INSTITUT FÜR INFORMATIK

DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Fortgeschrittenenpraktikum

Highlevel Service Handover in einem kontextuellen Framework

Matthias Röckl

Aufgabensteller: Prof. Dr. Claudia Linnhoff-Popien

Betreuer: Thomas Strang
Michael Schiffers

Abgabetermin: 15. August 2003

INSTITUT FÜR INFORMATIK
DER LUDWIG–MAXIMILIANS–UNIVERSITÄT MÜNCHEN

Fortgeschrittenenpraktikum

Highlevel Service Handover in einem kontextuellen Framework

Matthias Röckl

Aufgabensteller:	Prof. Dr. Claudia Linnhoff-Popien
Betreuer:	Thomas Strang Michael Schiffers

Abgabetermin: 15. August 2003

Hiermit versichere ich, dass ich das vorliegende Fortgeschrittenenpraktikum selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 15. August 2003

. . . .
(Unterschrift des Kandidaten)

Zusammenfassung

Eine grundlegende Eigenschaft verteilter Systeme ist die Interoperabilität. Ein Betrachtungswinkel der Interoperabilität ist die Ersetzbarkeit zweier Komponenten bezüglich einer Aufgabenstellung. Dabei wird der Vorgang der Ersetzung auch als Handover bezeichnet. Der Handover lässt sich allgemein anhand der Funktionalitäten der dynamischen Entitäten, dem Transparenzgrad, der Position der initiierenden sowie der kontrollierenden Entität, dem Zeitpunkt der Initiierung und der Anzahl der parallelen Kommunikationskanäle unterscheiden. Der eigentlichen Handoverdurchführung geht dabei die Handoverentscheidung voraus. Bei der Handoverentscheidung werden die zur Verfügung stehenden Partner mittels einer Kosten-/Leistungsanalyse bewertet und bei Bedarf dadurch ein Handover initiiert. Eine wichtige Einflussgröße ist dafür der Kontext als vierte Ebene der Interoperabilität. Schließlich folgt, dass jede Veränderung des Kontextes zu der Initiierung eines Handovers führen kann.
Ziel dieser Arbeit ist im Speziellen auf den Highlevel Service Handover einzugehen. Durch Highlevel Services wird eine allgemeine Dienstleistung zur Verfügung gestellt. Wird bei einer Dienstnutzung ein anderer Dienstanbieter als der aktuelle als optimal bewertet, so muss aufgrunddessen ein Handover intiiert werden. Zur Demonstration des Highlevel Service Handovers wird anhand eines Frameworks ein Beispielszenario vorgestellt. Bei der Konzeptionierung des Frameworks wird dabei besonders der Einfluss des Kontextes auf die Dienstnutzung und vor allem auf die Dienstauswahl fokussiert. Anschaulich zeigt dies das Beispielszenario, bei dem ein Dienst zur Entwicklung von digitalen Bildern, erzeugt z.B. durch ein Mobiltelefon mit integrierter Kamera, implementiert wurde.

Inhaltsverzeichnis

Abbildungsverzeichnis

Kapitel 1

Interoperabilität als Basis des Handovers

In der Ära des Ubiquitous Computing [Weis 91] umgeben den Menschen eine Vielzahl von Computersystemen. Dazu zählen Komponenten, wie Handys, Notebooks, PDAs, sowie in naher Zukunft vielleicht Wearables [Mann 98] oder sogar in den Körper implantierte Systeme (implanted devices). Diese einzelnen Komponenten bilden jedoch nicht ein für sich geschlossenes System, sondern können bzw. müssen mit anderen Komponenten interagieren. Die Interoperabilität ist die Grundlage für die Interaktion zwischen verteilten Komponenten (vgl. [StLP 03]). Nach [VHT] ist Interoperabilität die Fähigkeit von zwei oder mehr Entitäten zu kommunizieren und zu kooperieren unabhängig von den Programmiersprachen, in denen die Software der Entitäten implementiert wurde, unabhängig von der Ausführungsumgebung und unabhängig von der „Model Abstraction".
Eine Betrachtung der Interoperabilität von zwei oder mehr Entitäten kann auf verschiedenen Ebenen geschehen. Dabei unterscheidet man die Ebenen Signatur, Semantik, Protokoll und Kontext (vgl. [StLP 03]). In der Ebene der Signatur wird das bei der Interaktion eingesetzte Vokabular syntaktisch beschrieben, das von den teilnehmenden Entitäten benutzt wird. Die Bedeutung des durch die Signatur festgelegten Vokabulars ergibt sich aus der semantischen Ebene. Das Protokoll spezifiziert die Anordnung des in der Signatur definierten Vokabulars in seiner sequenziellen Anordnung. Schließlich charakterisiert der Kontext die Summe der externen Einflüsse auf die Interaktion zwischen den beteiligten Entitäten.
Nach [VHT , StLP 03] kann die Interoperabilität bezüglich der zwei Betrachtungswinkel Kompatibilität (Compatibility) und Ersetzbarkeit (Substitutability) analysiert werden. Während die Kompatibilität für die korrekte Durchführbarkeit einer Kommunikation zwischen zwei Entitäten sorgt, ergibt sich aus der Ersetzbarkeit, dass eine Entität durch eine andere ausgewechselt werden kann. Ein ähnliches Bild ist uns durch die menschliche Sprache gegeben. Ein deutschsprechender Tourist kann einen deutsch sprechenden Menschen nach dem Weg fragen (Kompatibilität). Ist der Tourist zudem noch der italienischen Sprache mächtig, kann er auch einen italienisch sprechenden Menschen nach dem Weg fragen. Gegebenenfalls kann dieser den deutsch sprechenden Wegweiser ersetzen, wenn er über ein entsprechendes Wissen, wie zum Beispiel den noch zu beschreibenden Weg, besitzt. Passend zu diesem Beispiel ergibt sich die Abbildung [StLP 03].
Hierbei ist die Entität C kompatibel zu den beiden Entitäten A und B und die Entität A ersetzbar durch die Entität B.

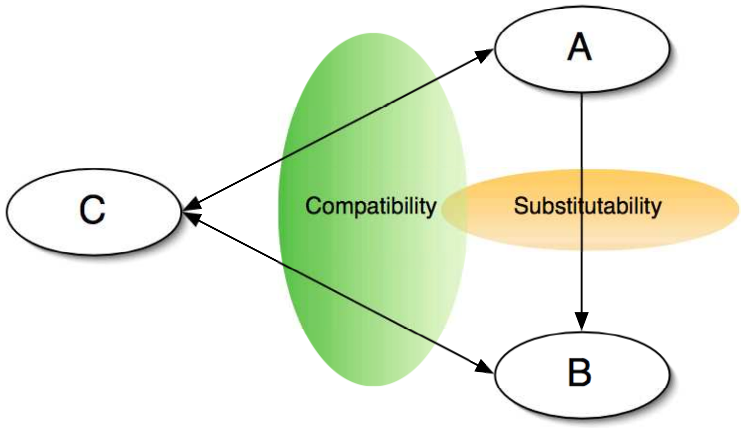

Abbildung 1.1: INTEROPERABILITY: COMPATIBILITY & SUBSTITUTABILITY

1.1 Kompatibilität

[VHT] beschreibt Kompatibilität als die Fähigkeit von zwei Entitäten korrekt zusammenzuarbeiten, d.h. dass alle zwischen diesen Entitäten ausgetauschten Nachrichten und Daten von der Entität, die diese empfängt, verstanden werden. Kompatibilität ist in dieser Betrachtungsweise eine symmetrische Relation (dargestellt in Abb. 1.1 durch den zweiseitigen Pfeil), da an dieser Stelle grundsätzlich von einem Dialog und nicht von einem Monolog zwischen den Entitäten ausgegangen wird. Kompatibilität ist die Grundlage für eine (gleichberechtigte) Kommunikation zwischen den beteiligten Entitäten. Deshalb wird im Folgenden eine Kompatibilität immer bezüglich einer Kommunikation zwischen zwei Entitäten betrachtet. Um nun Kompatibilität zwischen den beteiligten Entitäten zu erreichen, müssen die Entitäten bezüglich der zugrundeliegenden Kommunikation in allen vier Ebenen kompatibel sein. Kompatibilität in einer Ebene bedeutet, dass die beteiligten Entitäten bezüglich dieser Ebene ein gemeinsames Verständnis (shared understanding) haben.

- Dazu müssen die Entitäten grundsätzlich eine korrespondierende **Syntax** besitzen. Eine korrespondierende Syntax ergibt sich aus einem syntaktisch passenden gegenseitigen Verständnis. Am Beispiel der Entitäten A und C (vgl. Abb. 1.1) bedeutet dies, dass A das von C zur Kommunikation mit A benutzte Vokabular als syntaktisch korrekt erkennt und, da es sich um eine symmetrische Relation handelt, auch C das von A zur Kommunikation mit C benutzte Vokabular als syntaktisch korrekt erkennt.

- Desweiteren müssen die bei der Kommunikation beteiligten Entitäten die gleiche Bedeutung des zur Kommunikation verwendeten Vokabulars, d.h. dessen **Semantik** kennen, um ein gemeinsames Verständnis zu erlangen. Folglich müssen die Entitäten A und C ein identisches semantisches Verständnis des bei der Kommunikation zwischen A und C eingesetzten Vokabulars besitzen.

- In der Ebene des **Protokolls** wird ein gemeinsames Verständnis dadurch erreicht, dass die bei der Kommunikation beteiligten Entitäten eine identische Anordung des in der Ebene der Signatur eingeführten Vokabulars nutzen. Entität C muss dafür das zur Kommunikation mit A verwendete Vokabular in der Reihenfolge benutzen, in der es von A als korrekt erkannt wird. Entsprechendes gilt für das von C zur Kommunikation mit A eingesetzte Vokabular.

- Letztlich besitzen die bei der Kommunikation beteiligten Entitäten ein gemeinsames Verständnis bezüglich des **Kontextes**, wenn sie ein identisches Verständnis der gegenseitigen externen Einflüsse besitzen.

Äquivalentes gilt für die Entitäten B und C aus Abbildung 1.1.

1.2 Ersetzbarkeit

[VHT] bezeichnet die Ersetzbarkeit relativ kurz als die Fähigkeit einer Entität eine andere zu ersetzen. Für eine detaillierte Definition der Ersetzbarkeit reicht ein Bezug auf die ersetzende und die zu ersetzende Entität nicht aus. Die Ersetzbarkeit bezieht sich grundsätzlich auf zwei Entitäten bezüglich einer Kommunikation zu einer dritten Entität. Dementsprechend lässt sich die Ersetzbarkeit definieren als die Fähigkeit einer Entität äquivalent zu einer anderen Entität bezüglich einer Kommunikation zu einer dritten Entität zu interagieren. In Abbildung 1.1 bedeutet dies, dass Entität A durch Entität B bezüglich der Kommunikation zu Entität C ersetzbar ist. Die Ersetzbarkeit ist im Gegensatz zur Kompatibilität eine nicht symmetrische Relation (dargestellt in Abb. 1.1 durch den gerichteten Pfeil). Ist eine Entität durch eine andere ersetzbar, so gilt dadurch nicht automatisch auch die Umkehrung. In Abbildung 1.1 ist die Entität A ersetzbar durch die Entität B, B aber ist nicht ersetzbar durch A.
Durch eine Ersetzung, dem sogenannten *Handover*, soll es für C möglich sein die ursprüngliche Kommunikation zu A durch eine äquivalente Kommunikation zu B fortsetzen zu können. Es stellt sich also die Frage, wann eine Entität durch eine andere ersetzbar ist. Im Allgemeinen müssen dazu die ersetzende und die zu ersetzende Entität eine äquivalente Funktionalität anbieten. So müssen zum Beispiel beide eine Basisstation für den Mobilfunk darstellen oder nach obigem Beispiel Kenntnis über die Örtlichkeit zur Wegbeschreibung haben. Wie die Wegbeschreibung im Folgenden abläuft ist nicht von Interesse. Ausschlaggebend ist ein richtiger Weg zum Ziel. In manchen Fällen sind auch Unterschiede in den Funktionalitäten erlaubt. Dies ist jedoch im Speziellen zu prüfen.
In bestimmten Fällen ist zusätzlich eine Kompatibilität zwischen der ersetzenden und der zu ersetzenden Entität nötig. Diese Kompatibilität ist Grundlage für den Informationsaustausch für den reibungslosen Vorgang der Ersetzung, d.h. des Handovers, bei zustandsbehafteten Kommunikationen. In obigem Beispiel der Wegbeschreibung wäre dies durch eine gemeinsames Verständnis zwischen der Person, die ersetzt wird, und der ersetzenden Person, um dieser mitteilen zu können, wie weit die Wegbeschreibung bereits fortgeschritten ist. Dies bedeutet, dass entweder die zu ersetzende Person, die deutsch spricht, oder die ersetzende Person, die italienisch spricht, die jeweils andere Sprache beherrscht. Allerdings kann an dieser Stelle auch der Einsatz eines Dolmetschers erfolgen, der eine Transformation der deutschen Sprache in die italienische Sprache ermöglicht. Die Transformation bezieht sich dabei sowohl auf die Signatur, wie auf die Semantik, das Protokoll und den Kontext:

- Auf der Ebene der **Signatur** muss dafür eine Abbildung des Vokabulars, das durch die zu ersetzende Entität verwendet wird, auf das von der ersetzenden Entität verwendete Vokabular erfolgen. Bei der Transformation der bereits erfolgten Wegbeschreibung müsste in diesem Fall beispielsweise eine Abbildung des Wortes „München“ auf das Wort „Monaco“ durchgeführt werden.

- Desweiteren muss auch auf die **Semantik** des Vokabulars geachtet werden. Dabei muss die Bedeutung des Vokabulars durch die Abbildung erhalten bleiben. Bei der Wegbeschreibung muss hierfür beispielsweise sichergestellt werden, dass mit dem Wort „Monaco“ auch die Stadt in Deutschland gemeint ist und nicht der europäische Stadtstaat.

- Auf der Ebene des **Protokolls** muss bei der Transformation die Reihenfolge des Vokabulars, das von der zu ersetzenden Entität verwendet wird, auf eine korrekte Reihenfolge des transformierten Vokabulars der ersetzenden Entität abgebildet werden. Bei der Transformation der bereits erfolgten Wegbeschreibung von deutsch auf italienisch muss dabei zum Beispiel auf eine korrekte Stellung der Nomen mit ihren dazugehörigen Adjektiven geachtet werden.
- Für den **Kontext** als vierte Ebene gilt, dass durch die Transformation eine Abbildung der externen Einflüsse der zu ersetzenden Entität auf äquivalente externe Einflüsse der ersetzenden Entität durchgeführt werden muss. Das bedeutet zum Beispiel bei der Wegbeschreibung, dass die verschiedenen Blickwinkel der Wegbeschreiber abgestimmt werden müssen, da der deutsch sprechende Wegweiser die Alpen möglicherweise im Süden sieht, der italienisch sprechende Wegweiser diese aber im Norden. Durch die Transformation wird dann eine Wegbeschreibung des deutschsprechenden Wegweisers Richtung Alpen, d.h. die Himmelsrichtung Süden, in die Himmelsrichtung Norden für den italienischen Wegweiser abgebildet.

Als dritte Bedingung für die Ersetzbarkeit gilt, dass die zu ersetzende sowie die ersetzende Entität kompatibel zur dritten Entität, mit der sie in Kommunikation treten, sein müssen. Soll A durch B bezüglich der Kommunikation zu C ersetzbar sein, muss C also einerseits kompatibel zu A und andererseits kompatibel zu B sein, um mit A sowie mit B eine Kommunikation eingehen zu können. Anhand des Beispiels der Wegbeschreibung bedeutet dies, dass einerseits der Tourist und der Wegbeschreiber des ersten Wegstückes deutsch als Sprache verstehen und sprechen, und andererseits für die Beschreibung des folgenden Wegstückes zwischen dem zweiten Wegbeschreiber und dem Tourist die italienische Sprache gesprochen und verstanden wird.

Kapitel 2

Handover

2.1 Allgemeine Definition

Der Prozess der Ersetzung eines Kommunikationspartners während einer bestehenden Kommunikation wird allgemein als Handover (oder als Handoff) bezeichnet. Dabei wird im Folgenden die Entität, die durch die Ersetzung einen neuen Partner bekommt, als statisch (*static entity*), charakterisiert. Die beiden anderen Entitäten, die durch den Handover betroffen sind, werden als dynamische Entitäten (*dynamic entities*) bezeichnet (vgl. Abb. 2.1). Desweiteren lassen sich bei den dynamischen Entitäten die zu ersetzende bzw. der Vorgänger (*predecessor*) und die ersetzende bzw. der Nachfolger (*successor*) unterscheiden.

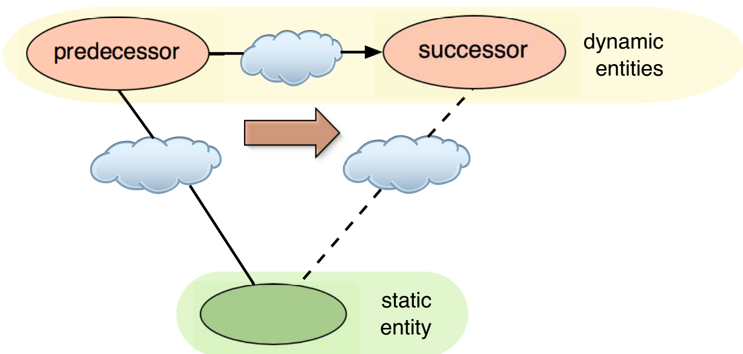

Abbildung 2.1: HANDOVER

Die Frage, welcher der beiden Kommunikationspartner ersetzt wird, ist vorerst nur von sekundärer Bedeutung, da dies stark von der zugrundeliegenden Problemstellung abhängt, bei der der Handover eingesetzt wird. Beispielsweise ist im Bereich Mobilfunk grundsätzlich nur von einem Handover die Rede, bei dem die wechselnde Partei in der Infrastruktur liegt (d.h. ein Wechsel der Basisstation). Denkbar wäre allerdings durchaus ein Handover bezüglich eines wechselnden Mobiltelefons, um eine gewisse Unabhängigkeit des Nutzers vom Endgerät zu erreichen (Personal Mobility) [KüLP 02].
Selbiges gilt im Rahmen des Mobile IP der IETF (Internet Engineering Task Force) (vgl. [MRP$^+$ 03]). Da durch die Lokalität, die durch die IP-Adressen der IP-Netze gegeben ist, einer Ortsveränderung im Allgemeinen eine Änderung der IP-Adresse folgt, wurde im Rahmen des Mobile IP folgende Strategie

eingeführt: Ein sogenannter Home Agent repräsentiert die IP-Adresse, unter der das mobile Endgerät ansprechbar ist. Befindet sich das mobile Endgerät, welches im Mobile IP den Namen Mobile Agent trägt, in einem fremden Netz, wird seine dortige Adresse - die sogenannte Care-Of-Adresse - dem Home Agent mitgeteilt. Empfängt der Home Agent nun eine Anfrage von einer Entität, dem sogenannten Correspondent Host, so leitet er diese mittels des Prinzips des Tunnelings an die bei ihm derzeit registrierte Care-Of-Adresse weiter. Der Handover entspricht in den vorhandenen Studien des Mobile IP grundsätzlich der Ersetzung des Foreign Agents bei einem Wechsel des Mobile Agents in ein anderes Fremdnetz, welches von einem anderen Foreign Agent verwaltet wird. Auf eine Ersetzung des Mobile Agents wird meist nicht weiter eingegangen.
Wie in der Definition angesprochen, muss die Kommunikation, auf die sich der Handover bezieht, zum Initiierungszeitpunkt bereits bestehen. Da eine Ersetzung eines Kommunikationspartners ohne eine bereits bestehende Kommunikation einem einfachen Kommunikationsaufbau zum neuen Partner entspricht, wird dies hier nicht als Handover bezeichnet. Überführt man beispielsweise ein Mobiltelefon im ausgeschalteten Zustand in ein Umfeld einer anderen Basisstation, so wird lediglich ein erneuter Anmeldevorgang durchgeführt, nicht jedoch ein Handover initiiert.

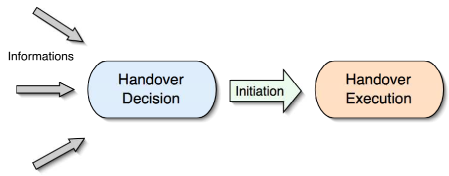

Abbildung 2.2: DECISION - EXECUTION

Der Handoverprozess ist im wesentlichen zweigeteilt. Der eigentlichen Handoverdurchführung (*Handover Execution*) geht die Handoverentscheidung (*Handover Decision*) voraus (vgl. [WoLi 97] und Abb. 2.2). Aufgabe der Handoverentscheidung (vgl. Kap. 2.3) ist die Initiierung eines Handovers, falls ein Kommunikationspartner nicht mehr als optimal angesehen wird. Bei der Handoverdurchführung (vgl. Kap. 2.4) muss daraufhin sichergestellt werden, dass alle Zustandsinformationen, die durch die Kommunikation zwischen Vorgänger und statischer Entität auf der Seite des Vorgängers entstanden sind, auch beim Nachfolger verfügbar gemacht werden. Dadurch ist ein nahtloser Übergang für die Kommunikation der statischen Entität beim Wechsel der dynamischen Entität möglich. Die Menge aller Zustandsinformationen wird dabei auch als *Session* bezeichnet. Derartige Zustandsinformationen bzw. Sessioninformationen können zum Beispiel Artikel eines bereits gefüllten Warenkorbs eines E-Commerce-Dienstes sein. Einen Handover, bei dem es zu einem Transfer von Sessioninformationen kommt, nennt man einen *stateful Handover* (vgl. [Snoe 03]). Entsprechend wird ein Handover ohne einen Sessiontransfer als *stateless* bezeichnet.

2.2 Klassifizierung

2.2.1 Funktionalitäten der dynamischen Entitäten

In Kapitel 1 wurde bereits angesprochen, dass für die Ersetzbarkeit beide dynamische Entitäten eine äquivalente Funktionalität bereitstellen müssen. Ein Beispiel hierfür ist eine Ersetzung eines Foreign Hosts im Mobile IP durch einen ähnlichen Foreign Host in einem anderen Fremdnetz. Unter

Umständen können die dynamischen Entitäten jedoch auch eine unterschiedliche Funktionalität zur Verfügung stellen. Um eine kurze Nachricht an einen Freund zu senden, wäre ein Beispiel für unterschiedliche Funktionalitäten ein Email- und ein SMS-Dienst. Beide bieten die Möglichkeit eine kurze Nachricht zu versenden, obwohl beim Email-Dienst auch weitaus mehr an Möglichkeiten verfügbar sind.
Sind die Funktionalitäten der dynamischen Entitäten identisch, ist zur Ersetzung ein *horizontaler Handover* ausreichend. Als *vertikaler Handover* bezeichnet man den Handover zwischen Entitäten, die eine unterschiedliche Funktionalität bereitstellen. Ohne Einschränkungen erfolgt die Ersetzung durch eine Entität, die eine höhere Funktionalität besitzt, da eine Nutzung der überschüssigen Funktionsmenge zwar möglich ist, jedoch keine Beeinträchtigung der ursprünglichen Funktionalität darstellt. Lassen sich Einschränkungen in Kauf nehmen, ergibt sich auch die Möglichkeit einen Handover zu einer Entität mit geringerer Funktionalität durchzuführen.
Ein Beispiel hierfür ist ein Handover in den sogenannten *Parked State*, der nur einen geringen Befehlssatz zur Verfügung stellt (z.B. {park,unpark}), um eine Session abzulegen und sie zu einem späteren Zeitpunkt wieder abzurufen. Ein Einsatzgebiet für diesen Handover bietet sich, wenn zum Zeitpunkt des Handovers kein Nachfolger mit einer vergleichbaren Funktionalität zur Verfügung steht. Eingesetzt wird der Handover in den Parked State zum Beispiel beim Parken von Telefongesprächen, die dann zu einem späteren Zeitpunkt wieder angefordert werden können (auch an einem anderen Telefonapparat).
In der Gegenwart bezieht man sich im Allgemeinen auf horizontale Handover. Zum Beispiel wird im GSM-Standard (vgl. [gsm]) nur ein Handover zu einer anderen Basisstation nach GSM-Standard angesprochen, obwohl auch die Möglichkeit besteht, als Nachfolger eine häusliche Telefonanlage (z.B. nach DECT-Standard) anzusprechen, die eine andere Funktionalität bereitstellt. Einschränkungen im DECT-Standard ergeben sich zum Beispiel durch das Verschicken von Kurznachrichten. Durch den Einsatz von UMTS bekommt jedoch auch der vertikale Handover einen höheren Stellenwert, da aufgrund der noch geringen Netzabdeckung in den Anfängen von UMTS oft Handover in das GSM Netz stattfinden werden.

2.2.2 Transparenz des Handover

Durch den Handover im Mobilfunk nach GSM-Standard ergibt sich durch die Ersetzung der Basisstation eine geringe, aber durchaus nachvollziehbare Verzögerung in der Sprachübertragung von wenigen Millisekunden. Allgemein lässt sich festhalten, dass ein Handover natürlich immer schnellstmöglich durchgeführt werden sollte. Einen Handover mit minimaler Verzögerung nennt man einen *Fast Handover* [MKS$^+$ 01]. Dabei bestimmt sich die Verzögerungszeit des Handovers aus der Zeitspanne, in der die statische Entität keine Kommunikationsmöglichkeit mit einer dynamischen Entität hat. Ziel ist die Zeitspanne der Unterbrechung der Kommunikation so gering wie möglich zu halten. Bei einer sanften Übergabe der Kommunikationsverbindung (vgl. Kap. 2.2.6) ist ein Handover sogar ohne Verzögerung erzielbar.
Desweiteren trägt bei der Handoverdurchführung die Minimierung des Datenverlustes, d.h. des Verlustes von Sessioninformationen, die beim Vorgänger durch die ursprüngliche Kommunikation entstanden sind, zur Transparenz des Handovers für die statische Entität bei. Gehen Sessioninformationen während des Sessiontransfers bei der Handoverdurchführung verloren, kann es zu schweren Fehlern in der Kommunikation mit dem Nachfolger führen. Eine Ersetzung ohne oder mit minimalem Datenverlust nennt man *Smooth Handover* [MKS$^+$ 01]. Ziel ist die Beeinträchtigung der Kommunikation für die statische Entität durch Datenverlust so gering wie möglich zu halten. Deshalb muss durch den

Handover für einen ordnungsgemäßen Transfer möglichst aller Sessioninformationen gesorgt werden. Während beim Fast Handover die höchste Priorität auf die Minimierung der Verzögerung durch den Handover gelegt wird, liegt beim Smooth Handover die höchste Priorität auf der Minimierung des Datenverlustes.
Optimal wäre eine vollständige Transparenz der Ersetzung des Kommunikationspartners für die statische Entität. Erzielt oder am besten approximiert wird dies durch einen *Seamless Handover* [MKS$^+$ 01], der sich aus einer geeigneten Kombination aus einem Fast und Smooth Handover ergibt. Nur in seltenen Fällen kann eine vollständige Transparenz erreicht werden. Deshalb muss oft je nach Art der Kommunikation abgewägt werden, welche Komponente der Transparenz als wichtiger anzusehen ist. Erhält die Minimierung der Verzögerungszeit durch den Handover Vorrang vor der Minimierung des Datenverlustes, ist möglicherweise ein Handover ohne Rücksicht auf den Sessiontransfer vorzuziehen. Ein Beispiel hierfür ist ein Handover im Bereich VoiceOverIP (vgl. [CaPa 98]) oder beim Mobilfunk, da ein kurzzeitiger Datenverlust eher annehmbar ist als eine starke Verzögerung in der menschlichen Sprache. Handelt es sich demgegenüber um eine Datenübertragung, bei der es auf Korrektheit der übermittelten Daten ankommt (z.B. File Transfer), ist ein Handover ohne vollständige Datenweiterleitung nicht denkbar.

2.2.3 Kontrollierende Entitäten

Bei sessionbehafteten Handovern muss ein Transfer der Session vom Vorgänger zum Nachfolger durchgeführt werden. Um Sessioninformationen, die bei der zu ersetzenden Entität entstanden sind, an deren Nachfolger zu übermitteln, muss zwischen der zu ersetzenden und der ersetzenden Entität eine Kommunikation stattfinden. Dafür müssen Vorgänger und Nachfolger nach Kapitel 1 kompatibel sein. Diese Kompatibilität ist bei sessionbehafteten Handovern Bedingung für die Ersetzbarkeit. Durch den aufgebauten Kommunikationskanal, dem sogenannten *Backlink*, erfolgt der Datenaustausch zwischen den dynamischen Entitäten. Dabei wird der Kommunikationskanal bei einem Datenaustausch, der von der nachfolgenden Partei angestoßen wird, als *Backward Backlink* und der entsprechende Handover als *Backward Handover* bezeichnet. „Backward“ bezieht sich auf die rückwärtige Nutzung des Backlinks vom Nachfolger zum Vorgänger. Im Gegensatz dazu gibt es den *Forward Backlink*, bzw. *Forward Handover*, bei dem der Kommunikationskanal für die Sessionübertragung in Richtung des Nachfolgers aufgebaut wird. Gibt es keine Möglichkeit zum Aufbau eines Kommunikationskanals zwischen den dynamischen Entitäten, nennt man dies eine *No-Backlink* Architektur.
Existiert ein Forward Backlink zwischen den dynamischen Entitäten, so kann der Vorgänger den Transfer der Session, die bei ihm durch die Kommunikation mit der statischen Entität entstanden ist, anstoßen. Der Vorgänger hat in diesem Fall die Kontrolle über die Durchführung des Handovers. Man nennt eine derartige Entität auch kontrollierende Entität. Bei Existenz eines Forward Backlinks ist deshalb ein vom Vorgänger kontrollierter Handover (*predecessor controlled*) einsetzbar. Liegt zwischen den dynamischen Entitäten demgegenüber ein Backward Backlink, so kann ein Handover mit Kontrolle beim Nachfolger (*successor controlled*) durchgeführt werden. Dabei fordert der Nachfolger die Session vom Vorgänger an.
Sind Vorgänger und Nachfolger nicht kompatibel bezüglich der Sessionübertragung, d.h. es existiert kein Backlink, kann eine Kompatibilität auch durch die Zuhilfenahme einer dritten Entität erreicht werden. Diese fungiert als eine Art Dolmetscher zwischen den beiden dynamischen Entitäten. D.h. sie muss einerseits kompatibel zum Vorgänger und andererseits kompatibel zum Nachfolger sein, um die Session vom Vorgänger anzufordern und an den Nachfolger weiterzuleiten. Diese Entität übernimmt dadurch die Kontrolle der Handoverdurchführung und wird deshalb, wie bei Existenz eines Back-

links, als kontrollierende Entität bezeichnet. Beim Transfer der Session muss sie eine Transformation zwischen den unterschiedlichen Kommunikationssprachen, die durch die dynamischen Entitäten eingesetzt werden, auf den vier Ebenen der Interoperabilität anwenden können. Ein Beispiel für eine syntaktische Transformation ist eine Umwandlung des Sessionidentifikators (vgl. Kap. 2.4.1) beim Vorgänger namens „SID“ in den Sessionidentifikator beim Nachfolger mit dem Namen „SessionID“. Beide Werte sind Identifikatoren für eine Session, besitzen jedoch bei den unterschiedlichen Entitäten eine unterschiedliche Syntax.
Den predecessor controlled Handovern und successor controlled Handovern kommen deshalb weitere Variationen bezüglich der kontrollierenden Entität hinzu. Möglich ist, dass die statische Entität (*static entity controlled*) oder eine externe Entität (*external controlled*) die Dolmetscherfunktion einnimmt (vgl. Abb. 2.3). Zum Transfer der Session kann sie die ursprüngliche Session vom Vorgänger anfordern und die modifizierte Session an den Nachfolger übergeben.
Neben der Übergabe der Session kann auch die Wahl des Nachfolgers bei der kontrollierenden Entität angesiedelt werden, wenn zum Zeitpunkt der Initiierung noch kein passender Nachfolger zur Verfügung steht (vgl. Kap. 2.2.5).

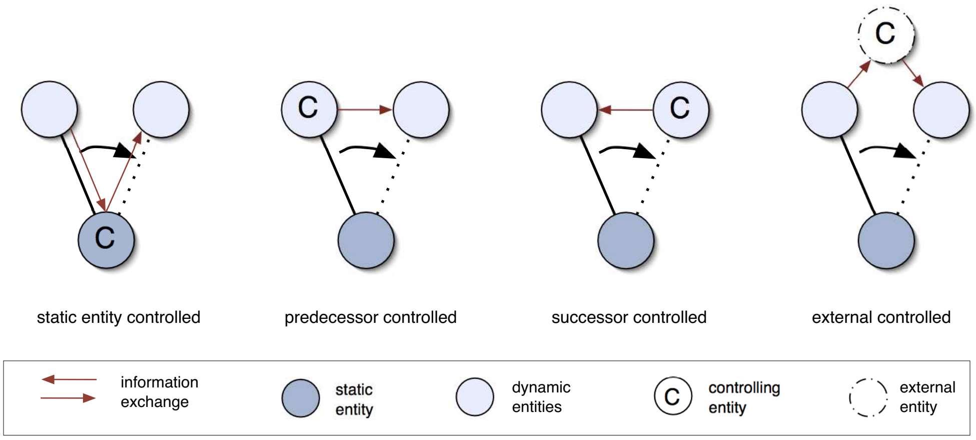

Abbildung 2.3: KONTROLLIERENDE ENTITÄTEN

2.2.4 Initiierende Entität

Einer Handoverdurchführung geht die Handoverinitiierung voraus. Die Handoverinitiierung ist Produkt der Handoverentscheidung und gilt als die die Durchführung einleitende Handlung. Anzusiedeln ist die Initiierung auf Seite der statischen Entität (*static entity initiation*) oder auf Seite der dynamischen Entitäten (*dynamic entity initiation*). Zweiteres bezieht sich im Allgemeinen auf die zu ersetzende Entität (*predecessor initiation*), aber auch eine Initiierung des Nachfolgers (*successor initiation*) ist in manchen Fällen anwendbar (vgl. [MKS$^+$ 01]). Wie bei der Kontrolle kann auch hier eine Initiierung von einer externen Entität veranlasst werden (*external initiation*).
Wird durch die initiierende Entität der Entschluss gefasst einen Handover auszulösen, wendet sie sich an die kontrollierende Entität, die den Handover daraufhin durchführt. Ist zur Zeit der Initiierung bereits ein Nachfolger bekannt, wird diese Information an die kontrollierende Entität weitergeleitet. Bei Unwissenheit über eine „bessere“ dynamische Entität, kann bei der Initiierung diese Information auch weggelassen werden, worauf die Wahl des Nachfolgers bei der kontrollierenden Entität durchgeführt

wird (vgl. Kap. 2.2.5).
Beispiel für die Initiierung ist im Mobilfunk eine Initiierung aufgrund der Abnahme der Empfangsstärke der derzeitigen Basisstation. Die Auslösung erfolgt im GSM-Standard von der ursprünglichen Basisstation (predecessor initiation).

2.2.5 Zeitpunkt der Initiierung

Der Initiierung kann die Wahl des Nachfolgers bereits vorausgegangen sein, d.h. ein Nachfolger wurde bereits bei der Handoverentscheidung festgelegt, oder die Wahl des Nachfolgers findet nachträglich bei der kontrollierenden Entität statt. Lassen sich allgemein Anzeichen für das Eintreten eines Problems erkennen und eine Gegenmaßnahme einleiten bevor das Problem auftritt, so spricht man von einem proaktivem Ansatz. Ist im Gegensatz dazu das Problem vor dem Eintreten nicht ersichtlich und/oder keine passende Gegenmaßnahme verfügbar, so kann nur noch ein reaktiver Ansatz eingesetzt werden. Die Einleitung einer Gegenmaßnahme kann erst nach dem Eintritt des Problems erfolgen. Äquivalent nennt man einen Handover, der aufgrund eines ex ante erkennbaren Problems initiiert wird, einen *proaktiven* Handover. Wenn eine Vermutung im Vorhinein ausgeschlossen ist, und man direkt mit dem Problemeintritt konfrontiert wird, muss ein *reaktiver* Handover (vgl. [MKS+ 01]) folgen. In der Literatur wird anstatt „proaktiv“ auch der Ausdruck „prediktiv“ verwendet (vgl. [Perk 00]). Das Wort „prediktiv“ soll in diesem Zusammenhang ausdrücken, dass man den Problemeintritt vorhersagen kann.
Bei genauerer Betrachtung lässt sich die Problemerkennung auf die Handoverentscheidung (vgl. Kap. 2.3) zurückführen. Genauer gesagt tritt das entscheidende Anzeichen auf die Problemsituation genau dann ein, wenn es einen „besseren“ Partner gibt als der derzeitige. Proaktiv bedeutet demzufolge, dass eine Entscheidung zugunsten eines alternativen Kommunikationspartners gefallen ist, und dieser nun für die weitere Nutzung bevorzugt wird. Es gibt also eine Lösung für das zu erwartende Problem, und die entsprechende Gegenmaßnahme kann vor dem Problemeintritt eingeleitet werden. Ist in diesem Sinne der Problemeintritt nicht vorhersehbar, muss nach Auftreten des Problems reagiert werden. Zur Zeit des Problemeintritts besteht noch keine Möglichkeit das Problem zu lösen.
Ein Beispiel für Proaktivität ist beim Zellenwechsel im GSM-Standard durch das periodische Messen der Empfangsstärke der 16 stärksten Basisstationen gegeben. Nimmt das Empfangssignal der eigenen Basisstation stetig ab, so ist in naher Zukunft mit dem Verlassen des Verfügungsbereichs dieser Basisstation zu rechnen. Existieren nun eine oder mehrere Basisstationen deren Empfangspegel einen höheren Wert aufweisen, kann dem Abbruch der Kommunikation durch geeignete Maßnahmen, wie dem Wechsel zu einer im Sinne der Empfangsstärke „besseren“ Basisstation, vorgebeugt werden. Tritt allerdings ein unerwarteter Einbruch der Bandbreite ein, und es existieren keine „besseren“ Basisstationen, z.B. durch das Befahren eines Tunnels, so kann dies nur zu einem reaktiven Handover führen, da das Problem des Verbindungsabbruchs eintritt bevor eine passende Lösung gefunden wurde. Im GSM-Standard ist keine Möglichkeit zur Durchführung eines reaktiven Handovers spezifiziert. Eine geeignete Maßnahme wäre ein Handover in den Parked State (vgl. Kapitel 2.2.1) bis das Ende des Tunnels erreicht ist.

2.2.6 Anzahl paralleler Kommunikationskanäle

Um eine bestimmte Problemstellung zu lösen, braucht eine Entität mindestens einen Partner, mit dem sie kommunizieren kann. Eine andere Möglichkeit ist, mehrere Kommunikationskanäle zu verschie-

denen Partnern parallel zu verwalten. Daraus folgt eine Unterscheidung anhand der Anzahl der parallelen Kommunikationskanäle, die bei einem Handover eingesetzt werden.
Ist die Kommunikation mit dem derzeitigen Partner nicht mehr als optimal einzustufen, ist es nötig, diesen Kommunikationskanal abzubauen und einen neuen Kommunikationskanal zu einem anderen Partner aufzubauen. Zu jedem Zeitpunkt existiert also nur ein Kommunikationskanal ausgehend von der statischen Entität. Es handelt sich in diesem Fall um einen *Hard Handover*, da es einen harten Einschnitt zwischen der ursprünglichen Kommunikation mit dem Vorgänger und der neuen Kommunikation mit dem Nachfolger gibt.

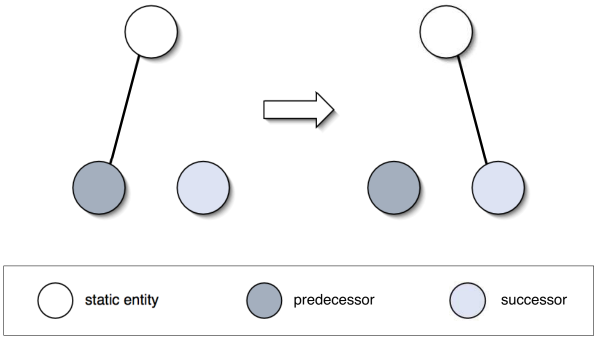

Abbildung 2.4: HARD HANDOVER

In besonderen Fällen ist eine Variante zum Hard Handover, anstatt den ursprünglichen Kommunikationskanal abzubauen und anschliessend den neuen Kommunikationskanal aufzubauen, den alten Kommunikationskanal parallel zum neuen aktiv zu halten. Danach kann, wenn das Problem eintritt, bei Bedarf der alte Kommunikationskanal abgebaut werden. Eine derartige Ersetzung nennt man wegen des sanften Übergangs *Soft Handover*. Zum Zeitpunkt der Ersetzung hat hier die statische Entität gleichzeitig mehrere Kommunikationskanäle zu verschiedenen dynamischen Entitäten. Die Menge aller Kommunikationspartner, die eine Kommunikation mit der statischen Entität betreiben, nennt man auch *active set*.

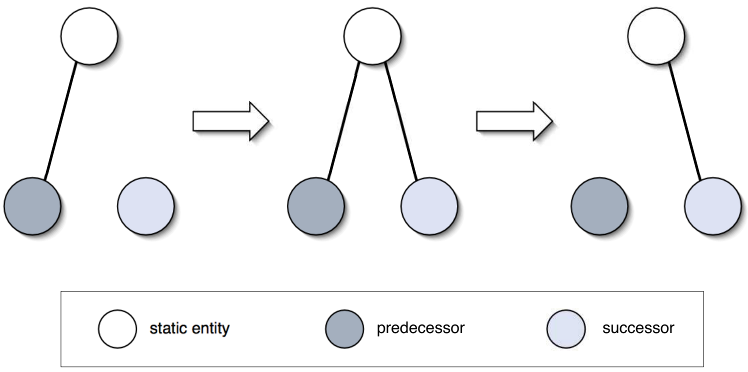

Abbildung 2.5: SOFT HANDOVER

Ein Beispiel, welches den Unterschied zwischen dem Hard Handover und dem Soft Handover veranschaulicht, ist in [WoLi 97] aufgeführt. Ein Hard Handover ist vergleichbar mit der Ersetzung eines Schwimmers durch einen anderen Schwimmer seiner Mannschaft bei einem Schwimmturnier. Nähert sich ein Schwimmer dem Ende seiner Strecke, muss sein Nachfolger warten bis er den Beckenrand

erreicht hat, bevor dieser starten darf. Anders ist es im Staffellauf. Dort laufen Vorgänger und Nachfolger noch eine kurze Strecke parallel, um das Staffelholz zu übergeben. Dies ist vergleichbar mit einem Soft Handover, bei dem ein Kommunikationskanal zur ursprünglichen und zur zukünftigen Entität parallel existieren kann.
Auch im Mobilfunk kann man diese Alternativen wiederfinden. Im GSM-Standard ist aufgrund der Netztechnologie nur der Hard Handover einsetzbar. Im UMTS-Standard hingegen ist auch die Aufrechterhaltung mehrerer paralleler Kommunikationskanäle zu unterschiedlichen Basisstationen möglich.

2.3 Handoverentscheidung

2.3.1 Grundlagen der Handoverentscheidung

Informationsbeschaffung und -evaluierung

Nachdem das Kapitel 2.2 sich allgemein mit den verschiedenen Arten des Handovers beschäftigte, soll nun der eigentliche Handovervorgang, der sich in Handoverentscheidung und Handoverdurchführung gliedert, betrachtet werden.
Grundlage für die Handoverentscheidung ist das Sammeln und Auswerten von Informationen mit dem Ziel, bei der Veränderung dieser Informationen gegebenenfalls den Handover zu initiieren. Informationen erhält die initiierende Entität zum Beispiel durch Sensordaten eines Context Observers (vgl. [StLP 03]). Eine Auswertung der Informationen kann entweder periodisch oder ereignisgesteuert vorgenommen werden. Für die periodische Auswertung muss die initiierende Entität in bestimmten Zeitabständen die entsprechenden Informationen anfordern. Bei der ereignisgesteuerten Auswertung erhält die initiierende Entität bei jeder Änderung der Informationen aktuelle Daten. Anhand der ausgewerteten Informationen entscheidet die initiierende Entität, ob eine Handoverinitiierung durchgeführt werden soll.

Kontextuelle Entscheidungsgründe

Nach [Caro 00] gibt es drei Gründe, die eine Handoverinitiierung herbeirufen können. Der erste Grund beruht auf den Eintritt des Problems, worauf gezwungenermaßen reagiert werden muss. [Caro 00] spricht in diesem Fall von einem *Forced Handover*. Der Ersetzungszwang tritt aufgrund des Verlustes von Qualities of Service (QoS) wie z.B. der unvorhergesehene Abbruch des Kommunikationskanals ein.
Tritt anstelle des Verlustes nur eine Veränderung der QoS, wird in [Caro 00] von einem *QoS-Aware Handover* gesprochen. Für die Initiierung des QoS-Aware Handover, d.h. zur Ersetzung des derzeitigen Partners, spricht eine durch den Wechsel erzeugte Leistungssteigerung bei gleichen Kosten oder die gleiche Leistung bei niedrigeren Kosten ([M.F.]).
Als dritte Möglichkeit wird bei [Caro 00] der *Location-Aware Handover* betrachtet. Dieser bezieht sich auf den Handover bei einer Aufenthaltsänderung der statischen Entität (Bsp.: Verlassen einer Zelle im Mobilfunk). Zur Vervollständigung sollte man hier nicht nur die Position der statischen Entität, sondern des gesamten örtlichen Umfeldes, d.h. alle ortsspezifischen Informationen, in Betracht ziehen.

Bei genauerer Analyse der eben beschriebenen Entscheidungsgründe (Forced -, QoS-Aware - & Location-Aware Handover) stellt man fest, dass es sich bei allen drei Punkten im Grunde genommen um eine Veränderung des Kontextes als Summe aller externen Einflüsse auf die Kommunikation (vgl. Kap. 1) handelt. Um die kontextuellen Entscheidungsgründe genauer zu spezifizieren, wird im Folgenden die Terminologie von [StLP 03, SLPF 03b]) verwendet. Nach [StLP 03, SLPF 03b] wird die Teilmenge aller bekannten Kontextinformationen, die eine spezielle Aufgabenstellung in irgendeiner Art und Weise beeinflusst, als relevant bezeichnet. Kontextinformationen sind hierbei Informationen, die einen Zustand bezüglich eines Aspektes und einer dazugehörigen Skala charakterisieren. Aspekte zeichnen sich durch eine Menge zusammengehöriger Skalen aus. Skalen geben Wertebereiche, d.h. alle gültigen Ausprägungen für Kontextinformationen vor.
Eine mögliche Darstellung des Kontextes bezüglich des Aspektes Ort könnte zum Beispiel auf der Skala der Gauss-Krüger-Koordinaten oder der WGS84-Koordinaten basieren, deren Wertebereiche alle gültigen Ausprägungen der Koordinaten sind. Eine Kontextinformation könnte in diesem Fall die Gestalt eines Tupels (Bsp.: GK(4472108,5333376) oder WGS84(44.586015,6.9536576)) annehmen. Der Aspekt Zeit findet eine repräsentative Darstellung in der Skala GPS-Time (Global Positioning System Time) [gps] oder UTC (Universal Time Coordinated) [utc].
Eine Projektion des Kontextes auf die Aspekte Ort oder Zeit sind jedoch meist nur eine geringe Teilmenge aller relevanten Kontextinformationen. Allerdings sind Ort und Zeit aufgrund der einfachen Bestimmung die Informationen, die als am repräsentativsten gelten und deshalb seit einigen Jahren stark in der Forschung (z.B. durch Location Based Services) vertreten sind. Genauso gut können Kontextinformationen bezüglich den Aspekten Wetter (dargestellt durch Temperatur (z.B. Kelvin-Skala), Niederschlag (in der Skala Millimeter/Quadratmeter)), Geräuschpegel (z.B. dB-Skala, phon-Skala), das allgemeine Wohlbefinden (dargestellt durch Krankheit, Hunger, usw.) für die Handoverentscheidung als relevant gelten.
Zusätzlich zur Ausprägung der Kontextinformationen (Content) sind gerade für die Entscheidung die dazugehörigen Metainformationen (Metadata) über die Kontextinformationen von Bedeutung. Dazu zählen Informationen über die Validität, Reliabilität und Objektivität der Kontextinformationen.
Validität beschreibt die Gültigkeit der Kontextinformationen. Es stellt sich die Frage, ob die Kontextinformationen das repräsentieren, was sie vorgeben zu sein. Eine typische Grösse, die für valide Kontextinformationen notwendig ist, ist die Aktualität. Sind Kontextinformationen, das Wetter betreffend, zum Beispiel im Flugverkehr nicht aktuell genug, kann dies zu schweren Fehlern führen. Metadaten über die Aktualität der Kontextinformationen sind in den meisten Fällen von großer Bedeutung und sind generell ein Entscheidungskriterium für den Handover, da die höchste Genauigkeit der Kontextinformationen keine Rolle mehr spielt, wenn diese überhaupt nicht mehr zutreffen.
Reliabilität steht für die Verlässlichkeit der Kontextinformationen. Verlässlichkeit bedeutet hier inwiefern man darauf vertrauen kann, dass die Daten exakt sind. Ein wichtiger Punkt ist dabei sicherlich die Genauigkeit der Kontextinformationen. Ziel ist es immer einen möglichst geringen Messfehler zu erhalten.
Die Objektivität ergibt sich aus der externen Beeinflussung der Kontextinformationen. In Kontextinformationen enthaltene externe Effekte tragen nicht unwesentlich zur unerwünschten Verfälschung der Realität bei.
Zusammenfassend lässt sich sagen, dass jede Änderung des Kontextes, d.h. des Contents oder der Metadaten, einen Grund für eine Handoverinitiierung darstellen können.

2.3.2 Handoverinitiierung

Verfügungsbereiche (Scopes)

Nachdem nun die notwendige Basis für die Handoverentscheidung detailliert betrachtet wurde, bleibt noch der Vorgang der Initiierung als Fragestellung offen. Dazu wird der Zeitraum, in dem der Handover eingeleitet wird, weiter eingeschränkt.
Anhand der für die zugrundeliegende Problemstellung als relevant anzusehenden Aspekte kann für jede Entität ein Verfügungsbereich (Scope) festgelegt werden. Durch den Verfügungsbereich wird eine Entität bezüglich ihrer relevanten Aspekte charakterisiert. Die Menge der relevanten Aspekte ist im Normalfall statisch, da die relevanten Einflußfaktoren meist spezifisch für diese Art der Problemstellung sind. Ein Routingdienst wird dementsprechend immer ortsabhängig agieren, eine Kalenderdienst immer abhängig von der Zeit. An den einzelnen Verfügungsbereichen der für diese Problemstellung möglichen Kommunikationspartner erkennt man Unterschiede sowie Gemeinsamkeiten der verschiedenen Kommunikationspartner. Beispielsweise unterscheiden sich ein Routendienst für Deutschland und ein äquivalenter Routendienst für ganz Europa nur durch ihren örtlichen Scope.
Je nach Art der Skala, in der bezüglich dieses Aspektes die Kontextinformationen beschrieben werden, kann der Scope kontinuierliche oder diskrete Werte vorgeben. Ein kontinuierlicher Wertebereich ergibt sich zum Beispiel bei einer Skala, die Entfernungen in Metern angibt, ein diskreter Wertebereich in einer Skala für Postleitzahlen.
Betrachtet man nun verschiedene Kommunikationspartner anhand ihrer Scopes, können sich folgende drei Konstellationen ergeben (vgl. [KaBl 02] und Abb. 2.6):

- **Disjunkte Verfügungsbereiche** (vgl. Abb. 2.6c):
  Disjunkte Verfügungsbereiche ergeben sich, wenn sich die Bereiche in keinem Punkt schneiden. Es handelt sich um völlig getrennte Scopes. Horizontale Handover sind bei dieser Konstellation in der Regel nicht möglich, da zeitweise Nichtverfügbarkeit vorprogrammiert ist. Alternativ könnte die Möglichkeit zur Durchführung eines vertikalen Handovers bestehen.

- **Überlappende Verfügungsbereiche** (vgl. Abb. 2.6a):
  Handelt es sich um überlappende Verfügungsbereiche, gibt es eine Schnittmenge, in der beide Kommunikationspartner zur Verfügung stehen. Die Durchführung eines Handovers muss in dieser Schnittmenge eingeleitet werden. Dabei bietet sich diese Konstellation an, um einen Soft Handover zu verwenden, da durch die Überlappung die Möglichkeit besteht, zu einem Zeitpunkt nicht nur einen Kommunikationskanal zu einem Partner offen zu halten, sondern gleichzeitig mit Vorgänger und Nachfolger verbunden zu sein. Ein sanftes Umschalten ist nur in diesem Szenario anwendbar. Die örtliche Überlappung der Verfügungsbereiche findet man beispielsweise im Mobilfunk. Dort sind die Zellen, die hier eine Projektion des gesamten Verfügungsbereiches auf den Aspekt Ort darstellen, zur Entscheidungsfindung von Interesse. Ein Handover wird stets in der durch die Überlappung entstehenden Schnittmenge durchgeführt. Der GSM-Standard bittet hier nur die Möglichkeit des Hard Handovers an. Im UMTS-Standard hingegen wird im Allgemeinen der Soft Handover eingesetzt.

- **Adjazente Verfügungsbereiche** (vgl. Abb. 2.6b):
  Da Redundanzen in den Verfügungsbereichen oft vermieden werden (z.B. im Mobilfunk), sind Scopes oft nicht überlappend, aber auch nicht disjunkt. Die dritte Konstellation, die sich dadurch ergibt, sind adjazente Verfügungsbereiche. Adjazente Verfügungsbereiche berühren sich an ihren Grenzen. Ein Handover zwischen den betreffenden Entitäten muss genau dort erfolgen.

Der eingesetzte Handover muss folglich ein Hard Handover sein, da nur ein harter Wechsel an den Grenzen möglich ist. Manchmal wird dabei von einem *idealen* Handover gesprochen.

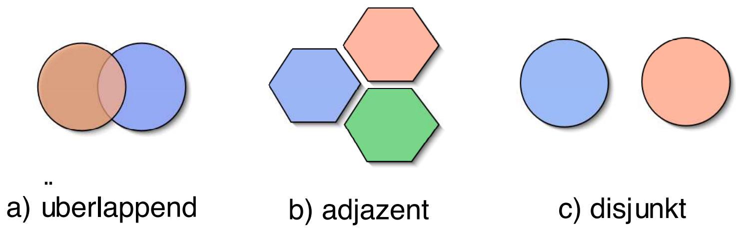

Abbildung 2.6: ARTEN DER KONSTELLATION MEHRERER SCOPES

Ein Beispiel für das Auftreten unterschiedlicher Scopekonstellationen stellen Angebote verschiedener Stromerzeuger dar. Unterschieden werden könnte dabei Strom von Solarzellen und von Nachtstromanbietern. Tagsüber ist eine Stromversorgung per Solarzellen möglich, nachts kann auf den günstigen Nachtstrom zurückgegriffen werden. Eine Entscheidung, welcher Stromanbieter gewählt wird, wird anhand der Tageszeit gefällt. Dazu passend gibt es den als relevant zu betrachtenden Aspekt Tageszeit, der eine Skala mit den diskreten Werten {Tag, Nacht} besitzt. Die Abnahme von Solarstrom hat den Verfügungsbereich Tag, hingegen gilt als Verfügungsbereich des Nachtstroms die Nacht. Die beiden Scopes wären dann adjazent. Wenn der Tag endet, beginnt unverzüglich die Nacht, und wenn die Nacht endet, beginnt der Tag.
Da man eine Entscheidung bezüglich dieser Skala als Mensch in der Regel schwer fällen kann, zieht man eine andere Skala zur Hilfe heran, eine Referenzskala, wie zum Beispiel die Skala, die die aktuelle Stunde des Tages {0,...,23} angibt. Nutzt man die für die Umrechnung zwischen verschiedenen Skalen eines Aspektes definierte Intraoperation (vgl. [SLPF 03b]), könnte man auf die Konstellation der Überlappung kommen, wenn der Tag eine Überschneidung mit der Nacht hat (Bsp.: 6-9 Uhr). Genauso gut könnten sich hier aber auch disjunkte Verfügungsbereiche ergeben, wenn bei der Umrechnung Werte des Wertebereichs nicht angenommen werden (Bsp.: Tag: 8-18 Uhr - Nacht: 20-6 Uhr).

Kosten-/Leistungsanalyse

In Kapitel 2.3.1 wurde bereits erwähnt, dass für eine Ersetzung des derzeitigen Partners eine durch den Handover erzeugte Leistungssteigerung bei gleichen Kosten oder die gleiche Leistung bei niedrigeren Kosten (vgl. [M.F.]) sprechen. Dies steht im Einklang mit dem ökonomischen Prinzip (vgl. [oek 95]), das aus der Betriebswirtschaft bekannt ist und besagt, dass bei einer Minimierung des Inputs bzw. dessen Kosten keine gleichzeitige Maximierung des Outputs, bzw. der Leistung möglich ist. Bei einer Extremwertbetrachtung existiert also nur die Möglichkeit, bei festem Input den Output zu maximieren (Maximalprinzip) oder bei festem Output den Input zu minimieren (Minimalprinzip). Angewendet auf die Handoverentscheidung ergibt sich der Input aus den aufzuwendenden Kosten, wie beispielsweise Wartezeiten, monetäre Kosten (z.B. Onlinekosten abgerechnet nach Zeit oder Datenvolumen), usw. Nicht zu vernachlässigen sind natürlich auch die durch den Handover entstehenden Kosten, die durch den zusätzlichen Datentransfer, die resultierende Wartezeit, etc. entstehen. Diese können einen beträchtlichen Anteil der Gesamtkosten ausmachen.
Folgt man weiter dieser Extremwertbetrachtung, so sollte nach der Veranschlagung eines festes Budget zur Lösung der Aufgabenstellung die aus diesem Einsatz größtmögliche Leistung folgen (Ma-

ximalprinzip). Ein Beispiel ist die bestmöglichste Genauigkeit bezüglich einer Ortsbestimmung zu erhalten, die mit einem festgelegten Kapital erzielbar ist. Andererseits ist es denkbar eine bestimmte Genauigkeit mit dem geringstmöglichen Kapitaleinsatz zu erzielen. Ergebnis ist eine feste Leistung mit dem minimalen Budget (Minimalprinzip).
Um eine Handoverentscheidung durchzuführen, müssen also alle eingehenden und ausgehenden Parameter (bzw. Returnwerte) einer jeden Verbindung abhängig des zugrundeliegenden Prinzips detailliert betrachtet werden. Will man Kosten minimieren, müssen die Kosten für die eingehenden Parameter analysiert werden, will man die Leistung maximieren, müssen die Leistungen der ausgehende Parameter untersucht werden. Das bedeutet nun, dass man für die Energieversorgung (vgl. obiges Beispiel) nach dem Minimalprinzip einen Anbieter kontaktiert, der beispielsweise für 10 KWh Kosten im Wert von 1 Euro veranschlagt - unabhängig von der Tageszeit - da dieser für eine Leistung von 10 kWh den günstigsten Preis berechnet. Möglicherweise gibt es jedoch auch teuere Anbieter, die aber auch den günstigeren Nachtstrom liefern je nachdem wann die Energie benötigt wird. Folglich fehlt in dieser recht statischen Analyse eine weiterere wichtige Komponente, damit eine vollständige Kosten-/Leistungsanalyse durchgeführt werden kann. Kosten und Leistungen sind abhängig vom Kontext. In diesem Fall ist der Kontext bezüglich des relevanten Aspektes Tageszeit zu betrachten, da je nachdem wann die Energie benötigt wird, unterschiedliche Kosten entstehen. Die Kosten-/Leistungsanalyse bezieht sich also auf die drei Komponenten Inputparameter, Kontext und Outputparameter (vgl. Abb. 2.7).

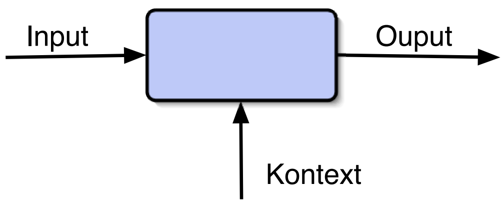

Abbildung 2.7: INPUT-KONTEXT-OUTPUT

Strategien zur Optimierung des Initiierungszeitpunktes

Ist nun der Fall eingetreten, dass eine Kosten-/Leistungsanalyse für die Ersetzung des Partners spricht, folgt eine Handoverinitiierung bei einer (vgl. [KüLP 02])

- **unmittelbaren Entscheidung**
  sofort bei Eintreten der Situation, d.h. sobald die Kosten eines anderen Partners niedriger sind als die des derzeitigen oder die Leistung höher ist.

- **schwellwertbasierten Entscheidung**
  nicht sofort bei Eintreten der Gründe, sondern erst wenn ein Schwellwert (Threshold) bei der Kostenanalyse über- bzw. bei der Leistungsanalyse unterschritten wurde und ein „besserer“ Partner zur Verfügung steht.

- **hysteresebasierten Entscheidung**
  sobald die Differenz der Kosten bzw. Leistung des derzeitigen und des zukünftigen Partners um einen bestimmten Wert (Hysterese) überschreitet.

- **schwellwert- und hysteresebasierten Entscheidung**
  wenn die Kosten (bzw. Leistungen) über (bzw. unter) einen Schwellwert steigen (bzw. sinken) und Kosten (bzw. Leistungen) des zukünftigen Partners diese um einen bestimmten Wert (Hysterese) unterbieten (bzw. übersteigen).

2.4 Handoverdurchführung

2.4.1 Session Transfer

Ist eine Handoverentscheidung herbeigeführt worden, geht man durch die Initiierung zur Durchführung über. Zur Durchführung zählen einerseits Aktionen, die den Kommunikationskanal zum ursprünglichen Partner abbauen und einen Kommunikationskanal zum zukünftigen Partner aufbauen, und andererseits alle Maßnahmen, um die ursprüngliche Session des Vorgängers beim Nachfolger wieder herzustellen.
Zum Transfer einer Session gehört die Generierung der Session beim Nachfolger sowie die Übergabe aller Sessioninformationen. Denkbar sind hierfür die Varianten:

- *Session Forwarding*
- *Session Recreation*

Session Forwarding bezeichnet den Mechanismus, eine Session als Ganzes vom Vorgänger zum Nachfolger zu transferieren. Die Session muss dafür durch eine Datenstruktur repräsentiert werden, in der alle für die Session relevanten Informationen abgelegt sind. Meist heißt dies, dass in der Datenstruktur ein eineindeutiger Identifikator für diese Session (im Folgenden mit *SessionID* bezeichnet) und die Möglichkeit zur Sicherung sessionspezifischer Daten vorhanden sein muss. Zum Transfer muss die Session nicht komplett neu angelegt werden, sondern kann mit ihrer derzeitigen Datenstruktur an den Nachfolger übergeben werden. Aufgrunddessen ist die Session weiterhin unter der gleichen SessionID referenzierbar.
Im Gegensatz dazu ist Session Recreation mit einem Neuaufbau der Session beim Nachfolger verbunden. Die bestehende Session wird dabei nicht als Ganzes übertragen, sondern sukzessive neu generiert. Folglich muss zuallererst eine passende Datenstruktur angelegt werden, um daraufhin die sessionspezifischen Daten in diese Session zu importieren. Es handelt sich deshalb im Grunde genommen nur um eine Kopie der ursprünglichen Session. Aufgrund der Eineindeutigkeit der SessionID ist dies mit dem Erhalt einer neuen SessionID für die Kopie der Session verbunden, mit der Folge, dass zur Fortführung der Kommunikation die statische Entität die neuen SessionID zur Referenzierung der Session benutzen muss.

2.4.2 Handoverprotokoll

Initiierung

Anhand der in Kapitel 2.2 angesprochenen Alternativen in der Durchführung des Handovers gibt es eine Vielzahl verschiedener Kombinationsmöglichkeiten bei der Wahl des Handoverdesigns. Zulässige bzw. sinnvolle Kombinationen sowie daraus resultierende Protokolle sollen in diesem Kapitel genauer betrachtet werden.

Um die Entscheidung über einen Handover durchzuführen, muss die initiierende Entität Kontextinformationen sammeln und auswerten, wie es in Kapitel 2.3.1 beschrieben wurde. Durch die Handoverentscheidung sendet die initiierende Entität bei gegebenem Anlass eine Initiierungsnachricht an die kontrollierende Entität. Ausschlaggebend für die Anordnung der Initiierungsnachricht in das Handoverprotokoll ist die Position der initiierenden Entität. Man differenziert dabei static entity initiation, predecessor initiation, successor initiation und external initiation (vgl. Kap. 2.2.4). In den meisten Fällen erfolgt eine Entscheidung bei der statischen Entität oder beim derzeitigen dynamischen Partner, da diese beiden Entitäten direkt miteinander kommunizieren und deshalb am besten abwägen können, ob der dynamische Partner der optimale ist, oder doch eine Kommunikation zu einem anderen Partner „besser" wäre. Es ist aber auch denkbar, dass eine Initiierung beim zukünftigen Partner erfolgt, wenn dieser erkennt, dass er ein „besserer" Partner wäre als der derzeitige. Auch eine Initiierung durch eine externe Entität kann in manchen Fällen eingesetzt werden, wenn diese genug Kenntnis über die bestehende Kommunikation zwischen der dynamischen und der statischen Entität besitzt, um eine Handoverentscheidung herbeizuführen.

Proaktivität/Reaktivität

Ausschlaggebend für die Art der Intiierungsnachricht ist der Zeitpunkt der Initiierung (vgl. Kap. 2.2.5). Handelt es sich um ein proaktives Szenario, so entsteht die Handoverinitiierung, wenn ein „besserer" Partner gefunden wurde. Die Handoverinitiierung erfolgt deshalb zu einem festen Partner. Bei Reaktivität ist dies nicht der Fall, da auf das Eintreten des Problems reagiert werden muss, ohne dass ein Nachfolger für den derzeitigen dynamischen Partner feststeht. Eine Initiierungsnachricht wird deshalb, ohne einen festen Nachfolger zu spezifizieren, an die kontrollierende Entität gesendet. Oft muss in diesem Fall dann ein vertikaler Handover eingesetzt werden. Beispiel wäre der in Kapitel 2.2.1 beschriebene Handover in den Parked State. Durch das eingetretene Problem ist hier nur ein vorübergehendes Sichern der Session zum Ziel gesetzt. Die Entscheidung über den Nachfolger bei Reaktivität wird bei der kontrollierenden Entität getroffen.

Kontrolle

Ist in der Initiierungsnachricht kein fester Nachfolger vorhanden, ist es die Aufgabe der kontrollierenden Entität diesen auszuwählen. Notwendig dazu ist eine Abwägung aller möglichen Nachfolger mit einer Kosten-/Leistungsanalyse (vgl. 2.3.2). Nachdem eine weitere Aufgabe der kontrollierenden Entität der Transfer der Session vom Vorgänger zum Nachfolger ist, muss es eine Kommunikationsmöglichkeit zwischen Vorgänger und Nachfolger geben. Wie in Kapitel 2.2.3 beschrieben unterscheidet man den Forward Backlink und den Backward Backlink, sowie No Backlink:

- Liegt ein Forward Backlink zwischen Vorgänger und Nachfolger vor, kann eine Kontrolle auf der Seite des Vorgängers eingesetzt werden. Der Nachfolger muss dementsprechend architekturell so ausgestattet sein, dass er Sessions von anderen Entitäten annehmen, und diese weiter verarbeiten kann. Dies folgt aus einer Kompatibilität zum Informationsaustausch, wie sie in Kapitel 1 angesprochen wurde.
- Ist demgegenüber ein Backward Backlink zwischen Vorgänger und Nachfolger vorhanden, so kann der Nachfolger als kontrollierende Entität eingesetzt werden. Dieser kann die Session nach den Prinzipien Session Forwarding oder Session Recreation von seinem Vorgänger anfordern

und danach bei sich weiter verarbeiten. Dabei ist wieder eine entsprechende Kompatibilität zwischen Vorgänger und Nachfolger nötig. Nicht möglich ist diese Art der Kontrolle bei reaktiven Ansätzen, da dort der kontrollierenden Entität die Wahl des Nachfolgers auferlegt ist, der Nachfolger aber die kontrollierende Entität ist. D.h. eine Kontrolle durch die nachfolgende dynamische Entität ist nur bei proaktiver Initiierung möglich.

- Existiert zwischen Vorgänger und Nachfolger keine Kompatibilität bezüglich des Informationsaustausches, besteht keine Möglichkeit der direkten Kommunikation zwischen den dynamischen Entitäten. Als Ersatz wurde in Kapitel 2.2.3 eine Art Dolmetscherentität eingeführt. Bei dieser Entität kann es sich um die statische Entität oder eine externe Entität handeln. Diese können die komplette Session (beim Session Forwarding) oder die einzelnen Sessioninformationen (bei Session Recreation) vom Vorgänger anfordern und dann an den Nachfolger übergeben, der diese zur Weiterverarbeitung aufnimmt. Notwendig ist dafür einerseits eine Kompatibilität zwischen Vorgänger und kontrollierender Entität und andererseits zwischen kontrollierender Entität und dem Nachfolger.
  Sinnvoll sind externe oder statische kontrollierende Entitäten oft bei vertikalen Handovern, da hier möglicherweise eine direkte Kommunikation aufgrund der Heterogenität der Entitäten nicht anwendbar ist, eine vermittelnde Entität aber in der Lage ist zu beiden dynamischen Entitäten eine Kommunikation aufzubauen. Der entscheidende Einsatz einer vermittelnden Entität ergibt sich aus der Transformation der Session, um die bereits erzeugte Session beim Vorgänger in eine durch den Nachfolger verständliche Session zu überführen (vgl. Kap. 2.2.3).

Anzahl paralleler logischer Verbindungen

Während des Transfers der Session zwischen den dynamischen Entitäten, kann eine Kommunikation zwischen der statischen Entität und deren ursprünglichem Partner weiter bestehen. Zusätzlich bietet sich für die statische Entität die Möglichkeit, parallel zum Kommunikationskanal mit ihrem ursprünglichen Partner einen Kommunikationskanal mit dem zukünftigen Partner aufzubauen. Nach Kapitel 2.2.6 nennt man dies einen Soft Handover. In den meisten Fällen reicht jedoch der Einsatz des Hard Handovers aus. Durch die Durchführung eines Soft Handovers lässt sich im Allgemeinen die Verzögerungszeit reduzieren. Aber auch eine Verbesserung der QoS ist durch die Nutzung mehrerer paralleler Kommunikationskanäle möglich, soll aber an dieser Stelle nicht weiter verfolgt werden, da es nicht direkt den Handovervorgang beeinflusst. Bedingung für den Soft Handover, d.h. der parallelen Nutzung der Kommunikationskanäle zu beiden dynamischen Entitäten, ist die bereits erfolgte Wahl des zukünftigen Partners. Ist durch eine Reaktivität zum Zeitpunkt der Initiierung der Nachfolger noch nicht bekannt, kann auch keine Kommunikation mit ihm und der statischen Entität durchgeführt werden.

Zusammenfassung

Insgesamt sieht man nun, dass eine Vielzahl von Handovervarianten existieren (vgl. Abb. 2.9). Davon muss bei der Handoverkonzeptionierung eine bzw. mehrere Alternativen ausgewählt werden, die von der Architektur unterstützt werden sollen. Ein Beispiel wäre ein static entity initiated (proactive) - predecessor controlled - Hard - Handover, dessen Sequenzdiagramm in etwa folgendem Aufbau entspricht:

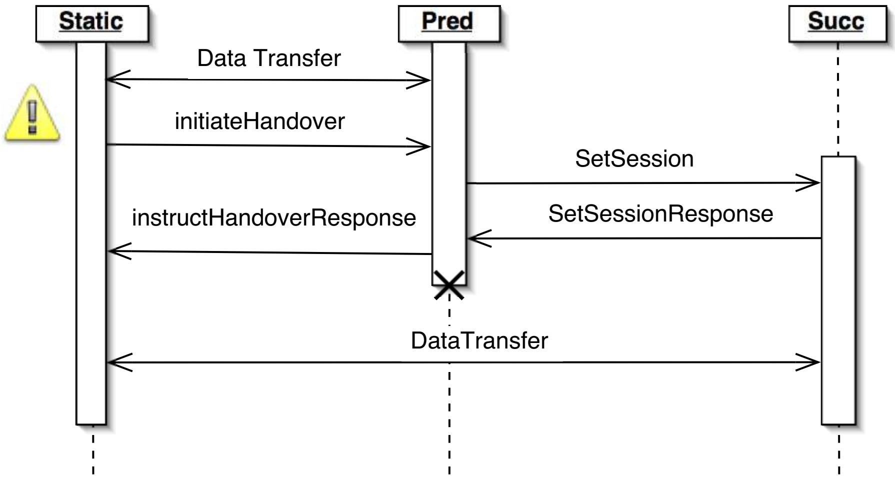

Abbildung 2.8: STATIC ENTITY INITIATED (PROACTIVE) - PREDECESSOR CONTROLLED - HARD - HANDOVER

Einige andere Beispiele finden sich auch in GSM, Mobile IP, usw. Im GSM-Standard wird ein Handover durch ständiges Messen der Empfangsstärke des Mobiltelefons zu den umliegenden Basisstationen von der derzeitigen Basisstation initiiert und dann durch diese ein harter Wechsel kontrolliert. Es handelt sich also um einen predecessor initiated (proactive) - predecessor controlled - Hard - Handover. Im Bereich Mobile IP wird einerseits ein proactive - static entity initiated - successor controlled - Hard - Handover [KAP+ 00] und andererseits ein proactive - static entity initiated - predecessor controlled - Hard - Handover [KAP+ 00] eingesetzt.

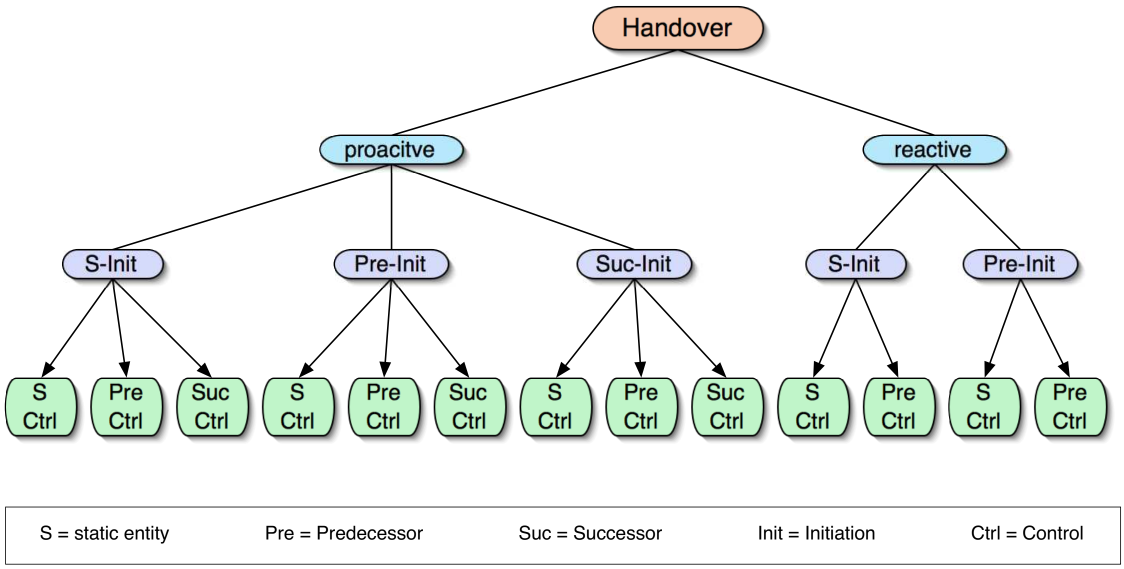

Abbildung 2.9: HANDOVEROPTIONTREE

Kapitel 3

Highlevel Service Handover

3.1 Highlevel Services

Die im vorigen Kapitel beschriebenen theoretischen Grundlagen des Handovers basieren auf den bereits erfolgten Studien über den Handover aus dem GSM-Standard und aus dem Bereich Mobile IP. Ziel dieser beiden Systeme ist dem Nutzer einen Dienst (Service) anzubieten. Im GSM-Standard ergibt sich der Dienst aus der Bereitstellung eines Gesprächs- oder Datenkanals zu einem oder mehreren anderen Nutzern. Im Bereich Mobile IP wird dem Mobile Agent die Möglichkeit geboten, einen Datenaustausch mit dem Correspondent Host über den Umweg des Home Agents durchzuführen. In beiden Systemen bezieht sich das Dienstangebot auf die Bereitstellung eines Kommunikationskanals. Man nennt diese Art von Diensten *Carrier Services*.
An dieser Stelle soll ein Dienst auf einer abstrakteren Ebene betrachtet werden. Im Vordergrund steht dabei nicht die Kommunikationsverbindung, sondern eine allgemeine Dienstleistung, die vom Anbieter bereitgestellt wird. Man nennt eine solche Art von Diensten *Non Carrier Services* oder *Highlevel Services*. Im folgenden wird unter einem Dienst immer ein Highlevel Service verstanden. Beispiele hierfür sind ein Routendienst, ein Wetterdienst oder ein Photoentwicklungsdienst, wie er in Kapitel 5 beschrieben wird. Highlevel Services erhalten in der heutigen Zeit immer größere Anerkennung, vor allem durch den ständigen Zuwachs von Diensten wie E-Commerce, E-Marketing, usw. im World Wide Web.
Um einen derartigen Highlevel Service bereitzustellen, muss mindestens ein Dienstanbieter existieren, der diese Leistung erbringen kann. Andererseits muss das Gegenstück zum Dienstanbieter, der Dienstnutzer, die Funktionalität besitzen, um einen optimalen Dienstanbieter zu finden und Nachrichten mit ihm austauschen zu können. Der Nachrichtenaustausch wird auch als Dienstnutzung bezeichnet. Bezüglich einer gewissen Problemstellung kann ein Dienstnutzer Anfragen an den Dienstanbieter stellen, der daraufhin die Anfrage entsprechend beantwortet. Diese Antworten können wie im Beispiel des Routendienstes eine Wegbeschreibung enthalten oder nach dem Beispiel des Photoentwicklungsdienstes aus einer Anzahl entwickelter Bilder bestehen.

3.2 Handover bei Highlevel Services

Bezieht man sich bei der Interaktion zweier Entitäten auf die eben beschriebene Dienstnutzung kann es wie bei Carrier Diensten zu einer Art Handover kommen. Dieser Handover wird als *Highlevel Service Handover* bezeichnet. In Analogie zur Definition des Carrier Service Handovers aus Kapitel 2.1 wird durch den Highlevel Service Handover eine Ersetzung eines Dienstanbieters während einer bestehenden Dienstnutzung beschrieben. Prinzipiell ist die im vorigen Kapitel entstandene Klassifizierung des Handovers, die auf Carrier Services basierte, auch bei Highlevel Services einsetzbar. Dementsprechend unterscheidet man Highlevel Service Handover nach den Funktionalitäten der dynamischen Entitäten, nach der Nutzertransparenz, nach der Position der kontrollierenden und der initiierenden Entität, nach dem Zeitpunkt der Initiierung und nach der Anzahl paralleler Kommunikationskanäle. Dabei sind die dynamischen Entitäten in diesem Fall durch verschiedene Dienstanbieter vertreten, und die statische Entität entspricht dem Dienstnutzer.
Ein Unterschied zwischen dem Carrier Service Handover und dem Highlevel Service Handover ist bei den Gründen zur Initiierung eines Handovers (vgl. Kap. 2.3.1) zu finden. Während bei Carrier Services der (erwartete) Abbruch des Kommunikationskanals für den Handover verantwortlich ist, erfolgt bei Highlevel Services der Handover allgemein aufgrund einer Veränderung des Kontextes. Zur Analyse des Kontextes durch die in Kapitel 2.3.2 vorgestellte Kosten-/Leistungsanalyse gehören alle für die Dienstnutzung notwendigen Informationen, d.h. alle Kontextinformationen der relevanten Aspekte. Das bedeutet auch, dass Informationen über die Qualität des Kommunikationskanals mit in die Berechnung einfließen müssen, wenn diese für die Dienstnutzung als relevant spezifiziert wurden. Daraus folgt, dass auch die Kosten-/Leistungsanalyse aus Kapitel 2.3.2 nicht auf Carrier Services oder Highlevel Services bezogen ist, sondern nur auf die Art des Dienstes, d.h. die für diesen Dienst als relevant geltenden Aspekte.

Kapitel 4

Ein kontextuelles Framework für Highlevel Service Handover

Um einen Highlevel Service zu nutzen, ist es im Allgemeinen nötig im Vorhinein einen passenden Dienstanbieter zu finden, der eine passende Funktionalität entsprechend der zugrundeliegenden Aufgabenstellung bereitstellt. Wie bereits in Kapitel 1 beschrieben, kann ein Dienstnutzer mit einem Dienstanbieter nur kommunizieren, wenn dieser eine passende Schnittstelle bereitstellt, d.h. kompatibel bezüglich der Kommunikation zur Lösung der Aufgabe ist. Dazu muss eine Beschreibung dieser Schnittstelle existieren, die alle für die Dienstnutzung notwendigen Informationen dem Dienstnutzer zugänglich macht. Die Informationen betreffen die in Kapitel 1 bereits erwähnte Ebene der Signatur, der Semantik, des Protokolls und des Kontextes.
Da in einem kontextuellen Framework besonderen Wert auf den Kontext als vierte Ebene der Interoperabilität gelegt wird, soll an dieser Stelle eine detaillierte Betrachtung des Einwirkens des Kontextes auf die Suche eines Dienstanbieters und vor allem auf die Dienstnutzung erfolgen. Die wesentlichen Einsatzbereiche des Kontextes sind *Context Obligations*, *Contextual Service Discovery* und *Context Binding* (vgl. [SLPF 03a]).
Desweiteren wird im darauffolgenden Kapitel eine Betrachtung der für den Handover notwendigen Eigenheiten dieses Frameworks durchgeführt. Dafür steht das Paradigma der Highlevel Services auch für den Handover im Vordergrund.

4.1 Kontextuelle Besonderheiten des Frameworks

4.1.1 Context Obligation

Context Obligations finden ihre Anwendung in der kontextuellen Beschreibung eines Dienstanbieters. Ziel ist eine Beschreibung in den für die Nutzung dieses Dienstanbieters relevanten Aspekten. Dies entspricht der Darstellung des in Kapitel 2.3.2 definierten Verfügungsbereichs (Scope). Context Obligations spiegeln den Verfügungsbereich eines Dienstanbieters in all seinen Dimensionen wieder. Ein Dienst könnte zum Beispiel die relevanten Aspekte Ort und Zeit, die seine örtliche sowie zeitliche Verfügbarkeit spezifizieren, besitzen. Zur kontextuellen Beschreibung dieses Dienstes sind dann Kontextinformationen bezüglich des Aspektes Ort und Zeit in entsprechenden Skalen anzugeben. Neben der direkten Angabe von Kontextinformationen ist nach [SLPF 03a] in den Context Obligations auch eine Bereichsangabe von Kontext Informationen sinnvoll. Prädikate wie LessOrEqual sind dabei für Skalen, die eine relative Ordnung besitzen, einsetzbar. Besondere Beachtung benötigen auch die in Kontextinformationen enthaltenen Metadaten, die Angaben zur Qualität des entsprechenden Contents preisgeben. Eine Anwendung hierfür wäre ein Routendienst einer Stadt, dessen Geltungsbereich mittels örtlicher Kontextinformationen angegeben wurde, diese Kontextinformationen aber nur eine Genauigkeit bis zu 100 Metern besitzen.

4.1.2 Contextual Service Discovery

Aus der Menge aller möglichen Dienstanbieter, d.h. der Dienstanbieter, die eine entsprechende Funktionalität bereitstellen und die zum Dienstnutzer kompatibel sind, muss daraufhin der optimale Dienstanbieter ausgewählt werden. Die Suche nach einem entsprechenden Dienstanbieter nennt man auch *Service Discovery*. Im Gegensatz zum herkömmlichen Service Discovery, welches zur Auswahl des optimalen Dienstanbieters keinerlei Informationen über den Kontext in die Entscheidung aufnimmt, zieht das sogenannte *Contextual Service Discovery* auch Kontextinformationen zur Entscheidung heran. Die Entscheidung kann zum Beispiel anhand der in Kapitel 2.3.2 eingeführten Kosten-/Leistungsanalyse erfolgen. Eine detaillierte Betrachtung der Kosten und Leistungen ist meist nur möglich, wenn aktuelle Kontextinformationen zur Verfügung stehen. Beispielsweise kann ein Dienstanbieter für einen Routendienst nur gültige Ergebnisse für Dienstnutzer, die sich im Raum München aufhalten, liefern. Zur Auswahl dieses Dienstanbieters müssen Informationen über die aktuelle Position des Dienstnutzers bereitstehen. Dadurch kann bei der Kosten-/Leistungsanalyse eine Entscheidung für oder gegen diesen Dienstanbieter anhand des von ihm spezifizierten Verfügungsbereichs und der aktuellen Position des Dienstnutzers als örtliche Kontextinformation geschehen.
Zur Auswahl eines Dienstanbieters ist aber nicht nur eine Betrachtung der relevanten Aspekte des Dienstanbieters sinnvoll, sondern auch relevante Aspekte anderer Entitäten wie zum Beispiel des Dienstnutzers müssen in die Entscheidung miteinfließen. Beispiel hierfür wäre ein Dienstnutzer der den Aspekt Wetter als relevant spezifiziert. Zur Kosten-/Leistungsanalyse muss in diesem Fall das aktuelle Wetter und die vom Dienstnutzer zu definierenden Präferenzen bezüglich des Wetters hinzugezogen werden.

4.1.3 Context Binding

Wurde ein passender Dienstanbieter ausgewählt, kann ein Dienstnutzer Anfragen bezüglich der zugrundeliegenden Aufgabenstellung an den Dienstanbieter senden. Ein Beispiel hierfür ist die Nutzung eines Routendienstes, der den Weg von der aktuellen Position des Dienstnutzers zu einer Zielposition bestimmt. Dabei ist bei jeder Anfrage des Dienstnutzers an den Dienstanbieter neben der Zielposition die aktuelle Position des Dienstnutzers anzugeben. Ist der Dienstnutzer nicht oder nur bedingt fähig derartige Informationen, wie die aktuelle Position als örtliche Kontextinformation des Dienstnutzers, zu bestimmen, so kann im Rahmen des sogenannten *Context Bindings* dieser Wert in die Anfrage eingesetzt bzw. unvollständige Angaben ersetzt werden. Meist sind dazu Informationen über die Identität des Dienstnutzers nötig, um Werte wie die aktuelle Position bestimmen zu können. Diese Identitätsinformationen müssen dann in der Anfrage enthalten sein.

4.1.4 Context Provider

Bei der Suche eines Dienstanbieters und bei der Dienstnutzung sind ständig Kontextinformationen für Contextual Service Discovery und Context Binding nötig. Im Allgemeinen besitzt der Dienstnutzer und der Dienstanbieter jedoch nicht die Fähigkeit all diese Kontextinformationen immer aktuell bereitzustellen. Deshalb muss eine Entität, ein sogenannter *Context Provider*, eingeführt werden, der diese Aufgabe übernimmt.
Der Context Provider liefert einerseits auf Anfrage entsprechende Kontextinformationen oder erlaubt andererseits eine Registrierung (Subscribing) bezüglich eines Aspektes, wodurch eine Benachrichtigung bei einer Änderung von Kontextinformationen dieses Aspektes erfolgt. Derartige Dienste werden auch Push-Dienste genannt (vgl. [LP 02]), da eine Interaktion durch das Eintreten eines Ereignisses vom Dienstanbieter, d.h. in diesem Fall vom Context Provider angestoßen wird. Pull-Dienste hingegen werden vom Dienstnutzer angefordert und sind dementsprechend nicht abhängig von der Veränderung des Kontextes.
Die Menge aller bei der Bereitstellung des Kontextes beteiligten Entitäten wird in der *Context Provider Domain* zusammengefasst. Dazu gehört neben Sensoren zur Erfassung der Informationen eine Schnittstelle, die dem Dienstanbieter, dem Dienstnutzer, sowie dritten Entitäten bereitgestellt wird. Diese wird als *Context Management Access Point (CMAP)* [SLPF 03b] bezeichnet und dient den Entitäten als Zugangspunkt zur Spezifikation von Relevanzbedingungen, relevanten Aspekten zur Dienstnutzung und Einschränkungen bezüglich der Güte der Kontextinformationen (vgl. [SLPF 03b]).

4.2 Besonderheiten durch die Handoverfunktionalität

4.2.1 Dienst zur Handoverdurchführung

Anhand des Contextual Service Discovery kann zu Beginn der Interaktion die optimale Wahl des Dienstanbieters erfolgen. Aber auch während der Dienstnutzung muss überprüft werden, ob der aktuelle Dienstanbieter noch optimal ist. Andernfalls kommt es durch die initiierende Entität zur Einleitung eines Handovers. Durch den Handover erfolgt eine Ersetzung des derzeitigen Dienstanbieters durch einen im Sinne der Kosten-/Leistungsanalyse „besseren" Dienstanbieter. Aus der Sicht der initiierenden Entität kann die Handoverdurchführung deshalb ebenfalls als eine Art Highlevel Service angesehen werden. Der Dienstnutzer ist in diesem Fall die initiierende Entität des Handovers (vgl. Kap. 2.2.4). Der Dienstanbieter entspricht der kontrollierenden Entität des Handovers (vgl. Kap. 2.2.3). Dabei ist es möglich, dass die Handoverinitiierung von verschiedenen Entitäten ausgeführt werden kann. Auch eine mehrfache Bereitstellung der kontrollierenden Entität kann in manchen Fällen sinnvoll sein. Ein Beispiel dafür wäre, unterschiedliche kontrollierende Entitäten einzusetzen, je nach ihrer Fähigkeit proaktive oder reaktive Handover durchführen zu können. Desweiteren sind vor allem bei vertikalen Handovern meist kontrollierende Entitäten speziell für die Funktionalitäten der dynamischen Entitäten zugeschnitten. Für den Handover in bzw. aus dem Parked State kann beispielsweise eine kontrollierende Entität zuständig für den Handover zwischen Dienstanbietern zur Photoentwicklung und dem Parked State sein, und eine weitere kontrollierende Entität für das Wiederherstellen der Session aus dem Parked State für die Photoentwicklung eingesetzt werden.
Ein Framework, welches den Highlevel Service Handover unterstützt, muss dementsprechend mindestens zwei Dienste bereitstellen. Der erste Dienst ist zuständig für die eigentliche Aufgabenstellung und zweiterer zur Durchführung des Handovers.

4.2.2 Dienst zum Sessiontransfer

Muss zur Durchführung des Handovers eine Session vom Vorgänger zum Nachfolger transferiert werden, müssen zusätzliche Dienste eingesetzt werden. Der Vorgänger muss einen Dienst bereitstellen, um die bereits erzeugte Session zu exportieren, und der Nachfolger muss über einen Dienst zur Importierung von Sessions verfügen. Der Dienstnutzer ist in diesem Fall die kontrollierende Entität, die eine Session anfordert und diese dann an den Nachfolger weiterleitet bzw. bei diesem neu aufbaut. Nach Kapitel 2.4.1 unterscheidet man an dieser Stelle Session Forwarding und Session Recreation je nachdem, ob die Session als Ganzes übergeben wird oder mittels ihrer einzelnen Bestandteile beim Nachfolger neu aufgebaut wird. Gegebenenfalls muss dazu die Session durch die kontrollierende Entität in eine vom Nachfolger verständliche Form transformiert werden. Nach der Übertragung der Session kann mit der Dienstnutzung zur eigentlichen Aufgabenstellung fortgefahren werden. Der Zustand beim Nachfolger enthält nun alle Informationen, die den Zustand des Vorgängers vor dem Abbruch der Dienstnutzung definierten. Gegebenenfalls tritt dabei jedoch eine Veränderung von dienstanbieterspezifischen Daten auf. Beispielsweise kann dem Dienstnutzer eine neue SessionID durch den Aufbau einer neuen Session bei Session Recreation zugewiesen werden, da diese ein eineindeutiger Identifikator für die Session ist, und ein Neuaufbau deshalb einen neuen Identifikator mitsichbringt.

4.3 Framework zur direkten Dienstnutzung

Wie bereits angesprochen benötigt man für die Nutzung eines Highlevel Services einerseits Dienstanbieter, d.h. Entitäten, die einen derartigen Dienst zur Verfügung stellen und andererseits ein entsprechendes Gegenstück auf der Seite der Dienstnutzer, die an einer Nutzung des Dienstes interessiert sind. Die auf der Seite eines Dienstnutzers beteiligten Entitäten werden in der *Customer Domain* zusammengefasst. Auf der Seite der Dienstanbieter stehen die *Service Provider Domains* der Customer Domain gegenüber, die durch einzelne Dienstanbieter aufgespannt werden. Eine Menge aller zur Verfügung stehenden Dienstanbieter wird von einem sogenannten *Service Directory* verwaltet, bei dem sich Dienstanbieter mit ihren Dienstbeschreibungen registrieren (Announcement), und Dienstnutzer dadurch entsprechende Dienstanbieter aufsuchen können (Contextual Service Discovery). Zusätzlich ist eine Context Provider Domain (vgl. Kap. 4.1.4) nötig, durch die die Kontextinformationen für Contextual Service Discovery und Context Binding zur Verfügung gestellt werden.
Ein Zusammenwirken dieser Bereiche ist in Abbildung 4.1 dargestellt. Zur Dienstnutzung kann eine Entität aus der Customer Domain durch das Contextual Service Discovery den für den zu erfüllenden Zweck optimalen Dienstanbieter finden. Dazu bekommt sie Zugriff auf alle zur Verfügung stehenden Dienstanbieter durch das Service Directory und die zur Entscheidung notwendigen Kontextinformationen durch den Context Provider. Bei erfolgreicher Suche kann der Dienstnutzer dann den Nachrichtenaustausch mit dem Dienstanbieter anstoßen (Invocation). Bei Bedarf werden unvollständige Parameter einer Nachricht dabei durch das Context Binding vervollständigt. Auch während der Dienstnutzung führt der Dienstnutzer periodisch, oder nachdem er vom Context Provider auf eine Veränderung des Kontextes aufmerksam gemacht wurde, das Contextual Service Discovery durch. Gleichzeitig können auch Dienstanbieter oder dritte Entitäten das Contextual Service Discovery praktizieren. Entschließt sich eine Entität einen Handover zu einem anderen Dienstanbieter zu initiieren (Handover Initiation), da der aktuelle Dienstanbieter nicht mehr als optimal eingestuft wurde, so sendet diese eine Initiierungsnachricht an die kontrollierende Entität (Handover Controller). Diese führt daraufhin den Sessiontransfer vom Vorgänger zum Nachfolger aus. Schließlich kann der Dienstnutzer

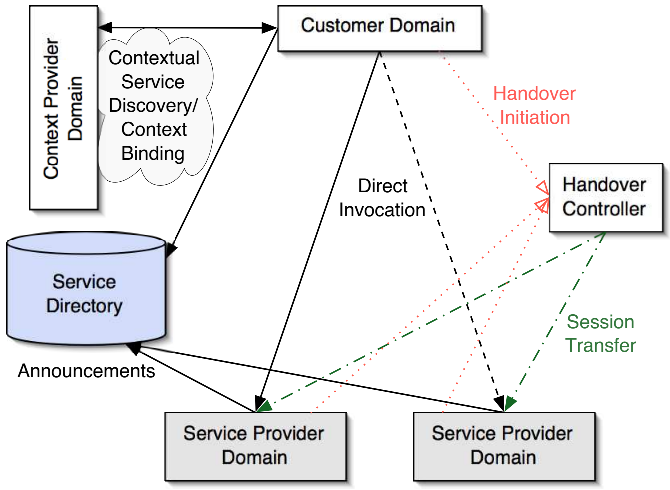

Abbildung 4.1: MODELL ZUR DIREKTEN DIENSTNUTZUNG

den Nachrichtenaustausch mit dem neuen Dienstanbieter fortsetzen. Gegebenenfalls muss jedoch die Dienstnutzung zum neuen Dienstanbieter gegenüber der alten modifiziert werden, da sich durch den Handover inhaltliche Änderungen dienstanbieterspezifischer Daten (z.B. SessionID) oder grundlegende Änderungen der Dienstnutzungsschnittstelle ergeben können.

4.4 Framework zur delegierten Dienstnutzung durch einen Intermediate

Eine Variation des oben angeführten Frameworks ergibt sich aus dem Einsatz einer Middleware zwischen Dienstnutzer und Dienstanbieter, dem sogenannten *Intermediate*. Der Intermediate fungiert als Proxy in diesem Framework. Er empfängt Nachrichten des Dienstnutzers und sendet diese weiter an einen von ihm gewählten Dienstanbieter. In entgegengesetzter Richtung empfängt er Nachrichten von einem Dienstanbieter und leitet diese weiter zum entsprechenden Dienstnutzer. Vorteil beim Einsatz eines Intermediates ist, dass durch ihn eine vollständige Transparenz für den Dienstnutzer entsteht. D.h. der Dienstnutzer hat keine Kenntnis darüber, mit welchem Dienstanbieter er eine Dienstnutzung eingeht. Seine Anfragen richten sich ausschließlich an den Intermediate, der diese Anfragen erhält und entsprechende Antworten darauf an den Dienstnutzer zurücksendet. Ein durch den Handover wechselnder Dienstanbieter, betrifft deshalb in keinster Weise die Interaktion zwischen Dienstnutzer und Intermediate. Im Gegensatz zur direkten Dienstnutzung, bei der der Dienstnutzer sich an die Situation, die durch den Handover entstanden ist, anpassen muss, ergibt sich für den Dienstnutzer bei der delegierten Dienstnutzung durch den Intermediate keine Veränderung. Da der Intermediate die vollständige Kontrolle über die Suche nach einem passenden Dienstanbieter und die anschließende Dienstnutzung übernimmt, braucht der Dienstnutzer keine Funktionalität für das Contextual Service Discovery und Context Binding besitzen. Diese Funktionen fallen in den Aufgabenbereich des Intermediates.
Desweiteren bietet sich an, die Handoverinitiierung beim Intermediate zu platzieren, da er bereits

über die Möglichkeiten zur Handoverentscheidung durch das Contextual Service Discovery verfügt. Da der Dienstnutzer über den Handover uninformiert bleibt erfolgt auch keine Initiierung mehr von dieser Entität (vgl. Abb. 4.2).

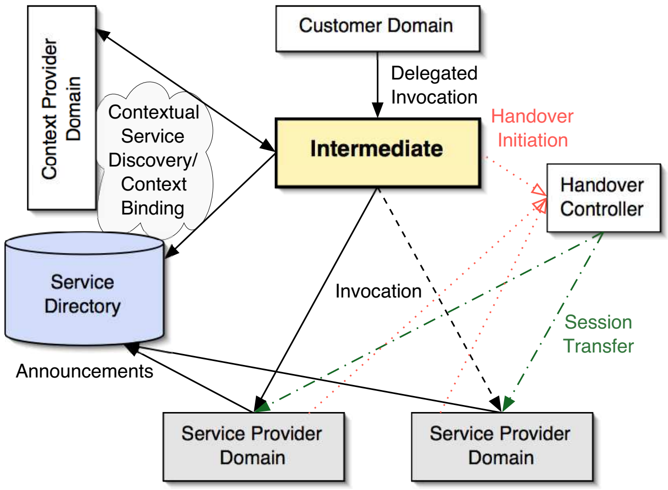

Abbildung 4.2: MODELL ZUR DELEGIERTEN DIENSTNUTZUNG

Kapitel 5

Ein Beispielszenario

5.1 Grundlagen des Photoentwicklungsdienstes

Für den konkreten Einsatz dieses kontextuellen Frameworks mit Handoverfunktionalität soll im Folgenden ein Photoentwicklungsdienst als Grundlage dienen, an den Photos in digitaler Form übermittelt werden können, um diese in Papierformat zu erhalten. Denkbar ist eine Dienstnutzung durch ein mobiles Endgerät (z.B. ein Mobiltelefon mit integrierter Kamera). Bei Bedarf können die erzeugten Photos an einen geeigneten Photoentwicklungsdienst weitergeleitet werden. Dazu muss man sich zuerst mit seinem Namen (und ggf. Adresse, Kreditkartennummer,...) beim Dienstanbieter registrieren. Jeder Nutzer erhält daraufhin eine eineindeutige Nummer und kann ab diesem Zeitpunkt Bilder an den Dienstanbieter senden. Eingeleitet wird der Datentransfer, bei dem jedes Bild einzeln übertragen wird, mit einer Anmeldung, bei der eine eineindeutige Nummer für diesen Auftrag generiert wird. Sie entspricht der in Kapitel 2.4.1 definierten SessionID. Sind alle Photos erfolgreich übermittelt, kann der Dienstnutzer den Auftrag und damit die Session beenden. Zu einem erfolgreichen Abschluss der Session gehört allerdings noch die Wahl des Abholortes. Während einer Session kann dieser, sowie der Preis für die Entwicklung und die Dauer der Entwicklung jederzeit abgefragt werden. Der Abschluss der Session erfolgt dann durch eine Auftragsbestätigung.
Einzelne Operationen wie Methoden zum Setzen und Abfragen des Abholortes oder der Phototransfer werden als atomare Transaktionen angesehen. Zwischen je zwei Transaktionen liegen Checkpunkte, zu denen jederzeit zurückgekehrt werden kann (Rollback). Folglich kann bei Unterbrechungen während Transaktionen immer an einem konsistentem Zustand fortgefahren werden.
Verschiedene Dienstanbieter unterscheiden sich primär durch die Funktionalität, die sie dem Dienstnutzer bereitstellen und sekundär durch die Eigenschaften ihres Dienstangebotes. Letzteres ist beispielsweise die Dauer der Entwicklungszeit, die Qualität der Photos oder die monetären Kosten für die Entwicklung. Desweiteren unterscheidet man Dienstanbieter, bei denen man die fertigen Photos nach der Dauer der Entwicklungszeit selbst abholen muss, von denen, die gegen Aufpreis die fertigen Photos nach Hause liefern. Im Vordergrund soll hier jedoch die Selbstabholung stehen. Dazu kann durch das Context Binding der Parameter für den optimalen Abholort automatisch an einen Wert gebunden werden. Ausschlaggebend für die Wahl des optimalen Abholortes ist dabei die Position des Dienstnutzers in Verbindung mit allen möglichen Abholorten. Die Dienstnutzung ist also kontextabhängig. Dabei handelt es sich in diesem Fall um den örtlichen Kontext des Dienstnutzers. Man nennt einen derartigen Dienst auch einen Location Based Service. Bei Location Based Services

spielt der örtliche Kontext mobiler Endgeräte die entscheidende Rolle, da sie meist einer ständigen Positionsänderung unterliegen. Aber auch andere Aspekte des Kontextes können in einem derartigen Szenario als relevant gelten. Beispielsweise wird bei schlechtem Wetter ein entfernter Abholort sehr viel schlechter eingestuft als ein Abholort in der Nähe.
Kontextabhängigkeit besteht jedoch nicht nur in der Dienstnutzung, sondern auch in der Dienstauswahl. Hier muss ein Contextual Service Discovery, wie es in Kapitel 4.1.2 eingeführt wurde, für die Wahl des optimalen Photoentwicklungsdienstanbieters sorgen. Eingesetzt wird das Contextual Service Discovery zu Beginn einer Dienstnutzung, sowie während der Dienstnutzung bei Veränderung des Kontextes, um gegebenenfalls einen Handover einzuleiten.

5.2 Nachrichtenaustausch zur Dienstnutzung

Die für die Dienstnutzung grundlegende Funktion ist die Möglichkeit des Nachrichtenaustausches zwischen den beteiligten Entitäten. Meist wird dieser Nachrichtenaustausch zwischen entfernten Entitäten eingesetzt, die durch eine Netzinfrastruktur miteinander verbunden sind. Eine mögliche Art des Nachrichtenaustausches, passend zu diesem Szenario, ist der entfernte Funktionsaufruf. Dieser wird auch als *Remote Procedure Call (RPC)* bezeichnet. Es handelt sich dabei um einen lokalen Aufruf einer Funktion, die bei einer entfernten Entität bearbeitet wird, und das Ergebnis dann als Rückgabewert des lokalen Funktionsaufrufs erscheint. Das hier recht einfach beschriebene Verfahren basiert auf der Technik des Serialisierens von Funktionsaufrufen, dem sogenannten Marshalling, und der entsprechenden Deserialisierung, dem Unmarshalling. Durch das Marshalling wird der Funktionsaufruf in eine Form gebracht, in der er durch die Netzinfrastruktur versendet werden kann. Auf der korrespondierenden Seite bewirkt das Unmarshalling den Wiederaufbau des Funktionsaufrufs aus der serialisierten Form. Dazu muss bei Plattformunterschieden auf Differenzen im Aufbau von Datenstrukturen geachtet werden. Eine schematische Darstellung des entfernten Funktionsaufrufs entspricht der Abbildung 5.1.

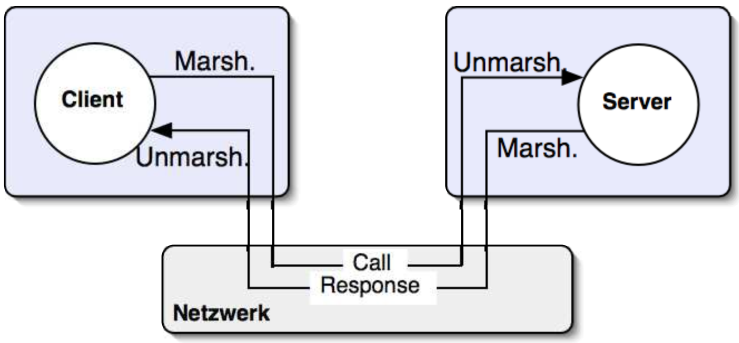

Abbildung 5.1: REMOTE PROCEDURE CALL

Derzeit existieren einige Implementierungen dieses Verfahrens zum Beispiel in SUNs Java 1.4.1 durch Remote Method Invocation (RMI) [jav a] oder der ONC RPC von SUN [onc]. Beide Implementierungen nutzen eine eigens entwickelte serialisierte Form des Funktionsaufrufs. Ein entfernter Funktionsaufruf zwischen diesen Implementierungen ist deshalb nicht direkt möglich. Abhilfe schafft das durch das World Wide Web Consortium (W3C) [w3c] entwickelte Simple Object Access Protocol (SOAP) [BEK^{+} 00], welches ein einheitliches Protokoll zum Nachrichtenaustausch spezifiziert. Es stellt eine Syntax zur Verfügung, um entfernte Funktionsaufrufe mit ihren Parametern in eine XML-basierte (vgl. [w3c 03]) Form zu überführen. Ein SOAP-RPC ist in einen SOAP-Envelope eingebettet (vgl. [BEK+ 00]). Als direkte Kindelemente befinden sich im SOAP-Envelope der optionale SOAP-Header und der zwingend vorgeschriebene SOAP-Body. Der SOAP-Body enthält den eigentlichen Funktionsaufruf mit seinen Parametern in serialisierter Form. Ein SOAP-RPC für den Photoentwicklungsdienst könnte folgendermaßen aussehen:

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
     xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <SOAP-ENV:Body>
  <ns1:beginUpload xmlns:ns1="http://demo.heywow.com/wsdl/PhotoShop.wsdl">
   <id xsi:type="xsd:int"
       SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
     410
   </id>
  </ns1:beginUpload>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

In diesem Beispiel handelt es sich um einen Funktion namens beginUpload, auf deren Signatur durch ihren Namespace ns1 verwiesen wird, und einem Parameter namens id vom Typ xsd:int mit dem Wert 410. Eine entsprechende Antwort auf diese Anfrage könnte folgende Gestalt annehmen:

```
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope
     xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
     xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <SOAP-ENV:Body>
  <ns1:beginUploadResponse xmlns:ns1="http://demo.heywow.com/wsdl/PhotoShop.wsdl">
   <Return xsi:type="xsd:int"
       SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
     0
   </Return>
  </ns1:beginUploadResponse>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

5.3 SOAP-basierende Architektur

Eine Implementierung, entsprechend der W3C SOAP Spezifikation, in der Programmiersprache Java (vgl. [jav b]), namens Apache SOAP (vgl. [BCC+ 02]), wurde im Jahre 2000 von der Apache Foundation Group eingeführt. Sie besitzt ein Application Programming Interface (API) für den gewöhnlichen Nachrichtenaustausch, sowie für den speziellen Nachrichtenaustausch in Form des entfernten Funktionsaufrufs. Dieser wird durch die Klassen Call für den Aufruf und Response für die entsprechende Antwort repräsentiert. Neben diesen steht eine Vielzahl an Klassen für die Bearbeitung, das Marshalling, das Unmarshalling, usw. bereit.

Apache SOAP stellt jedoch nicht nur Klassen für den Nachrichtenaustausch zur Verfügung, sondern kann auch als *Web Application Archive (WAR)* für den von der Apache Foundation Group stammenden Webserver Apache Tomcat (vgl. [tom]) eingesetzt werden, um auf der Seite des Dienstanbieters die Verwaltung verschiedener Dienste, die Wahl des entsprechenden Dienstes, Marshalling und Unmarshalling, usw. durchzuführen.
Durch die Nutzung des Apache Tomcats kann mit einfachen Mitteln ein *Webservice* im World Wide Web angeboten werden, der eine Vielzahl an Möglichkeit bereitstellt. [web] beschreibt Webservices wie folgt: „The use of Web services on the World Wide Web is expanding rapidly as the need for application-to-application communication and interoperability grows. These services provide a standard means of communication among different software applications involved in presenting dynamic context-driven information to the user“.
Die Apache SOAP Implementierung kann auf das *Hyper Text Transfer Protocol (HTTP)* [FGM+ 99] oder das *Simple Mail Transfer Protocol (SMTP)* [smt] aufbauen. Diese können bei der Initialisierung des Verbindungsprotokolls ausgewählt werden. Da es sich bei diesem Dienst um einen Webservice handeln soll, bietet sich eine Übertragung per HTTP an, da für eine externe Dienstnutzung, basierend auf HTTP, eine entsprechende Infrastruktur, wie eine für den HTTP-Nachrichtenaustausch geöffnete Firewall, im Normalfall durch die Webservice-Architektur bereits existiert.
Der Apache Tomcat Webserver in Verbindung mit der entsprechenden SOAP-Implementierung ermöglicht die Nutzung von Funktionen auf Dientstanbieterseite durch den SOAP-RPC. Das auf Dienstanbieterseite vorhandene Funktionsangebot muss dafür keine spezielle Form oder Funktionalität zur Unterstützung des SOAP-RPCs besitzen. Prinzipiell ist es möglich jede Funktion unabhängig der zugrundeliegenden Programmiersprache durch den SOAP-RPC aufzurufen. Dafür muss die Klasse, die diese Funktion enthält, zuvor mittels eines Deployment Descriptors beim Webserver registriert werden. Der Deployment Descriptor enthält alle Informationen die für die Ausführung eines SOAP-RPCs notwendig sind. Ein für den Photoentwicklungsdienst entsprechender Deployment Descriptor enthält folgende Gestalt:

```
<?xml version="1.0" encoding="UTF-8"?>
<isd:service xmlns:isd="http://xml.apache.org/xml-soap/deployment"
             id="http://demo.heywow.com/wsdl/PhotoShop.wsdl">
  <isd:provider
    type="com.heywow.photoshop.PhotoShopProvider"
    scope="Application"
    methods="getTitle register beginUpload transfer order
             getPrice getDate getLocation setLocation">
    <isd:java class="com.heywow.photoshop.PhotoShop1" static="false" />
  </isd:provider>
</isd:service>
```

Insgesamt erkennt man, dass durch den Apache Tomcat in Verbindung mit der Apache SOAP Implementierung eine Architektur bereitgestellt wird, um recht komfortabel ein Dienstangebot durch einen Webservice zu erstellen, um dann eine entfernte Dienstnutzung mittels des SOAP-RPCs durchführen zu können.

5.4 Dienstbeschreibung

Wie bereits mehrfach angesprochen, muss für die Dienstsuche und Dienstnutzung eine Beschreibung des Dienstanbieters existieren. Diese Beschreibung betrifft die in Kapitel 1 eingeführte Ebene der

Signatur, der Semantik, des Protokolls und des Kontextes. Bezüglich der einzelnen Ebenen wird im Folgenden anhand von XML-basierten Sprachen dargestellt, welche Informationen zur Beschreibung von Webservices notwendig sind.

5.4.1 Dienstbeschreibung auf der Ebene der Signatur

Auf der Ebene der Signatur muss eine Beschreibung eines Webservices die Funktionsnamen der bereitgestellten Funktionen sowie den Typ und die Reihenfolge der Parameter einer Funktion enthalten. Zum Einsatz kommt an dieser Stelle zum Beispiel die *Web Service Description Language (WSDL)* [CCMW 01], eine XML-basierte Sprache zur Beschreibung der Signatur von Webservices und deren Zugriffspunkten. Der WSDL Wurzelknoten beinhaltet fünf Kindelemente. Diese sind types, message, portType, binding und service. Letzteres spezifiziert mögliche Zugriffspunkte von Dienstanbietern, die den Dienst mit dieser Signatur zur Verfügung stellen. Die restlichen vier Elemente geben Informationen zu Zusammengehörigkeit von Nachrichten, wie Request und Response, zu den Namen der Nachrichten, sowie deren Parameter und zu Typ und Reihenfolge der Parameter. Zusätzlich können abhängig von der Art des zugrundeliegenden Webservices (z.B. bei Nutzung mittels SOAP) spezifische Informationen hinzugefügt werden.
Durch die Erweiterbarkeit der Sprache ist es an dieser Stelle möglich, auch über die obigen Informationen hinausgehende Angaben zu einem Webservice preiszugeben. Sinnvoll erscheint hier Referenzen zu anderen Ebenen herstellen zu können. Beispielsweise könnte so ein Funktionsparameter einer bestimmten Skala eines Aspektes zugeordnet werden. Daraufhin ist es beim Context Binding möglich, zu überprüfen, ob für den Parameter bezüglich des Parametertyps ein korrekter Wert angegeben wurde, und gegebenenfalls diesen Wert anzupassen.

5.4.2 Dienstbeschreibung auf der Ebene der Semantik

Desweiteren wird eine semantische Dienstbeschreibung benötigt, in der Informationen über die Bedeutung der auf Signaturebene spezifizierten Ausdrücke vorhanden sind. In diesem Bereich ist bereits einige Arbeit im Rahmen des *Semantic Webs* ausgeübt worden. Daraus sind einige XML-basierte Sprachen wie DAML+OIL oder DAML-S (vgl. [dam]) entstanden, die eine deklarative Beschreibung der Semantik eines Webservices ermöglichen.

5.4.3 Dienstbeschreibung auf der Ebene des Protokolls

Die Ebene des Protokolls muss eine Beschreibung erhalten, die die relative Ordnung der Funktionsaufrufe angibt. An dieser Stelle sind auch Angaben über den Aufruf von Funktionen anderer Dienste, Blockierungsbedingungen und Regeln für die Interaktion mit anderen Diensten (vgl. [VHT]) festzulegen. Eine entsprechende Funktionalität wird durch das *Web Service Choreographie Interface (WSCI)* [wsc] bereitgestellt. Es basiert auf XML und beschreibt den Ablauf des Nachrichtenaustausches von Web Services, die eine choreographierte Interaktion mit anderen Diensten betreiben (vgl. [wsc]).

5.4.4 Dienstbeschreibung auf der Ebene des Kontextes

Zur Beschreibung eines Web Services auf der Ebene des Kontextes müssen alle für die Dienstnutzung relevanten Kontextinformationen spezifiziert werden. Darunter wird der Verfügungsbereich eines Webservices verstanden. Eine Sprache, die die nötige Ausdrucksstärke aufweist ist die *Context Ontology Language (CoOL)* [SLPF 03b]. Sie ermöglicht die Angabe von relevanten Kontextinformationen in einer Skala eines Aspektes. Derartige Beschreibungen entsprechen den in Kapitel 4.1.1 angesprochenen Context Obligations.

5.4.5 Dienstsuche

Nachdem nun Webservices anhand der bereits aufgeführten Sprachen auf allen vier Ebenen der Interoperabilität beschrieben werden können, muss ein Anbieter eines Webservices sein Dienstangebot mit den entsprechenden Beschreibungen für andere zugänglich machen. Dazu muss eine Registry existieren, welches alle Dienstanbieter mit ihren Zugangspunkten und (Verweise auf) Beschreibungen enthält.
Ein Beispiel für eine derartige Registry ist *Universal Description, Discovery and Integration (UDDI)* [udd 01]. UDDI ist eine Gruppe von web-basierten Registries, die Informationen über Unternehmen und andere Entitäten und ihrer technischen Schnittstelle (APIs) bereitstellen (vgl. [udd 01]). Auf diese Registries kann einerseits von den Dienstanbietern zugegriffen werden, um ihre Angebote zu veröffentlichen (Announcement), und andererseits von den Dienstnutzern eine Dienstsuche durchgeführt werden (Discovery). Ein Zugriff auf UDDI kann ebenfalls mittels SOAP geschehen.

5.5 Highlevel Service Handover

5.5.1 Handoverentscheidung

Die Handoverentscheidung beruht auf der bereits beschriebenen Dienstauswahl während der Dienstnutzung. Aus der Menge an Dienstanbietern, die von der UDDI Registry verwaltet wird, muss ein Dienst ausgewählt werden. Die Menge der Dienstanbieter kann sich hierbei über die Zeit verändern. Dazu muss stets erneut die UDDI Registry nach potenziellen Dienstanbietern befragt werden. Legt man bei der Kosten-/Leistungsanalyse das Minimalitätsprinzip zugrunde, muss das Minimum einer Kostenfunktion bei fester Leistung berechnet werden. Parameter der Kostenfunktion könnten die für die Entwicklung aufzuwendenden monetären Kosten sein. Für die Entscheidung ist also eine Angabe der Bearbeitungskosten jedes Dienstanbieters nötig. Desweiteren könnten Onlinekosten, die man an seinen Internetprovider zu zahlen hat, ein zusätzlicher Parameter der Kostenfunktion sein. Da diese nicht direkt für jeden Dienstanbieter angegeben werden können, muss hier von der entscheidenden Entität bei einer Abrechnung nach der Onlinezeit zuerst das Minimum der Übertragungsgeschwindigkeiten der beteiligten Entitäten ermittelt werden, und diese dann mit dem Datenvolumen sowie den dafür aufzuwendenden Kosten verrechnet werden. Vom Dienstanbieter muss dafür die angebotene Übertragungsgeschwindigkeit, die Bearbeitungskosten, etc. öffentlich zugänglich gemacht werden. All diese Informationen müssen in der kontextuellen Beschreibung des Dienstanbieters enthalten sein. Natürlich spielt an dieser Stelle auch der zeitliche Kontext eine wichtige Rolle. Die dafür berechneten Kosten könnten hierbei möglicherweise nur Werte zwischen 0 und ∞ annehmen, je nachdem, ob der

Dienstanbieter zur Zeit verfügbar ist oder nicht.
Desweiteren muss die Dienstauswahl anhand der Funktionalitäten zum Sessiontransfer zwischen den dynamischen Entitäten getroffen werden. Das heißt, dass ein in Frage kommender Nachfolger eine entsprechende Kompatibilität zum Vorgänger besitzen muss. Gegebenenfalls kann diese Kompatibilität auch durch die Transformation bei der kontrollierenden Entität erzielt werden. Im Beispiel des Photoentwicklungsdienstes wurde jeder Dienstanbieter mit der Fähigkeit zur Importierung und Exportierung von Sessions ausgestattet. Dabei handelt es sich bei den Sessions um eine einheitliche Datenstruktur, sodass eine Kompatibilität zum Sessiontransfer stets gegeben ist.

5.5.2 Handoverdurchführung

Hat nun eine Handoverentscheidung stattgefunden, wird durch die Initiierung zur Durchführung übergegangen. Dazu muss eine entsprechende Initiierungsmeldung an die kontrollierende Entität gesendet werden. In diesem Beispiel wurde dabei auf eine Kontrolle beim Vorgänger zurückgegriffen. Je nachdem, ob reaktiv oder proaktiv initiiert wurde, muss dieser zuerst noch einen passenden Nachfolger auswählen, bevor zum Sessiontransfer übergegangen wird. Die Übergabe der Session kann per Session Forwarding oder Session Recreation vollzogen werden. An dieser Stelle wurde auf das Session Forwarding, d.h. der Übergabe der Session als Ganzes zurückgegriffen. Alle bereits übertragenen Bilder sowie Informationen über Dienstnutzer, Abholort, etc. sind dafür in einem Objekt gespeichert, welches beim Marshalling in eine serialisierte XML-basierte Form transformiert werden kann. Zum Session Transfer sendet der Vorgänger die Session an den Nachfolger. Der Nachfolger nimmt die Session entgegen und fügt sie seiner aktuellen Umgebung hinzu. Bei Erfolg kann nun der Dienstnutzer weitere Photos zur Entwicklung transferieren. Es handelt sich dementsprechend bei der Anzahl der parallelen Kommunikationskanäle um einen Hard Handover.

5.6 Einsatz eines Intermediates

Wie in Kapitel 4.4 beschrieben, kann zur Erhöhung der Nutzertransparenz ein Intermediate zwischen Dienstanbieter und Dienstnutzer eingesetzt werden. Dieser hat dann die Aufgabe alle Anfragen, die der Dienstnutzer zuvor direkt an den Dienstanbieter gestellt hat, als eine Art Proxy anzunehmen und dann an einen von ihm gewählten Dienstanbieter weiterzuleiten. Der Dienstnutzer versieht alle Anfragen dafür mit einem für Proxies erstellten HTTP-Header und sendet alle Pakete an den Intermediate. Der Intermediate entfernt den an ihn gerichteten HTTP-Header und führt ein Unmarshalling des serialisierten SOAP-Envelopes aus, um die Art der Anfrage zu analysieren und das Context Binding durchführen zu können. Zusätzlich ersetzt er die SessionID für den Dienstnutzer durch eine vom Intermediate generierte Pseudo-SessionID. Diese wird zu Beginn einer Session vom Intermediate erzeugt und dem Dienstnutzer anstatt der echten - vom Dienstanbieter generierten - SessionID weitergeleitet. Die SessionID des Dienstanbieters speichert der Intermediate in den Sessioninformationen, die er für jeden Nutzer anlegt. Dies ist eine weitere Stufe um den Handover für den Client transparent zu halten. Bei einer Änderung der SessionID durch einen Handover mit Session Recreation ändert sich die SessionID nur in den Sessioninformationen des Intermediates, nicht aber für den Dienstnutzer.

Kapitel 6

Zusammenfassung und Ausblick

Anhand der Implementierung des Photoentwicklungsdienstes ergibt sich ein Modell, um den Highlevel Service Handover, wie er in Kapitel 3 eingeführt wurde, praktisch in einem Dienstnutzungsverhältnis zu testen. Dabei wurden die Einflüsse und Auswirkungen des Kontextes während der Dienstauswahl und der Dienstnutzung fokussiert. Für den Handover ist dabei im Speziellen die kontextuelle Wahl des optimalen Dienstanbieters von Interesse, da jede Veränderung des Kontextes zu einem Wechsel des optimalen Dienstanbieters und damit zu einem Handover führen kann. Zur Vereinfachung wurde im aufgeführten Beispielszenario der Highlevel Service Handover nur zwischen dynamischen Entitäten, die identische Funktionalitäten besitzen, durchgeführt. Zur Vollständigkeit ist auch der Highlevel Service Handover zwischen Entitäten mit unterschiedlichen Funktionalitäten in Betracht zu ziehen. Ein Beispiel ist der Handover in den Parked State, bei dem das Ziel nur ein vorübergehendes Sichern der Session ist. Um einen Dienstanbieter für die temporäre Sessionaufbewahrung zu finden, muss eine andere Art der Dienstregistrierung und der Dienstsuche, als sie im beschriebenen Beispielszenario verwendet wurde, herangezogen werden, da ein Auffinden von Diensten mit unterschiedlichen Funktionalitäten mit den verwendeten Komponenten nicht direkt möglich ist.
Desweiteren ist eine detailliertere Betrachtung der Handoverbedingungen zwischen Photoentwicklungsdiensten verschiedener Organisationen nötig. Vergleichbar sind diese Bedingungen mit den Roaming-Abkommen im Mobilfunk. Dabei müssen Regelungen zu Abrechnungsverfahren, Nutzerverwaltung, Sicherheitsmaßnahmen, etc. getroffen werden. Diese wurden vorerst außer Betracht gelassen, sind aber für ein ordnungsgemäßes Dienstnutzungsverhältnis unumgänglich.
Wie bereits erwähnt, stellt der Kontext als die Summe aller externen Einflüsse die entscheidende Position des Beispielsszenarios dar. Basis zur Darstellung des Kontextes ist dabei die Context Ontology Language, die einen wesentlichen Fortschritt in der syntaktischen Beschreibung des Kontextes darstellt. Kontext, der in den Context Obligations, dem Context Binding und dem Contextual Service Discovery eingesetzt wird und natürlich als Basis für die Handoverentscheidung gilt, erhält so eine greifbare Form für die Dienstauswahl und die Dienstnutzung, wie im Beispiel des Photoentwicklungsdienstes.
Der Kontext ist dabei neben der Signatur, der Semantik und des Protokolls eine der Betrachtungsebenen der Interoperabilität. Durch den Highlevel Service Handover, der auf der Grundlage der Ersetzbarkeit basiert, ist hiermit ein weiterer Schritt hin zur Interoperabilität verteilter Komponenten gesetzt worden.

Literaturverzeichnis

[BCC+ 02] BEDI, HARMEET, JONATHAN CHAWKE, FRANCISO CURBERA, GLEN DANIELS, DOUG DAVIS, MATTHEW J. DUFTLER, KEVIN MITCHELL, WILLIAM NAGY, SCOTT NICHOL, SAM RUBY, JAMES SNELL und SANJIVA WEERAWARANA: *Apache SOAP*. 2002, http://ws.apache.org/soap .

[BEK+ 00] BOX, DON, DAVID EHNEBUSKE, GOPAL KAKIVAYA, ANDREW LAYMAN, NOAH MENDELSOHN, HENRIK FRYSTYK NIELSEN, SATISH THATTE und DAVE WINER: *Simple Object Access Protocol (SOAP)*. http://www.w3.org/TR/SOAP/, May 2000.

[CaPa 98] CACERES, RAMON und VENKATA N. PADMANABHAN: *Fast and Scalable Wireless Handoffs in Support of Mobile Internet Audio*. Band 3, Seiten 351–363, 1998, citeseer.nj.nec.com/caceres98fast.html .

[Caro 00] CAROLIS, ANDREA DE: *QoS-Aware Handover for Mobile IP: Secondary Home Agent*. Mobile IP - Internet Draft, November 2000, http://www.alternic.org/drafts/drafts-d-e/draft-decarolis-qoshandover-01.pdf .

[CCMW 01] CHRISTENSEN, E., F. CURBERA, G. MEREDITH und S. WEERAWARANA: *Web Services Description Language (WSDL)*. http://www.w3.org/TR/wsdl, 2001.

[dam] *The DARPA Agent Markup Language*. , http://www.daml.org .

[FGM+ 99] FIELDING, R., J. GETTYS, J. MOGUL, H. FRYSTYK, L. MASINTER, P. LEACH und T. BERNERS-LEE: *Hypertext Transfer Protocol – HTTP/1.1*. June 1999, http://ietf.org/rfc/rfc2616.txt .

[gps] , http://www.galleon.eu.com/GPS-Time.htm .

[gsm] *Handover Procedures*. , $http://www.3gpp.org/ftp/Specs/archive/03_series/03.09/0309-700.zip$.

[jav a] *Java RMI*. , http://java.sun.com/products/jdk/rmi .

[jav b] *SUN Java*. , http://java.sun.com .

[KaBl 02] KAMMANN, JENS und TIM BLACHNITZKY: *Split-Proxy Concept for Application Layer Handover in Mobile Communication Systems*. In: *Proceedings IEEE MWCN 2002*, Stockholm, Sweden, September 2002. .

[KAP+ 00] KHALIL, MOHAMED, HASEEB AKHTAR, KRISH PILLAI, EMAD QADDOURA und NORTEL NETWORKS: *AAA Interface for IPv6 Handoff*. October 2000.

[KüLP 02] Küpper, Dr. A. und Prof. Dr. C. Linhoff-Popien: *Mobilkommunikation I*. 2002.

[LP 02] Linhoff-Popien, Prof. Dr. C.: *Verteilte Systeme / Ubiquitous Computing*. 2002.

[Mann 98] Mann, Steve: *What is a wearable computer*. May 1998, http://about.eyetap.org/fundamentals/ .

[M.F.] M.Femminella und F.Pugini: *Mobile IP Performance in high mobility environments*. , http://conan.diei.unipg.it/netweb/MIPperf.pdf .

[MKS$^+$ 01] Manner, J., M. Kojo, T. Suihko, P. Eardley, D. Wisely, BT, R. Hancock und N. Georganopoulos: *Mobility Related Terminology*. Internet Engineering Task Force - Internet Draft, Juli 2001, http://www.alternic.org/drafts/drafts-m-n/draft-manner-seamoby-terms-00.pdf .

[MRP$^+$ 03] Montenegro, Gabriel, Phil Roberts, Basavaraj Patil, Thomas Narten, Erik Nordmark und Thomas Narten: *P Routing for Wireless/Mobile Hosts (mobileip)*. February 2003, http://www.ietf.org/html.charters/mobileip-charter.html .

[oek 95] *Betriebswirtschaftslehre für Ingenieure und Informatiker*. Seiten 39–52. Verlag Vahlen, 1995.

[onc] *ONC RPC*. , http://www.onc-rpc-xdr.com .

[Perk 00] Perkins, Charles E.: *Fast Handovers for Mobile IPv6*. Internet Engineering Task Force - Internet Draft, November 2000, http://www.alternic.org/drafts/drafts-p-q/draft-perkins-mobileip-handover-00.pdf .

[SLPF 03a] Strang, Thomas, Claudia Linnhoff-Popien und Korbinian Frank: *Applications of a Context Ontology Language*. 2003.

[SLPF 03b] Strang, Thomas, Claudia Linnhoff-Popien und Korbinian Frank: *A Context Ontology Language to enable Contextual Interoperability*. 2003.

[smt] *Simple Mail Transfer Protocol (SMTP)*. , http://www.ietf.org/rfc/rfc0821.txt .

[Snoe 03] Snoeren, Mark Alexander Connell: *A Session-Based Architecture for Internet Mobility*. 2003.

[StLP 03] Strang, Thomas und Claudia Linnhoff-Popien: *Service Interoperabilität auf Kontextebene (Contextual Service Interoperability)*. In: Eckstein, Rainer und Robert Tolksdorf (Herausgeber): *Proceedings of Workshop XML-Technologien für Middleware / Middleware für XML-Anwendungen (XMIDX2003)*, GI-Edition Lecture Notes in Informatics (LNI), Seite 95ff, Berlin/Germany, February 2003. Gesellschaft für Informatik (GI), http://www.kn.op.dlr.de/ strang/paper/xmidx2003/XMIDX2003full.pdf .

[tom] , http://jakarta.apache.org/tomcat/index.html .

[udd 01] *Universal Description, Discovery and Integration (UDDI)*. http://www.uddi.org, 2001.

[utc] , $http : //www.bipm.fr/enus/5_Scientific/c_time/time_server.html$.

[VHT] Vallecillo, Antonio, Juan Hernandez und Jose M. Troya: *Component Interoperability*. .

[w3c] W3C: *World Wide Web Consortium.* , http://www.w3.org .

[w3c 03] W3C: *Extensible Markup Language.* 2003, http://www.w3.org/XML .

[web] *Web Services Activity.* , http://www.w3c.org/2002/ws .

[Weis 91] WEISER, MARK: *The Computer for the 21st Century.* September 91, http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html .

[WoLi 97] WONG, DANIEL und TENG JOON LIM: *Soft Handoffs in CDMA Mobile Systems.* 1997.

[wsc] *Web Service Choreography Interface.* , http://w3.org/TR/wsci .