

Context-Aware Choreography Adaptation: A Survey

Senait Behre
bla..

Markus Schütterle
bla..

Marcel Zeller
bla.

Abstract—The abstract goes here.

1. Einleitung

Menschen besitzen die Fähigkeit aus einem Gespräch den Kontext einer Situation oder Handlung zu verstehen. Dadurch erhöht sich der Informationsgehalt enorm. Computer sind dagegen nicht ohne Weiteres in der Lage aus einer Interaktion mit einem Menschen den Kontext zu extrahieren und zu nutzen. Um trotzdem basierend auf dem Kontext arbeiten zu können, muss eine Anwendung andere Informationsquellen bemühen. Dabei kann der Kontext als jede Information definiert werden, die verwendet werden kann, um die Situation einer Entität zu beschreiben. Eine Entität kann ein Ort, eine Person oder ein Objekt sein, das relevant für die Anwendung ist. Dies schließt Benutzer und die Anwendung selbst ein [1]. Nutzt eine Anwendung den Kontext bei der Durchführung ihrer Arbeit, ist sie demnach Kontext-sensitiv. Eine Service-Choreographie ist eine globale Beschreibung der teilnehmenden Services, die durch den Austausch von Nachrichten, Interaktionsregeln und Vereinbarungen zwischen zwei oder mehr Endpunkten definiert wird. Eine Choreographie verwendet einen dezentralisierten Ansatz für die Zusammenstellung von Dienstleistungen. Das bedeutet, dass die Services sich untereinander selbst organisieren. Eine Choreographie beschreibt im Gegensatz zur Orchestrierung die Interaktionen zwischen mehreren dezentralen Diensten. Wohingegen die Orchestrierung einen zentralen Dienst hat, von dem andere Services aufgerufen werden. Dabei wird das Zusammenspiel aller teilnehmenden Services vom zentralen Service gesteuert.

Da sich eine Infrastruktur ähnlich wie Businessentscheidungen ändern können, muss sich die Choreographie daran anpassen. Dies trifft auf funktionale Anforderungen wie auf einen zusätzlichen Service, sowie auf nicht-funktionale Anforderungen, wie Änderungen am Service-Level-agreement, zu [2]. Je nach Anforderung ist außerdem eine manuelle oder automatische Anpassung nötig. Insbesondere bei automatischen Anpassungen muss das System selbstständig entscheiden, was diese Anpassung im Detail erreichen soll und wie diese durchgeführt wird. Wird dazu der Kontext der Services herangezogen, ergibt sich eine Kontext-sensitive Choreographieanpassung.

Ziel der Arbeit ist die Identifizierung und Analyse des Stands der Technik in der kontextsensitiven Choreographie Anpassung. Die Identifizierung der relevanten Literatur erfolgt durch eine systematische Literaturrecherche [3], [4].

Im Gegensatz zum üblichen Prozess der Literaturrecherche verringert die systematische Literaturrecherche die Verzerrung und folgt einer genauen und strengen Abfolge von methodischen Schritten, die sich auf ein genau definiertes Protokoll stützen.

2. Forschungsmethode

Systematic literature review.
Budgen [3]
Dybaa [5]
Kitchenham [4]

2.1. Forschungsfrage

Welche Frage stellen wir?

Folgende Fragen sind an Leite [2] orientiert:

RQ1: Welche Strategie wählte die ausgewählte Studie für Context Aware Choreography Adaptation?

RQ2: Wie wählt eine ausgewählte Studie ihre Adaptionstrategie gemäß den folgenden Aspekten aus?

(i) *Ziel:* Unterstützt die Anpassung funktionale oder nicht funktionale Anforderungen?

(ii) *Erforderlicher Eingriffsgrad:* Wird die Anpassung automatisch durchgeführt? Oder ist menschliches Eingreifen notwendig?

(iii) *Kontextsensitivität:* Welchen Kontext bezieht die Choreography Adaption Strategie in ihre Entscheidungsfindung mit ein? Wie stark wird die Entscheidungsfindung durch den Kontext beeinflusst?

(iv) *Auswirkungen auf die Skalierbarkeit:* Ist die Strategie für die Skalierbarkeit der Choreographie diskutiert? Ist eine solche Diskussion informell oder enthält sie formale Beweise / Experimente?

(v) *Implementierungen:* Wird der vorgestellte Ansatz von einem Tool oder Prototyp implementiert? Ist die Implementierung des Systems verfügbar? Wenn ja, ist es Open-Source-Software?

(vi) *Zugrundeliegende Modelle:* Welche Choreography Modelle oder Standards werden in der Strategie verwendet?

RQ3: Was sind die Hauptlimitationen und offene Forschungsfragen der Ansätze von "Context-aware choreography adaption"?

Alternative Fragen:

- (i) Gibt es ein System, dass auf Situationen reagiert ohne Einfluss eines Menschen?
- (ii) Was sind die Vor- und Nachteile eines adaptiven Systems
- (iii) Welche Methoden gibt es, um ein System vom Kontext abhängig zu machen?
- (iv) Welche Grenzen werden einem Situations-abhängigen System gegeben?

2.2. Datenquellen

Um die Reproduzierbarkeit des durchgeführten systematischen Literatur Reviews, wurde der Suchprozess durch Skripte automatisiert. Die Datenquellen wurden mit einem benutzerdefinierten Python Skript durchsucht. Dazu wurden die folgenden wissenschaftlichen Quellen mit einer definierten Zeichenabfolge durchsucht: IEEE Xplore¹ SpringerLink² und Google Scholar³

2.3. Abfragezeichenfolge

Die Suchen wurden mit Hilfe von Suchanfragen durchgeführt. Dabei wurden Titel und Abstract der Paper durchsucht. Die Suchanfragen sind an der Suchanfragen der SLR von Leite [2] orientiert und wird durch Suchwörter zu Context Aware erweitert.

Aus der Kombination entstanden 3 Suchblöcke. Der erste Block ist für *context-ware*, der zweite für *choreography* und der dritte Block für *adaption*. Die verschiedenen Strings aus einer Gruppe sind mit ODER (OR) verbunden. Die Suchblöcke werden mit UND (AND) verbunden. Dadurch entstehen Anfragen wobei aus jedem Block ein String enthalten sein muss. Daraus wurden sämtliche Möglichkeiten gebildet. Die Suchanfrage ist die folgende:

(- Context-aware computing
Context-aware/Context aware
- Context-specific computing
Context-specific/Context specific OR
- Context-dependent computing
Context-dependent/Context dependent OR
- Context-sensitive computing
Context-sensitive/Context sensitive OR
- Situation-aware computing
Situation-aware/Situation aware
"location" OR
"situational" OR
"environment" OR)// ab hier vgl. auch Leite2013)
("choreography" OR
"decentralized composition" OR
"decentralized service composition" OR
"distributed composition" OR

- 1. <https://ieeexplore.ieee.org/Xplore/home.jsp>
- 2. <https://link.springer.com/>
- 3. <https://scholar.google.de/>

) AND
(adapt* OR
self-config* OR
auto-config* OR
reconfig* OR
"sensitiv" OR
"behaviour" OR
)

2.4. Einschluss und Ausschluss Kriterien

Nach Suche waren nach entfernen der Duplikate insgesamt 4435 Ergebnisse vorhanden.

Alle Paper vor 2000 wurden ausgeschlossen. 2000-2010 mindestens 10 Zitationen. 2011 9 Zitationen, 2012 8 Zitationen, 2013 7 Zitationen, 2014 6 Zitationen, 2015 5 Zitationen, 2016 4 Zitationen, 2017 3 Zitationen, 2018 2 Zitationen, 2019 1 Zitationen. Reultat: 2722 Paper

Hier fehlt wie wir entschieden haben ob wir es auswählen oder nicht.

Manuelle Überprüfung wobei Titel und im Zweifelsfalls Abstract gelesen wurde um die Zahl weiter zu reduzieren. Rslutat: X Paper

Nach lesen sämtlicher Abstracts bleiben X Paper übrig.

TABLE 1. STUDIENAUSWAHLVERFAHREN

Stufe	Beschreibung
Stufe 1	Die Suchanfragen wurden auf allen angegeben Datenquellen angewendet und die Ergebnisse gespeichert.
Stufe 2	Duplikate und ungültige Paper wurden ausgeschlossen.
Stufe 3	Die Inklusions- und Exklusionskriterien wurden auf die Titel der Paper und bei Unklarheit auf den Abstract angewendet.
Stufe 4	Die Inklusions- und Exklusionskriterien wurden auf Abstract und Zusammenfassung angewendet.
Stufe 5	Die Inklusions- und Exklusionskriterien wurden auf den ganzen Text angewendet.

2.5. Datenextrahierung

Die Daten wurden mithilfe eines automatisierten Suchskriptes extrahiert. Jede der Anfragen wurde einzeln auf jeder der genannten Suchmaschinen ausgeführt. Die Ergebnisse der einzelnen Suchanfragen wurden jeweils in einer CSV-Datei gespeichert. Dabei wurden die folgenden Metadaten extrahiert:

- Titel
- Author(en)
- Veröffentlichungsjahr
- Papertyp
- Link zum Ergebnis
- Link zum PDF
- Zugehörige Suchanfrage
- Datenquelle
- Zahl der Zitationen
- Digitale Objekt ID (DOI) des Ergebnisses

Die Ergebnisse der einzelnen Suchfragen wurden anschließend in einer einzelnen CSV-Datei zusammengefasst, wobei 9168 Paper als Ergebnis resultieren. Aufgrund der Suchanfragen auf verschiedenen Suchmaschinen sind darin Duplikate enthalten. Die Duplikate wurden deshalb aus den Ergebnissen entfernt.

Nach dem Entfernen der Duplikate bleiben 4435 Ergebnisse übrig.

2.6. Protokoll Evaluierung und Limitierungen

Braucht man nach Kitchenham

3. Charakterisierung der ausgewählten Studien

Hier kommen allgemeine Sachen über die gefundenen Paper hin. Dies gibt eine Übersicht über die gefundenen Paper. Siehe SLR von Leite [2].

Von welcher Datenquelle kommen wieviele Paper?

Wie wurden die Paper veröffentlicht?

In welchen Jahren wurden sie veröffentlicht?

In welche Kategorie kann man die Paper einordnen?

4. Context-Aware Choreography Adaptation Strategien

Dieser Abschnitt präsentiert die synthetisierte Daten, um die Forschungsfragen zu beantworten. Zunächst wird ein Überblick über kontextsensitive Choreographie-Adaptionsstrategien gezeigt, die **RQ1** beantworten. Anschließend werden die Ergebnisse über die kontextsensitive Adaptionsstrategie zusammengefasst, was **RQ2** beantwortet. Am Ende wird **RQ3** beantwortet und die Hauptlimitationen und offenen Forschungsfragen aufgezeigt. Die ausgewählten Studien wie bei Leite [2] nach ihrem Fokus in Kategorien gruppiert: modellbasierte Ansätze, messungsbasierte Ansätze, Multiagenten basierte Ansätze, formale Methoden basierte Ansätze, semantisches Denken basierte Ansätze und Proxylayer basierte Ansätze.

4.1. Automated Context-Aware Adaptation of Web Service Executions

Narendra und Gundugola [6] zeigen einen Ansatz zur Anpassung von zusammengesetzten Web Service Ausführungen, wobei funktionale Anforderungen angepasst werden. Dabei zeigen sie, wie die Anpassungen mithilfe von Kontextontologien modelliert werden können. Zur automatischen Anpassung werden Kontextinformationen auf verschiedenen Ebenen der Web Services identifiziert. Diese werden in verschiedene Typen klassifiziert: C-Kontext für den zusammengesetzten Service, W-Kontext für den Web Service Provider und I-Kontext für einzelne Service-Instanzen, die von den Web Service Providern erstellt werden [6]. Anpassungen in einem Web Service werden

immer dann benötigt, wenn sich die Spezifikation einer Komponente ändert [6]. Dies kann beispielsweise der Austausch eines Services durch einen anderen mit erweiterten Anforderungen sein. Als Beispiel für die Workflow-Anpassung wird die Ausnahmebehandlung genannt, die sich mit Mechanismen zur Wiederherstellung des Zustands des Workflows im Falle eines Fehlers während der Workflow-Ausführung befasst [6]. Dieser Fall tritt ein, wenn eine der Web-Service-Komponenten nicht ausgeführt wird [6]. Zur automatischen Anpassung werden Sphären verwendet, welche eine Sammlung von Aufgaben repräsentieren, die entweder komplett oder überhaupt nicht ausgeführt werden [6]. Sphären werden mit einem Algorithmus identifiziert. Der Algorithmus zur Anpassung des Workflows, spiegelt die Rückwärtsbewegungen des Sphären-Bestimmungsalgorithmus wider [6]. Der Ansatz soll in Zukunft noch in einem Prototyp implementiert werden, um die Machbarkeit zu beweisen. Außerdem sollen die Anpassungstechnik weiter erweitert werden, um weiteren Kontext einzubeziehen.

4.2. Modeling of Context-Aware Self-Adaptive Applications in Ubiquitous and Service-Oriented Environments

Der Ansatz Geijs et al. [7] zielt auf Umgebungen mit dynamisch verfügbaren Diensten ab, die von Anwendungen genutzt werden können. Dabei ist das Ziel die Anpassung der funktionalen Anforderungen. Die Anpassungsentscheidungen hängen dabei nicht nur von den Kontext-Eigenschaften ab, sondern auch von der Verfügbarkeit der Dienste. Um Dienste dynamisch anzupassen, muss der Adaptionsansatz verschiedene Anforderungen erfüllen. Dazu gehören die Variabilität der Anwendung, eine dynamische Serviceerkennung, Heterogenität und die Integration von Service- und Kontext-Eigenschaften in die Planung [7]. Zur Adaption werden bei einem signifikanten Kontextwechsel alle verfügbaren Anwendungsvarianten von der Middleware ausgewertet und auf Basis verschiedener QoS-Metadaten verglichen, die den beteiligten Komponentenrealisierungen zugeordnet sind [7]. Komponententypen sind Variationspunkte, die einer bestimmten Teilefunktionalität der Anwendung entsprechen. Die Komponententypen werden rekursiv ausgewählt und der Prozess wird gestoppt, wenn ein atomarer Realisierungsplan ausgewählt wurde. Wenn Dienste für die Middleware nicht mehr verfügbar sind, wird der Realisierungsplan verworfen und ein neuer Anpassungsprozess wird ausgelöst [7]. Der Ansatz ist bereits teilweise implementiert und wurde verwendet, um erste Prototypen zu implementieren. Weitere Forschungsmöglichkeiten gibt es im Bezug auf Performance und Skalierbarkeit.

4.3. Semantic Web Service Adaptation Model for a Pervasive Learning Scenario

Lau et al. [8] liefern einen halbautomatischen Adaptionsansatz zur Abstimmung und Anpassung von relevanten

Web-Services. Um einen Service anzupassen oder durch einen anderen zu ersetzen wird eine *Service Requirement Specification* verwendet, in welcher beschrieben wird, wie ein Service auszusehen hat. Außerdem werden Dienste mit einem sogenannten *Service Descriptor* beschrieben. Die Anpassung besteht aus zwei Schritten die hintereinander ablaufen. Zuerst werden relevante Dienste basierend auf der Beschreibung des Services aus einer Datenbank gesucht, um die Serviceanforderungen für eine Aufgabe zu erfüllen. Die relevanten Services dienen als Input für die folgende Adaptionsphase, die dazu dient die relevanten Dienste an die aktuelle Situation und den Interessen der Nutzer anzupassen. Der Adaptionsprozess besteht aus den drei Phasen Bewertung/Klassifizierung, Filterung und Ranking [8]. Bei der Klassifizierung werden die Services entsprechend der aktuellen Situation und Klassen eingeteilt. In der Phase des Filterns werden irrelevante Dienste über die Klassifizierung herausgefiltert. Anschließend wird basierend auf ausgewählten Merkmalen ein Ranking erstellt, wobei die Services in eine Rangfolge gebracht werden. Der Ansatz wurde in einem Prototyp implementiert, um die Funktionalität zu validieren.

4.4. A Conceptual Model for Adaptable Context-aware Services

Autili et al. [9] schlagen einen Ansatz für anpassbare kontext- und serviceorientierte Anwendungen vor. Dieser basiert auf einer 2-Layer-Architektur, welche aus Service-Layer und Komponenten-Layer besteht. Der Komponenten-Layer bildet die Rechenressourcen für die vernetzten Dienste ab, welche auf dem Service-Layer abgebildet werden. Softwarekomponenten können fliegend bereitgestellt, aktualisiert oder entfernt werden, ohne die Funktionalität zu beeinflussen. Das Mapping zwischen Komponenten und Services wird auf Service-Layer Ebene realisiert. Die Adaption der Services soll funktionale und nicht funktionale Anforderungen verbessern, wobei das Ziel ist, einen optimalen Kompromiss zwischen Bedürfnissen der Nutzer und den aktuellen verfügbaren Ressourcen zu erreichen [9]. Jeder Service hat eine Beschreibung, wobei zusätzliche QoS (Quality of Service) Attribute hinzugefügt werden. Alle verfügbaren Services werden in einer Service-Registry gespeichert. Services werden durch Serviceanfragen nachgefragt, wobei mehrere Anforderungen an den Service enthalten sind. Wenn ein Service aus der Registry passt wird dieser verwendet. Die Anfrage enthält zusätzliche Informationen wie verfügbare Speicher, Größe des Bildschirms oder der Typ der Netzwerkverbindung, was Informationen zum Kontext sind in der der Service laufen wird. Diese werden verwendet, um den nachgefragten Service an die Eigenschaften des Gerätes anzupassen, auf dem er läuft. Der Ansatz wurde als Prototyp implementiert. Es gibt weitere Forschungsmöglichkeiten bei der Verfeinerung des Kontextes.

4.5. Context-Aware Service Composition A Methodology and a Case Study

Bastida et al. [10] zeigen einen Ansatz und die notwendigen Schritte, um kontextsensitive Servicekompositionen zu entwickeln. Dieser besteht aus sechs Schritten. Im ersten Schritt wird die Architekturspezifikation entwickelt, wobei auch Architekturalternativen identifiziert und bewertet werden, um die Systemanforderungen zu erfüllen. Im zweiten Schritt wird die funktionale Spezifikation ausgearbeitet, wobei alle Subfunktionalitäten identifiziert werden, um die endgültige Funktionalität der Komposition abzudecken [10]. Im dritten Schritt wird die geeignete Komposition zur Designzeit identifiziert, sodass funktionale als auch nicht-funktionale Anforderungen erfüllt werden. Im vierten Schritt werden variable Punkte in der Komposition identifiziert. Dabei wird zur Designzeit identifiziert, welche Teile der Komposition sich ändern können und Anforderungen zur Anpassung definiert [10]. Das Ziel dabei ist die Funktionalität der Komposition zur Laufzeit aufrechtzuerhalten. Im fünften Schritt werden geeignete Services anhand der abstrakten Beschreibungen ausgewählt, um die endgültige ausführbare Komposition zu erhalten. Im letzten Schritt wird verifiziert, dass die ausgewählten Dienste ihre spezifizierte Funktionalität und QoS-Anforderungen erfüllen [10]. Die Anpassung der Komposition wird mithilfe von Regeln realisiert. Bei Variationspunkten werden die Regelbedingungen ausgeführt und ein konkreter Dienst für den Variationspunkt ausgewählt. Dies geschieht auf Basis von definierten Varianten oder Alternativen für einen Variationspunkt. Der Ansatz wurde mithilfe einer Fallstudie evaluiert, wobei das Hauptproblem die Notwendigkeit einer stabilen Plattform war, welche dynamische Kompositionen zur Design- und zur Laufzeit unterstützt [10]. Außerdem gibt es Verbesserungsmöglichkeiten des Ansatzes, um die möglichen Varianten dynamisch anzupassen. Zudem bestehen Forschungsmöglichkeiten bei der nachträglichen Anpassung der Komposition, bei der Teile durch eine Subkomposition ersetzt werden, um die gleiche Funktionalität zu erhalten.

4.6. Towards Context-aware Semantic Web Service Discovery through Conceptual Situation Spaces

Um Kontextanpassungsfähigkeit zu erreichen, schlagen Dietze et al. [11] *Conceptual Situation Spaces* (CSS) vor, welche die Beschreibung von situationsabhängigen Charakteristiken in geometrischen Vektorräumen erlauben. Dabei beschreibt ein CSS einen bestimmten Kontext für eine bestimmte Situation. Die CSS sind dabei auf semantische Web Services (SWS) ausgerichtet. Die semantische Ähnlichkeit zwischen Situationen wird dabei in Form ihrer euklidischen Entfernung innerhalb eines CSS berechnet [11]. Dabei werden die am besten geeigneten Ressourcen, wie Dienste oder Daten, basierend auf der semantischen Ähnlichkeit identifiziert. Dieser werden aufgrund einer vorgegebenen realen Situation aus Ressourcenbeschreibungen ausgewählt [11].

Der Ansatz wurde in einem Prototyp implementiert, mit dem die Machbarkeit nachgewiesen wurde. Der Ansatz wird als Schritt nach vorne gesehen, wobei weitere Forschungsmöglichkeiten bei der Priorisierung der verschiedenen Dimensionen eines Vektorraumes genannt werden, um den Ressourcenallokationsprozess besser auf die Präferenzen des Benutzers anzupassen. Außerdem sollen die Beschreibung weiterer relevanter Kontextinformationen durch das CSS-Modell ermöglicht werden.

4.7. Context aware service composition

Vukovic [12] präsentiert einen Ansatz, um kontextsensitive Anwendungen zu entwickeln. Dabei werden die Anwendungen als dynamische Zusammensetzung von Diensten betrachtet. Ändert sich der Kontext, in dem die Anwendung läuft, kann dies zu einer dynamischen Anpassung der Zusammensetzung führen. Dabei wird der aktuelle Kontext des Benutzers der Anwendung verwendet. Fragt der Nutzer einen Service nach, wird in der Anfrage zum einen eine Beschreibung der Aufgabe übermittelt und zum anderen kontextabhängige Parameter, wie das Gerät von dem die Anfrage kommt, oder die Information das der Nutzer sich gerade auf einer Straße mit einem Fahrzeug bewegt. Der Ansatz zielt auch auf Ausfälle von Diensten ab. Tritt ein Fehler bei der Erstellung der Zusammensetzung auf, wird versucht die Anfrage in eine Alternative Zusammensetzung zu ändern. Änderungen von Kontext oder Ausfälle werden auch während der Ausführung überwacht und die Zusammensetzung der Services wird entsprechend angepasst. Das Ziel der Anpassung sind folglich die funktionale sowie nicht-funktionale Anforderungen. Um Ausfälle zu kompensieren und den Kontext zu erfassen wird ein System namens GoalMorph vorgestellt, welches bei Ausfällen auf alternative Zusammensetzungen wechselt. Der Ansatz beschreibt die Implementierung eines Frameworks für kontextsensitive Service Kompositionen. Der Ansatz wurde als Prototyp implementiert. Als weitere Forschungsmöglichkeiten wird eine benutzerdefinierte Anpassung an die Wünsche, eine Priorisierung der Anfragen sowie eine Verbesserung der QoS. Zudem werden Datenschutz, Sicherheit und Vertrauen als weitere Forschungsmöglichkeiten in Umgebungen mit Services von verschiedenen Anbietern genannt, da Kontextinformationen von Nutzern oft sensibel sind.

4.8. Behavioural Self-Adaptation of Services in Ubiquitous Computing Environments

Camara, Canal und Salaün [13] stellen einen Ansatz zur Selbstanpassung von Services vor. Dabei wird sowohl die Zusammensetzung der Services zur Designzeit als auch die dynamische Anpassung der Zusammensetzung zur Laufzeit thematisiert. Dabei zielt der Ansatz auf das Protokoll oder die Verhaltensregeln ab. Dazu schlagen die Autoren ein Service Schnittstellenmodell vor, dass sowohl die Signatur als auch das Verhalten beinhaltet. Der Ansatz zur Anpassung ist ein dynamisches System, bei dem sich Dienste

zur Laufzeit dynamisch an- und abmelden können. Der Ansatz ist vollständig automatisiert und wird durch eine Anforderungsbeschreibung vom Benutzer gesteuert [13]. Wenn ein neuer Service der Choreography beitrifft oder sie verlässt laufen drei Aufgaben ab. Im ersten Schritt wird ein Vektor aus mehreren Serviceschnittstellen durch abstrakte Operationssignaturen instanziiert [13]. Im zweiten Schritt wird eine Liste von stabilen Zuständen ermittelt, welche Dienstprotokolle und Vektoren enthalten. Ein stabiler Zustand des Systems ist einer, bei dem sich jeder der Dienste im System in einem stabilen Zustand befindet [13]. Dienste können nur in solch einem stabilen Zustand hinzugefügt oder entfernt werden. Nach den ersten beiden Schritten wird die Ausführung des Systems gestartet. Dazu wird eine Laufzeit-Engine vorgestellt, die solch ein System ausführt. Zur Selbstanpassung werden Regeln vorgeben, die beispielsweise die Sicherheit betreffen. Dort wird beispielsweise definiert, was nicht passieren darf, wenn ein Service mit dem restlichen System kommuniziert. Zudem werden sogenannte Lebendigkeitseigenschaften definiert, was passieren soll, wenn das System mit dem Rest interagiert. Diese gewährleisten, dass der Ablauf eines Services an einem gewünschten Punkt gehalten wird. Die Anpassung durch Hinzufügen oder Entfernen von Services im Ablauf geschieht immer unter Einhaltung der Regeln. Von dem vorgeschlagenen Ansatz existiert kein Prototyp, wobei die Entwicklung eines solchen in den weiteren Forschungsmöglichkeiten thematisiert wird.

4.9. Context-aware Adaptive Service Mashups

Dorn et al. [14] stellen einen Ansatz zur kontextabhängigen Selbstanpassung von Service Kompositionen vor, bei dem die Fähigkeiten der Services mit den permanent aktualisierten Anforderungen an die Fähigkeiten abgeglichen werden. Der Anpassungsprozess besteht aus zwei Phasen, wobei in der ersten Phase Kontextänderungen und der betroffene Teil der Service Zusammensetzung ermittelt wird. Eine Konfiguration wird neu bewertet, wenn Anforderungsregeln durch die Kontextänderung betroffen sind. Die Anforderungsregeln generieren neue Fähigkeitseinschränkungen und eine Neuanpassung wird gestartet, wenn die bestehende Service Konfiguration nicht mehr mit den Einschränkungen übereinstimmt [14]. Mit den Anforderungen wird eine Liste aus Services zusammengestellt, deren Fähigkeiten die Anforderungen am besten erfüllen. Die Services werden nach Gruppen geordnet und die Services jeder Gruppe bewertet, aus denen der Entwickler dann die beste Kombination auswählt. Der Ansatz ist folglich halbautomatisch. Dies soll durch zukünftige Arbeiten durch automatisches generieren der Anforderungen automatisiert werden. Als Kontext zur Anpassung wird die Reputation der Services, die Organisationen die Service-Zusammensetzung verwenden und der physische Ort verwendet. Der Ansatz wurde mit einer Fallstudie evaluiert.

4.10. Towards Context-Aware Adaptable Web Services

Keidl und Kemper [15] zeigen einen Ansatz für ein Kontext-Framework, das die Entwicklung und Bereitstellung von kontextabhängigen und anpassbaren Webservices erleichtert. Der verwendete Kontext beinhaltet alle Informationen über den Client eines Web Services, die vom Web Service verwendet werden können, um die Ausführung und Ausgabe anzupassen und dem Client ein individuelles und personalisiertes Verhalten zu ermöglichen [15]. Der Kontext des Clients wird in der Anfrage an den Service übergeben, wobei der übergebene Kontext optional ist. Wenn der angefragte Service während seiner Ausführung einen anderen Service aufruft, wird der aktuelle Kontext automatisch in die Anfrage an den weiteren Service eingefügt. Die Services nehmen sich jeweils die relevanten Kontextinformationen heraus. Der Kontext wird in der Serviceantwort zurückgeschickt. Bei dem Ansatz wird zwischen explizitem und automatischen Kontext unterschieden. Expliziter Kontext sind Informationen, die in den Anfragen übergeben wurden und automatischer Kontext wird automatisch aus dem expliziten Kontext generiert. Der Kontext wird durch verschiedenen Kontexttypen definiert. Es werden Typen wie Standort, Client, Verbraucher und Verbindungseinstellungen verwendet. Der Typ Client enthält Informationen, wohingegen der Typ Verbraucher Informationen wie Name oder E-Mail enthält. Der Typ Verbindungseinstellungen enthält wichtige Informationen zur Verbindung selbst. Das Kontext-Framework der Prototyp sind in Java implementiert und basieren auf Standards wie XML, SOAP, UDDI und WSDL [15]. Bei dem Ansatz bestehen weitere Forschungsmöglichkeiten bei zusätzlichen Kontexttypen und einer genaueren Verarbeitung des Kontexts. Außerdem gibt es weiteren Forschungsbedarf bei den Sicherheitsrichtlinien, sodass festgelegt werden kann, welche Services Zugriff auf welchen Kontext haben und welche Dinge damit gemacht werden dürfen.

5. Diskussion

6. Verwandte Arbeiten

Leite [2]

7. Zusammenfassung

References

- [1] A. Dey, "Understanding and using context, personal and ubiquitous computing, vol. 5," 2001.
- [2] L. A. Leite, G. A. Oliva, G. M. Nogueira, M. A. Gerosa, F. Kon, and D. S. Milojicic, "A systematic literature review of service choreography adaptation," *Service Oriented Computing and Applications*, vol. 7, no. 3, pp. 199–216, 2013.
- [3] D. Budgen and P. Brereton, "Performing systematic literature reviews in software engineering," in *Proceedings of the 28th international conference on Software engineering*. ACM, 2006, pp. 1051–1052.
- [4] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering—a systematic literature review," *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.
- [5] T. Dybå and T. Dingsøyr, "Empirical studies of agile software development: A systematic review," *Information and software technology*, vol. 50, no. 9–10, pp. 833–859, 2008.
- [6] N. C. Narendra and S. Gundugola, "Automated context-aware adaptation of web service executions," in *IEEE International Conference on Computer Systems and Applications*, 2006. IEEE, 2006, pp. 179–187.
- [7] K. Geihs, R. Reichle, M. Wagner, and M. U. Khan, "Modeling of context-aware self-adaptive applications in ubiquitous and service-oriented environments," in *Software engineering for self-adaptive systems*. Springer, 2009, pp. 146–163.
- [8] B. S. Lau, C. Pham-Nguyen, C. Lee, and S. Garlatti, "Semantic web service adaptation model for a pervasive learning scenario," in *2008 IEEE Conference on Innovative Technologies in Intelligent Systems and Industrial Applications*. IEEE, 2008, pp. 98–103.
- [9] M. Autili, V. Cortellessa, A. Di Marco, and P. Inverardi, "A conceptual model for adaptable context-aware services," in *International Workshop on Web Services—Modeling and Testing (WS-MaTe 2006)*, 2006, p. 15.
- [10] L. Bastida, F. J. Nieto, and R. Tola, "Context-aware service composition: a methodology and a case study," in *Proceedings of the 2nd international workshop on Systems development in SOA environments*. ACM, 2008, pp. 19–24.
- [11] S. Dietze, A. Gugliotta, and J. Domingue, "Towards context-aware semantic web service discovery through conceptual situation spaces," in *Proceedings of the 2008 international workshop on Context enabled source and service selection, integration and adaptation: organized with the 17th International World Wide Web Conference (WWW 2008)*. ACM, 2008, p. 6.
- [12] M. Vukovic and P. Robinson, "Context aware service composition." Ph.D. dissertation, University of Cambridge, UK, 2007.
- [13] J. Camara, C. Canal, and G. Salaun, "Behavioural self-adaptation of services in ubiquitous computing environments," in *2009 ICSE Workshop on Software Engineering for Adaptive and Self-Managing Systems*. IEEE, 2009, pp. 28–37.
- [14] C. Dorn, D. Schall, and S. Dustdar, "Context-aware adaptive service mashups," in *2009 IEEE Asia-Pacific Services Computing Conference (APSCC)*. IEEE, 2009, pp. 301–306.
- [15] M. Keidl and A. Kemper, "Towards context-aware adaptable web services," in *Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters*. ACM, 2004, pp. 55–65.