

Der Einfluss verschiedener Mobilitätsgrade auf die Architektur von Informationssystemen

Matthias Book, Volker Gruhn, Malte Hülder, Clemens Schäfer

Lehrstuhl für Angewandte Telematik/e-Business,* Universität Leipzig
Klostergasse 3, 04109 Leipzig
{book, gruhn, huelder, schaefer}@ebus.informatik.uni-leipzig.de

Abstract: Bei der Entwicklung von mobilen Informationssystemen stehen die Entwickler oft vor immer wiederkehrenden Entwurfsentscheidungen, die von einer Anzahl noch unstrukturierter Kriterien abhängen. Den kompletten Entscheidungsprozess für jedes einzelne Projekt von vorne bis hinten zu durchlaufen ist ineffizient und fehleranfällig, trotzdem gibt es noch keine umfassende Sammlung von „Best Practices“, die diesen Prozess verkürzen könnte. Wir präsentieren daher die Grundlagen eines Klassifikationsschemas für mobile Informationssysteme, das Entwicklern hilft, Anwendungen anhand von Anforderungen auf höherer Ebene zu klassifizieren und entsprechende Architekturentscheidungen zu treffen. Im Anschluss an die Diskussion der vorgestellten Kriterien schlagen wir Erweiterungen des Klassifikationsschemas und Folgerungen, die daraus gezogen werden können, vor.

1 Einleitung

In den letzten Jahren hat sich das Verlangen der Benutzer gesteigert, mit ihren Anwendungen immer flexibler arbeiten zu können – idealerweise sollte jeder Dienst auf jedem Gerät an jedem Ort und zu jeder Zeit verfügbar sein. Die weite Verbreitung von drahtlosen Netzwerken und die Verfügbarkeit von Endgeräten haben die Entwicklung von mobilen Anwendungen ermöglicht, die uns Weisers Vision des „ubiquitous computing“ [We91] einen großen Schritt näher bringen. In diesem noch reifenden Feld besteht Bedarf dafür, die Charakteristika und Prinzipien aufzuzeigen, die mobilen Systemen zugrunde liegen. Wenn man bedenkt, dass es sich immer noch um ein recht junges Gebiet handelt, dessen wichtigste Technologien (wie GSM, GPRS und IEEE 802.11 WLAN) erst im Laufe der 90er Jahre eingeführt wurden oder (wie UMTS) heute erst eingeführt werden, wird schnell klar, dass wesentliche Methoden für die Entwicklung von mobilen Anwendungen wenig ausgereift sind und Optimierungspotenzial besitzen.

Wir beschränken uns hier auf die Architektur von mobilen Informationssystemen mit ausgeprägten Client-Server-Charakteristika, da eine große Anzahl solcher Informationssysteme heute bereits im Einsatz ist, wenn auch vielfach ohne mobile Unterstützung. (Offen-

*Der Lehrstuhl für Angewandte Telematik/e-Business ist ein Stiftungslehrstuhl der Deutschen Telekom AG.

sichtlich existieren auch andere Klassen von mobilen Anwendungen ohne explizite Client-Server-Charakteristika (z.B. Peer-to-Peer Anwendungen in mobilen Ad-hoc-Netzwerken). Der Klarheit wegen beschränken wir uns hier allerdings auf mobile Informationssysteme. Eine Erweiterung des Klassifikationsschemas um andere Anwendungsarten wird ein wichtiger Aspekt unserer zukünftigen Forschung sein, wie im Abschnitt 5 beschrieben.) Viele Organisationen, z.B. im Versicherungssektor, evaluieren derzeit, ob durch mobilen Zugriff die Geschäftsprozesse, die bisher in diesen Legacy-Anwendungen implementiert sind, effizienter gestaltet werden können. Dementsprechend sind die Implikationen für die Architektur solcher Systeme, die durch die Einführung eines mobilen Zugriffs entstehen, ein wichtiges Thema für die Industrie.

Wir sehen es als wahrscheinlich an, dass einige Aspekte der Entwicklung von mobilen Informationssystemen der Entwicklung von traditionellen Informationssystemen sehr ähnlich sind, während andere Aspekte zwar spezifischer für den mobilen Einsatz, aber immer noch unabhängig von einer konkreten, individuellen Anwendung sind. Daher nehmen wir an, dass eine Klassifikation der mobilen Aspekte von Informationssystemen Einblicke in diejenigen Charakteristika liefert, die einen effizienteren Entwicklungsprozess ermöglichen. Konkrete Maßnahmen, um architekturelle, technische und prozessorale Entwurfsentscheidungen auf Basis des vorgestellten Klassifikationsschemas zu unterstützen, sind Teil unserer derzeitigen Forschungsaktivitäten, um ein ehrgeiziges Ziel zu erreichen: Mit Hilfe eines solchen Klassifikationsschemas müssten die Entwickler nicht mehr jede architekturelle, technische und prozessorale Entwurfsentscheidung für jede Anwendung von Grund auf neu treffen, sondern könnten einige der benötigten Entwurfsmuster, Methoden und Prozesse anhand der Klasse ableiten, zu der die zu implementierende Anwendung gehört.

In den folgenden Abschnitten werden wir einen kurzen Überblick über existierende Klassifikationsschemata geben, die bereits vorgeschlagen wurden (Abschnitt 2). Danach werden wir unsere Definitionen verschiedener Aspekte von Mobilität und Konnektivität vorstellen, die zentrale Bedeutung für unser Klassifikationsschema haben (Abschnitt 3), und anhand einiger Praxisbeispiele zeigen, wie aus der Klassifikation Architekturentscheidungen abgeleitet werden können (Abschnitt 4). Abschließend diskutieren wir die bisher eingeführten Kriterien und zeigen weitere Möglichkeiten auf, mit deren Hilfe das Klassifikationsschema verfeinert werden kann (Abschnitt 5).

2 Verwandte Arbeiten

In der Literatur zu findende Klassifikationsschemata für mobile Anwendungen basieren in der Regel auf fachlichen Kriterien, d.h. sie klassifizieren Anwendungen nach Aufgaben und Prozessen, die der Anwender mit ihnen ausführen kann. Varshney und Vetter [VV02] unterscheiden beispielsweise mobiles oder drahtloses Finanzmanagement, Werbung, Bestandsverwaltung, Produktsuche und -bestellung, Dienstverwaltung, Re-Engineering, Auktionen, Unterhaltung, Büro, Fernunterricht und Rechenzentrumsanwendungen.

Die Kriterien, nach denen man spezifische Anwendungen mit diesen Klassen assoziieren

soll, sind nicht explizit erläutert, und die Klassen selbst scheinen nicht disjunkt zu sein. Das lässt die Klassifikation unklar erscheinen und macht es schwer, präzise Schlussfolgerungen aus ihr zu ziehen. Varshney und Vetter zeigen eine Liste von Netzwerkanforderungen auf, die für Anwendungen bestimmter Klassen erfüllt sein müssen (z.B. Lokationsmanagement, Multicast-Unterstützung, Zuverlässigkeit des Netzwerks, Dienstgüte und Roaming über Netzgrenzen hinweg). Da einige dieser Anforderungen für jede Klasse aufgeführt werden, scheinen sie nicht wirklich in Korrelation zu den Charakteristika einer Klasse zu stehen, sondern vielmehr grundsätzliche Anforderungen zu sein, die jede mobile Anwendung in stärkerer oder schwächerer Form stellt.

Klassifikationsansätze, die technologiezentrierte Kriterien bemühen, wie z.B. die zugrunde liegenden Netzwerkprotokolle, erscheinen auf den ersten Blick hilfreich zu sein, da die ausgewählte Technologie weitreichende Konsequenzen für das Design der Anwendung haben kann. Beispielsweise impliziert die Entscheidung für HTTP als Protokoll zur Datenübertragung, dass die Client-Anwendung das Pull-Paradigma benutzen muss, um mit dem Server zu kommunizieren, während der Server von sich aus keine Kommunikation mit dem Client initiieren kann. Dadurch ergeben sich Einschränkungen in der Art und Weise der Interaktion, auf die Nutzer mit der Anwendung arbeiten können [Ri96]. Als weiteres Beispiel müssen Entwickler von GPRS-basierten Anwendungen im Auge behalten, dass die Latenzzeiten ca. 25 Sekunden für den Aufbau einer Verbindung und weitere 2 Sekunden für jeden Request-Response-Zyklus zusätzlich zur tatsächlichen Übertragungszeit der Daten betragen [CP02]. Für eine Anwendung, die viele Request-Response-Zyklen benötigt (z.B. um eine Webseite aus mehreren Elementen zusammzusetzen), kann diese Einschränkung, die aus den technischen Eigenschaften einer unteren Ebene des Protokollstapels herrührt, drastische Auswirkungen auf deren Benutzbarkeit haben.

Während Anwendungen sicherlich sehr exakt anhand der zugrunde liegenden technischen Kriterien klassifiziert werden können, propagieren wir, dass dies nicht *a priori* getan werden sollte, da die technischen Charakteristika anhand der Aufgaben definiert werden sollten, die ein Anwender mit der Anwendung ausführen möchte und nicht umgekehrt. Unsere Motivation ist daher, Anwendungen so weit wie möglich anhand nichttechnischer Kriterien zu klassifizieren, sodass wir die notwendigen technischen Anforderungen daraus ableiten können.

Aus einer weiteren Perspektive können wir verschiedene Grade der Mobilität unterscheiden und sie für unsere Klassifikation von Anwendungen nutzen. In diesem Kontext unterscheidet Pandya [Pa00] Gerätemobilität, Nutzermobilität und Dienst-Portabilität: Gemäß seiner Definition ist Gerätemobilität dann gegeben, wenn ein Gerät mit einem Netzwerk verbunden bleibt, während es sich physisch in Bewegung befindet. Nutzermobilität bedeutet, dass der Benutzer nicht an ein bestimmtes Gerät gebunden ist, sondern verschiedene Geräte benutzen kann, um einen Dienst zu nutzen. Schließlich erfährt ein Nutzer Dienstportabilität, wenn er einen Dienst überall benutzen kann, unabhängig von seinem Aufenthaltsort.

Dieser Ansatz, von konkreten fachlichen und technischen Charakteristika zu abstrahieren, scheint ein Schritt in die richtige Richtung zu sein. Allerdings meinen wir, dass der Begriff der Gerätemobilität anders definiert werden sollte, um unterschiedliche Mobilitätsgrade zu berücksichtigen, wie sie im Abschnitt 3 beschrieben werden. Pandyas Nutzermob-

bilität sollte in heutigen Anwendungen üblicherweise schon gegeben sein: Ein Gerät, das so eng mit seinem Benutzer verbunden ist, dass es von niemand anders bedient werden kann, und dass der Benutzer selbst kein anderes Gerät benutzen kann, um die gleiche Aufgabe damit zu erledigen, ist praktisch unvorstellbar. Auch wenn der Benutzer Daten lokal auf einem Gerät speichert (z.B. zu Personalisierungszwecken), bleibt die Anwendung prinzipiell auch mit anderen Geräten nutzbar, wenn auch unter Verzicht auf die lokal gespeicherten Daten (z.B. Personalisierungsprofile). Auch in diesem Fall ist daher eine (allenfalls auf Ebene der Geschäftslogik eingeschränkte) Nutzermobilität gegeben. Sie kann daher nur schwer als Unterscheidungskriterium für mobile Anwendungen herangezogen werden. Schließlich erscheint Pandyas Definition der Dienstportabilität wie eine Kombination aus Geräte- und Nutzermobilität, was sie quasi äquivalent zur Gerätemobilität macht, da die Nutzbarkeit eines Dienstes an beliebigen Aufenthaltsorten nicht davon abhängt, ob der Benutzer an diesen Orten verschiedene Geräte oder immer das eigene verwendet. Insbesondere impliziert die Dienstportabilität nicht, dass die Software, die den Dienst zur Verfügung stellt, selbst mobil ist.

In ihrer „Roadmap“ machen Roman, Picco und Murphy [RPM00] die Unterscheidung zwischen physischer Mobilität (Bewegung des mobilen Endgerätes) und logischer Mobilität (mobiler Code und dessen Zustandsdaten). Aus unserer Sicht ist die Situation hier ähnlich zu Pandyas Definition: Unserer Meinung nach ist der Begriff der physischen Mobilität von Roman et al. zu grob und sollte verfeinert werden. Das Gleiche gilt für die Auffassung von Mobilität durch Issarny et al. [Is04]: Hier wird der Begriff der Computer- und Nutzer-Mobilität benutzt, jedoch werden diese Begriffe nicht ordentlich definiert und bieten somit keine solide Grundlage.

Aufgrund dieser Probleme mit den existierenden Klassifikationsansätzen, wollen wir die existierenden Definitionen schärfen und präsentieren daher einen alternativen Ansatz, der von konkreten fachlichen und technischen Kriterien abstrahiert und stattdessen auf Interaktionsmustern höherer Ebene, Geräteeigenschaften und Anforderungen an die zu implementierenden Dienste basiert.

3 Klassifikationskriterien

In den folgenden Abschnitten stellen wir die drei Klassifikationskriterien Benutzermobilität, Gerätemobilität und Dienstkonnektivität vor und diskutieren ihre Korrelation.

Da heutige Anwendungen ihren Nutzern i.d.R. ein ganzes Spektrum verschiedener Dienste anbieten (wobei ein *Dienst* als ein Teil einer Anwendung definiert sei, der ein bestimmtes Feature bereitstellt oder einen bestimmten Geschäftsvorfall unterstützt), kann es schwierig sein, die gesamte Anwendung unzweideutig einer einzigen Kategorie zuzuordnen. Unsere Kriterien erlauben daher die unabhängige Klassifikation einzelner Dienste innerhalb einer Anwendung.

3.1 Benutzermobilität

Da unser Ziel ist, Architektur-, Infrastruktur- und Implementierungsaspekte eines Dienstes aus seiner Klassifikation abzuleiten, wollen wir die Klassifikationskriterien möglichst wenig technisch formulieren, sondern uns stattdessen auf Benutzeranforderungen und Nutzungsmuster konzentrieren. Infolgedessen drückt das erste Kriterium den Grad der Orts- und Bewegungsfreiheit aus, der einem Benutzer gegeben wird, während dieser mit einem Dienst arbeitet (solange der Benutzer *nicht* mit dem Dienst arbeitet, sind Ort und Bewegung des Benutzers nicht relevant, da das System weder Kenntnis davon hat noch davon beeinflusst wird). Wir definieren vier Grade der *Benutzermobilität*:

- Ein *lokal* arbeitender Benutzer kann den Dienst nur am Ort der Dienstbereitstellung nutzen.
- Ein *verteilt* arbeitender Benutzer kann den Dienst von einem entfernten Ort aus nutzen.
- Ein *mobil* arbeitender Benutzer kann den Dienst von verschiedenen entfernten Orten aus nutzen.
- Ein *in Bewegung* arbeitender Benutzer kann den Dienst nutzen, während er den entfernten Ort wechselt.

Bei der Betrachtung der Mengen aller lokal, verteilt, mobil und in Bewegung arbeitenden Benutzer (gekennzeichnet mit U_{loc} für *local*, U_{dis} für *distributed*, U_{mob} für *mobile* und U_{mot} für *in-motion user*) stellen wir fest, dass die Mengen der lokal und verteilt arbeitenden Benutzer offensichtlich disjunkt sind ($U_{\text{loc}} \cap U_{\text{dis}} = \emptyset$). Mobile Benutzer sind hingegen Spezialfälle von verteilt arbeitenden Benutzern, und in Bewegung arbeitende Benutzer sind mobile Benutzer mit zusätzlichen Anforderungen. Folglich sind die verteilt arbeitenden Benutzer eine Obermenge der mobil arbeitenden Benutzer, die wiederum eine Obermenge der in Bewegung arbeitenden Benutzer sind ($U_{\text{dis}} \supset U_{\text{mob}} \supset U_{\text{mot}}$). Die Unterscheidung zwischen letzteren Gruppen wird klar, wenn wir die Differenzen der Mengen betrachten, d.h. verteilt, aber nicht mobil arbeitende Benutzer ($U_{\text{dis}} \setminus U_{\text{mob}}$) sowie mobil, aber nicht in Bewegung arbeitende Benutzer ($U_{\text{mob}} \setminus U_{\text{mot}}$):

- Ein verteilt, aber *immobil* arbeitender Benutzer kann den Dienst nur von genau einem entfernten Ort aus nutzen.
- Ein mobil, aber *ruhend* arbeitender Benutzer kann den Dienst nur nutzen, während er statisch an einem entfernten Ort verbleibt.

Dabei ist zu beachten, dass ein in Bewegung arbeitender Benutzer das zur Arbeit verwendete Gerät ständig mit sich tragen muss (z.B. einen PDA). Wenn der Benutzer jedoch immer ruht, während er mit dem Dienst arbeitet, so muss er kein eigenes Gerät mit sich tragen, sondern kann beliebige Geräte nutzen, die an den Orten, an denen er arbeiten will, bereitgestellt werden (z.B. in einem Konferenzzentrum verteilte Terminals).

Nach der Definition der Benutzermobilität konzentrieren wir uns nun auf die Mobilität der Geräte, die Benutzer zur Arbeit mit einem Dienst verwenden.

3.2 Gerätemobilität

Intuitive Definitionen von Objektmobilität heben üblicherweise darauf ab, ob es möglich ist, ein Objekt physisch zu bewegen, z.B. durch Definition von mobil als „beweglich, nicht an einen festen Standort gebunden“ [Du01]. Aus Sicht der Softwaretechnik ist diese Definition von physischer Mobilität jedoch nicht ausreichend, weil sie nicht aussagt, wie die „Standorte“ voneinander zu unterscheiden sind und damit festzustellen ist, dass sich ein Objekt bewegt hat – wir benötigen ein Bezugssystem, in Relation zu dem sich die Objekte bewegen.

Kommunikationsnetzwerke stellen solch ein Bezugssystem in Form ihrer Zugangspunkte (z.B. GSM-Mobilfunkzellen, WLAN-Hot-Spots etc.) zur Verfügung. Jeder dieser Zugangspunkte deckt ein bestimmtes Gebiet ab, innerhalb dessen das Gerät sich befinden muss, um Zugang zum Netzwerk zu bekommen. In diesem Szenario können wir Mobilität als *Fähigkeit, sich leicht zwischen den Abdeckungsgebieten der Zugangspunkte eines Netzwerks zu bewegen bzw. bewegt zu werden* definieren. Mit dieser Definition lassen sich vier Grade der *Gerätemobilität* unterscheiden:

- Ein *lokal* funktionierendes Gerät kann sich nicht mit dem Netzwerk verbinden.
- Ein *verteilt* funktionierendes Gerät kann sich mit dem Netzwerk verbinden.
- Ein *mobil* funktionierendes Gerät kann sich mit verschiedenen Netzwerkzugangspunkten verbinden.
- Ein *in Bewegung* funktionierendes Gerät kann sich mit verschiedenen Netzwerkzugangspunkten verbinden, während sein Benutzer es verwendet.

Die Mengen aller lokal, verteilt, mobil und in Bewegung funktionierenden Geräte (D_{loc} , D_{dis} , D_{mob} und D_{mot}) weisen dieselben Beziehungen wie die verschiedenen Grade der Benutzermobilität auf: $D_{\text{loc}} \cap D_{\text{dis}} = \emptyset$ und $D_{\text{dis}} \supset D_{\text{mob}} \supset D_{\text{mot}}$. Die Unterscheidung zwischen den netzwerkfähigen Geräten wird wiederum klarer, wenn wir die Differenzmengen $D_{\text{dis}} \setminus D_{\text{mob}}$ und $D_{\text{mob}} \setminus D_{\text{mot}}$ betrachten, die immobile und ruhende Geräte definieren:

- Ein verteilt, aber *immobil* funktionierendes Gerät kann sich immer nur mit ein und demselben Netzwerkzugangspunkt verbinden.
- Ein mobil, aber *ruhend* funktionierendes Gerät muss mit dem gleichen Netzwerkzugangspunkt verbunden bleiben, solange der Benutzer es verwendet.

Zu beachten ist, dass in Gegenwart mehrerer unterschiedlicher Kommunikationsnetze mit verschiedener Zugangspunktdichte ein Gerät mobil in Bezug auf ein Netzwerk sein kann

(d.h. sich zwischen den Abdeckungsgebieten verschiedener Zugangspunkte bewegt), während es zur gleichen Zeit immobil in Bezug auf ein anderes Netzwerk ist (d.h. jederzeit innerhalb des Abdeckungsgebiets eines anderen Zugangspunkts bleibt). Die Wahl des Netzwerks, das als Bezugssystem dienen soll, bedarf daher besonderer Überlegung.

Da praktisch jedes Objekt mit hinreichend großem Aufwand zwischen verschiedenen Orten bewegt werden kann, bedeutet das Wort „leicht“ eine wichtige Beschränkung in der obigen Mobilitätsdefinition: Es spezifiziert nicht absolut, unter welchen Bedingungen ein Objekt als mobil betrachtet werden kann, sondern erfordert, dass der zu seiner Bewegung erforderliche Aufwand in Relation zu anderen Metriken (wie z.B. dem Nutzen seiner Bewegung oder dem Aufwand zur Bewegung anderer Objekte) bewertet werden muss. Allgemein gesprochen kann dieser Aufwand durch Gewicht oder Befestigung des betreffenden Objekts bestimmt sein. In der netzwerkspezifischen Definition ist zu berücksichtigen, dass die Verbindung zu verschiedenen Netzzugangspunkten einen bestimmten (einmaligen oder wiederkehrenden) Konfigurationsaufwand erfordern kann, der das Gerät in dem Sinne immobil macht, dass es sich nicht „leicht“ zwischen den Abdeckungsgebieten verschiedener Netzzugangspunkte bewegen kann, weil der zur Verbindung notwendige Konfigurationsaufwand unverhältnismäßig hoch wäre.

Wir drücken die Mobilität von Geräten in Form ihrer Fähigkeit, sich leicht mit verschiedenen Netzzugangspunkten zu verbinden, aus, da aus Sicht der Softwaretechnik kleine physische Bewegungen innerhalb des Abdeckungsgebiets eines Zugangspunkts nicht feststellbar und daher irrelevant sind.¹ Größere physische Bewegungen, die das Gerät aus dem Abdeckungsgebiet eines Zugangspunkts und in das Gebiet eines anderen führen, erfordern einen Handover-Prozess, der entweder transparent von der Netzinfrastruktur oder explizit von der Anwendung gehandhabt werden muss. Infolgedessen ist Gerätemobilität nicht nur ein Merkmal des Geräts, sondern auch der Netzinfrastruktur und u.U. der Anwendung.

3.3 Dienstkonnektivität

Aus dem OSI-Referenzmodell [ISO94] folgt, dass die Mobilität eines Dienstes von der Mobilität des Geräts bestimmt wird, das den Dienst zum entfernten Benutzer transportiert. Dienstmobilität ist daher kein unabhängiges Kriterium, sondern an die Gerätemobilität gekoppelt. Dienste müssen jedoch dafür vorbereitet sein, mit einem Seiteneffekt der Gerätemobilität umzugehen: Da sich die Abdeckungsgebiete der Zugangspunkte eines Netzwerks nicht notwendigerweise überall überlappen müssen, können Gebiete ohne Netzabdeckung existieren. Infolgedessen kann ein mobil funktionierendes Gerät an bestimmten Orten u.U. nicht in der Lage sein, sich mit dem Netzwerk zu verbinden, und ein in Bewegung funktionierendes Gerät kann zeitweise seine Netzwerkverbindung verlieren, wenn es sich durch nicht abgedecktes Gebiet bewegt. Die Vorkehrungen, die von einem Dienst getroffen wer-

¹Dies gilt offensichtlich nur, solange wir uns mit dem Einfluss von Mobilität auf die Anwendungsarchitektur und Kommunikationsinfrastruktur beschäftigen. Für ortsbewusste Anwendungen kann eine kleine physische Bewegung sehr wohl wichtige Konsequenzen haben – diese liegen jedoch im Bereich der Geschäftslogik. Eine Untersuchung der Beziehungen zwischen Gerätemobilität und Ortsbewusstsein ist ein weiteres Forschungsthema, wie im Abschnitt 5 beschrieben.

den müssen, um diese Situationen zu handhaben, richten sich danach, wie stark er von der Netzverbindung abhängt. Wir definieren vier Grade der *Dienstkonnektivität*:

- Ein *Offline-Dienst* benötigt niemals eine Netzverbindung.
- Ein *Hybrid-Offline-Dienst* benötigt gelegentlich eine Netzverbindung.
- Ein *Hybrid-Online-Dienst* benötigt meistens eine Netzverbindung.
- Ein *Online-Dienst* benötigt ständig eine Netzverbindung.

Die Mengen aller Offline-, Hybrid-Offline, Hybrid-Online und Online-Dienste (bzw. *services*, gekennzeichnet durch S_{off} , S_{hoff} , S_{hon} und S_{on}) sind disjunkt ($S_{\text{off}} \cap S_{\text{hoff}} \cap S_{\text{hon}} \cap S_{\text{on}} = \emptyset$).

Es lässt sich einwenden, dass die Definitionen der hybriden Konnektivitätsgrade nicht völlig trennscharf sind, was sicher berechtigt ist. Ähnlich wie die „Leichtigkeit“ der Bewegung im vorherigen Abschnitt lässt sich das begrenzende Verhältnis von verbundenem zu unverbundenem Betrieb kaum absolut spezifizieren. Vielmehr muss der tolerierbare Grad des unverbundenen Betriebs in Bezug zu anderen Metriken wie dem Grad der Autonomie gesetzt werden, den der vom Dienst dargestellte Geschäftsprozess sowie gegebene Architektur- oder Infrastrukturaspekte erlauben.

Typischerweise wird ein Hybrid-Offline-Dienst dem Benutzer erlauben, einen Geschäftsprozess offline durchzuführen, und nur gelegentlich und kurzzeitig eine Netzverbindung benötigen, um die Ein-/Ausgabedaten des Prozesses zu übertragen. Im Gegensatz dazu wird ein Hybrid-Online-Dienst häufig über das Netzwerk kommunizieren, während der Benutzer einen Geschäftsprozess bearbeitet, vorübergehende Verbindungsabbrüche jedoch ohne Beeinträchtigung des Betriebs überbrücken (z.B. durch das Zwischenspeichern von Ein-/Ausgabedaten, die übertragen werden, sobald die Verbindung wieder verfügbar ist, oder durch die autonome Steuerung trivialer Aktivitäten ohne Verbindungsnotwendigkeit).

3.4 Korrelationen zwischen den Klassifikationskriterien

Bei der Betrachtung möglicher Kombinationen der drei Kriterien Benutzermobilität, Gerätemobilität und Dienstkonnektivität innerhalb einer Anwendung zeigt sich, dass nur bestimmte Kombinationen praktikabel sind, während andere sich widersprechen.

Wenn ein in Bewegung arbeitender Benutzer (U_{mot}) z.B. versucht, ein Gerät der Kategorie „mobil, aber ruhend“ ($D_{\text{mob}} \setminus D_{\text{mot}}$) zu verwenden (d.h. ein Gerät, das sich zwar an beliebigen Punkten mit dem Netzwerk verbinden kann, jedoch nicht in Bewegung), so wird er nicht in der Lage sein, einen Online-Dienst (S_{on}) zu benutzen, da diese Dienstklasse eine permanente Netzverbindung erfordert, die das mobil, aber nur ruhend funktionierende Gerät nicht bieten kann, während der Benutzer in Bewegung ist. Ein in Bewegung arbeitender Benutzer *kann* jedoch ein mobil, aber nur ruhend funktionierendes Gerät verwenden, wenn er mit einem Hybrid-Online- oder Hybrid-Offline-Dienst (S_{hon} oder S_{hoff}) arbeitet, der

	D_{loc}	$D_{dis} \setminus D_{mob}$	$D_{mob} \setminus D_{mot}$	D_{mot}
U_{loc}	S_{off}	S_{off}	S_{off}	S_{off}
$U_{dis} \setminus U_{mob}$	\emptyset	$S_{hoff} \cup S_{hon} \cup S_{on}$	$S_{hoff} \cup S_{hon} \cup S_{on}$	$S_{hoff} \cup S_{hon} \cup S_{on}$
$U_{mob} \setminus U_{mot}$	\emptyset	S_{hoff}	$S_{hoff} \cup S_{hon} \cup S_{on}$	$S_{hoff} \cup S_{hon} \cup S_{on}$
U_{mot}	\emptyset	S_{hoff}	$S_{hoff} \cup S_{hon}$	$S_{hoff} \cup S_{hon} \cup S_{on}$

Tabelle 1: Korrelationen zwischen den Klassifikationskriterien.

keine permanente Netzverbindung benötigt, sondern auch einen gewissen Grad unverbundenen Betriebs zulässt.

Tabelle 1 fasst Beziehungen wie diese zusammen: Wenn ein Benutzer zu einer bestimmten Gruppe aus der ersten Spalte gehört und ein Gerät aus einer bestimmten Gruppe in der ersten Zeile nutzt, dann muss die praktikable Dienstkonnektivität aus der Menge in der zugehörigen Tabellenzelle stammen: Für einen Benutzer der Gruppe U_{mot} und ein Gerät der Gruppe $D_{mob} \setminus D_{mot}$ ist die Menge aller möglichen Dienstkonnektivitätsgrade z.B. $S_{hoff} \cup S_{hon}$.

In Weiterführung dieser Diskussion gilt: Wenn ein in Bewegung arbeitender Benutzer (U_{mot}) ein verteilt, aber nur immobil funktionierendes Gerät ($D_{dis} \setminus D_{mob}$) benutzt, d.h. eines, das nur von einem einzigen, bestimmten Ort aus eine Netzverbindung herstellen kann, so kann der Dienst nur hybrid-offline (S_{hoff}) sein, d.h. er darf die meiste Zeit über (während der Benutzer sich bewegt) keine Netzwerkverbindung erfordern, und nur kurzzeitig eine Verbindung aufbauen, wenn der Benutzer an dem spezifischen Ort ruht, von dem sich das Gerät verbinden kann. Wenn ein in Bewegung arbeitender Benutzer jedoch ein in Bewegung funktionierendes Gerät benutzt, dass die Verbindung mit verschiedenen Netzzugangspunkten aufrecht erhalten kann, sogar wenn der Benutzer sich zwischen ihnen bewegt (D_{mot}), so kann er Online-Dienste, die eine permanente Verbindung benötigen, ebenso benutzen wie Hybrid-Online-Dienste, wenn er immun gegen gelegentliche Verbindungsabbrüche sein möchte. Er kann natürlich ebenfalls Hybrid-Offline-Dienste nutzen, auch wenn diese die Netzwerkfähigkeiten des Geräts nicht voll ausnutzen würden.

Bei näherer Betrachtung der Tabelle fällt auf, dass die Korrelation zwischen den drei Kriterien (mit Ausnahme der Ausprägungen U_{loc} , D_{loc} und S_{off}) den folgenden Regeln gehorcht:

- Wenn für eine bestimmte Kombination von Benutzer- und Dienstmobilität eine bestimmte Dienstkonnektivität möglich ist, so sind alle schwächeren Konnektivitätsgrade für diese Kombination ebenfalls möglich.
- Um Online-Dienstkonnektivität zu erreichen, muss das Gerät in der gleichen oder einer höheren Mobilitätsklasse als der Benutzer sein.
- Wenn das Gerät in einer niedrigeren Mobilitätsklasse als der Benutzer ist, wird die mögliche Dienstkonnektivität (u.U. überproportional) reduziert.

Lokal arbeitende Benutzer, lokal funktionierende Geräte und Offline-Dienste (U_{loc} , D_{loc} und S_{off}) folgen diesen Regeln nicht, da ihre Mengen disjunkt von den netzwerkfähigen

Graden der drei Kriterien sind. Wenn ein lokaler Benutzer einen Dienst nur auf dem gleichen Gerät benutzt, auf dem dieser auch läuft, so ist keine Netzverbindung erforderlich, und der Dienst wird als offline klassifiziert (auch wenn das Gerät netzwerkfähig wäre, wird dieses Feature schließlich nicht vom Dienst genutzt). Umgekehrt kann auf keiner Geräteklasse ein Offline-Dienst verfügbar sein, wenn der Benutzer nicht lokal arbeitet, da grundsätzlich eine Netzverbindung notwendig wäre, um ihn zu erreichen.

Es ließe sich einwenden, dass ein in Bewegung befindlicher Benutzer doch einen Offline-Dienst (z.B. einfache Rechenfunktionen) auf einem mobilen Gerät nutzen könnte, das er mit sich führt. Da wir unter Benutzermobilität jedoch Bewegung verstehen, die entfernt von einem Dienst stattfindet, und Gerätemobilität als Bewegung relativ zu den Zugangspunkten eines Netzwerks definiert haben, sind Benutzer, Gerät und Dienst in diesem Szenario alle lokal und stationär zueinander. Aus technischer Sicht ist die physische Mobilität dieses geschlossenen Systems nicht relevant – daher gehört es ebenfalls zur Kategorie mit lokalem Benutzer und lokalem Gerät.

4 Technische Implikationen der Klassifikation

Bislang erlauben unsere Klassifikationskriterien den Entwicklern, geeignete Grade der Dienstkonnektivität ausgehend von Rahmenbedingungen – gegeben in Form der gewünschten Benutzermobilität und der verfügbaren Gerätemobilität² – abzuleiten. Oft benötigen die verschiedenen Konnektivitätsgrade bei der Implementierung der Dienste jedoch unterschiedliche technische Vorkehrungen.

Daher ist es in einem zweiten Schritt nötig, architekturelle Implikationen der unterschiedlichen Arten der Konnektivität zu identifizieren. Hierbei konzentrieren wir uns auf Informationssysteme mit Client-Server-Charakteristik – andere Arten mobiler Anwendungen (wie z.B. Telemetrie- oder Peer-to-Peer-Anwendungen [VV02]) können durchaus andere Anforderungen an die Architektur aufweisen, was Gegenstand weiterer Forschung ist.

4.1 Verteilung der Architekturschichten

Informationssysteme speichern und verarbeiten Daten entsprechend den Anforderungen einer bestimmten Anwendungsdomäne. Für ihre Architektur ist ein Drei-Schicht-Aufbau üblich, bei dem zwischen Präsentationslogik, Geschäftslogik und Datenhaltungsschicht unterschieden wird. In einer Client-Server-Umgebung können diese Schichten nach unterschiedlichen Strategien verteilt sein.

Abbildung 1 verdeutlicht, wie diese Verteilungsmuster vom Grad der Dienstkonnektivität abhängen. Für jeden Konnektivitätsgrad zeigt sie, welche Teile der drei Schichten auf

²In Abhängigkeit von den verfügbaren Informationen kann die Korrelationstabelle benutzt werden, um den Grad der Benutzermobilität aus einer gegebenen Kombination aus Gerätemobilität und Dienstkonnektivität abzuleiten, oder um den Grad der Gerätemobilität zu bestimmen, der nötig ist, um einen gegebenen Dienst Benutzern mit dem erforderlichen Mobilitätsgrad zur Verfügung zu stellen.

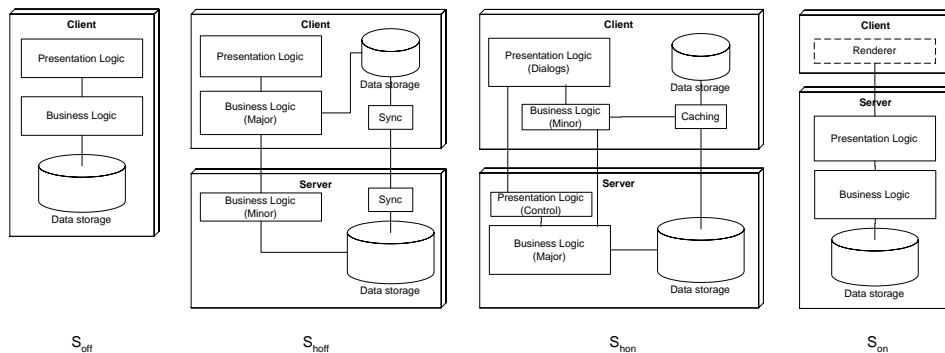


Abbildung 1: Verteilungsmuster für Informationssysteme, abhängig von Dienstkonnektivität.

den Client und welche auf den Server verteilt sein müssen und welche Informationen zwischen diesen beiden Seiten kommuniziert werden müssen. Die gezeigten Verteilungsmuster basieren auf der Überlegung, welche *minimalen Client-Anforderungen* und welche *maximalen Server-Anforderungen* für die zugeordnete Dienstkonnektivität benötigt werden. Beispielsweise könnte im Fall, dass mehr Präsentationslogik auf der Client-Seite eines Online-Dienstes implementiert ist, der Grad der Konnektivität derselbe bleiben. Falls jedoch keine Daten auf der Client-Seite eines Hybrid-Online-Dienstes gespeichert werden, würde eine Online-Konnektivität für den Datenhaltungsschicht erforderlich. Folgende Regeln drücken diese Zusammenhänge aus:

- Verschiedene Mobilitätsklassen eines Dienstes können unterschiedliche Grade von Konnektivität unterstützen.
- Das Verschieben von Logik vom Server zum Client innerhalb einer Mobilitätsklasse erlaubt einen niedrigeren Konnektivitätsgrad für diese Mobilitätsklasse.
- Das Verschieben von Logik vom Client zum Server innerhalb einer Mobilitätsklasse erfordert einen höheren Konnektivitätsgrad.
- Der gesamte Konnektivitätsgrad eines Dienstes wird durch den höchsten Grad in allen seinen Mobilitätsklassen bestimmt.

4.2 Beispiele

Um den Klassifikationsprozess zu illustrieren, wenden wir ihn nun auf verschiedene Praxisbeispiele aus aktuellen Industrieprojekten an. Nach der Identifikation der Anforderungen und der Ableitung der verfügbaren architekturellen Wahlmöglichkeiten gemäß der hier vorgeschlagenen Methodik vergleichen wir die Ergebnisse mit den Architekturen, die in den jeweiligen Projekten tatsächlich gewählt wurden.

Als erstes Beispiel betrachten wir ein Informationssystem, das für ein Wasserversorgungsunternehmen implementiert wurde. Es erlaubt Bauleitern das Mitführen elektronischer Kopien von Projektdokumenten auf einem Tablet-PC, während sie sich auf einer Baustelle aufhalten. Auch wenn die Dokumente nicht editierbar sind, können die Bauleiter dennoch geschäftsprozessrelevante Daten aufnehmen und in das System eingeben. Da der Bauleiter sich auf der Baustelle frei bewegen kann, während er mit den Dokumenten arbeitet, kann er als in Bewegung (U_{mot}) angesehen werden. Der Tablet-PC kann durch seine Ethernet-Karte mit jedem LAN gekoppelt werden und ist somit ein mobil, aber ruhend ($D_{\text{mob}} \setminus D_{\text{mot}}$) funktionierendes Gerät. Nach Tabelle 1 erlaubt diese Konfiguration entweder eine Hybrid-Offline- oder eine Hybrid-Online-Dienstkonnektivität. Tatsächlich wurde für das System eine Hybrid-Offline-Architektur (S_{hoff}) gewählt, da der Bauleiter die meiste Zeit nicht an seinem Büroarbeitsplatz ist, sondern das System auf Baustellen benutzt. Entsprechend Abbildung 1 ist die gesamte Präsentationslogik sowie die Geschäftslogik auf dem Client implementiert. Dieser übernimmt auch die Speicherung einer Untermenge des Datenbestands, die von Zeit zu Zeit über die Netzwerkverbindung mit dem kompletten Datenbestand des Servers synchronisiert wird.

Als zweites Beispiel dient ein Informationssystem für ein Speditionsunternehmen, das Fahrer (die meistens in Bewegung sind und damit zur Kategorie U_{mot} gehören) über eine On-Board Unit (OBU), die mit dem GSM-Netzwerk verbunden ist (d.h. ein in Bewegung funktionierendes Gerät der Kategorie D_{mot}) ständig mit aktuellen Dispatch-Informationen, Straßenzuständen und Routen-Informationen versorgt. Diese Kombination würde die gesamte Bandbreite an Hybrid-Offline-, Hybrid-Online- und Online-Dienstkonnektivität erlauben. Da jedoch Teile des Systems (wie z.B. das Routing) auch in Gegenden ohne Netzabdeckung funktionieren müssen, wurde eine Architektur gewählt, die nicht ausschließlich von der jeweiligen Netzabdeckung abhängig ist: Während die gesamte Präsentationslogik und die vom Fahrer benötigte Geschäftslogik auf dem Client vorhanden ist (wie für einen Hybrid-Offline-Dienst), versucht das System immer mit aktuellen Daten, die vom Server empfangen wurden, zu arbeiten und nutzt nur dann zwischengespeicherte Daten, wenn die Netzwerkverbindung unterbrochen ist. Da dies eine Hybrid-Online-Konnektivität in der Datenhaltungsschicht repräsentiert, betrachten wir das gesamte System als ein Hybrid-Online-System (S_{hon}).

Im Gegensatz dazu erfüllt ein Lotteriewebportal, das Benutzern die Teilnahme an Lotto-Spielen mittels WAP-fähiger Mobiltelefone erlaubt, auch die Kriterien U_{mot} (Benutzer in Bewegung) und D_{mot} (Geräte in Bewegung). Um jedoch den Dienst so unabhängig wie möglich von den Endgeräten der Benutzer zu halten, wurde das System in einer komplett Serverzentrierten Architektur realisiert. Nur die Beschreibung der anzuzeigenden WML-Seiten wird an den Client übertragen, was einer Online-Dienstkonnektivität (S_{on}) entspricht.

Auf dasselbe Portal kann auch über einen weiteren Kanal via Short Message Service (SMS) zugegriffen werden. So kann ein Benutzer in Bewegung (U_{mot}) seinen Lottotipp in eine SMS auf seinem Mobiltelefon (ein D_{mot} -Gerät) eingeben. Die Nachricht kann unabhängig von der Netzwerk-Verfügbarkeit verfasst werden und wird versandt, wenn Netzabdeckung vorhanden ist. Der Server erzeugt dann eine Bestätigungsnachricht, die zum Mobiltelefon zurückgesendet wird, sofern dieses mit dem Netz verbunden ist. Das System bleibt daher während kurzer Netzausfälle immer noch benutzbar (d.h. Nachrichten können

verfasst und gelesen werden). Geht die Netzverbindung jedoch für eine längere Zeit verloren, wird das System immer unbenutzbarer, da der Benutzer keine Tipps mehr platzieren oder wichtige Bestätigungsnachrichten mehr empfangen kann. Da auf dem Client keinerlei Geschäftslogik vorhanden ist und die client-seitigen Daten (d.h. Nachrichten) nur eine Art von Pufferung repräsentieren, kann der Dienst als hybrid-online (S_{hon}) angesehen werden.

5 Zusammenfassung

In den vorhergehenden Abschnitten haben wir drei Kriterien für die Klassifikation von mobilen Informationssystemen vorgestellt, die es ermöglichen, die mögliche Dienstkonnektivität anhand von zwei Rahmenbedingungen abzuleiten: grobe Interaktionsmuster (im Sinne der Nutzer-Mobilität) und grundlegende Geräteeigenschaften (im Sinne der Geräte-Mobilität). Indem wir die Dienstkonnektivität als Richtlinie nehmen, können wir dann Entscheidungen darüber treffen, wie die Präsentations-, Geschäftslogik und Persistenzschicht über die Client-Server-Architektur zu verteilen sind.

In Rahmen der weiteren Forschung betrachten wir zwei Fragen ganz besonders: Erstens, welche anderen Klassen von Anwendungen sollten noch berücksichtigt werden? Können wir irgendeine existierende Anwendungsklassifikation benutzen und unsere neuen Kriterien darauf anwenden, um die Implikationen des mobilen Einsatzes darzustellen, oder gibt es andere allgemeine Charakteristika mobiler Anwendungen, die die Verteilungsstruktur von z.B. Telemetrie- oder Peer-to-Peer-Anwendungen bestimmen? Und zweitens, wenn eine bestimmte Kombination von Nutzer- und Geräte-Mobilität eine Auswahl zwischen verschiedenen Dienstkonnektivitätsgraden erlaubt, welche Kriterien bestimmen dann, welcher Konnektivitätsgrad in jeder Ebene gewählt werden sollte?

Bei der Suche nach solchen Klassifikationskriterien, müssen wir ein Gleichgewicht zwischen den Charakteristika mobiler Anwendungen finden, deren Wesen eher das von Anforderungen als von Konsequenzen oder Eigenschaften ist. (Beispielsweise ist hohe Bandbreite eine Eigenschaft bestimmter Kommunikationskanäle – eine verwandte Anforderung, die als Klassifikationskriterium besser geeignet ist, ist hohes Datenvolumen). Als Softwaretechniker tendieren wir auch dazu, uns auf technische Eigenschaften zu beschränken, wir sollten aber auch nach relevanten Kriterien aus der Anwendungsdomäne suchen – ein guter Kandidat für ein solches Kriterium ist sicherlich die Kenntnis über den aktuellen Aufenthaltsort.

Weiterhin sollte die Granularität erforscht werden, die möglich und zugleich sinnvoll für ein solches Klassifikationsschema ist. Die in diesem Artikel vorgestellten Kriterien gelten auf Dienstebene und nicht auf Anwendungsebene, d.h. dass verschiedene Dienste zu einer Anwendung zusammengefasst werden können; innerhalb einer Anwendung können die verschiedenen Schichten somit (mit gewissen Einschränkungen) unterschiedliche Konnektivitätsgrade für die unterschiedlichen Dienste aufweisen. Welche Auswirkungen die Zusammenführung von Diensten mit unterschiedlichen Konnektivitätsgraden in einer Anwendung auf die Architektur derselben haben, muss ebenfalls noch erforscht werden.

Bisher haben wir nur die Mobilität von Nutzern und Geräten in Bezug auf physische Netzwerke betrachtet. Letztendlich sollten die Klassifikationskriterien und Regeln für ihre architekturellen Auswirkungen auch virtuelle Aspekte der Mobilität umfassen – d.h. Mobilität in Bezug auf virtuelle Netzwerke und logische Mobilität von Code, der zwischen den Geräten ausgetauscht werden kann.

Auf der Grundlage der Kriterien, die in diesem Artikel vorgestellt wurden, sollten die oben genannten Fragen einen Ausgangspunkt für interessante Forschung in verschiedenen Richtungen darstellen, die uns hilft, die Charakteristika mobiler Anwendungen besser zu verstehen und somit solche Anwendungen effektiver zu entwickeln.

Literatur

- [CP02] Rajiv Chakravorty und Ian Pratt. WWW Performance over GPRS. In *Proceedings of the IEEE International Conference on Mobile and Wireless Communication Networks*, 2002.
- [Du01] Dudenverlag. *Duden Bd. 5 – Das Fremdwörterbuch*, 7. Auflage, 2001.
- [Is04] V. Issarny, F. Tartanoglu, Jinshan Liu und F. Sailhan. Software Architecture for Mobile Distributed Computing. In *Proceedings of the Fourth Working IEEE/IFIP Conference on Software Architecture (WICSA'04)*, Seiten 201–210. IEEE, 2004.
- [ISO94] International Organization for Standardization. *ISO/IEC Standard 7498-1: Information Processing Systems – OSI Reference Model – The Basic Model*, 1994.
- [Pa00] R. Pandya. *Mobile and personal communication systems and services*. IEEE Press, 2000.
- [Ri96] J. Rice, A. Farquhar, P. Piernot und T. Gruber. Using the web instead of a window system. In *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI '96)*, 1996.
- [RPM00] G.-C. Roman, G. P. Picco und A. L. Murphy. Software Engineering for Mobility: A Roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, Seiten 241–258. ACM Press, 2000.
- [VV02] Upkar Varshney und Ron Vetter. Mobile commerce: framework, applications and networking support. *Mobile Networks and Applications*, 7(3):185–198, 2002.
- [We91] M. Weiser. The computer for the twenty-first century. *Scientific American*, 265(3):94–104, 1991.