungsfähiges Software System kann Änderungen in der Umwelt ohne einen externen Eingriff vertragen.

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

2015-02-02

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

└ Anpassungsfähigkeit

└ Anpassungsfähigkeit

Anpassungsfähigkeit

Definition (Anpassungsfähiges Software System)

Ein anpassungsfähiges Software System kann Änderungen in der Umwelt ohne einen externen Eingriff vertragen.

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

- Quantifizierung des Grads der Anpassungsfähigkeit wichtig
- Über heuristische Verfahren kann eine automatische Anpassung der Architektur erfolgen, hin zu einer Architektur, welche die Qualitätsmerkmale erfüllt oder nah dran ist
- Die Ziele des Papers sind:
 - Eine erweiterte Menge von architekturellen Metriken die zur Evaluierung der Anpassungsfähigkeit des Systems verwendet werden können
 - Der Ansatz benutzt diese Metriken um die Beziehung zwischen Anpassungsfähigkeit und Qualitätswerten zu definieren, damit hilft dieser Ansatz bei der Begründung des Designs
 - Ein Hilfsmittel bereitstellen um den Ansatz zu benutzen

Beispiel

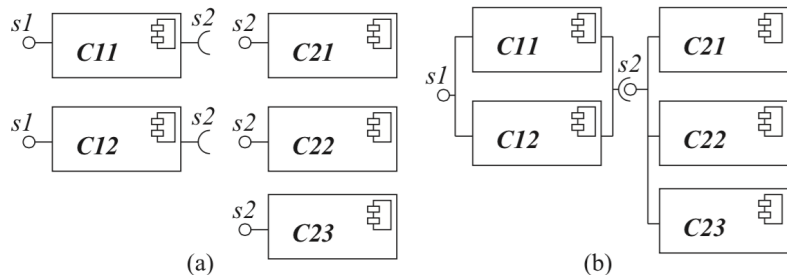


Abbildung : Beispiel Component-and-Connector Ansicht

2015-02-02

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

└ Anpassungsfähigkeit

└ Beispiel

Beispiel

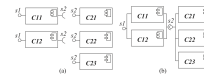


Abbildung : Beispiel Component-and-Connector Ansicht

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

Metriken

Definition (UC_i)

Komponenten, die den Dienst i bereitstellen

Definition (C_i)

Komponenten, die den Dienst i bereitstellen können

2015-02-02

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

└ Metriken

└ Metriken

Metriken

Definition (UC_i)

Komponenten, die den Dienst i bereitstellen

Definition (C_i)

Komponenten, die den Dienst i bereitstellen können

Metriken

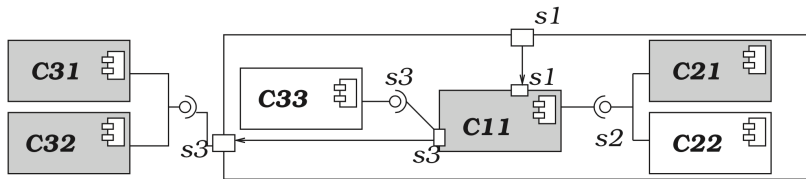


Abbildung : Beispielarchitektur

2015-02-02

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

└ Metriken

└ Metriken

Metriken



Abbildung : Beispielarchitektur

Metriken

- AAS und RAS
- MAAS und MRAS
- LSA

2015-02-02

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

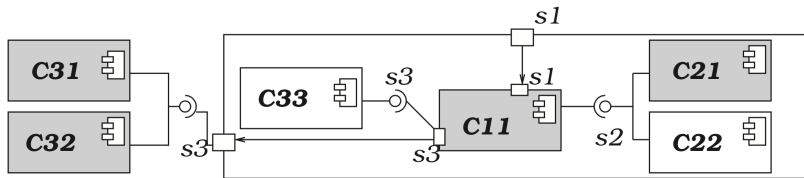
└─Metriken

└─Metriken

AAS

Definition (Absolute adaptability of a service)

$$AAS_i = |UC_i|$$

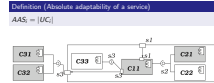


2015-02-02

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

- └ Metriken
 - └ AAS und RAS
 - └ AAS

AAS

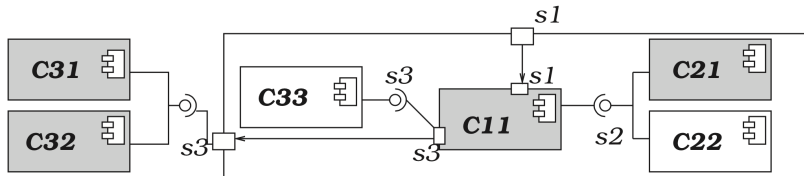


AAS misst die Anzahl der benutzten Komponenten, welche gewisse Dienste bereitstellen.

RAS

Definition (Relative adaptability of a service)

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



2015-02-02

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

└ Metriken

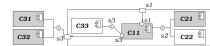
└ AAS und RAS

└ RAS

RAS

Definition (Relative adaptability of a service)

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

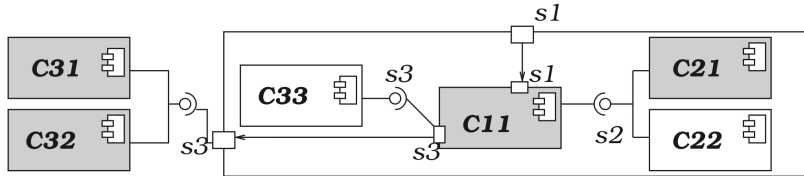


RAS misst die Anzahl der verwendeten Komponenten, welche einen gegebenen Service bereitstellen in hinsicht auf die Anzahl der Komponenten, die tatsächlich solchen Service anbieten.

MAAS

Definition (Mean of absolute adaptability of service)

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



2015-02-02

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

- └ Metriken
 - └ MAAS und MRAS
 - └ MAAS

MAAS

Definition (Mean of absolute adaptability of service)

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

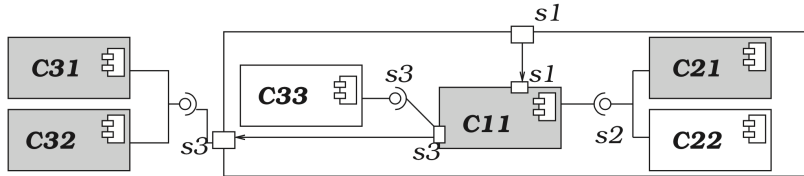


MAAS misst die durchschnittliche Anzahl der genutzten Komponenten pro Dienstleistung.

MRAS

Definition (Mean of relative adaptability of service)

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



2015-02-02

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

- └ Metriken
 - └ MAAS und MRAS
 - └ MRAS

MRAS

Definition (Mean of relative adaptability of service)

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

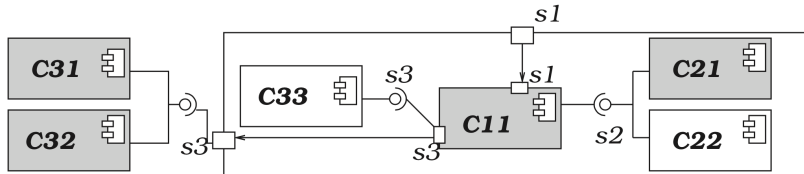


MRAS misst den Durchschnitt des RAS (Relative Adaptability of a service).

LSA

Definition (Level of system adaptability)

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



2015-02-02

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

└ Metriken

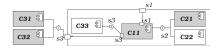
└ LSA

└ LSA

LSA

Definition (Level of system adaptability)

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



LSA misst die Anzahl der Komponenten die benutzt werden um das System abzubilden im Verhältnis zu der Anzahl der Komponenten die die anpassungsfähigste Architektur nutzen würde

Adapt⁻ und Adapt⁺

Definition (*Adapt⁻*)

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

2015-02-02

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

└─ Adapt⁻ und Adapt⁺

└─ Adapt⁻ und Adapt⁺

A_i sind zunehmende Werte für die gewählte Metrik der Anpassungsfähigkeit.

Adapt⁻ und Adapt⁺

Definition (*Adapt⁻*)

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

Adapt⁻ und Adapt⁺

Definition (*Adapt⁺*)

Das niedrigste A_i für dessen Grenzen $Q_{A_i,U}$ und $Q_{A_i,L}$ die Anforderungen erfüllen.

2015-02-02

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

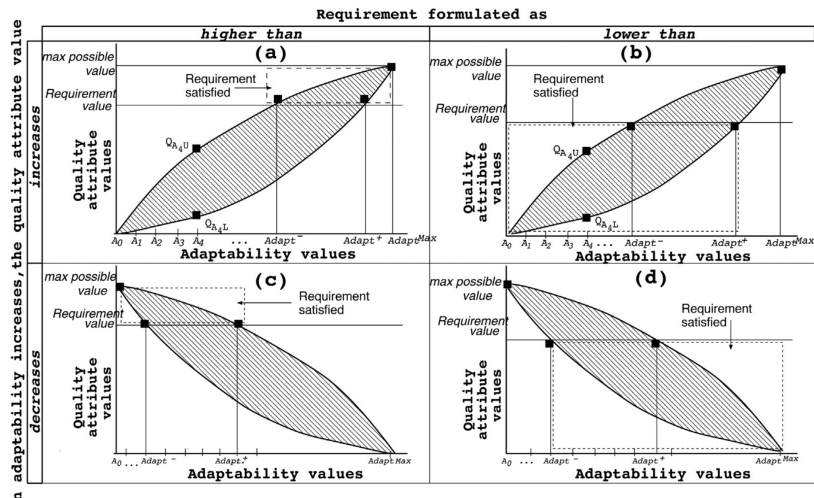
- └ Adapt⁻ und Adapt⁺
 - └ Adapt⁻ und Adapt⁺

Adapt⁻ und Adapt⁺

Definition (*Adapt⁺*)

Das niedrigste A_i für dessen Grenzen $Q_{A_i,U}$ und $Q_{A_i,L}$ die Anforderungen erfüllen.

Adapt⁻ und Adapt⁺

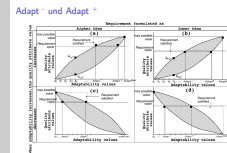


2015-02-02

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

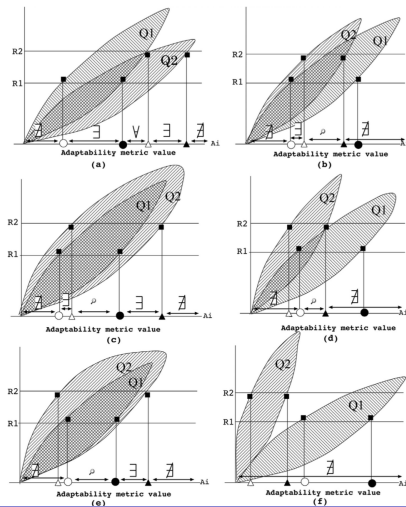
└ Adapt⁻ und Adapt⁺

└ Adapt⁻ und Adapt⁺



- In (a) und (d) ist Adapt⁻ das niedrigste 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



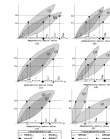
2015-02-02

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

└ Beispiel

└ Beispiel

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.

2015-02-02

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

Analyse des Ansatzes

Definition des Ansatzes

Definition des Ansatzes

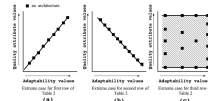


Abbildung : Beziehungen der QoS zur Anpassungsfähigkeit

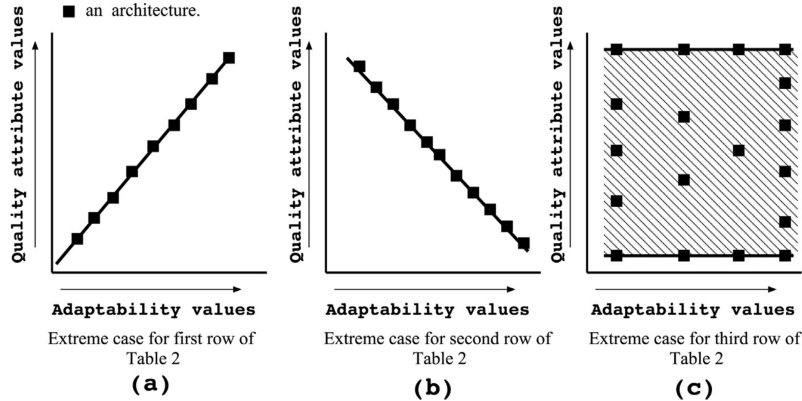


Abbildung : Beziehungen der QoS zur Anpassungsfähigkeit

- Für jedes System gelten unterschiedliche Beziehungen zwischen QoS und Anpassungsfähigkeit
- Wissen über die Beziehungen ermöglicht es den besten Kompromiss zu finden zwischen Anpassungsfähigkeit und Zielerforderung
- 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
- 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)

Analyse des Ansatzes

- **Ziel:** Zu zeigen, dass es eine Reihe von Möglichkeiten gibt mithilfe des Ansatzes ein System zu entwerfen, welches die Anforderungen erfüllt und manchmal auch die gesamte QoS und / oder Anpassbarkeit zu verbessern.
- Ansatz dauert länger als bisherige, aber das Resultat ist auch bei Änderungen weiterhin nutzbar.

2015-02-02

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

└ Analyse des Ansatzes

└ Analyse des Ansatzes

- dauert länger als andere Ansätze, aber Erkenntnisse aus den anderen Ansätzen nutzlos sobald sich die Anforderungen ändern, hier nicht.
- Es muss lediglich die Asymptote der Anforderungen neu gezeichnet werden und die neuen Komponenten entsprechend ausgewählt werden.
 - neue Komponente: Ja, da es neue Möglichkeiten gibt
 - Komponente zerstört: Ja, wenn in Architektur
 - Komponente ändert QoS: Wenn es in der Architektur ist bei Verschlechterung Ja, ansonsten nein. Falls es nicht in der Architektur ist sollte er angewendet werden.
 - Die Anforderungen ändern sich: Wenn die Anforderungen strikter werden und die Anforderungen nicht mehr eingehalten muss der Ansatz genutzt werden, ansonsten nicht.

- Ziel: Zu zeigen, dass es eine Reihe von Möglichkeiten gibt mithilfe des Ansatzes ein System zu entwerfen, welches die Anforderungen erfüllt und manchmal auch die gesamte QoS und / oder Anpassbarkeit zu verbessern.
- Ansatz dauert länger als bisherige, aber das Resultat ist auch bei Änderungen weiterhin nutzbar.

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

2015-02-02

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

└ Beschränkungen

└ Beschränkungen

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

- 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 durchgehenden 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



José Merseguer Diego Perez-Palacin Raffaella Mirandola. “On the relationships between QoS and software adaptability 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/sqm.pdf>.

2015-02-02

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

└ Beschränkungen

└ Literatur

Literatur

■ José Merseguer Diego Perez-Palacin Raffaella Mirandola. “On the relationships between QoS and software adaptability 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/sqm.pdf>.