



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Piotr Wendt

Trailblazers - eine Plattform zur
Community-gestützten kollaborativen
Erweiterung von Navigationskarten

Piotr Wendt

Trailblazers - eine Plattform zur
Community-gestützten kollaborativen Erweiterung
von Navigationskarten

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Kai von Luck
Zweitgutachter : Dipl.Inform. Birgit Wendholt

Abgegeben am 16. Mai 2007

Piotr Wendt

Thema der Masterarbeit

Trailblazers - eine Plattform zur Community-gestützten kollaborativen Erweiterung von Navigationskarten

Stichworte

Kollaborative Kartenannotation, Kollaboratives Geographisches Informationssystem, Kollaborative Navigation, Soziale Software, Fußgänger Navigationssystem

Kurzzusammenfassung

Navigationssysteme für Nichtautofahrer sind aktuell Gegenstand der Forschung. Für die Realisierung solcher Systeme bedarf es eines speziellen Kartenmaterials, welches kostspielig in seiner Beschaffung und zudem statisch in seiner Beschaffenheit ist. Auf der anderen Seite setzen sich zunehmend Ansätze durch, die auf den Prinzipien der Sozialen Software aufsetzen. Der Kerngedanke ist, dass eine sich selbst organisierende Gemeinschaft in der Lage ist, mit Hilfe von geeigneten Werkzeugen das fehlende Kartenmaterial zu erstellen. Diese Arbeit stellt ein System vor, welches eine Community-gestützte Erstellung von Kartenmaterial unterstützt. Das hierbei erstellte Kartenmaterial zielt dabei speziell auf die Bedürfnisse von mobilitätseingeschränkten Menschen ab.

Piotr Wendt

Title of the master thesis

Trailblazers - a platform for community driven collaborative creation of navigation maps

Keywords

Collaborative Mapping Annotation, Collaborative Geographic Information Systems, Cooperative Navigation, Social Software, Pedestrian Navigation Systems

Abstract

Navigation Systems for pedestrians are still under development. These systems are based on geographical data which is expensive and also static. On the other side new systems are emerging which are based on communities and the principles of Social Software. These communities are able to generate the required geographical data on behalf of their own by using appropriate software tools. As a result, there are new maps which can be used for navigation purposes. In this thesis a system will be developed which facilitates the community driven creation of geographical data. In particular the focus is on the creation of navigation maps suitable for mobility impaired people.

Danksagung

Ich möchte mich zunächst bei meinen Eltern bedanken und ihnen gleichzeitig diese Arbeit widmen. Sie haben mich zu dem geprägt was ich heute bin. Hierbei hat auch meine kleine Schwester Ewelina ihren Anteil geleistet und mich stets unterstützt.

Im Verlauf meines Studiums und in der anschließenden schriftlichen Phase der Erstellung dieser Arbeit wurde ich durch konstruktive Kritik und Anregungen von Prof. Dr. Kai von Luck und Dipl. Inform. Birgit Wendholt unterstützt. Für ihren vielseitigen fachlichen Rat möchte ich ihnen meinen Dank aussprechen.

Diese Arbeit resultiert aus einem Projekt in dessen Rahmen ein Team zur Realisierung des Systems entstand. Den Trailblazers-Teammitgliedern Sven Stegelmeier, Martin Stein und Mark Thome möchte ich für die konstruktive und kameradschaftliche Zusammenarbeit danken.

In den etwas schweren Phasen dieser Arbeit standen mir die Kommilitonen Thies Rubarth, Nils Bertling und Lars Westphal immer unterstützend zur Seite. Diese stärkten mein Durchhaltevermögen und waren gewissermaßen meine kollaborative Plattform zur Navigation durch die deutsche Rechtschreibung.

Inhaltsverzeichnis

Tabellenverzeichnis	x
Abbildungsverzeichnis	xi
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	3
1.3 Inhaltlicher Aufbau der Arbeit	4
2 Grundlagen	6
2.1 Entstehung Community-gestützter Systeme	6
2.1.1 Soziale Software	7
2.1.2 Web 2.0	16
2.2 Technologische Betrachtung	22
2.2.1 Technologien Community-gestützter Systeme	23
2.2.2 Ausprägungen Community-gestützter Systeme	39
2.2.3 Merkmale und Leitsätze Community-gestützter Systeme	47
2.3 Ambient Assisted Living	63
2.4 Was ist Trailblazers	66
2.4.1 Szenario	66
2.4.2 Community	68
2.4.3 Daten	69
2.4.4 Aktualität	70
2.4.5 Benutzergerechte Navigation	71
2.4.6 Neuartige Daten	71
2.4.7 Selbstbestimmend Leben	72
2.4.8 Trailblazers - Collaborative Mapping	73
2.5 Zusammenfassung	74
3 Analyse	75
3.1 Konkretisierung des Anwendungsszenarios	75
3.2 Funktionale Analyse (Fachanalyse)	77
3.2.1 Die Systemidee	77

3.2.2	Anforderungen eines geographischen Informationssystems	79
3.2.3	Identifizierung der Anwendungsfälle	81
3.2.4	Resultierende Sicht auf die abstrakten Anwendungsfälle	99
3.2.5	Nichtfunktionale Anforderungen	101
3.3	Nichtfunktionale Analyse	107
3.3.1	Trailblazers - Herausforderungen und Risiken	107
3.3.2	Zu kleiner Markt	109
3.3.3	Barrieren	109
3.3.4	Aggregation von Weginformationen	110
3.3.5	Handhabbarkeit	110
3.3.6	Privacy	111
3.3.7	Tracking	111
3.3.8	Kontraproduktive Verwendung	112
3.3.9	Kartographieren und Lokation	112
3.3.10	Point Of Interest	113
3.3.11	Gemeinsame Erarbeitung von Material	114
3.3.12	Verteilung der Sichtweise	115
3.3.13	Echtzeit	115
3.4	Anforderungen an die Benutzerschnittstelle	116
3.4.1	Angemessenheit der Aufgaben	116
3.4.2	Selbstbeschreibungsfähigkeit	116
3.4.3	Steuerbarkeit	117
3.4.4	Erwartungskonformität	117
3.4.5	Fehlertoleranz	117
3.4.6	Individualisierbarkeit	117
3.4.7	Lernförderlichkeit	118
3.4.8	W3C Web Content Accessibility Guidelines	118
3.4.9	Rollstuhlfahrgerechte Informationsdarstellung	120
3.5	Wahrnehmung der Umgebung durch Rollstuhlfahrer	121
3.5.1	Zugänglichkeit	122
3.5.2	Parkplätze	122
3.5.3	Beschaffenheit der Eingänge	122
3.5.4	Aufzüge	123
3.5.5	Sanitäre Orte	123
3.5.6	Zusammenfassung	123
3.6	Zusammenfassung	124
4	Entwurf	125
4.1	Schichtenmodell	126
4.1.1	Dialog (Präsentationsschicht)	128

4.1.2	Dialogsteuerung (Anwendungsschicht)	128
4.1.3	Web Service Zugriffskomponente (Fachkonzept-Zugriffsschicht)	129
4.1.4	Web-Service-Fassade (Fachkonzept-Zugriffsschicht)	129
4.1.5	Dialog-Agent (Fachkonzeptschicht)	129
4.1.6	Fachliche Fassade (Fachkonzeptschicht)	130
4.1.7	Fachliche Komponente (Fachkonzeptschicht)	130
4.1.8	Adapter-Komponente (Datenhaltungs-Zugriffsschicht)	131
4.1.9	Datenzugriffs-Komponente (Datenhaltungs-Zugriffsschicht)	131
4.1.10	Datenhaltungsschicht	131
4.1.11	Externe Komponenten	131
4.2	Verteilungsmodell	132
4.2.1	Hardwarekomponenten	133
4.2.2	Softwarekomponenten	133
4.3	Systemtechnische Anwendungsfälle	134
4.3.1	Erstellung von Navigationskarten (Erstellung)	136
4.3.2	Nachbearbeitung der Navigationskarten (Nachbearbeitung)	139
4.3.3	Integrierung bestehender Karten	143
4.3.4	Annotierung der Navigationskarten (Annotierung)	146
4.3.5	Sharing der Navigationskarten (Sharing)	153
4.4	Anwendungsfallmodell	158
4.5	Identifizierung der Fachklassen	160
4.5.1	Navigationskarte	162
4.5.2	Spezialisierte Karte	163
4.5.3	Route	164
4.5.4	Trampelpfad	164
4.5.5	Point of Interest	165
4.5.6	Spezialisierte Information	166
4.5.7	Dynamische Information	166
4.5.8	Benutzergruppe	167
4.5.9	Benutzer	167
4.5.10	Rollstuhlfahrer	168
4.5.11	Administrator	168
4.5.12	Benutzerrecht	169
4.5.13	Ressource	169
4.6	Fachliches Subsystemmodell	170
4.6.1	Fachliche Komponenten	170
4.6.2	Externe Systeme	173
4.6.3	Spezifizierung der Anwendungsfallsteuerung	174
4.6.4	Spezifizierung der Beziehungen zwischen den Komponenten	175
4.6.5	Identifizierung der Subsysteme	176

4.7	Konkretisierung der Interaktionen	177
4.8	Zugrunde liegendes Datenbankmodell	180
4.8.1	Modularisierung der Datenbank	180
4.8.2	Darstellung der geographischen Informationen	181
4.8.3	Eigenschaften eines GIS	181
4.8.4	Trampelpfad	183
4.8.5	Spezialisierte Karte	184
4.8.6	Benutzer	184
4.8.7	Point of Interest	185
4.8.8	Dynamische Information	185
4.8.9	Rechte	186
4.8.10	Spezialisierte Information	186
4.8.11	Ressource	186
4.8.12	Navigationskarte	187
4.8.13	Diskussion	187
4.9	Zusammenfassung	188
5	Realisierung	190
5.1	Technischen Rahmenbedingungen	190
5.1.1	Entwicklungsumgebung	190
5.1.2	Server	191
5.1.3	Client	192
5.1.4	Testumgebung	193
5.1.5	Testläufe	193
5.1.6	Durchführung der Testläufe	197
5.2	Gewonnen Erkenntnisse	199
5.2.1	Server	199
5.2.2	Client	199
5.3	Offenen Punkte	200
5.3.1	Benutzer-Schnittstelle	200
5.3.2	Regelung der kollektiven Arbeit	201
5.3.3	Sicherheits-Modell	201
5.3.4	Suche	202
5.3.5	Mobilität	202
5.3.6	Datensammlung	202
5.3.7	Notifikation	203
5.3.8	Performance	203
5.3.9	Virtual Earth	204
5.4	Fazit	204
6	Fazit	207

6.1 Erreichte Ziele	208
6.2 Ausblick	209
6.3 Zusammenfassung	210
Literaturverzeichnis	214

Tabellenverzeichnis

3.1	Abstrakter Anwendungsfall - Navigation	83
3.2	Abstrakter Anwendungsfall - Kollaborative Kartenerstellung	84
3.3	Abstrakter Anwendungsfall - Navigieren	85
3.4	Abstrakter Anwendungsfall - Änderung der Stecke	86
3.5	Abstrakter Anwendungsfall - Synchronisation der Navigationskarten	88
3.6	Abstrakter Anwendungsfall - Betrachtung von POI Informationen	90
3.7	Abstrakter Anwendungsfall - Echtzeit Informationen abrufen	91
3.8	Abstrakter Anwendungsfall - Erstellen von Navigationskarten	92
3.9	Abstrakter Anwendungsfall - Nachbearbeitung der Navigationskarten	94
3.10	Abstrakter Anwendungsfall - Integration bestehender Karten	96
3.11	Abstrakter Anwendungsfall - Annotierung der Navigation	97
3.12	Abstrakter Anwendungsfall - Sharing der Navigationskarten	99
4.1	Systemtechnischer Anwendungsfall - Neue Navigationskarte erstellen	137
4.2	Systemtechnischer Anwendungsfall - Neue Route erstellen	138
4.3	Systemtechnischer Anwendungsfall - Ändern der Eigenschaften der Navigationskarte	140
4.4	Systemtechnischer Anwendungsfall - Ändern der berechneten Route	141
4.5	Systemtechnischer Anwendungsfall - Ändern des POI	142
4.6	Systemtechnischer Anwendungsfall - Ändern der integrierten spezialisierten Karte	143
4.7	Systemtechnischer Anwendungsfall - Ändern des Trampelpfades	144
4.8	Systemtechnischer Anwendungsfall - Spezialisierte Karte auf das System laden	145
4.9	Systemtechnischer Anwendungsfall - Verschmelzen der spezialisierten Karte mit der Navigationskarte	146
4.10	Systemtechnischer Anwendungsfall - Neuen POI erstellen	149
4.11	Systemtechnischer Anwendungsfall - Neue Zusatzinformationen erstellen	150
4.12	Systemtechnischer Anwendungsfall - Neue dynamische Informationen erstellen	152
4.13	Systemtechnischer Anwendungsfall - Neue Struktur der Annotierung erstellen	154
4.14	Systemtechnischer Anwendungsfall - Eine Benutzergruppe bilden	157
4.15	Systemtechnischer Anwendungsfall - Recht an eine Ressource vergeben	158

Abbildungsverzeichnis

1.1	Übersicht über Trailblazers-System	2
2.1	Ausprägungen der Sozialer Software unterteilt nach Anwendungsdomänen (Vgl. Kalz, 2006)	15
2.2	Die Evolution der Computer Architektur (Vgl. Hashimi und Steffan, 2005)	20
2.3	Verschieden Verteilungsausprägungen des Client/Server Modell nach Tanenbaum (Hashimi und Steffan, 2005)	33
2.4	Drei Schichten Architektur mit Ajax nach (Woolstone, 2006)	34
2.5	Die Korrelierenden Abhängigkeiten zwischen Benutzerzahl und Datenqualität (vgl. Symeonidis und Mitkas, 2005)	49
2.6	Systemunterteilung nach verschiedenen Domänen des Living Assistance (Nehmer u. a., 2006)	64
2.7	Aufgabenbereiche für das Ambient Assisted Living (Huch, 2006)	64
2.8	Rollstuhlfahrer als Hauptnutzer des Trailblazers-Systems	66
2.9	Treppe als Ausprägung einer unüberwindbaren Barriere für einen Rollstuhlfahrer	67
2.10	Trailblazers-System auf einem Smartphone die der Rollstuhlfahrer konsultieren kann um Barrieren-freie Wege zu bekommen.	68
2.11	Speziell für die Rollstuhlfahrer konzipierte Rampe die von dem Trailblazers-System erkannt und dem Rollstuhlfahrer vorgeschlagen wurde.	68
2.12	Trailblazers gestaltender Prozess der Karten-Erstellung und deren Nutzung	69
2.13	Collaborative Mapping und seine Eigenschaften nach nach (Balram und Dragicevic, 2006)	73
3.1	Skizzierter Verlauf des Anwendungsszenarios	75
3.2	Identifizierte Anwendungsfälle im Trailblazers-System	82
3.3	Beziehungen zwischen den einzelnen abstrakten Anwendungsfällen	101
3.4	Gestaltungsstruktur des Prototypen angelehnt an (Ringbauer und Schlegel, 2006).	121
4.1	Zugriffswege auf beziehungsweise von der Trailblazers-Plattform	127
4.2	Schichtenmodell des webbasierten, Server-seitigen Trailblazers Werkzeug zur Bearbeitung der Navigationskarten.	128
4.3	Einsatz- und Verteilungsdiagramm für Trailblazers	132

4.4	Für die kollaborative Erstellung relevanten abstrakten Anwendungsfälle	135
4.5	Mögliche systemtechnische Anwendungsfälle für die Erstellung von Navigationskarten	136
4.6	Mögliche systemtechnische Anwendungsfälle für die Nachbearbeitung	139
4.7	Mögliche systemtechnische Anwendungsfälle für die Integrierung von bestehenden Karten	144
4.8	Mögliche systemtechnische Anwendungsfälle für die Annotierung	148
4.9	Mögliche systemtechnische Anwendungsfälle für das Sharing	156
4.10	Anwendungsfallmodell mit primären und resultierenden sekundären Anwendungsfällen	160
4.11	Abhängigkeitsbeziehung zwischen den übergeordneten Anwendungsfällen .	161
4.12	Zentrale fachliche Klassen und ihre Beziehungen	162
4.13	Analyse-Klassmodell welcher die Aufteilung der Klassen in Komponenten darstellt	172
4.14	Veranschaulichung der Beziehungen anhand eines Komponentenmodells für fachliche Komponenten	176
4.15	Modell des Subsystems „Annotierung“	177
4.16	Veranschaulichung der Erstellung eines Point of Interest in einem Kommunikationsdiagramm	178
4.17	ER-Diagramm zur Trampelpfad und davon abhängigen Entitäten	183
4.18	ER-Diagramm zur Spezialisierte Karte und ihre Abhängigkeiten	184
4.19	ER-Diagramm zur Benutzer und seine Abhängigkeiten	184
4.20	ER-Diagramm zur Point of Interest und seine Abhängigkeiten	185
4.21	ER-Diagramm zur Dynamischen Information und ihre Abhängigkeit	186
4.22	ER-Diagramm zur Rechten und ihre Abhängigkeiten	186
4.23	ER-Diagramm zur Spezialisierte Information und ihre Abhängigkeiten	186
4.24	ER-Diagramm zur Ressource und ihre Abhängigkeiten	187
4.25	ER-Diagramm zur Navigationskarte und ihre Abhängigkeiten	188
4.26	ER-Diagramm zur Positions- und ressourcenbezogene Diskussion angelehnt an (Rinner, 1999)	188
5.1	Skizze der technischen Rahmenbedingungen	191
5.2	Aktivitätsdiagramm - Entfernen eines Trampelpfades	194
5.3	Aktivitätsdiagramm - Spezifizierung eines geschützten Bereiches	195
5.4	Aktivitätsdiagramm - Berechnung der Wegbeschreibung	196
5.5	Darstellung der Point of Interest Informationen	197

1 Einleitung

Überall im Internet bilden sich Gruppen von Menschen, sog. Communities, mit ähnlichen Interessen. Der heute allgegenwärtige Zugang zum Web ist der Web-Browser. Tim Berners-Lee stellte im März 1991 den ersten Vertreter dieser Softwaregattung vor. Der ewig wähernden Kritik, dass Webbrowser nicht genügend Vielfalt bei der Gestaltung komplexer Benutzerschnittstellen ermöglichen, wird durch die jüngsten Entwicklungen entgegengewirkt. Der Zeitpunkt, bei dem webbasierte Anwendungen dieselbe Bedienungsvielfalt wie Desktopanwendungen erlauben, steht kurz bevor. Die neuen Möglichkeiten von Webanwendungen begünstigen die Bildung weiterer Communities im Internet.

1.1 Motivation

In der Europäischen Union leben schätzungsweise 37 Millionen an einen Rollstuhl gebundene Menschen. Die USA verzeichnen eine vergleichbare Statistik. An einen Rollstuhl gebunden zu sein, bedeutet eine drastische Einschränkung der Mobilität und damit der persönlichen Freiheit. Artefakte wie z.B. Bürgersteige oder Treppen werden von Nichtrollstuhlfahrern oft gar nicht mehr wahrgenommen. Für Rollstuhlfahrer stellen sie allerdings meistens unüberwindbare Hindernisse dar. Das gleiche gilt auch für Veränderungen in unserer Umgebung. Bei der Absicherung einer Baustelle wird z.B. nicht immer an mobilitätseingeschränkte Menschen Rücksicht genommen. Diese Barrieren führen dazu, dass Betroffene in ihrer Bewegung und somit auch in ihrer Selbstbestimmung eingeschränkt werden. Denn *„Behindert ist man nicht, behindert wird man“* (Stein, 2006). Diese daraus resultierenden Einschränkungen sind aufgrund des enormen technischen Fortschritts des 21en Jahrhunderts inakzeptabel.

Das Internet hat sich in den letzten Jahren mehr und mehr in die Richtung entwickelt, für die es ursprünglich konzipiert war. Es sollte nicht nur zum Datenabruf und zur Durchführung von Geschäftstransaktionen verwendet werden, sondern als ein Werkzeug fungieren, mit dem sich Menschen gegenseitig helfen können. Letzteres unterstützt die Bildung von sog. Community-getriebenen Anwendungen. Die Anwendung selber fungiert hier nur als das Werkzeug, welches von Mitgliedern der Community auf verschiedenste Weisen verwendet

werden kann. Blogs und Wikis sind zwei der neueren Vertreter dieser Werkzeuge. Beide dienen zur Unterstützung der Kommunikation innerhalb größerer Gemeinschaften.

Der Zusammenschluss vieler Menschen bewirkt große Synergieeffekte, welche dem Wohl aller beteiligten dienen. Der Aufbau einer Community, dessen Synergieeffekte zur Erhöhung der Mobilität von Rollstuhlfahrern führen soll, ist daher naheliegend. **Trailblazers** ist ein System mit diesem Ziel. Es ist ein für Rollstuhlfahrer geeignetes Community-getriebenes mobiles Navigationssystem. Der Grundgedanke ist die Bereitstellung eines Navigationsdienstes, welcher Wissen über barrierefreie Routen verfügt. Somit können mobilitätseingeschränkte Menschen ihr Ziel erreichen, ohne dabei auf Hindernisse wie Treppen oder Kantsteine zu stoßen.

Abbildung 1.1 zeigt die Architektur des Trailblazers-System. Mobile Navigationssysteme fallen in die Kategorien On- und Offboard. Während sich erstere durch eine Speicherung sämtlicher Kartendaten auf dem mobilen Gerät auszeichnet, übernimmt bei letzterem ein zentraler Server diese Aufgabe. Trailblazers zählt zur Offboard-Kategorie. Dies ist aufgrund des Community-Ansatzes notwendig.

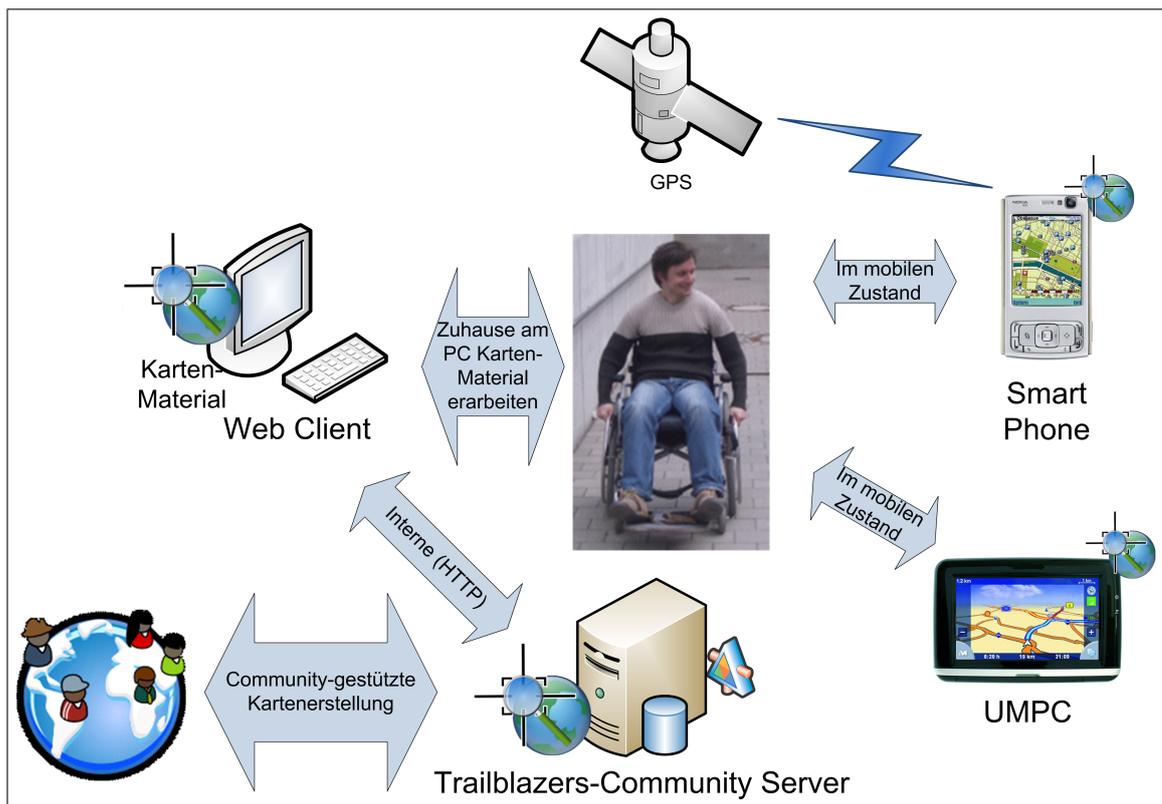


Abbildung 1.1: Übersicht über Trailblazers-System

Die Haupthürde, die es zu nehmen gilt, um barrierefreie Navigation bereitzustellen, liegt in der Beschaffung der dafür notwendigen Daten. Die von heutigen Herstellern wie Navteq (Navteq, 2007) und TeleAtlas (Atlas, 2007) angebotenen Navigationsdaten umfassen nur für Fahrzeuge geeignete Routen. Diese sind weder für Fußgänger, und schon gar nicht für Rollstuhlfahrer geeignet. Um die entsprechenden Daten zu beschaffen besitzt Trailblazers den oben schon erwähnten Community-Aspekt. Ein mit einer am Systemstart leeren Datenbank ausgestatteter Community-Server soll von dieser nach und nach mit den notwendigen Daten gefüllt werden. Hier kommt die oben schon erwähnte Metapher zum Tragen: Trailblazers ist das Werkzeug, mit dem sich die Mitglieder der Community gegenseitig helfen. Mit letzterem ist das Sammeln von barrierefreien Navigationsdaten gemeint. Letzteres soll hier nur skizziert und wird in einem späteren Kapitel genauer erläutert werden.

Der Grundgedanke der datensammelnden Community ist ein Trackingverfahren, welches barrierefreie Wege auf Basis der Position und des Profil eines Nutzers erkennt. Dieses Verfahren wird nicht so perfektioniert werden können, dass die gelieferten Navigationsdaten ausreichend sind. Aus diesem Grund bietet Trailblazers der Community ein Werkzeug in Form eines **Editors** für die gesammelten Daten. Mit dessen Hilfe wird es allen Mitgliedern ermöglicht fehlerhafte Daten über das Internet zu korrigieren. Er ist Bestandteil des in Abbildung 1.1 skizzierten Webclients und kann über einen handelsüblichen Internetbrowser ausgeführt werden.

Ein Editor zur Bearbeitung geographischer Daten stellt eine Herausforderung für die Gestaltung von Weboberflächen dar. Neben den technischen Aufgaben erfordern die Nutzer in der Community, aufgrund ihrer unterschiedlichen Fähigkeiten, auch die Durchführung ergonomischer Betrachtungen. Alles in allem ist der Editor das letzte Bindeglied für den Prozess der Sammlung von Navigationsdaten durch die Community und folglich zwingend für den Erfolg von Trailblazers notwendig.

1.2 Zielsetzung

Ziel dieser Arbeit ist es, einen webbasierten Editor für geographische Navigationsdaten zu entwickeln. Es soll das oben schon erwähnte fehlende Bindeglied für die Trailblazers-Plattform darstellen. Er ist ein wichtiges Werkzeug für den Erfolg der Community und damit des Systems. Die Komplexität des Einsatzszenarios und der technologischen Herausforderungen rechtfertigen eine breiter angelegte Betrachtung der folgenden Sachverhalte.

Selbstbestimmung in allen Lebenslagen schließt die Fortbewegung mit ein. Umso mehr trifft dies auf Menschen zu, die in ihrer Mobilität eingeschränkt sind. Diese zentrale Problemstellung soll anhand verschiedener Gesichtspunkte betrachtet werden. So sollte die Integration

bestehender Technologien mit sozialen Aspekten untersucht und deren Auswirkungen bewertet werden. Des Weiteren sollten Effekte identifiziert werden, welche durch die Einbeziehung von öffentlichen Diensten durch die Trailblazers-Plattform entstehen. Letzteres soll Helfen die Umwelt besser wahrzunehmen.

Das Trailblazers-System fällt in eine Kategorie, welche in der Literatur mit *Distributed Collaborative Work* bezeichnet wird. Letzteres bezeichnet Systeme, welche das gemeinsame aber verteilte Arbeiten auf dem ganzen Erdball ermöglichen. In den letzten Jahren wurden diese Sachverhalte verstärkt untersucht und mehrere Systeme erstellt. Die hierbei gesammelte Erfahrung soll extrahiert und zur Verbesserung der Aufgabe genutzt werden.

Neue Möglichkeiten eröffnen neue Wege des Einsatzes von Computern. Ein junger Trend, welcher in (O'Reilly, 2005) einleitend vorgestellt wird ist „Web 2.0“. Er wurde auf der Web 2.0 Konferenz (O'Reilly, 2006) definiert und anhand von Beispielen verdeutlicht. Dieser neue Ansatz zeigt neben den neuen Einsatzmöglichkeiten von Technologie im Internet auch andere Einsatzgebiete von Computern. Ein Begriff welcher sich in diesem Zusammenhang ebenfalls etabliert hat ist „*Social Software*“, zu Deutsch: Soziale Software. Letzterer bilden eine Obermenge der Prinzipien des Web 2.0 und fügt noch einige Begriffe hinzu.

Ein weiterer Trend im europäischen Raum ist die verstärkte Forschung im Bereich der „*Ambient Intelligence*“ und des „*Ambient Assisted Living*“. Diese Forschungszweige zeigen viele Ähnlichkeiten zu den Prinzipien der oben genannten Ansätze. Letztere sollen im Rahmen dieser Arbeit auf Korrelation untersucht werden. Dabei sollen die Unterschiede ersichtlich werden und entsprechend in die Weiterentwicklung der hier zugrunde liegenden Trailblazers Software mit einfließen.

1.3 Inhaltlicher Aufbau der Arbeit

Nach dieser Einleitung beginnt die Arbeit mit der Einführung in die notwendigen Grundlagen. Zu Beginn wird die geschichtliche Entstehung von Community-basierten Systemen erläutert. Des Weiteren wird auf die Entstehung der obigen Begriffe des **Web 2.0** und der **Social Software** eingegangen. Anschließend wird auf die technologische Wandlung und die damit verbundenen Konsequenzen eingegangen. Zusätzlich werden Einsatzbereiche der vorgestellten Technologien beschrieben und diskutiert.

Den Kernpunkt soll eine Analyse der Leitsätze und Merkmale von Community-basierten Anwendungen bilden. Letztere sollen aus den vorangegangenen Abschnitten extrahiert und hier diskutiert werden. Das Ergebnis wird hier eine Liste von Prinzipien sein, welche als Basis für die Erstellung der Editor-Software dienen werden. Anschließend wird für die erstellten Prinzipien ein Szenario für das Trailblazers-System vorgestellt. Den Abschluss der Grundlagen wird eine detaillierte Beschreibung des Trailblazers-Systems bilden.

In der darauffolgenden Analyse werden, durch den Einsatz von objektorientierten Techniken, weitere dem System zugrunde liegende Aspekte erörtert. Mit deren Hilfe sollen die fachlichen Aspekte näher beleuchtet und die daraus resultierenden Vorgaben für die Realisierung des Systems extrahiert werden. Zu Beginn werden abstrakte Anwendungsfälle auf Basis einer gründlichen Untersuchung des Systems extrahiert. Zusätzlich wird die Wahrnehmung des Systems durch den Benutzer in Hinsicht auf Aspekte, welche nicht direkt mit der durch das System bereitgestellten Funktionalität zusammenhängen, untersucht. Die Analyse schließt mit einer Betrachtung von ergonomischen Richtlinien, welche für die Realisierung der Editor-Software zu Einsatz kommen, ab.

Nachdem die wichtigsten Vorgaben für den Entwurf des hier betrachteten Systems in der Analyse erarbeitet wurden, soll deren Entwurf und Realisierung diskutiert werden. Dabei soll in erster Linie die Architektur und die aus den Vorgaben entstehenden Architekturentscheidungen beschrieben und anschließend diskutiert werden. Als Ergebnis werden Architekturvorgaben erstellt, die bei der Realisierung zum Tragen kommen werden. Um hierbei eine saubere funktionale Trennung zu schaffen, welche als Basis für den darauffolgenden Komponentenentwurf dient, werden die in der Analyse identifizierten Anwendungsfälle noch einmal unter technischen Gesichtspunkten betrachtet.

Um die gesammelten Daten der Community kohärent und redundanzfrei zu halten, wird zu diesem Zweck ein Datenbankschema entwickelt. Als Grundlage hierfür werden die aus der Datenbanktheorie und aus der geographischen Kartographie bekannten Ansätze verwendet.

In dem Abschnitt Realisierung soll dann anhand zuvor festgelegter Testfälle die Kernfunktionalität der Editor-Software verdeutlicht werden. Anhand hierbei durchgeführter Diskussion soll die prototypische Umsetzung anschließend bewertet werden. Zuletzt sollen die bei der Realisierung identifizierten Lücken durch konstruktive Vorschläge geschlossen werden.

Diese Arbeit schließt mit einer Diskussion der gewonnenen Ergebnisse ab.

2 Grundlagen

2.1 Entstehung Community-gestützter Systeme

Seit geraumer Zeit werden vermehrt neue webbasierte Anwendungen eingesetzt, die unter dem Begriff *Soziale Software* zu klassifizieren sind. Dieser Begriff soll eine neue Art der Informationsverarbeitung umschließen. Betrachtet man die sich dahinter befindliche Erneuerungen genauer, werden von dieser zwei Arten sichtbar. Die eine Gruppe betrifft die Änderungen in der rechner-gestützten Verarbeitung und ist damit technologisch bedingt. Die zweite und möglicherweise noch wichtigere Art von Änderungen sind bei den Nutzern selbst zu beobachten.

Soziale Software stellt also in der Weise eine grundlegende Änderung der Sichtweise dar, dass sie Anwender durch das Wissen anderer Anwender unterstützt. Hierzu muss natürlich eine solche Wissensbasis zunächst aufgebaut werden, was allerdings wiederum durch die Anwendung selbst erledigt werden soll. Das interessante und gleichzeitig geniale hierbei ist, dass die Daten nicht etwa vom Bereitsteller der Anwendung sondern von anderen Nutzern selbst kommen. Hierdurch wird der bisher gültige Top-Down Ansatz der Daten-Bereitstellung verlassen.

Soziale Software muss darauf konzipiert sein Daten gemeinsam zu sammeln und verarbeiten zu können. Letzteres fördert die Partizipation der Community. Dabei werden verschiedene Vorgehensweisen eingesetzt, welche das Wissen und die damit verbundenen Daten zu strukturieren hilft. Freie Annotierung, Bewertungs-Methoden und Semantisches Web sind die gängigen Schlagwörter die bei der Auszeichnung von Daten auftauchen.

Diese neue Art von Anwendungen hat aber auch eine Änderung der technischen Sichtweise bewirkt. Die Fortentwicklung der Anwendung ist in die Hände der Nutzer delegiert worden. Soziale Software zeichnet sich durch offene, vermehrt leichtgewichtige Schnittstellen aus. Diese Schnittstellen, welche deutlich einfacher in der Verwendung sind, erlauben den Benutzern diese Soziale Anwendung anderweitig zu verwenden. Dabei werden vermehrt Anwendungen dieser Art zu neuen Anwendungen integriert, obwohl diese ursprünglich für andere Zwecke entworfen worden. Damit unterliegen die darin enthaltenen Daten ebenfalls einer anderweitigen Verwendung. Eine andere technische Änderung ist die fortlaufende Entwicklung Community-gestützter Systeme. Diese werden hierbei fortlaufend weiterentwickelt. Der

Benutzer selber wird dann durch starten der Anwendung sofort mit den neuesten Änderungen konfrontiert. Dieser Umstand führt zu einigen Vor- und Nachteilen wie im Verlauf des Abschnittes gezeigt werden wird.

Im Umfeld der Sozialen Software wird von einer *Intelligenten Community* gesprochen. Es werden verschiedene Menschen mit verschiedenen Sichtweisen und Erfahrungen, die verteilt auf der ganzen Welt leben, zusammen zu einer Community geschlossen um an verschiedenen Problemstellungen zusammen zu arbeiten. Dabei basieren diese Anwendungen auf Erkenntnissen die die Soziologie untersucht und zum Teil begründet darlegt. Es zeigt sich nämlich, dass durch Gemeinschaften erarbeitetes Wissen in vielerlei Hinsicht Vorteile aufweist, die durch eine kleine Gruppe von Spezialisten nicht erbracht werden können. Um diese Vorteile jedoch wahrnehmen zu können bedarf es bestimmter Voraussetzungen die erfüllt werden müssen, damit eine gesunde Community gedeihen kann. Welche dies sind wird im Verlauf dieses Abschnittes ergründet.

Technische Änderungen spielen ebenfalls eine wichtige Rolle im Umfeld Sozialer Software. Es hat sich herausgestellt, dass diese neuen Anwendungen durch verschiedene technische Eigenschaften gekennzeichnet sind. Am wichtigsten ist die Einfachheit in der Benutzung aber auch in der anderweitigen Verwendung. Diese Einfachheit in der Benutzung wird zum einen durch das proaktive Verhalten der Anwendung gegeben. Zum anderen aber auch durch deren Anpassung an die Arbeitsweise des Nutzers. Ebenfalls sehr wichtig ist die Möglichkeit des hardwareübergreifenden Einsatzes dieser Anwendung. Diese Einfachheit in der Nutzung, die Offenheit in der Gestaltung der Anwendung, wird noch durch ein weiteres Merkmal der „freien Nutzung“ verstärkt. Diese und weitere Aspekte einer Community-gestützten Software sollen nun im Folgenden näher beleuchtet werden.

2.1.1 Soziale Software

Der Begriff *Soziale Software* findet immer öfter Verwendung in Verbindung mit neuen web-basierten Anwendungen. Dieser wird von unterschiedlichen Menschen verschiedenartig verwendet um ein, wie es scheint neues Paradigma bzw. einen neuen Wandel zu beschreiben, welcher sich aus informationstechnischer sowie soziologischer Sicht vollzieht. Betrachtet man genauer den Ursprung des Begriffs wird dessen Entstehung auf die Anfänge des Computerzeitalters zurückgeführt. Zudem wird bei dieser Betrachtung deutlich, dass mindestens zwei Betrachtungsweisen auf Prinzipien, welche den Begriff der Sozialen Software umfassen, existierten. Die eine Betrachtungsweise ist es, den Rechner als Kommunikationsmedium zu sehen. Mit anderen Worten, soll der Rechner dazu dienen Menschen zusammenzubringen und dabei das Wissen dieser Menschen gegenseitig zugreifbar zu machen. Eine andere Sichtweise sieht das Computing eher aus der Perspektive der kommerziellen Verwendung. Dabei verfolgte Ziele orientieren sich hauptsächlich an der Generierung von

Kapital. Für was steht nun eigentlich die Soziale Software? Wie und wann wurde dieser Begriff geprägt? Diese und weitere Fragen möchte der Autor im Folgenden klären.

Entstehung

Erste Vision: Memex Die erste Verwendung des Begriffes Soziale Software wird auf eine Veröffentlichung von *Vannevar Bush*¹ zurückgeführt. In seiner visionären Veröffentlichung hatte dieser bereits 1945 die Aufgaben eines Rechners, welcher mit dem heutigen PC verglichen werden kann, skizziert. Darüber hinaus verfeinerte er seine Vision und prophezeite, dass Rechner über weite Strecken mit Benutzern interagieren werden.

Vannevar Bush sah damals die eigentlichen Aufgaben eines Rechners als menschenunterstützend an. Solch ein Rechner sollte nach seiner Ansicht dazu dienen, die Kommunikation von Menschen untereinander zu erleichtern bzw. zu ermöglichen. Damit sollte der Rechner eine Grundlage für die interpersonelle Interaktion darstellen, auf dessen Basis gemeinschaftliches Arbeiten ermöglicht werden kann. Solch ein menschenunterstützendes System sollte dann dazu verwendet werden den Austausch von Wissen mit anderen Menschen zu fördern.

Bei der näheren Untersuchung des von Vannevar Bush skizzierten Verarbeitungsmodells werden Merkmale sichtbar, welche dem Konzept des Hypertexts sehr ähnlich sind. In dem von ihm beschriebenen System sind Konzepte wiederzuerkennen, welche in den heute wiederum so erfolgreichen Systemen enthalten sind. So wird davon gesprochen, dass ein Benutzer dazu imstande sein soll Inhalte selber zu erstellen, diese nach Belieben zu verändern und ebenfalls zu kommentieren. Darüber hinaus sollte der Benutzer die von ihm erstellten Informationsartefakte miteinander durch Referenzen in Beziehung setzen können ([Möller, 2006](#)).

Spätere wissenschaftliche Auseinandersetzungen in diesem Umfeld basierten auf dieser visionären Arbeit. Im Mittelpunkt der Betrachtung lag dort die Identifizierung der neuen Möglichkeiten, die durch den Hypertext erst möglich wurden. Des Weiteren setzte man sich damit auseinander, wie eine mit Hypertext zur Verfügung gestellte Information vom Leser interpretiert wird.

¹In der Publikation „*As We May Think*“ hatte 1945 Vannevar Bush eine Skizze des heute bekannten PC vorgestellt und diesen damals den Namen memex gegeben. In dieser Visionären Veröffentlichung hatte Vannevar Bush damals bereits die Zusammenarbeit, die durch PC ermöglicht werden sollen prophezeit ([Bush, 1945](#)).

Xanadu Modell *Ted Nelson* hatte 1960 das Xanadu Modell entwickelt welches die Idee des Hypertexts neu aufleben ließ. In diesem Konzept sah Nelson vor, dass jede Informationsressource nur einmal in System existieren solle und eindeutig durch eine Adresse auffindbar zu sein hat. Alle in Xanadu gespeicherten Dokumente sollten nicht gelöscht und nicht kopiert werden. Vielmehr sollten diese nur durch bidirektionale Verweise referenziert werden können. Wurde eine neue Version eines Dokumentes veröffentlicht, so blieb die alte Version unberührt referenzierbar, die Unterschiede zwischen den verschiedenen Versionen wurden aber leicht sichtbar gemacht ([Nelson, 2006](#)).

Die erste Maus Eine ähnliche Vision, wie ein Computer dazu beitragen kann das immer größer werdende Wissen vernünftig zu strukturieren (und zu verarbeiten) hatte 1962 *Douglas Engelbart* beschrieben. Dieser hatte in der Veröffentlichung „*Augmenting Human Intellect: A Conceptual Framework*“ ([Engelbart, 1962](#)) ein Konzept vorgestellt wie ein Computersystem den Menschen helfen kann Informationen zu organisieren. In seiner Arbeit waren ebenfalls Konzepte wie Hyperlinks enthalten, welche an jeder Stelle im Text eingefügt werden konnten und auf andere Informationsressourcen verwiesen. Neben den neuen Möglichkeiten der Textverarbeitung bot das System Möglichkeiten an, Informationsressourcen einfach zu strukturieren und zu kategorisieren. 1968 wurde der erste Prototyp dieses innovativen Systems erfolgreich vorgestellt. Dabei wurde u.a. ein speziell für diesen Zweck von Engelbart entwickeltes Gerät eingesetzt: Die erste Computermaus. Das System war so sehr seiner Zeit voraus, dass es von vielen nicht verstanden werden konnte und sich damit zunächst nicht durchsetzen konnte.

Mainframes Die 70er Jahre haben die Mainframes hervorgebracht. Diese wurden über Terminals dem Menschen zugänglich gemacht. Damit wurde der erste Schritt in Richtung des dezentralisierten Rechnens getan. Parallel hierzu tauchten die ersten Ansätze zur Entwicklung von Alert-Diensten ([Wikipedia, 2006a](#)) auf. Diese stellten mehr oder weniger die Vorreiter der heutigen Realisierungen des Publisher-Subscriber Modells.

Computer Supported Cooperative Work und Groupware Der Begriff Soziale Software tauchte in den 80er Jahren in Gestalt zweier neuer Begriffe wieder auf, der *Groupware* und dem *Computer Supported Cooperative Work*. Der erste Begriff umschließt die damals eingesetzten, die Gruppenarbeit unterstützenden Systeme. Zitiert in ([Christopher, 2004](#)) definierten Peter und Trudy Johnson-Lenz diese Groupware als „*intentional group processes plus software to support them*“.

Kooperative Arbeit wurde parallel dazu als Forschungsgebiet von der akademischen Gemeinde entdeckt. Dabei wurden Untersuchungen über die der Gruppenarbeit förderlichen Nutzung von Computern durchgeführt. Die dabei erarbeiteten Forschungsergebnisse sollten

nicht unter dem immer häufiger verwendeten Begriff Groupware veröffentlicht werden. Aus diesem Grund entstanden Begriffe wie Computer Supported Collaborative Work und Computer Mediated Communication ([Morville, 2005](#)), ([Christopher, 2004](#)).

Chat - Echtzeit Kommunikation Im Umfeld der kollaborativen Arbeit entstanden weitere wichtige Ansätze die eine Zusammenarbeit von mehreren Menschen deutlich vereinfachen sollten. Einer dieser Ansätze war das *Bitnet Relay* welches als Grundlage für den *Internet Relay Chat* ([Wikipedia, 2006b](#)) diente und eine computergestützte Echtzeitkommunikation erlaubte. Durch diese neue Kommunikationsmöglichkeit konnten sich Arbeitsgruppen, die sich an unterschiedlichen Orten befanden, miteinander durchgehend austauschen, wie dies bei lokalen Gruppen bisher möglich war. Echtzeitkommunikation war ein weiterer wichtiger Schritt in Richtung der Ermöglichung der Zusammenführung von Menschen.

WWW Es folgten weitere interessante Ansätze, welche Menschen immer mehr erlaubten miteinander zu kommunizieren, Wissen gemeinsam zu sammeln und kollektiv zu evaluieren. 1989 hatte *Tim Berners-Lee* ein Hypertext-System vorgestellt, mit dessen Hilfe Informationsressourcen dezentral organisiert werden konnten ([Berners-Lee, 1989](#)). Kurze Zeit später entwickelte er den ersten Browser (World Wide Web), welcher nach dem WYSIWYG Prinzip² funktioniert und damit jedem weniger erfahrenen Benutzer das Publizieren von Informationen im Web vereinfachen sollte.

Der mächtige Browser konnte sich nicht, wie ursprünglich geplant, verbreiten, sodass das Web sich als passives Medium, entgegengesetzt zur Vision von Berners-Lee, ausbreitete. Tim Berners-Lee nahm seine Anregungen aus dem Xanadu Projekt, jedoch übernahm er zu wenig der dort vorgestellten Konzepte. Dieses hätte nach ([Möller, 2006](#)) dazu beigetragen, dass das Web wesentlich interaktiver gestaltet werden konnte.

Definition

Die erste Verwendung des Begriffes Soziale Software wird von Christopher Allen ([Christopher, 2004](#)) auf die 80 Jahre auf Erik Drexler zurückgeführt, welcher diesen in seiner Arbeit „Hypertext“ ([Drexler, 1997](#)) verwendete. Wie bereits erwähnt, wurde parallel in den 80er Jahren der Begriff Groupware verwendet, was als Vorläufer der Sozialen Software angesehen werden kann. Um die Definition des Begriffes Soziale Software besser zu spezifizieren, sollten die damalige Ausprägung dieser Bezeichnung für vergleichbare Arten von Systemen näher untersucht werden. Eine Definition des Begriffes wurde von Peter und Trudy Johnson-Lenz gegeben, welche besagt das Groupware für „*intentional group processes plus software*

²What You See Is What You Get

to support them“ steht. Douglas Engelbart, der sich in den 60 Jahren mit gleichen Fragestellungen beschäftigt hatte, beschrieb Groupware als „*A co-evolving human-tool system*“ (Christopher, 2004). Betrachtet man die beiden Definitionen wird in beiden auf Softwaresysteme verwiesen, deren Aufgabe es ist, die Zusammenarbeit von Menschen zu unterstützen.

Erst gegen Ende 2002 wurde der Begriff Soziale Software wieder aktueller und gewann immer mehr an Bedeutung. Seitdem wird dieser Begriff zunehmend verwendet. Seine Definition wurde aber bisher nur näherungsweise wissenschaftlich definiert. Dies war der Anlass für Clay Shirky diesen unscharfen Begriff zu festigen. Eine der ersten auf Soziale Software abzielenden Definition hatte dieser wie folgt formuliert, wonach Soziale Software als „*software that support group interaction*“ (Shirky, 2003) beschrieben wird. Damit sah der Autor die Unterstützung von Gruppeninteraktionen als Hauptmerkmal an.

Eine weitere Konkretisierung des Begriffes Soziale Software wird in (Cho, 2006) gegeben. Dort beschreibt der Autor Soziale Software als „*verschiedenartige, lose verbundene Arten von Anwendungen, welche eine Kommunikation zwischen den Individuen ermöglichen und dabei die Möglichkeit zur Verfügung stellen, den Kommunikationsverlauf wiederzugeben*“ (Cho, 2006, übersetzt). Ein weiterer Definitionsversuch wird in Thomas Burg (Cho, 2006, zitiert aus, übersetzt) unternommen, wo „*Soziale Software alle Informations- und Kommunikationstechnologien umfasst, welche das digitale Vernetzen von Individuen und Gruppen ermöglichen*“.

Wie also aus den beiden Definitionen ersichtlich wird, handelt es sich in erster Linie immer um die Herstellung der Kommunikation zwischen Menschen. Betrachtet man zusätzlich zu den beiden genannten Definitionen die von Shirky, so wird ersichtlich das die drei Autoren immer eine spezielle Art von Softwaresystemen nennen, welche dazu dienen sollen Interaktionen zwischen miteinander verbundenen Menschen zu erleichtern.

Mit Systemen, welche unterstützend bei Interaktionen zwischen Arbeitsgruppen wirken, hatte sich ebenfalls in den 80er Jahren das oben genannte Forschungsgebiet „*Computer Supported Collaborative Work*“ auseinandergesetzt. Dieser, sehr viele Parallelen aufweisende, Forschungszweig wurde beispielweise in (München, 1997) wie folgt definiert: „*While Groupware refers to the real computer-based systems, the notion CSCW means the study of tools and techniques of Groupware as well as their psychological, social and organizational effects*“. Bei dieser Definition werden nicht nur die, wie in den Oben bereits zitierten Definitionen erwähnten unterstützenden Systeme angesprochen. Der Autor geht hier einen Schritt weiter und fasst in seiner Definition ebenfalls die Einflüsse die solch eine Software auf die Zusammenarbeit haben wird. Damit ist zu entnehmen, dass die dort eingesetzten Systeme stark mit dem Benutzerverhalten gekoppelt sind, diesen beeinflussen oder sogar selber beeinflusst werden.

Werden nun alle die bis jetzt untersuchten Definitionen zusammengetragen um daraus einen Konsens zu bilden, so ist nach Meinung des Autors die in (Coates, 2005) formulierte Defi-

nition am zutreffendsten, wo demnach „*Social Software can be loosely defined as software which supports, extends, or derives added value from, human social behaviour - messageboards, musical taste-sharing, photo-sharing, instant messaging, mailing list, social networking*“. Hierbei werden bereits einige Ausprägungen der Sozialen Software genannt. Die Funktionsweise und die daraus resultierenden Prinzipien werden im weiteren Verlauf der Arbeit beschrieben. An dieser Stelle sollte erstmals grob die Eigenschaften der Sozialen Software identifiziert werden.

Eigenschaften

Die Betrachtung der Eigenschaften Sozialer Software möchte der Autor mit folgender Aussagen beginnen: „Soziale Software spielt vor allem dann ihre Stärken aus, wenn der Community-Gedanke im Vordergrund steht“ (Himpsl, 2006). Demnach ist Soziale Software nicht nur ein neuartiges Werkzeug um eine Community zu bilden. Es handelt sich um einen Katalysator, welcher für dessen Entstehung und der Sicherung ihres Fortbestandes notwendig ist. Folglich kann sich das System nur dann voll entfalten, wenn dahinter eine Community steht. Somit zeichnet sich eine auf Sozialer Software basierende Idee durch eine Art Symbiose zwischen diesen beiden Entitäten aus.

Für die gesunde Funktionsweise ist also eine funktionierende Community unabdingbar. Was zeichnet aber eine Community aus? Nach heutigem Verständnis versteht man unter einer Community eine Gruppe von Individuen, welche in erster Linie ein gemeinsames Interesse verfolgen. Dabei im Vordergrund stehende Prozesse werden durch die neuartige Soziale Software verstärkt, indem die Individuen zwar räumlich voneinander entfernt, nichtsdestotrotz aber ungehindert miteinander kommunizieren können bzw. in einem sozialen Netzwerk interagieren.

Demzufolge handelt es sich bei Sozialer Software in erster Linie um den Aufbau und damit verbundener Pflege der in einer Community entstehenden sozialen Beziehungen. Durch diese bekommt der Teilnehmer die Möglichkeit zum einen seine Beziehungen festzuhalten, damit diese nicht verloren gehen. Zum anderen kann dieser seine Kontakte vereinfacht strukturieren. Untersucht man die verschiedenen neuartigen Anwendungen stößt man auf ein weiteres interessantes Merkmal von Sozialer Software. Es scheint als müsse die Soziale Software bei der Bildung der Gemeinschaft nicht nur unterstützend und fördernd sein, sondern auch nicht als hinderlich wahrgenommen werden. Damit ist gemeint, dass Soziale Software sich dadurch auszeichnet, dass diese keine Strukturvorgaben definiert. Vielmehr überlässt sie diese anspruchsvolle Aufgabe der Community. Dabei entstehenden Strukturen basierend auf Prinzipien, welche schon aus den Selbstorganisierten Systemen (Wikipedia, 2006f) bekannt sind. Eine detaillierte Ausführung der Funktionsweise von Selbstorganisierenden

Systemen verdeutlicht Surowiecki in seinem Buch ([Surowiecki und Beckmann, 2005](#)). Diese Eigenschaft ist außerdem deshalb so essentiell für die Soziale Software, weil hierdurch die Sicherstellung der Autonomie der Community noch verstärkt wird.

Analysiert man die als Soziale Software klassifizierten Anwendungen so können diese in zwei Arten unterteilt werden ([Alby, 2006](#)):

- Systeme die als Ziel haben Kommunikation zwischen Menschen bereitzustellen.
- Systeme die der Community unterstützend bei der Erstellung von Inhalten zur Seite stehen, also auf die kollaborative Arbeit abzielen.

Natürlich ist die erste Eigenschaft als Basisfunktionalität für die zweite zu sehen. Betrachtet man aber die zweite Eigenschaft genauer wird hierbei ersichtlich, dass diese heute im Mittelpunkt der Communities liegt. Aus dieser Eigenschaft ergeben sich zwei weitere Aspekte, welche die Soziale Software so attraktiv machen:

- Benutzergenerierter Inhalt
- Kombination von Datenquellen

Mit anderen Worten geht es bei Sozialer Software u.a. um die kollaborative Sammlung, Evaluierung und Verwaltung von Daten, deren Verknüpfung mit anderen Datenquellen und damit Erschließung von neuartigen Zusammenhängen (Wissen). Diese neuen Möglichkeiten begeistern immer mehr Menschen und ermutigen diese zur einer Teilnahme.

Bei der großen Menge von gesammelten Daten wird vieles über den Benutzer offengelegt. Auf der anderen Seite möchten viele Menschen ihre Privatsphäre nicht verlieren. Ungeachtet dessen werden immer mehr Menschen teil einer Community und somit transparenter. Es stellt sich die Frage welche starken Beweggründe führen dazu, dass Menschen immer zunehmend ihre persönlichen Daten zur Verfügung stellen. Es müssen weitere, soziologisch begründete, Beweggründe existieren, welche Menschen dazu bewegen sich soweit nach außen zu offenbaren. Mit den Gründen für die Teilnahme an verschiedenen Sozialer Software Systemen haben sich ([Huffaker und Calvert, 2005](#)) und ([Alby, 2006](#)) auseinander gesetzt. Demnach sind hierbei die folgenden vier Motive identifiziert worden:

- Teil der Gemeinschaft zu werden, zu dieser einen Beitrag leisten und auch selber Nutzen aus dieser ziehen
- Anerkennung und Aufmerksamkeit zu erlangen
- Ein Gefühl etwas verändern zu können und damit die Welt zu verbessern
- Selbstreflexion und Identitätsfindung (Selbstdarstellung, Identitätskonstruktion)

Nach Ansicht des Autors stellen diese eben genannten Motive den Hauptgrund für die so große Teilnahme und steigender Offenheit der Menschen. Soziale Software ist hierbei als Werkzeug für das Informations-, Identitäts- und Beziehungsmanagement zu sehen. Damit zielt diese darauf ab persönliche sowie berufliche Beziehungen transparent nach außen zu machen.

Zuvor wurde bereits erwähnt, dass die Erstellung des Inhalts sehr wichtig in Zusammenhang mit Sozialer Software ist. Aber nicht nur das alleinige Merkmal der kollektiven Sammlung von Daten und die Daten selbst sind wichtig. Auch die Betrachtung der Wertentwicklung der Datenbestände für die Community, sowie aus kommerzieller Sicht weist ein interessantes Merkmal auf. Die gesammelten Daten erlangen bei zunehmender Benutzerzahl, parallel eine Wertsteigerung (Schmidt, 2006).

Nach Aussage von (Schmidt, 2006) wird Soziale Software durch Entwicklungen wie "continuous computing" (Roush, 2005) oder „ubiquitous computing“ (Weiser, 1991) zusätzlich angetrieben. Durch diese kann sie nahtlos in das tägliche Leben integriert werden und damit dieses noch weiter verändern (im Positiven sowie auch Negativem Sinne). Um diesen Ansprüchen gerecht zu werden, muss sich die Soziale Software nicht nur durch die Offenheit in der Benutzung und Gestaltung auszeichnen. Damit die, wie weiter oben genannte, Kopplung von mehreren Datenressourcen möglich wird müssen sich die entsprechenden Anwendungen durch offene Schnittstellen auszeichnen. Diese Offenheit in der technischen Sichtweise impliziert gleichzeitig das von O'Reilly identifizierte Merkmal. Durch diese Offenheit entwickeln sich die Anwendungen kontinuierlich weiter und erreichen somit mit zunehmender Zeit immer mehr die Benutzervorgaben.

Abschließend möchte der Autor auf die von Tom Coates (Coates, 2002) spezifizierten, drei wesentlichen Funktionen der Sozialen Software verweisen, die das eben besprochene nach Ansicht des Autors am besten wiedergeben:

- 1. removing the limitations placed on social contact by external factors such as language and geography.*
- 2. compensating for the overloads that this removal of limitations might generate.*
- 3. uncovering and improving on the mechanisms that people use in their social interactions with one another - making rational decisions about which are still appropriate and which can be replaced by software or technology.*

Zusammenfassung

Bis hierher hatte der Autor versucht dem Leser die wichtigsten Aspekte die hinter der Sozialen Software stehen, dem Leser näherzubringen. Betrachtet man die heute innovativen

webbasierten Anwendungen, so verbinden diese Merkmale wie, „frei“ und „von jedem erweiterbar“. Tim Berners-Lee hatte 1999 in einer Rede seine Vision des Webs wie folgt formuliert: „I wanted the Web to be what I call an interactive space where everybody can edit. And I started saying interactive, and then I read in the media that the Web was great because it was interactive, meaning you could click. This was not what I meant by interactivity.“ (Berners-Lee, 1999). Durch den zunehmenden Einsatz von Sozialer Software kann diese Vision möglicherweise Realität werden. Ungeachtet dessen wie die einzelnen Ausprägungen Sozialer Software aussehen, bieten alle eine effiziente Kommunikations- und Interaktionsmöglichkeit, welche die Teilnehmer dazu motiviert selbst etwas dazu beizutragen und damit das Web mitzugestalten.

Soziale Software nimmt heute die Schlüsselrolle bei der Ermöglichung zwischenmenschlicher Kommunikation (Cho, 2006) ein. Die dabei ermöglichte kontinuierliche Interaktion und Zusammenarbeit zeichnet sich durch einfache Onlinedienste wie Mail, Instant Messaging und weitere Kommunikationsarten aus. Durch diese Dienste sind Menschen Werkzeuge gegeben worden durch die sie in die Lage versetzt werden miteinander an gemeinsamen Zielen zu arbeiten. Die verschiedenen Anwendungsdomänen sowie einige Ausprägungen werden sehr schön in (Kalz, 2006) verdeutlicht, wie Abbildung 2.1 zeigt.

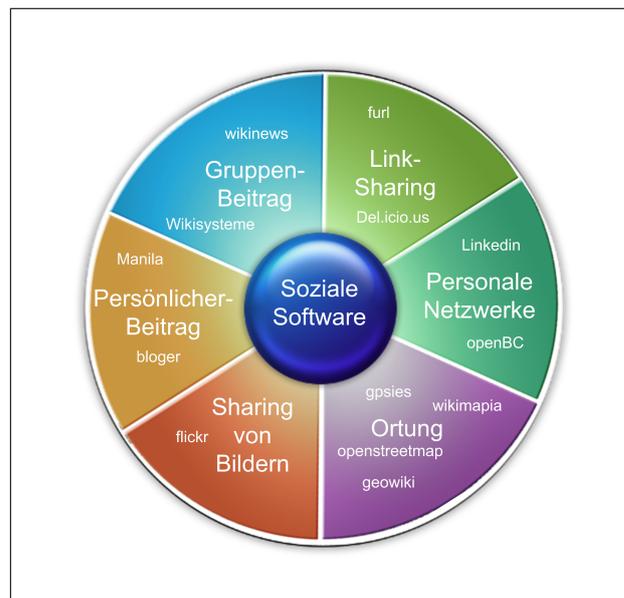


Abbildung 2.1: Ausprägungen der Sozialer Software unterteilt nach Anwendungsdomänen (Vgl. Kalz, 2006)

Die hierbei sechs identifizierten verschiedenen Anwendungsdomänen umschließen immer eine Community. Letztere stellt auch einen verarbeitenden Kanal zwischen diesen, und überführt die verschiedenen Daten in immer neue Ausprägungen des Kollektiven Wissens. Was sich hierdurch für Potentiale eröffnen ist schwierig abzuschätzen.

Der Begriff Soziale Software zielt auf das allgemeine Verständnis des Sozialen. Zu was diese Art von Software beitragen wird hängt letztendlich von dem Menschen selber ab. Viele positive Möglichkeiten eröffnen sich erst jetzt. Ungeachtet dessen existiert auch die Kehrseite der Medaille. Einige negative Szenarien, wie die digitale Spaltung, können genau das entgegengesetzte bewirken (Schmidt, 2006).

Abschließen möchte der Autor diesen Abschnitt mit Worten die Erik Möller in (Möller, 2006) in Zusammenhang mit der Betrachtung von Auswirkungen der Änderungen in den Medien gesagt hat: „*Es gibt kein Individuum und keine Schlüsseltechnologie, von der alles Weitere abhängt. Der Erfolg des Systems ist eine Eigenschaft des Systems.*“

2.1.2 Web 2.0

In dieser Arbeit soll ein Navigationssystem weiterentwickelt und verfeinert werden. Dabei soll sich die Arbeit darauf konzentrieren ein für eine Gemeinschaft geeignetes Werkzeug zu entwickeln, welches den heutigen Standards genügt. Wichtig ist, die neuesten Entwicklungen und Trends zu berücksichtigen. Einer dieser heutigen Trends ist das *Web 2.0*. Bevor jedoch mit den Einzelheiten begonnen wird sollte die geschichtliche Entstehung wie bei der Sozialen Software kurz überflogen werden.

Vorgeschichte

Analysiert man die Mainframe Ära, sind die Terminals eine notwendige Maßnahme gewesen um die damals noch teuren Ressourcen mit anderen Benutzern zu teilen. So gesehen war diese Verteilung damals auch in höchstem Maße gefordert. In Hinsicht auf den Web 2.0 Ansatz und seiner Prinzipien war dies eine der ersten Ausprägungen dieser dahinterstehenden Leitsätze. Die Zentralisierung von Rechenkapazität und Daten auf der einen und die Zugänglichkeit auf der anderen Seite findet sich alles in heutigen Web 2.0 Anwendungen wieder.

In den 90er Jahren erschien ein neuer Ansatz namens *Thin-Client*, welcher die bisher bekannten Architekturen in den Schatten stellen sollte. Letzterer sollte günstig sein und die darauf laufenden Anwendungen sollten webbasiert (und aus dem Web beziehbar) sein. Damit waren diese neuartigen Anwendungen von einer Internetverbindung abhängig. Da das Internet zur damaligen Zeit noch nicht so weit fortgeschritten war (gemeint sind hierbei die Bandbreiten und die Abdeckung) setzte sich diese erste Äußerung des neuartigen Denkens zunächst nicht durch. In den 90er Jahren tauchte aber noch ein anderer Trend auf, welcher mobilen Nutzern Anwendungen zugänglich machte. Der Thin-Client hatte selber keine Mobilität unterstützt. Dieses wird in der Literatur als ein weiterer Grund für das Scheitern dieser Technologie betrachtet.

Die 90er Jahre waren ebenfalls sehr ergiebig aus Sicht der heute in den Web 2.0 Konzepten verwendeten Technologien. 1995 wurde von *Ward Cunningham* ([Wikipedia, 2006h](#)) das erste Wiki entwickelt. Parallel hierzu hatte Netscape die Vorversion einer Skriptsprache veröffentlicht die nach kurzer Zeit den Namen JavaScript ([Wikipedia, 2006d](#)) bekam. Ein interessanter Aspekt bei der Entstehung von JavaScript ist, dass die dort verwendeten Ansätze aus der Praxis kamen und weniger von einer übergeordneten Instanz aufgezwungen wurden. Es dauerte nicht lange bis eine weitere Technologie, die Web Services auftauchten. Nach kurzer Zeit zeigte sich, dass die ersten webbasierten Anwendungen Probleme bei der Versorgung mit Daten hatten. Bei jeder Anfrage musste die gesamte Seite vom Server angefordert werden. Die Antwort darauf war der von Microsoft entwickelte *XMLHttpRequest* ([W3C, 2006](#)), welcher auch die technologische Grundlage für Ajax darstellte. Durch diese Technologie wurde eine asynchrone Datenversorgung für Webanwendungen entwickelt. Das ständige erneute Laden von kompletten HTML-Seiten entfiel somit.

Basierend auf dem Konzept der Alert-Dienste entstand 1999 das Really Simple Syndication (RSS), eine Technik die das Abonnieren von Web-Ressourcen erlaubt. Daneben breitete sich mit großer Geschwindigkeit parallel die Bannerwerbung aus. Letztere baute auf der Kooperation zwischen sich gegenseitig aufrufenden Webseiten auf. Folglich also die erste Realisierung des Mashup-Konzeptes ([Alby, 2006](#)). 2001 startete dann das bis heute noch erfolgreiche Projekt Wikipedia. Obwohl die Marktkrise seit 2001 immer noch anhielt, entstanden immer mehr webbasierte Anwendungen, welche später in ([O'Reilly, 2005](#)) als die Vorreiter der heute als Web 2.0 klassifizierten Anwendungen gelten. O'Reilly hatte in seiner Ausarbeitung ebenfalls darauf hingewiesen, dass sich ab 2002 eine neue Sichtweise auf das Web entwickelte. Diese Annahme beruht auf einer Untersuchung der gemeinsamen Merkmale der bis heute entwickelten Webanwendungen.

Web 2.0 - Entstehung des Begriffes

Die Auseinandersetzung in der oben zitierten Publikation ([O'Reilly, 2005](#)) basiert auf der, wie sich später herausstellte, ersten Web 2.0 Konferenz, welche im Oktober 2004 stattfand. Bei dieser Konferenz, bei der MediaLive International und O'Reilly beteiligt waren, hat man gemeinsam versucht die Auswirkungen und Veränderungen, die nach dem Zusammenbruch des Marktes nach dem Herbst 2001 zur Vorschein kamen, zu analysieren. Die dahinterstehenden, auf dieser Konferenz extrahierten, Merkmale wurden unter dem Begriff Web 2.0 zusammengefasst.

Bei der Diskussion stellte sich heraus, dass eine bestimmte Art von Webanwendungen, bzw. die die bestimmte Merkmale aufwiesen, sich im Web etablieren konnten. Diese Art von Webanwendung hatte erstaunlicherweise den Kollaps des Marktes überlebt. Die Identifizierung

dieser Merkmale wurde dann der Gegenstand dieser Diskussion. Als Ergebnis dieser Debatte wurden Prinzipien und Vorgehensweisen, welche hinter dem Begriff des Web 2.0 stehen, identifiziert.

Die dabei erkannten Änderungen der im Web etablierten Regeln und Geschäftsmodelle, welche die Web 2.0 Anwendungen charakterisieren, berühren verschiedene Gesichtspunkte. Das Web spielt dabei eine zentrale Rolle und wird in diesem Konzept als die Basisplattform betrachtet. Auf dieser entstehen Anwendungen, welche als austauschbare Werkzeuge betrachtet werden können. Dieses wird durch die neue, datenzentrische Sicht des Web 2.0 motiviert, da Daten und nicht die Anwendungen selber hierbei das Kapital der Unternehmen darstellen.

Ein weiteres sehr wichtiges Merkmal bei diesem Ansatz ist die kollaborative Arbeit und damit die an jeden Teilnehmer angebotene Möglichkeit des Mitwirkens zu sehen. Dieses äußert sich nicht nur in der Verarbeitung von Daten, sondern auch bei der Mitgestaltung des Systems selber. Hierfür sind die Web 2.0 Dienste technisch sehr einfach konzipiert und offen in Hinsicht auf ihren Verwendungszweck. Diese Tatsache ermöglicht den Teilnehmern diese Dienste beliebig miteinander zu verschachteln und somit neuartige Geschäftsprozesse abzubilden. Das System wächst somit ständig und erreicht unter der Beachtung der Prinzipien des Web 2.0 nie einen finalen Zustand.

Gegenwart

Bis zu den Neunzigern war das Internet nur eine Ablage und Versorgungsplattform für Dokumente gewesen. Langsam erkannte man, dass das Internet mehr zu bieten hat und so entstanden langsam neue Technologien, die heute unter dem Web 2.0 Begriff zusammengefasst sind. Für den richtigen Einsatz dieser Technologien steht aber erst heute die notwendige Bandbreite bereit. Diese Tatsache führt dazu, dass diese Techniken erst im heutigen Web vermehrt ihren Einsatz finden. Heute trifft man nahezu überall auf Web 2.0 basierte Ansätze und darauf basierende Soziale Software.

Neue Technologien eröffnen neue Wege für neue Möglichkeiten und bewirken außerdem Änderungen die nicht unbedingt technologischer Natur sein müssen. Nach ([Wikipedia, 2006i](#)) ist seit 2005 eine Veränderung des Webs durch folgende Merkmale zu beobachten:

- Verschmelzung von lokalen und zentralen Daten
- Verschmelzung von lokalen und internetbasierten Anwendungen
- Trend vom Passiven zum Aktiven Benutzer, vom Privaten zum Öffentlichen
- Verschmelzung von Diensten (Offenheit gegenüber Nutzung und Veränderung)

- Komplexitätsminderung der Dienste und somit Offenheit gegenüber fachlich weniger ausgebildeten Benutzern
- Vereinfachte Meinungsäußerung

In ([Adler, 2005](#)) wird von der kollabierenden Begrenzung gesprochen. Der Benutzer ist heute im Stande nicht nur formale Daten mit dem Server auszutauschen. Vielmehr können diese miteinander integrieren. Diese Art der Kommunikation ist heute auch als asynchrone Kommunikation bekannt. Durch diese kann sich eine Webseite so verhalten als wäre es eine Desktopanwendung. Der Unterschied hierbei kann wie folgt verdeutlicht werden: Die Desktopanwendung hatte immer zwei Rollen übernommen - diese umklammert die Darstellungs- und Anwendungslogik. Bei den webbasierten Anwendungen sind diese Rollen verteilt. Der Browser übernimmt die Verantwortung für die Darstellung, also umfasst dieser die Darstellungslogik. Die Anwendungslogik wird auf dem Server ausgelagert und unabhängig von der Entfernung dem Browser zur Verfügung gestellt.

Wenn man sich die Mainframes und die daran angeschlossenen Terminals nochmal näher anschaut, zeigen diese zum Teil eine Korrelation mit dem heutigen Web 2.0 Ansatz. Wie zuvor erwähnt wurde die Verteilung bei den Mainframes als eine Notwendigkeit erachtet. Im Web 2.0 wird die Verteilung anders motiviert. Durch die immer stärker werdende Forderung nach Mobilität ist die Verteilung einer der Grundvoraussetzungen. Eine Analogie zwischen dem Mainframe und dem Server, sowie zwischen dem Terminal und dem Webbrowser ist ersichtlich.

Durch Anbieten der Erstellungsmöglichkeit an ein breites Publikum wurde eine Tür zum Neuen eröffnet. Neben den Programmierern können nun nahezu alle Internet-Benutzer an der Evolution des Webs beteiligt werden. Dieser Umstand führte dazu, dass eine vorausgesetzte kritische Masse erreicht wurde und folglich erst jetzt neue Synergieeffekte entstehen und damit neue Dimensionen einnehmen können. In diesem Zusammenhang bezeichnet ([Millard und Ross, 2006](#)) das neue Web als verwildert. In weiteren Ausführungen wird davon gesprochen, dass Systeme und das Eigentum nicht mehr kontrollierbar sind. Welche Auswirkungen das auf die heutige Gesellschaft haben wird ist noch nicht ersichtlich.

Browser sind an sich sehr schlanke Anwendungen die für die Darstellung von Dokumenten konzipiert wurden. Mit dem Auftauchen von JavaScript hatte sich diese Gegebenheit geändert. Der Browser wurde dadurch befähigt anspruchsvollere Front-Ends darzustellen. Durch JavaScript wurde die Programmierung des Browsers ermöglicht. Dadurch konnten Rechenoperationen durchgeführt, neue Fenster geöffnet, sowie einige andere Aufgaben auf den Client verlagert werden, wodurch neue Möglichkeiten geschaffen wurden. Ab da konnten die im Internet abgelegten Webseiten Programmcode auf dem Clientbrowser ausführen. Verfolgt man die Änderungen bis zu dem Stand-Alone Anwendungen zurück, wird eine Evolutionskette dieser bis zu heutigen serviceorientierten webbasierten Anwendungen ersichtlich. Diese wird in [Abbildung 2.2](#) von Hashimi & Steffan verdeutlicht.

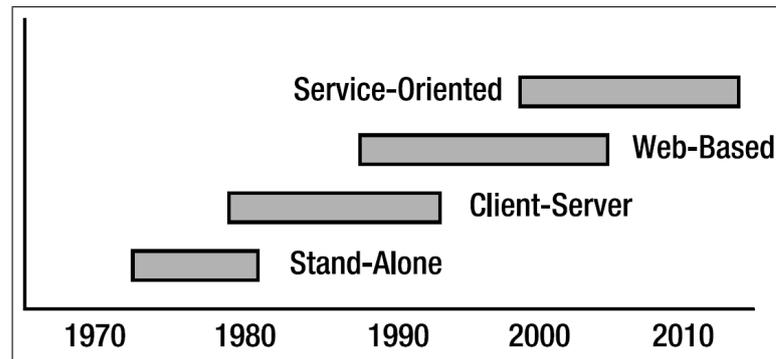


Abbildung 2.2: Die Evolution der Computer Architektur (Vgl. [Hashimi und Steffan, 2005](#))

Web 2.0 umschließt nicht nur die Software, sondern auch die Hardware, die immer mehr in den Hintergrund rückt. Es entstehen immer mehr intelligente Geräte die autonom im Hintergrund agieren können. So z.B. können diese bei Notwendigkeit eine Verbindung mit dem Internet aufbauen und somit von den dort zur Verfügung gestellten Ressourcen profitieren.

Im Gegensatz zu dem ursprünglichen Design ist das Web 2.0 Modell verstärkt heterogen, spontan und entwickelt sich evolutionär. Auf der anderen Seite ist es pragmatisch und robust. Hierdurch wird die Entfaltung neuer Anwendungen begünstigt (vgl. [Millard und Ross, 2006](#)).

Zukunft

Es ist zu erwarten, dass sich der Wettbewerb zwischen bereits etablierten und neuen Firmen zuspitzen wird. Dieses wird durch die Offenheit dieses Ansatzes bewirkt, die gegenüber neuen Ideen die treibende Kraft darstellt. In einem Szenario in dem Anwendungen über das Internet bereitgestellt werden, entfällt die Problematik der Lizenzierung. Jede Anwendung wird als Dienstleistung bereitgestellt. So wird es dem Benutzer immer freigestellt sein, die Dienstleistung bei dem besten Anbieter in Anspruch zu nehmen. Die Kosten der Anschaffung einer teuren Desktopanwendung entfallen. Die weiteren hierbei verbundenen Risiken wie das Veralten, oder einfach die beim Kauf nicht berücksichtigten Mängel, entfallen bei einer Dienstorientierung.

In ([Adler, 2005](#)) wird davon gesprochen, dass das Betriebssystem in der Zukunft eine untergeordnete Rolle spielen wird. Diese Vision kann für Firmen wie Microsoft gravierende Auswirkungen nach sich ziehen. Um dieses vorzubeugen hat Microsoft die *.NET* Plattform entwickelt. Hinter dieser verbirgt sich eine Reihe von Technologien, welche auf einer Plattform wie *WebOS* ([Adler, 2005](#)) zum Tragen kommen könnten. Dies ist ein Versuch an der

neuen Entwicklung des Webs, also die vermehrte Entstehung und Etablierung von nicht Microsoft Technologien, teilhaben zu können. Mit anderen Worten stellt Microsoft mit der .NET Technologie eine Plattform für Entwicklung zukünftiger Web 2.0 basierten Anwendungen zur Verfügung, mit der Hoffnung dort den Anschluss nicht zu verlieren.

Es gibt natürlich auch eine Kehrseite der Medaille. Durch den Wechsel von einer Desktopanwendung zur einer an die Dienstleistung orientierten Web-Anwendung entstehen neuartige Risiken. Bis heute hatte der Benutzer seine Daten lokal auf seinem Rechner verarbeitet. Bei den Webanwendungen kommt ein neuer Trend, die Daten entfernt bei dem Dienstanbieter zu speichern. Hierdurch hat der Benutzer überall einen Zugriff auf diese. Auf der anderen Seite wird er zunehmend die Kontrolle über seine Daten verlieren. Ein vergleichbares Phänomen entsteht bei der Anwendung selbst. Hatte sich der Benutzer an eine Anwendung erstmals gewöhnt, beherrscht er sie meisterhaft. Ein Vorteil der Webanwendungen ist, dass diese durchgehend weiter verbessert werden. Die Gefahr die dabei entstehen kann ist, dass der Benutzer, der sich gerade an die jetzige Version erstmals gewöhnt hat und damit umgehen kann, durch eine neue Version aus dem Konzept gebracht wird. Es besteht die Gefahr, dass der Benutzer seine Anwendung einfach nicht beherrschen wird und mit ihr nicht zurechtkommt, da diese regelmäßigen Veränderungen unterliegt.

Tim Berners-Lee, der ein Vertreter des *Semantischen Web* ist, kritisiert Web 2.0. Er bewertet den hinter dem Begriff stehenden neuen Trend als eine Art neuer Verpackung für alte Technologien. Nach seiner Aussage soll hinter Web 2.0 nichts Neues stecken. Weitere Meinungen deuten darauf hin, dass Web 2.0 als neuer Versuch für die *New Economie* zu deuten ist, wie man diese vor der Konsolidierung des Marktes um 2000 fand (vgl. [Wikipedia, 2006i](#)).

Diese und weitere Gefahren beziehen sich mehr oder weniger auf die technische Sicht der neuartigen Anwendungen. Es gibt aber noch eine allgemeinere, von der in ([Adler, 2005](#)) gesprochen wird. Dort formuliert der Autor, dass es keine Frage der Technik sei, diese neue Sichtweise einzuführen. Vielmehr muss hierbei untersucht werden, wie der Bedarf an der Nutzung einer solchen Plattform aussieht. Ist das vielleicht eine Gefahr die unterschätzt wird?

Letzteren Endes spielen heute die Mobilität, dabei durchgängige Vernetzung (Verbindung mit dem Netzwerk) und so gesehen auch die Dienstorientierung eine übergeordnete Rolle. Dieses führt dazu, dass die alt bewährten und möglicherweise ausgereifteren Lösungen eher in den Hintergrund treten werden. Die Verfügbarkeit von Diensten steht in erster Linie im Interesse der Benutzer, sodass eine Veränderung in der Sichtweise und somit der Forderung nach neuen Möglichkeiten der Informationsverarbeitung zu erwarten sind.

Die Vision umfasst positive sowie negative Seiten der neuen Prinzipien die hinter Web 2.0 stehen. Die Entwicklung und die Auswirkungen dieses Ansatzes kann man sich nur ansatzweise ausmalen. Um die Möglichkeiten diese neue Denke und die dahinter stehenden Tech-

nologien besser einschätzen zu können, soll in den nächsten Abschnitten eine genauere Betrachtung dieses Ansatzes erfolgen. Die dabei resultierenden Erkenntnisse sollen helfen die Einsatzmöglichkeiten besser zu erfassen.

2.2 Technologische Betrachtung

In dem vorangegangenen Abschnitt wurde gezeigt wie die Begriffe Soziale Software und Web 2.0 chronologisch einzuordnen sind. Es zeigte sich, dass viele der Ansätze ihren Ursprung weiter in der Vergangenheit haben und damit bereits viel früher in den Köpfen genialer Pioniere der Informatik entstanden. Es ist nicht so wie von einigen angenommen wird, dass der Trend erst vor kurzen entstand. Aber was verbirgt sich nun genau hinter den beiden Begriffen die immer mehr neuartige webbasierte Anwendungen kennzeichnen? Sind das nur Bezeichnungen für einen Trend, welcher aktuell von immer mehr Unternehmen verfolgt wird? Oder steckt hinter diesem noch etwas mehr als das? Diese und einige weitere Fragen sollen in diesem Abschnitt beantwortet werden.

Dieser Abschnitt setzt sich wie folgt zusammen. Zu Beginn möchte der Autor näher auf die hinter den Community-gestützten Systemen liegenden Technologien eingehen. Hierzu werden die am häufigsten verwendeten aufgezählt und kurz erläutert. Die Absicht hinter diesem Abschnitt ist die Betrachtung der Basis und die Eigenschaften dieser Technologien. Der Leser sollte nach der Lektüre dieses Abschnittes mit diesen vertraut sein. Dieses wiederum ist wichtig um die spätere Untersuchung der allgemeinen Merkmale Community-gestützter Systeme und den Gedankengang des Autors verfolgen zu können.

Bevor jedoch die Merkmale identifiziert werden, sollen die heute gängigen Ausprägungen von Community-gestützten Systemen veranschaulicht werden. Dieses soll dem Leser die Art und Weise dieser Anwendungen näher bringen. Damit soll auch eine Haltungsänderung bei dem Bereitstellern und den Benutzern motiviert werden. Denn wie es aus dem weiteren Verlauf dieses Abschnittes ersichtlich sein wird, soll Web 2.0 eine neue Art der Denkweise darstellen. Dieser Überlegung soll in diesem Abschnitt auf den Grund gegangen werden.

Abschließend sollten in dem letzten Abschnitt die Merkmale von Community-gestützten Systeme an sich zusammen getragen und erläutert werden. Dieser Abschnitt stellt auch das zentrale Augenmerk dieser Untersuchung. Der Autor möchte mit diesem Abschnitt den Einsatz von Web 2.0 Technologien bei der Entwicklung des hier zugrunde liegenden Navigationssystems motivieren. Daneben sollen die hierdurch möglicherweise entstehenden Vorteile sowie Nachteile verdeutlicht werden.

2.2.1 Technologien Community-gestützter Systeme

An dieser Stelle sollten die zugrunde liegenden Technologien der Community-gestützten Systeme eingeleitet werden, welche für deren Realisierung heutzutage verwendet werden. Dabei möchte der Autor keine genauere Beschreibung dieser Technologien durchführen. Falls dieses notwendig erscheinen sollte, wird dieses im späteren Verlauf der Arbeit nachgeholt. Es soll vielmehr eine einführende Beschreibung erfolgen, durch die dem Leser ein grobes Verständnis ermöglicht werden soll.

REST

Rest steht für *Representational State Transfer* ([Wikipedia, 2006e](#)) und wurde in Rahmen einer Dissertation von Roy Fielding [4] als Architekturkonzept für Informationssysteme eingeleitet, welches auf Hypermedien aufsetzt. Die Hauptforderung an solch ein System war die Skalierbarkeit. Dieses Architekturkonzept abstrahiert von den Details der einzelnen Komponenten und deren Syntax. Vielmehr betrachtet es die Interaktion die zwischen den involvierten Komponenten stattfindet. Dabei werden die Rollen der Komponenten, sowie die Interpretation der bei der Interaktion verwendeten Daten betrachtet. Die hinter dem Konzept stehenden Prinzipien sind:

Zustandsloses Kommunikationsprotokoll: Diese Vorgabe entstand in erster Linie durch die Forderung ein skalierbares und lose gekoppeltes System zu entwickeln. Um dieses zu erreichen, wurden alle Zustandsinformationen, die für die Verarbeitung notwendig sind, in das Kommunikationsprotokoll verlagert. Durch diese Maßnahme wurde die Verantwortung der Zustandsüberwachung von dem Servern genommen, sodass diese von den hierzu notwendigen Aufgaben entlastet wurden. Diese werden in der hier zugrunde liegenden Architektur nur für die Beantwortung der an sie gestellten Anfragen eingesetzt. Falls eine Verarbeitungskette den Verarbeitungszustand kennen muss, wird dieser durch die Clientanwendung aufgenommen und mit der Anfrage an den Server gesendet. Http ist ein solches zustandsloses Protokoll. Dieses versendet alle notwendigen Informationen in der http-Nachricht.

Wohldefinierte Operationen: Durch diese Vorgabe sollen Methoden eindeutig definiert werden, die bei der Verarbeitung der zugrunde liegenden Ressourcen angewendet werden. Http definiert solche Methoden wie GET, PUT, POST, DELETE.

Allgemeine Ressourcenbeschreibung: In diesem Konzept wird davon ausgegangen, dass das Web Unmengen von Ressourcen enthalten wird. Ein Problem was dabei entsteht ist die eindeutige Identifizierung der Ressourcen und somit deren Auffindung. Um dieses zu gewährleisten wird eine Forderung nach einer Syntax gestellt, die zum

einen Allgemein zum anderen Ausdrucksstark ist um solch eine Ressource eindeutig zu beschreiben.

Hypermedia: Zu Beginn wurde bereits gesagt, dass Rest für ein Informationssystem steht, welches vollständig auf Hypermedien basieren soll. Dieses ist durch die Eigenschaften und Möglichkeiten, die diese mit sich bringen, zu motivieren. Ein solches System kann bei der Akquirierung von Informationen diese durch Links miteinander in Beziehung setzen. Diese können dann bei der Durchsicht der Informationen zum Navigieren verwendet werden. Durch die Aufnahme der Netzstruktur in die Ressourcenbeschreibung wird aber noch ein weiterer, sehr wichtiger Vorteil gewonnen. Da jede Ressource die Beziehung zu anderen Ressourcen in sich umschließt, ist eine zentrale Verwaltungskomponente, wie eine Registrierungsdatenbank nicht mehr erforderlich.

Der Rest Ansatz stützt sich auf der klassischen Client/Server Architektur. Diese Maßnahme bewirkt eine Teilung zwischen der Darstellung und den Daten. Die Datenhaltung kann dadurch einfacher entwickelt werden, da die Serverkomponenten frei gestaltet werden können. Der dadurch gewonnene Vorteil ist die bessere Skalierung eines so aufgebauten Systems. Zudem erhöht die eben erwähnte Trennung die Portabilität des Systems.

Konsequenzen zustandsloser Protokolle Durch die Verwendung von zustandslosen Protokollen ergeben sich drei Vorteile:

Sichtbarkeit: Die Durchführung einer Operation wird transparenter in ihrem Ausführungskontext. Diese resultiert daraus, dass nur eine Anfrage überwacht werden muss.

Zuverlässigkeit: Eine Anfrage kann schnell wieder aufgesetzt werden, da kein Verarbeitungszustand, welcher sich im Vorfeld aufgebaut hat, beachtet werden muss. Die Betriebssicherheit des Systems wird hierdurch erhöht.

Einfachheit: Die Systemkomponenten müssen keine Verarbeitungszustände verwalten. Hierdurch kann also bei der Entwicklung dieser Komponenten die damit verbundene Problematik vernachlässigt werden. Diese Tatsache macht diese in ihrem Aufbau einfacher und damit auch das gesamte System. Zum anderen wird hierdurch eine bessere Grundlage zur Skalierbarkeit des gesamten Systems geschaffen.

Die eben genannten Vorteile muss man sich aber durch zwei neue Nachteile erkaufen:

Performance: Da bei jeder nichterfolgreichen Durchführung einer Anfrage diese beliebig bis zum Erfolg wiederholt werden kann, kann das ganze Netzwerk überflutet werden und damit verlangsamt werden.

Kontrolle: Die lose Kopplung und die Abgabe der Überwachung des Verarbeitungszustandes an den Client führt dazu, dass auch die Semantik der Anwendung und somit ihr

Verhalten nicht kontrollierbar ist. Hierdurch kann die hinter der Anfrage stehende Anwendung in einem ganz anderen Kontext verwendet werden, als für den es ursprünglich konzipiert wurde³.

Ressourcen Eine Ressource ist in der Rest Architektur eine von der Zeit unabhängige Anlaufstelle, welche auf Informationseinheiten verweisen kann aber nicht muss. Um die Ressourcen voneinander zu unterscheiden, muss deren Semantik beständig sein. Durch diese Abstraktion gewinnt man folgende Vorteile:

- Allgemeingültigkeit
- Späte Bindung
- Abstrahierende Referenzierung

Rest baut darauf, dass die Verbindungen auf eine Ressource nicht zentral gespeichert werden, sondern durch den *Ressource Identifier* auffindbar werden. Damit fördert diese Architektur die Dezentralisierung von Ressourcen.

Zusammenfassung Das REST-Architekturmodell wurde für verteilte Informationssysteme konzipiert. Sie beschränkt sich nicht nur auf ein bestimmtes Kommunikationsprotokoll. Die Einschränkung hierbei ist die, dass die Kommunikationsprotokolle bei den Interaktionen auf der Semantik eines REST-Connectors basieren müssen. Die bei REST im Vordergrund stehende zustandslose Kommunikation führt ebenfalls dazu, dass die Schnittstellen zwischen den Komponenten sehr eingeschränkt werden. Abstrakt gesehen betrachtet REST Anwendungen als ein Ganzes, welches sich aus Informationen und Steuerungsmöglichkeiten zur Veränderung zusammensetzt. Durch das hier eingeführte Ressourcenkonzept wird eine Möglichkeit geschaffen Informationen dezentral zu halten und diese wiederauffindbar zu machen. Ein weiteres Nebenprodukt, was bei der Entwurfszeit als Nachteil identifiziert wurde, ist der Kontrollverlust über den Zustand, die Semantik sowie den Verwendungskontext einer Ressource. Dieses wird heutzutage als Vorteil erachtet und begünstigt hiermit das Mashup-Konzept. Durch diesen wird die freie Entfaltung des Webs unterstützt, sowie die auf dieser Plattform laufenden Anwendungen.

³Damals wurde von dem Autor diese Tatsache als Nachteil erachtet, heut wird diese aus Sicht des Mashup-Konzeptes als Vorteil angesehen. Man unterstützt so die freie Entfaltung des Webs sowie auf dieser Plattform laufenden Anwendungen. Was früher Nachteil war ist heute zum Vorteil mutiert.

Asynchrone Kommunikation mittels XMLHttpRequest

Gewöhnliche Webanwendungen basieren auf dem Request/Response Modell. Damit ist gemeint, dass die Clientanwendung nach dem Absetzen einer Anfrage auf die Antwort vom Server wartet und quasi nichts tut. Diese Tatsache führt zu zwei Nebenerscheinungen. Bei lang andauernden Berechnungen oder Überlastungen des Servers entstehen auf der Clientseite lange Wartezeiten. Die zweite noch gravierendere Erscheinung ist die Unterbrechung des Prozessflusses. Dies tritt auf, wenn der Server keine Antwort liefert, was unter Umständen zum Abbruch der Clientanwendung führen kann. Dieses wiederum hat zur Folge, dass der Benutzer die Prozesskette von vorne beginnen muss ([Mauerer, 2006](#)).

Um der oben geschilderten Problematik aus dem Wege zu gehen und beliebige Daten über das Http Protokoll senden zu können hatte Microsoft den *XMLHttpRequest* entwickelt und als ActiveX Komponente in den Internet Explorer integriert. Hierdurch kann eine Transferierung von Daten aus JavaScript oder vergleichbaren Skriptsprachen erfolgen. Der XMLHttpRequest ermöglicht, vergleichbar mit REST, mehrere Methoden der Anfragestellung, wie z. B. GET, POST und PUT. Durch XMLHttpRequest zurückgegebene XML Daten können in zwei Formaten repräsentiert werden. Dabei kann es sich um eine gewöhnliche textuelle Darstellung handeln, oder um eine DOM Baumstruktur.

Der XMLHttpRequest kann ohne direkte Aufforderung eine Datenverbindung zwischen dem Server und dem Client aufbauen und die notwendigen Daten transferieren. Für den Benutzer laufen damit verbundene Aufgaben transparent im Hintergrund ab. Durch dieses Verhalten bekommt der Benutzer einen Eindruck vermittelt, bei dem sich die Anwendung höchst dynamisch verhält, wie dies bei Desktopanwendungen zu beobachten ist. Neben der hohen Interaktivität der Anwendung besitzt diese einen weiteren wesentlichen Vorteil gegenüber einer Desktopanwendung. Eine so realisierte Anwendung ist mit dem Web verbunden und kann sich die dadurch entstehenden neuen Möglichkeiten zu Nutze machen ([McLaughlin, 2005](#)).

Durch diese Komponente wurde endlich ein Weg gefunden wie man die Darstellungslogik und die Anwendungslogik voneinander in Webanwendungen trennen kann. Hierdurch ist es nun möglich die wesentliche Anwendungslogik auf dem Server zu betreiben. Diese wird asynchron durch den XMLHttpRequest aufgerufen und die Ergebnisse dem Benutzer ohne visuelle Unterbrechung präsentiert ([McLellan, 2005](#)). Der wesentliche Vorteil hierbei ist, dass bei dem XMLHttpRequest nur relevante Daten angefordert werden. Diese Gegebenheit führt folglich nicht nur zur Auslastung des Servers, sondern auch zur effizienten Nutzung des Kommunikationsweges. Durch den asynchronen Charakter dieser Technik können zusätzlich die in der Anwendung enthaltenen Abläufe parallelisiert werden. Die genannten Merkmale dieser Technologie sind gleichzeitig auch die Merkmale des Web 2.0 Ansatzes, wie später im Verlauf dieser Arbeit gezeigt wird. Somit kann XMLHttpRequest als die Ausgangstechnologie für das Web 2.0 betrachtet werden.

JavaScript

JavaScript ist für das dynamische Verhalten der in einer Webseite enthaltenen Elemente zuständig. Diese plattformunabhängige Interpretersprache wurde von Beginn an objektorientiert gestaltet. JavaScript kommt auf der Clientseite vorwiegend in Webbrowsern zum Einsatz, welche einen Interpreter für diese Skriptsprache integriert haben. Durch das Interpreterkonzept können JavaScript Programme sofort ausgeführt werden. Eine vorherige Kompilation und Linken sind nicht erforderlich.

JavaScript wird in Verbindung mit HTML verwendet und ist in den entsprechenden HTML Dateien enthalten. Des Weiteren können JavaScript-Programme ebenfalls extern in dafür dedizierten Dateien abgespeichert werden, welche wiederum durch HTML Links oder im Header einer HTML Datei referenziert werden (Maier, 2006).

Durch den Einsatz von JavaScript wird eine saubere Trennung zwischen Darstellung und den Daten selber hergestellt. Das HTML Markup dient der Darstellung, wobei JavaScript den vom Server übermittelten Daten einen dynamischen Charakter verleiht. Durch diese Trennung wird es möglich abhängig von den Browsereigenschaften ein entsprechend angepasstes JavaScript-Programm zur Verfügung zu stellen.

In Zusammenhang mit JavaScript taucht das *Document Object Model* (DOM) auf. Mit Hilfe von DOM wird eine Repräsentationsmöglichkeit für XML Date zur Verfügung gestellt. Diese in den DOM überführten Daten können mit wesentlich effizienteren Methoden verarbeitet werden.

Der durch JavaScript eingeführte interaktive Charakter basiert auf einem darunterliegenden Ereignismechanismus. Durch diesen wird das in den Webseiten enthaltene Routing aktiviert indem die vom Browser zur Verfügung gestellten Ereignisse entsprechend interpretiert und abgearbeitet werden.

Basierend auf DOM und dem Ereignismechanismus, welcher die Ereignisse, die der Benutzer durch seine Eingaben auslöst, abfängt und der Verarbeitung zur Verfügung stellt, wird das statische HTML dynamisch. Mit Hilfe der CSS Technologie wird dem Ganzen noch ein letzter Schliff gegeben. CSS stellt eine Möglichkeit der vereinheitlichten Darstellung von Daten in den Webanwendungen zur Verfügung.

Wie bereits erwähnt werden JavaScript-Programme durch das Laden der HTML Seiten auf dem Client verfügbar und damit auch einsehbar gemacht. Der Benutzer kann also die darin enthaltene Logik und damit möglicherweise auch die Programmsemantik entnehmen. Sicherlich existieren hierbei Nachteile, wobei allerdings der Vorteil aus Sicht der sich hierdurch eröffnenden Möglichkeiten überwiegt. Damit ist gemeint, dass der Quellcode einer Anwendung verstanden werden kann, was gegebenenfalls zu einer Verbesserung der Nutzung des Dienstes führen wird.

Ajax

Ajax trat das erste Mal im Jahr 2005 in Erscheinung. In dem Artikel „*AJAX: A new Approach to Web Applications*“ hatte Jesse James Garrett ([Garrett, 2005](#)) Ajax vorgestellt. Die Kernidee war eine Vereinfachung der Gestaltung dynamischer Webanwendungen.

Ajax umfasst Ansätze und die dazu notwendigen Werkzeuge, mittels deren Daten asynchron, zwischen dem Client und Server ausgetauscht werden können. Außer JavaScript und XML gehören hierbei weitere Technologien wie DHTML, das zuvor besprochene XMLHttpRequest und damit verbundene DOM. Diese Techniken wurden bereits bei vielen Webanwendungen eingesetzt, noch bevor Ajax bekannt war.

Da Ajax u.a. auf den beiden zuvor genannten Technologien basiert, erbt dieser Ansatz natürlich auch deren Vorteile. Somit kommt durch den Einsatz von XMLHttpRequest auch der asynchrone Charakter zum Vorschein. Zudem können mit JavaScript und DHTML Benutzeroberflächen vergleichbar zu Desktopanwendungen aufgebaut werden. DOM formalisiert die Datenrepräsentation und hilft bei deren Verarbeitung. Darüber hinaus, durch die Einführung des Ereignismechanismus, kann eine für den Benutzer transparente Client/Server Interaktion stattfinden.

Außerdem bietet Ajax eine Laufzeitumgebung, welche alle Benutzerabfragen abarbeitet und den Server nur bei Bedarf kontaktiert. Des Weiteren realisiert Ajax einen Ereignismechanismus durch den Serverantworten an das notwendige Objekt weitergeleitet werden, ohne dass dabei der Prozessfluss der ganzen Anwendung gestört wird. Durch diesen Mechanismus kann jedes Objekt unabhängig seine Anfragen an den Server stellen und erst beim Eintreffen der Antworten darauf entsprechend reagieren.

Ajax basiert auf alt bewährten Technologien. Hierdurch kommt ein großer, wenn nicht sogar entscheidender Vorteil zum Vorschein. Durch diesen Umstand benötigen die neuartigen Anwendungen keine neuen Laufzeitumgebungen, Browser oder Plug-Ins. Sie bauen stattdessen auf bestehenden Technologien auf, welche im heutigen Web zu finden sind.

Zusammengefasst ist Ajax keine neue Technologie. Vielmehr ist es ein Sammelbegriff für mehrere Techniken und damit verbundene Heuristiken, durch die eine effiziente, webbasierte Anwendung entwickelt werden kann.

RSS/Atom Feed

RSS/Atom Feed ist eine konkrete Variante eines bereits bewährten Prinzips, welches als Alert-Dienst bekannt ist. Das dahinter stehende Modell setzt zwei Kommunikationspartner voraus, den Publisher und den Subscriber. In diesem Modell wird der Subscriber mit Informationen beliefert sobald diese vom Publisher veröffentlicht wurden.

Durch RSS und Atom Feed wird eine Möglichkeit bereitgestellt durch die andere Webressourcen bzw. ihr Zustand überwacht werden kann. Im Zusammenhang mit *NewsFeeds* taucht ebenso der Begriff der *Aggregatoren* auf. Dahinter verbergen sich kleine Programme (RSS-Parser), welche die erkannte Änderungen von Webressourcen in eine weitere integrieren (aggregieren) können. Damit kann eine Webressource entstehen die vollständig auf anderen Ressourcen basiert. Darüber hinaus können diese Programme die Daten in ein Format überführen, welches dann auf verschiedenartigen Geräten dargestellt werden kann. Die Aktualisierung der Daten erfolgt automatisch. Neben der Integration von *NewsFeeds* in eine bestehende Webseite können diese auch über ein explizit dafür dediziertes Programm in einer lesbaren Form zur Verfügung gestellt werden, einen sogenannten *Feed-Reader*.

Der hinter RSS/Atom Feed stehende Kerngedanke ist zum einen die Abstraktion von den eigentlichen Informationen, sodass diese darstellungsneutral sind. Um diese rohen Daten dann auf der Empfängerseite eindeutig interpretieren zu können, erfordert es noch die Etablierung eines Standards für die Datenrepräsentation. Diese beiden Aspekte realisiert die RSS/Atom Feed Technik. Um eine neutrale Darstellung von Daten zu erreichen, welche dann auch einfach interpretierbar ist, wird heute XML verwendet. Bei den Ressourcen, die hierdurch überwacht werden können, muss es sich nicht notwendigerweise nur um textbasierte Dienste handeln. Zunehmend werden heute Ressourcen wie Bilder, Musik und Videos abonniert.

Die Feed-Technik ist eine Art Verzweigungsmöglichkeit auf Webressourcen. Gegenüber einem konventionellen Link ist sie aber viel mächtiger, da eine Überwachungsmöglichkeit für Ressourcen gegeben wird. Damit sind Voraussetzungen geschaffen worden um eine schnelle Informationsverbreitung zu gewährleisten. Durch den Mashup Charakter von *NewsFeeds* entsteht bei der Verbreitung einer Information das Schneeballprinzip. Da die Benutzer nur bestimmte Informationen und nicht alle aggregieren, entsteht hierbei eine Art Soziale Klassifikation von Informationen. Aus diesem Grund wird der RSS-Feed nicht umsonst zu einer der ersten Anwendungen des Semantischen Webs gezählt.

Web Services

Die Schnittstellentechnik *Web Services* ist eine Art Middleware die zum einen die von Anwendungen zur Verfügung gestellten Schnittstellen, bzw. die dahinter stehende Anwendungslogik, auf einer plattformunabhängigen Weise zugreifbar macht. Zum anderen bietet sie die Möglichkeit dienstorientierte Architekturen zu realisieren und damit die nächste Evolutionsstufe Verteilter Systeme zu erreichen. SOAP, WSDL und UDDI sind die hierfür zum Einsatz kommenden Basistechnologien. Zudem abstrahiert die *Web Services* Technik vom Kommunikationsprotokoll, obwohl konkret vermehrt das *Http*-Protokoll als Transportmechanismus zum Einsatz kommt. Um die mit dieser Technik verfügbar gemachten

Dienste nutzen zu können, ist eine Laufzeitumgebung notwendig, genau wie das in anderen Middleware-Konzepten ebenfalls zu beachten ist. Diese übernimmt alle kommunikationsrelevanten Aspekte und ist somit unverzichtbar.

Mit Hilfe des *Simple Object Access Protocol* (SOAP) wird die eigentliche Nachricht in einer plattformneutralen Form übertragen und ist somit von jeder Plattform interpretierbar. Dieses ist zum Teil der XML-basierten Syntax von SOAP zuzuschreiben, durch welche die Daten in strukturierter Form übertragen werden können. Die Übertragung von SOAP Nachrichten erfolgt über das http-Transportprotokoll.

Ein Web Service kann intern in einer Anwendung verwendet, allerdings auch nach außen über das Internet veröffentlicht werden. Damit kann dieser von einer beliebigen Anzahl von Anwendungen verwendet werden. Des Weiteren wird ein Dienst durch eine standardisierte Schnittstelle bereitgestellt, was seine Nutzung durch heterogene Plattformen ermöglicht (Brown und Haas, 2003).

Web Services schaffen neue Strukturierungsmöglichkeiten für verteilte Systeme. Damit stellt die Web Service Technik eine Möglichkeit dar eine Service Oriented Architecture (SOA) zu realisieren. Letztere beschreibt ein Strukturmodell für verteilte Anwendungen. Die primären Merkmale der SOA sind Dienste, welche verwendet werden um große Anwendungen in kleine separate Module aufzuteilen. Durch Kombination einzelner Module lassen sich größere Anwendungen zusammenstellen. Ein weiteres Merkmal von SOA ist, dass diese sich aus drei primären Kommunikationsteilnehmern zusammensetzt. Es wird zwischen Dienstbereiter, Dienstanwender und Dienstregistrierungsinstanz unterschieden (MomentumSI, 2006).

Web Services bieten eine Möglichkeit an, eine plattformneutrale Schnittstelle für eine Softwarekomponente zu definieren und diese im Web zur Verfügung zu stellen. Damit ist nicht nur die Kommunikation sondern ebenfalls die neutrale Repräsentation von Daten gemeint. Das den Web Services zugrunde liegende Kommunikationsprotokoll erlaubt eine synchrone sowie eine asynchrone Kommunikation und unterstützt damit die lose Kopplung von Diensten.

Um die Schnittstellen plattformneutral beschreiben zu können wird hierzu ein XML-basiertes WSDL Dokument verwendet. Über das *Universal Description, Discovery and Interaction* (UDDI) werden dann Dienste im Web auffindbar gemacht.

Durch die Web Service Technik zur Verfügung gestellte Anwendungslogik kann auf weitere Dienste zugegriffen werden. Dadurch können nicht nur einfache, sondern auch komplexe Dienste entstehen.

Die Offenheit welche die Web Service Technik anbietet erlaubt es alte, bereits bewährte Softwarekomponenten als Dienste in das SOA Modell zu integrieren. Diese Eigenschaft dieser Technologie macht sie sehr attraktiv, da hierdurch im großen Maße die Wiederverwendung

unterstützt wird. Die Offenheit erlaubt aber auch wie in allen Web 2.0 Technologien Webressourcen sehr einfach zu nutzen und diese miteinander zu kombinieren.

Peer-to-Peer

Systemtechnisch existieren drei Systemarchitekturen: Mainframes, Client/Server und Peer-to-Peer (P2P). Ein System wird als P2P-System bezeichnet, wenn seine Netzarchitektur keinen expliziten Server vorsieht. In einem P2P Netz sieht die Rollenverteilung vor, dass jeder Teilnehmer die in diesem Netz zur Verfügung gestellten Dienste nutzen kann sowie gleichzeitig auch welche bereitstellen kann. Alle Netzteilnehmer sind dabei gleichberechtigt. Diese Verteilung der Rollen unter Beachtung der Gleichberechtigung erlaubt den einzelnen Knoten bei Bedarf einzelne Infrastrukturaufgaben zu übernehmen und somit das Netz aufrecht zu erhalten (Minar u. a., 2001).

Betrachtet man ein P2P System genauer, so stößt man auf mehrere, so ein System auszeichnende Merkmale. Im Vordergrund steht, natürlich wie nicht anders zu erwarten war, die Datenverteilung. So wird jede Information, die in diesem System gespeichert ist, einer Dezentralisierung unterzogen. Mit anderen Worten residiert eine Information nicht auf einem zentralen Knoten, sondern verteilt auf mehreren. Die Dezentralisierung greift aber nicht nur auf der Datenhaltungsebene ein. Wie zuvor bereits angedeutet ist jeder Teilnehmer gleichberechtigt, d.h. er kann Systemdienste nutzen aber auch bereitstellen. Damit ist jeder Knoten dazu prädestiniert auch Aufgaben zu übernehmen, welche die Infrastruktur des Systems betreffen. Mit anderen Worten findet hier eine Dezentralisierung der infrastrukturelevanten Aufgaben vorgenommen. Somit kann ein Knoten jeder Zeit, aus Sicht der Client/Server Architektur, ein Client und ein Server sein. Diese Rollenverteilung wird aber natürlich unter Beachtung der Autonomieität jedes einzelnen Teilnehmers durchgeführt, was ein weiteres Merkmal eines P2P Systems darstellt. Diese Rollenverteilung zieht auch eine Konsequenz nach sich, eine dynamische Infrastruktur. Damit ist gemeint, dass das Verhalten des P2P Systems nicht vorherbestimmt ist, da dieses sich höchst dynamisch verhält. Diese Dynamik ist in einen solchem System notwendig, da man jeden Teilnehmer in das Netz einbezieht, ungeachtet seiner eigentlichen Systemverlässlichkeit. Damit ist gemeint, dass ein P2P System in seiner Konzeption davon ausgeht, dass jeder Systemteilnehmer jederzeit ausfallen kann. Es geht also pessimistisch in der Hinsicht auf die systemtechnische Verlässlichkeit von Knoten vor, was eine weitere Eigenschaft eines P2P Systems kennzeichnet.

P2P-Systeme kann man nach deren Architekturart in drei Kategorien unterteilen. Die ersten P2P-Systeme haben eine zentrale Einheit, welche die notwendigen Verwaltungsaufgaben übernehmen um das Netz aufrecht zu erhalten. In gleicher Zeit sind P2P-Systeme aufgetaucht, welche vollständig auf eine zentrale Steuerungseinheit verzichtet haben. Beide Architekturen haben ihre Vorzüge, welche man dann in den sog. P2P-Systemen der zweiten

Generation einfließen ließ. So entstanden hybride Systeme die zur Laufzeit dynamisch die Verwaltungsaufgaben an Knoten weiter delegieren.

Der Hauptvorteil bei diesen Systemen ist, dass diese die Aufgaben auf einzelne Knoten verteilen. Durch die Aufgabenteilung steht natürlich gesamt betrachtet viel größere Rechenleistung, Speicherplatz und Bandbreite zur Verfügung. Durch dieses Merkmal verhalten sich P2P Systeme bei wachsender Teilnehmerzahl entgegen dem Verhalten der Client/Server Systeme. Die P2P-Systeme legen an Leistung zu während Client/Server diese verlieren.

P2P-Systeme haben die Grundannahme, dass jeder Knoten jederzeit ausfallen kann. Damit wird das System darauf vorbereitet beim Auftreten dieser Situation entsprechend reagieren und geeignete Auswege finden zu können. Durch die Übernahme der Verwaltungsaufgaben durch die einzelnen Teilnehmer, kann bei einem Ausfall sofort einer der anderen Knoten als Ersatz einspringen. Damit ist dieses System gegenüber einer Client/Server Architektur und dem damit verbundenen *Single Point Of Failure* wesentlich robuster.

Wenn man den Nutzen eines P2P-Systems betrachtet stößt man auf einige Einsatzgebiete, in denen diese Verwendung finden. Natürlich herrscht heute eine negative Meinung über deren Einsatz, da diese heute hauptsächlich zur Verteilung geschützter Ressourcen genutzt werden. Auf der anderen Seite werden aber diese Systeme zunehmend von Unternehmen zur Distribution ihrer Produkte verwendet. Durch die Eigenschaften eines P2P-Systems entlasten diese Unternehmen ihre Bandbreite und verteilen diese auf ihre Kunden. Neue Ansätze gehen in Richtung der Verteilung von Echtzeitdaten. Hierbei eröffnen sich neue Möglichkeiten in der Distribution von Audio und Video bei den Teilnehmern, welche diese Medien dazu nutzen sich miteinander zu verständigen. Eine ausführliche Betrachtung der P2P Technologien findet man in ([Minar u. a., 2001](#)).

Zusammenfassung

An dieser Stelle möchte der Autor rückblickend die Web 2.0 Technologien betrachten und technische Aspekte diskutieren. Dabei sollen die durch diese hervorgerufene architektonische Änderungen betont werden und die damit verbundenen Konsequenzen genannt werden.

Von Fatclient zum Thinclient In ([Tanenbaum und van Steen, 2002](#)) wird eine detaillierte Betrachtung verschiedener Ausprägungen bzw. die Gestaltung der Client/Server Architekturen vorgenommen. Hierbei werden drei logische Schichten identifiziert aus denen das Client/Server Modell besteht: Die Darstellungsschicht, die Verarbeitungsschicht und die Datenschicht. Des Weiteren können diese in der physikalischen Verteilung variieren und folglich verschiedene Ausprägungen des Client/Server Modells darstellen. Die dabei entstehenden

Ausprägungsmöglichkeiten werden in Abbildung 2.3 dargestellt. Des Weiteren wird zwischen einer vertikalen und horizontalen Verteilung unterschieden.

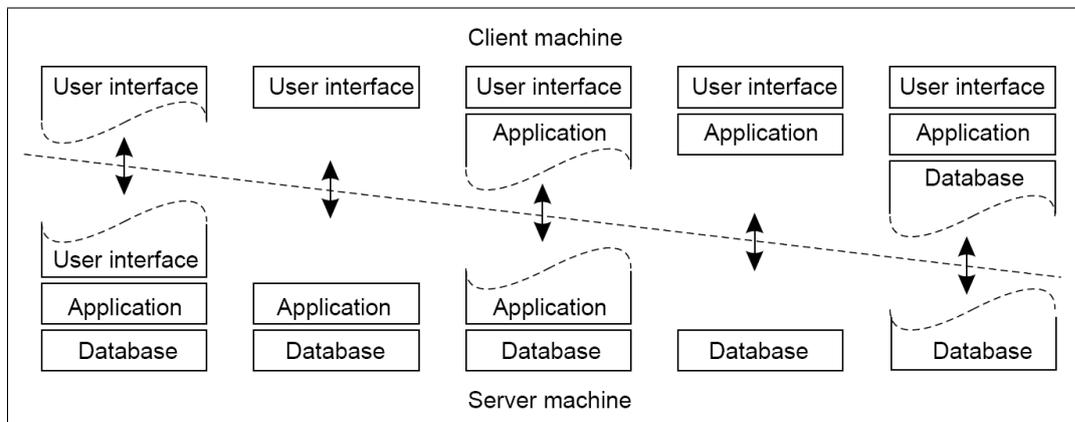


Abbildung 2.3: Verschieden Verteilungsausprägungen des Client/Server Modells nach Tanenbaum (Hashimi und Steffan, 2005)

Alle bis jetzt genannten Web 2.0 Technologien tragen mehr oder weniger zur Realisierung des Client/Server Modells bei. Ajax, und damit JavaScript und der XMLHttpRequest, sind Technologien welche auf der Clientseite residieren und damit für die Realisierung einer web-basierten, desktopähnlichen Anwendung verwendet werden können. Dabei wird die Steuerung der Benutzerschnittstelle und der Datenversorgung auf den Client verlagert und zielen vollständig auf die Belange der Darstellungsschicht und der horizontalen Verteilung ab.

Wie in Abbildung 2.4 zu sehen ist, kann Ajax als Bindeglied zwischen der Anwendungs- und Darstellungsschicht fungieren. Dies muss allerdings nicht so sein. Aus softwaretechnischer Sicht ist aber eine Trennung zwischen diesen beiden Schichten für die spätere flexible Nutzung von Anwendungen essentiell und wird hier als gefordert angesehen.

Atom/RSS Feed und WebServices bieten eine standardisierte und leichtgewichtige Schnittstelle zur Informationsverteilung an. Damit sind diese als Middleware zu deuten und sind als Bindeglied zwischen der Anwendungs- und Darstellungsschicht zu sehen.

P2P zielt auf die horizontale Verteilung ab. Dabei findet weniger eine Verteilung der drei genannten Schichten statt. Diese sind auf jedem Client vorhanden. Es geht vielmehr um die horizontale Verteilung von Ressourcen.

Kopplungsgrad bei der Kommunikation Durch die Einführung der losen Kopplung in einer Anwendungsarchitektur wird diese in ihrem Zusammenhalt soweit flexibilisiert, dass deren Erweiterung jederzeit erfolgen kann, ohne das gesamte Konzept neu überdenken

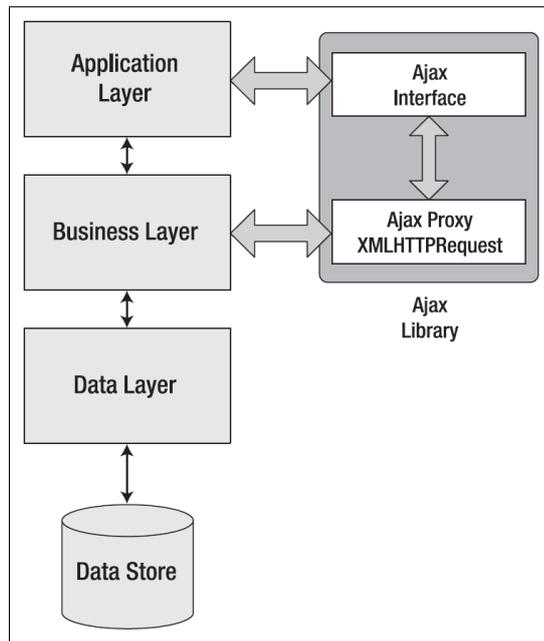


Abbildung 2.4: Drei Schichten Architektur mit Ajax nach (Woolstone, 2006)

zu müssen. Damit können neue Einheiten (Dienste) realisiert werden und jederzeit zu der eigentlichen Anwendung dazu geschaltet werden. Der Rest bleibt dabei unberührt.

Ist eine Anwendungsarchitektur lose gekoppelt gewinnt die Anwendung ebenfalls an Flexibilität in ihrem Laufzeitverhalten. Durch lose miteinander verbundene Dienste kann die Anwendung einfacher von Rechner zu Rechner bewegt werden und damit bei Bedarf neue Ressourcen beanspruchen. Damit werden die Aspekte der Skalierbarkeit leichter handhabbar. Folglich macht lose Kopplung ein System anpassungsfähig, zuverlässiger und flexibler.

Alle genannten Techniken kommunizieren auf der Anwendungsschicht über das Http Kommunikationsprotokoll. Einer der Vorteile der sich hierbei ergibt ist, dass dieser Kommunikationskanal in fast allen Netzwerkarchitekturen offen ist. Folglich kann durch diesen Umstand eine Kommunikation ungehindert stattfinden.

Durch die Web Service Schnittstellentechnik wird Anwendungslogik in Form von Diensten verfügbar gemacht, die neutral zur Plattform, Daten und Kommunikationsprotokoll ist. Zudem unterstützt diese außer der synchronen auch die asynchrone Kommunikation. Damit ist sie als eine Basis zur Gestaltung von dienstorientierten Anwendungen zu sehen, welche Dienste in Anwendungsarchitekturen kopplungsfrei integriert. Ajax mit dem darin enthaltenen XMLHttpRequest kann dann auf die so verfügbar gemachten offenen Dienste zugreifen und damit die durch Web Services gewonnenen Vorteile übernehmen.

Sichtet man die Literatur so entstanden viele weitere Middlewarekonzepte mit dem einen Ziel immer mehr die Bindung zwischen den darin enthaltenen Komponenten zu nivellieren. Inzwischen stellt Sun die *Jini* (Jini.org, 2006) Technologie in der zweiten Version bereit und damit auch die *JavaSpaces* ([Microsystems](#), 2003). *JavaSpaces* ist eine Möglichkeit zwischen verteilten Anwendungen Objekte auszutauschen. Dabei wird durch dieses Konzept eine Art *Virtual Shared Memory* zur Verfügung gestellt. Durch diese Technik wird ein Server praktisch überflüssig da die miteinander kommunizierenden heterogenen Anwendungen selbst ihr Verhalten koordinieren. Ein vergleichbares Konzept zur Entkopplung von verteilten Anwendungen wird ebenfalls durch *Java Messaging Service* (JMS) bereitgestellt. Zusammengefasst lässt sich aber sagen, dass die Forderung nach entkoppelten Systemen überall in der Entwicklung neuer Middlewarekonzepte zu erkennen ist. Web 2.0 Technologien stellen also hier keine Ausnahme dar, sondern folgen der aktuellen Entwicklung.

Plattformunabhängigkeit Die vielen Vorteile eines verteilten Systems sind in der Vergangenheit gründlich untersucht worden. Eine der wichtigsten ist die Plattformunabhängigkeit. Durch diese Eigenschaft wird eine Software mithilfe einer zusätzlichen Abstraktionsschicht nach Außen verfügbar gemacht. Diese wurde notwendig da mehr als nur eine Plattform, auf der solche Dienste residierten, existierten.

Die zwei heute wichtigsten Plattformen sind J2EE und .NET. Diese haben lange Zeit nebeneinander existiert aber nicht miteinander kooperiert. Durch die Integration der Web Service Technik auf beiden Plattformen hatte sich dieser Zustand verändert. Die heute verfügbaren Dienste werden auf beiden Plattformen entwickelt und anderen Diensten verfügbar gemacht. Damit können J2EE Dienste .NET Dienste nutzen und umgekehrt. Dadurch wurde eine Brücke zwischen zwei großen Welten geschlagen durch die bis dahin heterogene Anwendungen zusammengeführt werden.

Auch Ajax geht hierbei einen ähnlichen Weg. Es abstrahiert von den zugrunde liegenden Plattformen und greift auf Dienste zu, deren Schnittstellen Plattform neutral offengelegt sind. Zusätzlich hierzu abstrahiert Ajax bei der Realisierung von webbasierten Anwendungen vom Betriebssystem auf dem diese dann ausgeführt wird⁴. Dieses ist dadurch möglich, da die Anwendung in einem Webbrowser läuft. Da diese auf jedem System verfügbar und weitgehend alle notwendigen Techniken anbietet ist das Betriebssystem, auf welchem diese dann im Endeffekt laufen, irrelevant. Das Manko der Verfügbarkeit von Laufzeitumgebungen aller Plattformen für alle Betriebssysteme wurde durch die Web 2.0 Techniken behoben indem das Web selber zur einer Plattform wurde und wie in ([Weiss](#), 2005) zum WebOS mutierte.

⁴Dieses ist zumindest in der Theorie möglich. In der Praxis gibt es einige wichtige Aspekte, welche beachtet werden müssen, damit die verfügbare Implementierung des XMLHttpRequest richtig verwendet werden kann.

Die oben beschriebenen Technologien sind ein Weg zur vollständigen Migration der Informationstechnologie.

Vorzüge von Web 2.0 Technologien In ([Woolstone, 2006](#)), ([Interakt, 2005a](#)) und ([Interakt, 2005b](#)) werden verschiedene Vorteile von Ajax und damit auch der asynchronen Kommunikation identifiziert.

Eine effektive Ausnutzung der verfügbaren Leistung (Parallelität), um die für die Anwendung notwendigen Daten vom Server zu laden ist das Anfordern der gesamten Seite nicht erforderlich. Diese Gegebenheit zieht zwei erhebliche Vorteile nach sich. Zum einen wird bei der Anforderung der Daten durch Ajax die Benutzerschnittstelle weiterhin präsent und benutzbar. Durch die hierbei eingeführte Parallelität wird die Bandbreite effektiver genutzt.

Aus den eben genannten Konsequenzen, welche sich durch die Einführung von asynchroner Kommunikation ergeben resultiert eine weitere: Die gesteigerte Benutzerfreundlichkeit. Der Grund hierfür liegt in der zeitlichen Verfügbarkeit der notwendigen Daten. Diese werden nicht erst durch die explizite Aufforderung des Benutzers angefordert. Vielmehr erfolgt hier eine parallelisierte, präventive, in Hintergrund laufende Bereitstellung von Daten. Damit verkürzen sich die Wartezeiten bei der Benutzung der Anwendung. Außerdem wird durch das asynchrone Verhalten die Webanwendung weiterhin ansprechbar und nicht blockiert. Der Benutzer bekommt durch diese beiden Merkmale einen Eindruck von Kontinuität.

Ebenfalls unter der Benutzerfreundlichkeit einzuordnen ist die Reaktion bei Fehlerzuständen. Eine mit Ajax realisierte Benutzeroberfläche hat die Eigenschaft bei einer fehlerhaften Datenversorgung von Teilen der einen Anwendung nicht den Rest der Benutzerschnittstelle zu blockieren. Damit kann der Benutzer einen Teil der Anwendung weiterhin verwenden während eine erneute Anfrage an den Server gesendet werden kann.

Wie weiter oben bereit geschildert basiert die Ajax Technologie vollständig auf lang bewährten und seit längerem verwendeten Technologien. Diese Tatsache erlaubt eine einfache Nutzung dieser Technologie und was wichtiger ist, so realisierte Anwendungen können jeden Benutzer erreichen. Dies ist deshalb möglich weil für die Ausführung dieser Anwendung keine zusätzliche Software installiert werden muss. Natürlich kann z.B. die Unterstützung von JavaScript deaktiviert werden. Nach Aussage von W3C ([W3Schools, 2006](#)) tun dies aber nur 10% der Benutzer.

Durch die Realisierung der bis heute nur für den Desktopbereich existierenden Anwendungen als Webanwendungen werden immer mehr Werkzeuge überall verfügbar. Dies steigert in gleichem Maße die Mobilität des Nutzers, da dieser nur noch über einen Browser sowie eine Internetverbindung verfügen muss. Eine der wichtigsten Vorteile aber bei den hier vorgestellten Technologien ist die Ermöglichung der Zusammenarbeit mit mehreren Benutzern an einer Ressource und somit der Förderung der kollaborativen Arbeitsweise.

Die Funktionsweise der Benutzeroberfläche wird durch JavaScript realisiert. Die Implementierung der Benutzeroberfläche wird also beim Aufruf der Webanwendung geladen. Dieser Umstand bringt mit sich einen weiteren Vorteil, einen durchgängigen Aktualisierungsmechanismus. Denn wenn die Implementierung der Benutzeroberfläche sich ändert wird diese nach der Bereitstellung auf dem Server sofort bei der nächsten Webanfrage auf dem jeweiligen Client geladen. Damit wird jederzeit eine Verbesserung bzw. Beseitigung von Fehlern möglich, ohne dass die Benutzer damit konfrontiert werden müssen komplizierte Installationsaufgaben durchführen zu müssen.

Schwächen von Web 2.0 Technologien Natürlich bietet keine Technologie nur Vorteile sondern bringt auch Nachteile mit sich. In ([Woolstone, 2006](#)), ([W3Schools, 2006](#)) und ([Bosworth, 2005](#)) werden einige dieser Nachteile genannt. Im Folgenden sollen einige davon verdeutlicht werden.

Die Funktionsweise des Rückwärts-Buttons welcher in jedem Webbrowser integriert ist wird durch Ajax gestört beziehungsweise ganz ausgeschaltet. Dieses Fehlverhalten entsteht dadurch, dass Ajax neue Seiteninhalte generiert ohne dabei eine neue Seite zu erzeugen. Was oben als wichtiger Vorteil erkannt wurde, wird hier zu einem Problemfall. Durch diesen Umstand wird der Benutzer bei der Verwendung des Browsers irreführt, da dieser diesen nicht so benutzen kann, wie er es gewohnt ist. Damit wird gleichzeitig die Benutzerfreundlichkeit der Webanwendung verschlechtert.

Ein weiteres in dem gleichen Kontext entstehendes Fehlverhalten ist bei den URLs der Webseiten zu erkennen. Da wie zuvor angedeutet keine neue Seite erzeugt wird wenn diese ihren Zustand ändert, wird folglich auch keine neue URL erzeugt. Damit wird es gleichzeitig unmöglich eine Referenz auf den aktuellen Zustand der Seite zu bekommen. Konsequenterweise führt dies dazu, dass einer der am häufigsten verwendeten Funktionen des Webbrowsers, das sog. Bookmarks, fehlt. Der Benutzer kann also den Zustand einer Webseite nicht mehr durch eine URL bei Bedarf reproduzieren.

Verschiedene Implementierungen des XMLHttpRequest führen dazu, dass die Webanwendung zu Beginn immer prüfen muss, welche der Implementierungen im aktuellen Browser zur Verfügung stehen. Darüber hinaus unterscheiden sich die einzelnen Browser in ihrer Realisierung. Dieses führt dazu, dass diese neue Art von Webanwendungen mit den verschiedenen Browsern auf ihre Tauglichkeit verifiziert werden müssen. Dieses ist nicht nur dann notwendig wenn eine neue Webanwendung entsteht, sondern muss möglicherweise bei Ausweitung der Zugänglichkeit auf neuen Browsern durchgeführt werden.

Das oben als Vorteil identifizierte Merkmal der Aktualisierung der Programmlogik durch erneute Anforderung der Webseite wird hier gleichzeitig als Nachteil identifiziert. Durch eine erneute Anfrage auf dem Client geladene neue Version der Anwendung kann in sich so

gravierende Unterschiede bergen, dass diese nicht mehr von dem Benutzer verwendet werden kann und für diesen fremd erscheint. Dieses hat einen eher negativen Einfluss auf die Benutzerfreundlichkeit der zugrunde liegenden Anwendung.

Bei manchen Seiten ist es äußerst wichtig, dass auf Barrierefreiheit geachtet wird. Eine barrierefreie Webseite hat aber auch die Eigenschaft, dass diese ebenfalls bei ausgeschalteter JavaScript-Unterstützung den Inhalt immer noch ordnungsgemäß wiedergeben soll. Hierbei entsteht damit ein Konflikt mit der Ajax Technologie, da diese vollständig auf JavaScript aufsetzt.

Sicherheit ist in jeder Hinsicht eine der wichtigen Anforderungen an eine Anwendung. Mit Hilfe von JavaScript wird Steuerungslogik implementiert, welche in erster Linie das Verhalten der Benutzerschnittstelle definiert. Man muss sich im Klaren sein, dass die in JavaScript enthaltene Logik nach Außen für jeden sichtbar wird. Dieser Umstand führt dazu, dass kritische Implementierungsaspekte, also angefangen bei der Anwendungslogik, auf dem Server residieren sollten und seine Grenzen nicht verlassen sollten. Durch unsaubere Programmierung können diese Informationen nach Außen gelangen.

Asynchrone Kommunikation bei Ajax-basierten Anwendungen wurde zuvor als ein Grund für die effiziente Nutzung der zur Verfügung stehenden Bandbreite genannt. Die Kehrseite der Medaille ist, dass jede Ajax-Anwendung den Server immer wieder nach der Zustandsänderung der bearbeitenden Anfragen befragen muss, da der Server nicht in der Lage ist selbständig Antworten an den Client zu versenden. Damit wird im schlechtesten Fall die Bandbreite wie bisher ohne jeglichen Effizienzgewinn genutzt.

Eine weitere wichtige, durch die heutige leistungsstarke Hardware immer weniger sichtbare Problematik ist, dass durch Ajax-Anwendungen der Browser und damit die CPU immer mehr beansprucht wird. Mit steigender Komplexität der Webanwendung wird der Browser stärker ausgelastet, was dazu führen kann, dass dieser verlangsamt wird, was möglicherweise bei manchen nicht so leistungsstarken Rechnern zu sichtbaren Erscheinungen führen kann.

Die Kluft zwischen Benutzerschnittstellen (Browser) und Anwendungslogik (Server) schrumpft. Ajax kombiniert existierende Technologien und erweitert die Anwendungen um asynchrone Kommunikation. Durch diese laufen die Interaktionen zwischen den Browsern und den Servern asynchron. Ajax fügt einen Zustand zu dem sonst zustandslosen Web (Weiss, 2005). Des Weiteren werden durch Web Service Technologien viele Portale nach außen offen. Damit tragen diese zusätzlich dazu bei, dass eine kritische Masse an Benutzern angezogen wird um somit eine Wissensbasis zu aggregieren, welche gemeinschaftlich und selbstorganisierend entstehen kann. Die Betrachtung der Auswirkungen der neuen Technologien auf das heutige Web ist der Gegenstand des nächsten Abschnittes.

2.2.2 Ausprägungen Community-gestützter Systeme

An dieser Stelle möchte der Autor einige ausgewählte Ausprägungen Community-gestützter Systeme vorstellen die heute als Web 2.0 basierte Anwendungen oder Soziale Software bekannt sind.

Forum

Eine Webseite die als Diskussionsplattform dient wird als Forum bezeichnet. Dort durchgeführte Diskussionen sind einem Thema zugeordnet und werden als Beiträge von jedem Benutzer gespeichert. Gleichzeitig bietet ein Forum die Möglichkeit an jeden gespeicherten Beitrag wieder zur Diskussion zu stellen. Der Verlauf der Diskussionen wird baumartig gespeichert. Dieses ermöglicht das direkte beantworten von Artikeln.

Blog (Weblog)

Als eine der ersten Anwendungen des Web 2.0 kann wohl der Blog betrachtet werden. Ein Blog ist eine Webressource die einem Benutzer oder einer Benutzergruppe gehört und von dieser mit themenbezogenen Informationen beliefert wird. Damit kann dieser als ein Austauschmedium betrachtet werden, durch den der Verfasser und Leser gegenseitig Informationen tauschen können. Die dort eingefügten Beiträge werden in aufeinanderfolgenden umgekehrter chronologischen Folge sortiert. In einem Blog entstandene Informationen sind nach außen für jeden zum Lesen zugänglich und das durchgängig. Um die Beiträge aber verändern zu können muss man Mitglied der zugehörigen Community sein ([Kolbitsch und Maurer, 2006](#)).

Blogs können in zwei Arten unterteilt werden, den Tagebüchern und die, die als Filter dienen. In den sogenannten Tagebüchern werden Informationen über das private sowie berufliche Leben zusammengefasst. Bei dem Filtern handelt es sich um Aggregatoren von Referenzen auf andere Webressourcen. Dabei sind diese thematisch unterteilt und durch Zusammenfassungen eingeleitet.

Blogs können auf andere Blogs verweisen. Damit können Benutzer Veröffentlichungen aus einem Blog in ihrem eigenen referenzieren und den Inhalt dort zur Diskussion stellen. So sich gegenseitig referenzierende Blogs werden auch als Blogosphäre bezeichnet ([Alby, 2006](#)). Diese Blogosphären bilden offene informelle Kreise innerhalb dieser ein schneller Informationsaustausch stattfindet. Dieser Informationsaustausch basiert auf weiteren Techniken wie Kommentarfunktionen, die wiederum auf Permalinks oder TrackBack-Funktionen aufsetzen. Eine ebenfalls weit verbreitete Technik des Informationsaustauschs ist die Feed-Technik.

Kommentar Viele Blogs stellen die Möglichkeit dar einen Blogeintrag zu kommentieren. Dieses erlaubt jedem Benutzer kritisch jede Information zu bewerten und seine Meinung nach außen zu äußern. Damit wird durch diese einfache Möglichkeit einem Benutzer die Gelegenheit gegeben in die Rolle eines Kritikers zu schlüpfen.

Permalink Jeder Beitrag in einem Blog kann mit einem Permalink referenziert werden. Ein Permalink ist nichts anders als eine URL auf diesen Beitrag mit der Erweiterung, dass dieser immer verfügbar ist. Mit anderen Worten, unabhängig was mit dem Eintrag passiert ist, durch Permalink wird dieser für immer zugreifbar gemacht.

TrackBack Dieser Mechanismus ermöglicht die Sichtbarkeit einer Referenz von einem anderen Blog. Referenziert also ein Blog einen anderen, so wird dieses für den Autor und Leser des referenzierten Artikels kenntlich gemacht. Beide Beteiligten können nun so gegenseitig den Beitrag des anderen anschauen. Durch einen TrackBack kann nun eine bidirektionale Beziehung aufgebaut werden, da die beiden Teilnehmer gegenseitig für sich sichtbar gemacht werden. Damit ermöglicht die TrackBack Funktion die Wahrnehmung von Referenzen. Durch diese können Autoren erkennen ob, von wem und in welchem Kontext ihr Betrag referenziert wurde.

Blogrolls Unter Blogroll wird eine Linkliste in dem jeweiligen Blog verstanden, in der Links auf empfohlene Webseiten referenzieren. Dieses bietet eine weitere Möglichkeit zur Zusammenführung von Informationen des Ursprungsblogs mit Informationsquellen (anderen Blogs). Dabei bauen die einzelnen Blogs Referenzbeziehungen und damit eine Blogosphäre untereinander auf.

Feed (Abonnement-Dienst) Die Feed Technik dient der Publikation von zusammengefassten Inhalten einer Internetressource. Damit wird es möglich über die neu erschienenen Artikel in den referenzierenden Blogs informiert zu werden. Hierzu muss die Internetressource abonniert werden. Um diese Informationen empfangen und diese dann später lesen zu können sind Feed-Reader, welche auch als Aggregationssoftware bezeichnet werden, erforderlich. Diese nehmen die neuen Informationen wahr und bereiten die, in dem maschinenlesbaren Format enthaltene, Information soweit auf, dass es von Menschen gelesen werden kann.

Permalinks und Trackbacks sind Antworten auf die Lücken die das heutige Web immer noch aufweist. Durch die beiden Technologien können Artikel permanent in bidirektionaler Beziehung gesetzt werden. Durch die Feed Technik wandelt sich das heutige Web vom statischen

langsam zu einem dynamischen Medium um. Mit diesen drei Technologien ermöglichen die Blogs damit eine neue Art der Informationsverbreitung.

Instant Messaging

Durch Instant Messaging eröffnete sich eine neue Art der Kommunikation durch die Benutzer miteinander in Echtzeit Informationen austauschen können. In einem Instant Messaging Netzwerk existiert ein Server welche die Benutzer miteinander verbindet. Im Genaueren ist dieser Server wie bei hybriden P2P Netzen dazu da die Kommunikation zwischen einzelnen Knoten nur zu einzuleiten. Deren Funktionalität kann aber ausgeweitet werden um z.B. sogenannte *buddy lists* (Minar u. a., 2001) zu speichern. Damit wurde eine Basis für das verteilte Arbeiten geschaffen. Durch Instant Messaging können die Benutzer viel direkter miteinander interagieren und somit ihre Ideen und Ansichten austauschen.

Wissensmanagement

Sehr oft ist eine Wissensmanagement-Software komplex und schwer zu bedienen. Web 2.0 zeigt Ansätze die deutlich einfacher und dennoch geeignet genug für die Sicherung von Wissen sind. Damit wird möglicherweise ein Weg geschaffen wie man mit der immer größer werdenden Informationsmenge umgehen kann. Durch geeignetes Wissensmanagement kann dann endlich der Umbruch von einer Informationsgesellschaft zu einer Wissensgesellschaft vollzogen werden. Eine heute verstärkt eingesetzte Technik zur Sicherung von Wissen stellt das Wiki-Konzept dar.

Wiki Wikis sind Webseiten zur kollaborativen Erarbeitung von Informationseinheiten. Dort gesammelte Informationseinheiten können von jedem und jederzeit erstellt und verändert werden. Diese Offenheit der Wikis beruht auf Ansätzen, welche aus selbstorganisierenden Systemen stammen. Damit vertrauen diese darauf, dass die dahinter stehende Community selbstständig die Quintessenz aus den Informationen extrahiert. Diese Sichtweise trägt dazu bei, dass jeder Wiki-Teilnehmer in die Rolle des Verfassers schlüpfen kann. Mehr noch, jede Informationseinheit erreicht nie einen stabilen Zustand. Ein weiteres Merkmal ist die Tatsache, dass jeder Autor beim Veröffentlichenden eines Artikels davon ausgeht, dass dieser seine ursprüngliche Form nicht behält, da dieser jedem das Recht der Veränderung erteilt. Diese Maßnahme führt konsequenterweise zu einem weiteren Vorteil. Durch diese Offenheit und fortlaufende Veränderung wird ein zu Beginn unvollständig und möglicherweise fehlerbehafteter Artikel mit der Zeit zu einem qualitativ hochwertigen und fehlerfreien mutieren.

Bei genauerer Betrachtung erkennt man in einem Wiki viele Parallelen zu einem CMS. Der Unterschied liegt aber darin, dass beim Wiki kein echtes Rechtesystem existiert. Durch das

Fehlen von Hierarchien sind alle Teilnehmer gleichgestellt⁵. Bei der Erstellung von Artikeln wird eine schlanke Beschreibungssprache, die hinterher in HTML umgewandelt wird verwendet. So entstehende Sourcen werden dann in einer unterliegenden Datenbank versioniert gespeichert. Diese Zustände werden allen Benutzern verfügbar gemacht bzw. bei Notwendigkeit entsprechende Zustände zurückgespielt.

Durch Wikis wurde eine Möglichkeit geschaffen jenem, noch so ungeübten Benutzer dazu zu ermutigen an der kollaborativen Erarbeitung von Ressourcen teilzunehmen. Dieses wird durch eine einfache Syntax und Versionskontrolle verstärkt. Kritische Meinungen behaupten, dass in Wikis realisierte Konzept der Informationserarbeitung zu keiner qualitativ hochwertiger Wissensgenerierung führt kann. James Surowiecki hatte in seinem Buch ([Surrowiecki und Beckmann, 2005](#)) mehrmals verdeutlicht, dass solch ein Konzept dennoch in der Lage ist, diese Herausforderung zu bestehen. Natürlich ist so ein Konzept gegen Vandalismus anfällig. Die funktionierende Community wird aber dieses sehr schnell aufspüren können und wieder bereinigen. Explizite Methoden zur Steigerung der Qualität und dem Aufspüren von Vandalismus in Wiki-Systemen findet man in ([Möller, 2006](#)).

Wikipedia Wikipedia stellt eine konkrete Realisierung des Wiki-Prinzips. In diesem Wiki-System wird versucht das allgemeine Wissen, so wie es in Enzyklopädien vorzufinden ist, abzubilden. Der Unterschied zu dem kommerziellen Produkt ist zum einen, dass es kostenlos genutzt werden kann, zum anderen wird es von keinem ausgewählten Experten zusammengetragen, sondern von allen Teilnehmern.

Durch dieses Konzept entstehen folgende Vorteile:

- Flexibler als gedruckte Version
- Erreichte Genauigkeit vergleichbar mit gedruckter Version
- Durch Offenheit erreicht die Informationseinheit einen höheren Grad an Objektivität

Gleichzeitig birgt Wikipedia auch einige Schwächen in sich:

- Die Vertrauenswürdigkeit der Autoren ist unbekannt und damit auch die Qualität der Information
- Die Unterschiedlichkeit der Autoren kann negativen Einfluss auf die Objektivität haben
- Verletzung der Privatsphäre durch Publikation unseriöser Artikel
- Vandalismus und Oszillation von Artikeln
- Fälschliche Schlussfolgerungen durch Benutzer die das Wikikonzept nicht kennen

⁵Es existiert natürlich ein Administrativer Benutzer, dieser agiert eher in Hintergrund und stört die Zusammenarbeit der Teilnehmer nicht.

- Überladene Artikel

Zusammenfassend ist Wikipedia ein seit mehreren Jahren erfolgreiches Konzept, welches bis heute die verschiedenen Schwierigkeiten gut gemeistert hatte.

Wikinews Seit mehr als zwei Jahren existiert eine weitere Ausprägung von Wiki, indem kollaborativ aktuelle Neuigkeiten gesammelt werden. Nach ([Wikinews, 2006](#)) soll Wikinews ein Gegenstück zu den bekannten kommerziellen Nachrichtenagenturen sein. Dabei steht wieder wie in Wikipedia die Objektivität im Vordergrund. Da jeder einen Artikel bei Wikinews publizieren kann, können Neuigkeiten an die Öffentlichkeit treten, die bei konventionellen Medien möglicherweise weggefiltert oder durch beschränkte örtliche Meinungsfreiheit nicht veröffentlicht werden würden. Wikinews unterscheidet sich in einem Punkt gravierend zu Wikipedia und somit zu dem bekannten Wiki-Konzept. Nachdem ein Artikel erstellt und von der Community revidiert wird, wird dieser in einem persistenten Zustand versetzt und damit nicht mehr veränderbar. Er ist damit nur zum Lesen freigegeben. Eine weitere zusätzliche Einschränkung hierzu war die Unterbindung von jeglichen Möglichkeiten der Kommentierung von Beiträgen. Bei der Sichtung der existierenden Beiträge ist aber eine Schwäche zu erkennen. Viele in Wikinews veröffentlichte Artikel basieren auf üblichen Quellen und sind somit in der Hinsicht auf ihre Objektivität mit Vorsicht zu betrachten.

Semantisches Wiki Neben dem Web 2.0 existiert ein weiterer Ansatz, dass Semantische Web. Die Verbindung der beiden Welten wird in der Literatur als Web 3.0 gesehen. Dabei werden die Stärken der beiden Ansätze zusammengeführt und somit die Basis für die nächste Generation des Webs bereitgestellt. Einige sich hieraus ergebenden Möglichkeiten werden bereits im Semantischen Wiki wiedergegeben.

Nach ([Wikipedia, 2006g](#)) unterscheidet sich ein semantisches Wiki von dem ursprünglichen Wiki in der Wissensdarstellung. Damit werden nicht mehr rohe Daten im Wiki abgelegt, vielmehr werden diese einerseits durch semantische Informationen (Metadaten) beschrieben und mit anderen Informationen in Beziehung gesetzt. Diese Beziehung ist wiederum ebenfalls semantisch angereichert. Ein solch formalisiertes Wissen erlaubt dann, wie es im semantischen Web bekannt ist, dieses der automatisierten Verarbeitung verfügbar zu machen. Dabei ergeben sich speziell für Wikis neue Möglichkeiten der Validierung und Zusammenführung von Wissen, welches durch Inferenzmaschinen automatisch durchgeführt werden kann. Des Weiteren kann der Leser nicht nur aus der Information den Kontext erkennen, sondern aus den Metadaten eindeutig die einzelne Informationseinheit interpretieren. Dieses kann besonders hilfreich sein, wenn Wikis von sich sprachlich unterscheidenden Benutzern geführt werden. Die Sprachbarrieren können durch diese Metadaten (zum Teil durch Halbautomatismen) überwunden werden. Wikibenutzer könnten dann direkt Abfragen auf das im

Wiki enthaltene Wissen stellen, und nicht wie bisher üblich durch eine Volltextsuche, sondern auf der Basis von semantischen Informationen.

Eine konkrete Realisierung eines solchen Wikis stellt z. B. das Sematic MediaWiki ([Ontoworld, 2007](#)). Dieses ermöglicht das Typisieren von Relationen sowie eine Beschreibung von Ressourcen anhand typisierter Attribute. Durch so strukturierte Daten erlaubt dann das Semantic MediaWiki Anfragen anhand der dort enthaltenen Daten zu stellen also datenzentrisch zu gestalten. Zum anderen können Anfragen als neue Wissensartefakte in das Wiki eingetragen werden, sodass der Leser das resultierende Ergebnis der Anfrage liest. Diese trägt auch dazu bei, dass die Information verstärkt der Wiederverwendung unterzogen wird. Damit wird das Wissen noch weiter dynamisiert. Hiermit bekommen Schlussfolgerungen eine neue Dimension. Durch Anfragen, die wiederum auf Daten und Anfragen zurückgreifen, wird das Hinzufügen von neuen Informationen Einfluss auf die bestehenden haben. Der Community wird ein Werkzeug gegeben mit dem Diskussionen auf logische Zusammenhänge untersucht werden kann. Daraus entstandene Ergebnisse können durch die semantische Beschreibung leichter abstrahiert und aus der Wissensbasis extrahiert werden. Daten können dann zum Beispiel leichter zwischen einzelnen Wikis ausgetauscht werden. Auf dem Weg der Migration von existierenden Wikis anzutreffende Barrieren können damit überwunden werden. Damit können bis heute voneinander technisch getrennte Communities miteinander verbunden werden. Diese könnten dann zusätzlich auf das außerhalb ihrer Gemeinschaft gesammelte Wissen zurückgreifen. Die sich hierbei eröffnenden Möglichkeiten können enorme Auswirkungen auf diese haben. Hierbei kann nicht nur der Mensch davon profitieren sondern auch eine Anwendung, welche diese Daten automatisch weiter verarbeiten möchte. Weitere Beispiele der Realisierungen eines solches Wikis kann man in ([Ontoworld, 2007](#)) finden.

Soziale Netze

Unter dem Begriff Soziale Netze verbirgt sich ein Versuch Menschen zusammen zu bringen. Damit soll eine Möglichkeit geschaffen werden seine Bekanntschaften aufrecht zu erhalten oder neue zu erschließen. Zum anderen können über Soziale Netze alte, verlorene Kontakte wiedergefunden werden. Bei den genannten Aufgaben werden verschiedene Interessen verfolgt. Zum einen wird versucht Menschen beruflich zusammen zu bringen, auf der anderen Seite privat mit ähnlichen Interessen zu verbinden. Portale wie Friendster, MySpace, Orkut oder OpenBC sind Beispiele für solche Webplattformen.

Um ein Mitglied einer solchen Community zu werden, muss man vorher sein persönliches Profil erstellen und den anderen Benutzern verfügbar machen. Durch dieses Profil sind Benutzer in diesen Systemen identifizierbar. Mit einem gültigen Profil können diese dann Relationen zu anderen Teilnehmern definieren und somit eigene Soziale Netze nach außen offenbaren. Die Relationen zwischen den Benutzern werden manuell von dem Benutzer durch-

geführt. In manchen Anwendungen existieren Ansätze zur automatisierten Erstellung von interpersonellen Relationen. Zum anderen existieren Überlegungen wie man Soziale Netze dazu nutzt Ressourcen mit konkreten Benutzern in Verbindung zu setzen. Dadurch könnte eine personenabhängige, personalisierte Suche realisiert werden, um dabei Beziehungen zwischen den Teilnehmern und ihren Ressourcen in die Suchanfrage mit aufnehmen zu können.

Auch bei Sozialen Netzen kann das Semantische Web ein weiteres Beispiel der ergänzenden Zusammenarbeit mit dem Web 2.0 bieten, welches man in dem Projekt FOAF⁶ (Brickley und Miller, 2005) findet. In diesem werden Informationen über eine Person und Beziehungen zu anderen Personen formalisiert in einer Ontologie abgelegt und veröffentlicht. Auf der Basis dieser Beschreibungen können dann Programme diese Informationen auswerten und die daraus resultierenden Informationen über Personen sowie die sozialen Zusammenhänge zu anderen darstellen. Der Unterschied zu bestehenden Ansätzen ist die Dezentralisierung. Diese halten alle Daten über Personen zentral. Bei FOAF wiederum hat jeder sein Profil in der Hand.

Soziales Bookmarking

Kein Algorithmus kann so genaue Suchergebnisse liefern wie eine Suche, welche auf kooperativ gesammelten Metadaten basiert die von Menschen erstellt wurden [Bachle06:2006]. Diese Tatsache nutzen die Entwickler von del.icio.us aus. Dort wird das Konzept des Sozialen Bookmarkings zur gemeinschaftlichen Kategorisierung von Lesezeichen eingesetzt. Hierbei werden Deskriptoren zu jedem Lesezeichen als Beschreibung hinzugefügt und somit eine Wissensbasis über annotierte Webressourcen aufgebaut. Dieser Vorgang wird auch als Freie Verschlagwortung oder auch als Tagging bezeichnet. So gesammelte und mit Metadaten angereicherte Lesezeichen werden dann öffentlich gemacht und mit anderen Benutzern in Beziehung gesetzt. Die Verlinkung findet wie folgt statt:

- Benutzen zwei Benutzer den gleichen Link, so können diese miteinander verlinkt werden
- Findet eine Verlinkung statt, kann dem einen Benutzer die Deskriptoren (Tags) des anderen Benutzers zu dem gleichen Lesezeichen bereitgestellt werden

Die bei dem Sozialen Bookmarking entstehenden Metadaten und die damit entstandene Klassifizierung, die keiner Begriffssystematik unterliegt wird heute auch als Folksonomie bezeichnet. Diese zeichnet sich dadurch aus, dass dort verwendeten Klassifizierungsdeskriptoren von Benutzern ohne Einschränkungen frei gewählt werden können. Mit anderen Worten jeder Benutzer kann sein eigenes Begriffssystem definieren.

⁶Friend of a Friend

File Sharing

Hinter dem Begriff Filesharing verbergen sich nicht nur illegale Tauschbörsen. Neuerdings entstehen vermehrt Dienste durch die Benutzer verschiedene Ressourcen miteinander teilen können. Damit ist auch hierbei das Potential des Zusammenwirkens im Netz (Network collaboration) zu erkennen. Auf der anderen Seite zeigt Filesharing auch wie man nicht nur Ressourcen verteilen kann, sondern auch Benutzerinformationen. Diese werden nämlich über das gesamte Netz verteilt um die Koordination des Herunterladens zu ermöglichen zu können.

Flickr als Beispiel ist eine Art des Filesharing bei dem Bilder mit der Community geteilt werden. So unterscheidet sich dieser Ansatz von früheren. Es werden keine Ressourcen über eine Anwendung mit der Community geteilt. Dies geschieht durch webbasierte Anwendungen. Dabei werden die Ressourcen nicht wie zuvor auf dem Rechner des Teilnehmers gespeichert, sondern im Web.

Betrachtet man Flickr genauer, sind Fotos die dort gespeicherten Ressourcen, die anderen Teilnehmern zur Verfügung gestellt werden. Das Besondere ist, dass jeder Benutzer dabei helfen kann die Ressourcen eines Benutzers zu organisieren. So kann ein Benutzer jedes von ihm angeschauten Foto mit Metadaten durch Annotierung anreichern.

Podcast Beim Podcast handelt es sich um Audioressourcen die mit Hilfe der Feed Technik abonniert werden können. Statt Artikel werden Audiodateien zur Verfügung gestellt. Man kann sich bei Autoren eintragen lassen um dann bei erscheinen von neuen Audiodateien benachrichtigt zu werden. Ebenfalls besteht die Möglichkeit mehrere Audiodateien zusammenzuführen und als eine Abzurufen.

Durch eine Klassifizierung der Audiodateien vornehmen zu können, wird jede dieser Dateien durch Metadaten annotiert. Durch diese Maßnahme ist es dann dem Benutzer möglich die einzelnen Ressourcen nach Autoren und Themen zu sortieren.

Durch Podcasts wurde ein neues Medium erschlossen, durch das viele, die bisher als nur Zuhörer galten, selbst zu den Autoren wurden. Neben den neu ins Leben gerufenen Autoren entscheiden sich immer zunehmend professionellere Anbieter von Audiosendungen Ihre Beiträge auf dieser Art anzubieten.

Vodcast Hinter Vodcast steckt die gleiche Idee wie beim Podcast mit der Ausnahme, dass hier keine Audiodateien sondern Videos als Ressource fungieren. Auch hier kann jeder ein Autor von Videos werden und seine eigenen Videos an andere bereitstellen. Benutzer können dann ebenfalls Themen oder Autoren abonnieren und damit über den Feed-Mechanismus über neu verfügbare Ressourcen benachrichtigt werden.

2.2.3 Merkmale und Leitsätze Community-gestützter Systeme

In diesem Abschnitt sollen aus den zuvor vorgestellten Technologien und Anwendungen die wichtigsten Merkmale Community-gestützter Systeme extrahiert werden. Anschließend sollen diese dann nach verschiedenen Gesichtspunkten untersucht werden. Darauffolgend wird versucht die hierbei entstehenden Chancen aber auch die Risiken zu identifizieren.

Dieser Abschnitt orientiert sich auf der Publikation von Kolbitsch [Kolbitsch und Maurer \(2006\)](#) in der dieser versucht die Änderung, welcher die Informationsverarbeitung derzeit unterliegt und durch die kürzlich aufgetauchten neuartigen Technologien hervorgerufen wurde, zu untersuchen. Hierbei wird deutlich gemacht, wie die Unterscheidung zwischen dem Autor einer Informationsressource und deren Konsumenten immer mehr nivelliert wird, sodass die beiden Rollen von ein und derselben Person ausgeübt werden kann.

Im weiteren Verlauf des Abschnittes wird versucht deutlich zu machen, dass die hervorgerufene Änderung nicht durch die neuen Technologien verursacht wird. Vielmehr ist diese der sich wandelnden Sichtweise auf die Informationsverarbeitung zuzuschreiben. Zunehmend versuchen heute die Benutzer sich von der passiven Teilnahme weg zu bewegen und eine gestaltende Rolle anzunehmen. Die von Vannevar Bush beschriebene Kollektive Intelligenz tritt nun in Erscheinung.

Frühere Organisation des Webs: Top-Down Ansatz

Das bis heute bekannte Web hatte weniger die Aufgabe einen gewöhnlichen Benutzer bei der Wissenserarbeitung zu unterstützen. Dieser Benutzer hatte die Rolle eines Konsumenten eingenommen welcher die Informationsressourcen, die von einigen wenigen ausgewählten Menschen produziert wurden, konsumierte. Mit anderen Worten hatten nur einige wenige Menschen das Privileg bekommen als Informationsversorger zu fungieren und entsprechende Informationen zu publizieren. Damit hatte eine kleine Gruppe eine größere mit Informationen versorgt. Betrachtet man diese Gegebenheit genauer stellt man fest, dass sich auch hier bei der Informationsversorgung das Client/Server Konzept durchgesetzt hatte. Folglich wurde damit eine klare Hierarchie zwischen den Teilnehmern und damit der Informationsversorgung aufgebaut.

Wie in dem Buch von Surowiecki mehrmals zum Ausdruck kam, kann eine Struktur manchmal sinnvoll manchmal auch als hinderlich wahrgenommen werden. Zur Verdeutlichung der Aufhebung der Strukturen gibt der Autor dieses Buches das Beispiel der Zusammenarbeit von verschiedenen, auf der ganzen Welt verteilten Laboratorien, an. Durch die Auflösung der Hierarchien, die Möglichkeit sofortiger Publikation der Forschungsergebnisse und andere die kollaborative Arbeit unterstützende Methoden, kamen die Forscher in einer vorher nicht gekannten Geschwindigkeit zu der gesuchten Lösung. Durch die fehlende Zeit wurde

ein neuartiger Versuch der kollaborativen Zusammenarbeit durchgeführt, durch den quasi bewiesen wurde, dass das Aufheben von Hierarchien und die gegenseitigen Informationsversorgung positive Nebenwirkungen haben kann.

Änderung der Sichtweise: Bottom-Up Informationsversorgung

In der Bottom-Up Perspektive existiert eine ausgewählte Instanz, welche für die Informationserstellung zuständig ist, nicht mehr. Alle Aufgaben die diese Instanz ausgeübt hatte übernimmt nun die Community beziehungsweise jeder einzelne Teilnehmer. Dieser wird nun in die Lage versetzt selbstständig Informationseinheiten zu publizieren und für diese Strukturen zu definieren. Damit wird jeder in dieser Hinsicht gleichgestellt. Auf diese Weise wird eine Basis geschaffen, auf der neuartige Synergieeffekte erst möglich werden. Denn wie bereits von Surowiecki hinreichend geschildert wurde übersteigt die Weisheit einer Community die, ihres erfahrensten Teilnehmers. Aber was sind nun die Merkmale die eine gesunde bzw. eine intelligente Community auszeichnen, damit diese die gestellte Voraussetzung erfüllt und folglich qualitativ hochwertiges Wissen aufbauen kann? Welche Voraussetzungen müssen gestellt und erfüllt werden? Auf diese Fragen soll nun im Folgenden geantwortet werden.

Web als Plattform für Anwendungen

Vermehrt werden heute alle Desktopanwendungen zu webbasierten Anwendungen umgestaltet. Diese Änderung wird durch die neue Erkenntnis motiviert welche besagt, dass eine Webanwendung neuartige Einsatzmöglichkeiten bereitstellt durch die kollaborative Zusammenarbeit neuartig unterstützt werden kann. Durch die neuen Möglichkeiten können neue Wege begangen werden. Damit wandelt sich das Web von einem passiven Kommunikationsmedium zu einer Anwendungsplattform oder wie von Weis in [Weiss \(2005\)](#) bezeichnet zu dem WebOS.

Partizipation

Wie zuvor bereits geschildert, wurde das Web bis vor kurzem eher als statisches Medium zum Austausch von Medien betrachtet. Die Anwendungen die darauf basierten nutzten diesen hauptsächlich dazu, Daten zwischen den Servern und dem Client-Rechnern auszutauschen. Die hinter dem Web 2.0 stehenden Anwendungen gehen einen anderen Weg und bieten den Benutzern eine Möglichkeit nicht nur die verfügbaren Ressourcen zu nutzen. Nein, sie versuchen den Benutzer in verschiedenen Bereichen einzubeziehen und diesen zu ermutigen neue Ressourcen zu generieren. Mit anderen Worten der Benutzer wird nun

in die Lage versetzt das Web und die darauf basierenden Anwendungen nicht nur zu nutzen sondern kann selber dazu beitragen, dass diese mehr seiner Vorstellung entsprechen und damit besser funktionieren. Die Arten der Partizipation der Benutzer variieren in den verschiedenen Anwendungen und werden im Folgenden eingeleitet.

Daten In erster Linie versuchen die neuen Anwendungen den Benutzer dazu zu ermutigen neue Daten zu generieren. Die vom Benutzer eingefügten Daten variieren in ihren Verwendungszweck für diesen. Betrachtet man diese genauer so können hier zwei Arten von Daten unterschieden werden. Die eine Gruppe dient der Bindung der Benutzer an die jeweilige Anwendung. Solche bindenden Daten können z. B. Fotos, Videos oder Bloginhalte wie Tagebücher des Teilnehmers sein. Die zweite Art stellt ein neuartiges Wissen, welches sowohl für den Benutzer als auch für den Bereitsteller wertvoll ist, dar. Auf diesen Daten können neue Dienstleistungen angeboten werden.

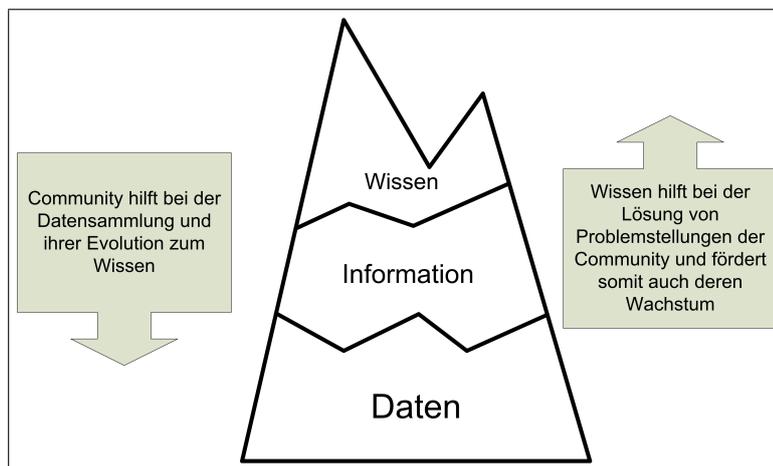


Abbildung 2.5: Die Korrelierenden Abhängigkeiten zwischen Benutzerzahl und Datenqualität (vgl. [Symeonidis und Mitkas, 2005](#))

Ein für die Partizipation bei der Sammlung von Daten sehr wichtiger Effekt ist deren Wertentwicklung. Es ist ersichtlich, dass durch die Zunahme der Nutzer solcher Anwendungen, der Datenbestand ebenfalls wächst. Gleichzeitig ist zu beobachten, dass die steigenden Benutzerzahlen einen positiven Einfluss auf die Qualität der Daten haben. Folglich steht der Wert der Daten für die Benutzer sowie für die Dienstbereitsteller mit dem Zuwachs der Teilnehmer in einer korrelierenden Abhängigkeit. Damit ist die Partizipation eine Antriebskraft für beide Seiten, da diese gegenseitig davon profitieren. Abbildung 2.5 veranschaulicht diese ergänzende Abhängigkeit.

Tagging: Freie Annotierung Eine spezielle Form der Daten wird durch die sogenannte Freie Annotierung von den Benutzern erstellt. Hierbei handelt es sich um Metadaten, welche die zugrunde liegende Ressourcen beschreiben. Diese Metadaten werden dazu verwendet zum einen Daten einfacher aufzufinden und zum anderen aber auch durch frei wählbare Merkmale zu klassifizieren. Mit diesen Daten kann der Benutzer seine Ressourcen besser unter Kontrolle bringen. Auf der anderen Seite wird aber mit diesen Daten ein neues Wissen und damit neue Zusammenhänge abgebildet. Dieses kann dann nicht nur vom Benutzer für die Wiederauffindung verwendet werden, sondern auch von dem Dienstanbieter zur Realisierung einer besseren Suchmaschine. Denn wie bereits angedeutet, übertrifft die auf menschlicher Analyse basierende Suche alle bis heute bekannten Suchverfahren.

Betrachtet man genauer die bestehenden Systeme bei denen Freie Annotierung eingesetzt wird, findet man dort ebenfalls Listen von häufig verwendeten Deskriptoren. Die haben nicht nur einen Übersichtscharakter um den Benutzer zu zeigen welche in der letzten Zeit verwendet wurden. Diese Listen haben eine wesentlich wichtigere Funktion. Verwendet ein Benutzer z. B. bei der Annotierung seiner Bilder einer oder mehrere der häufig verwendeten Deskriptoren, so kann er sicher davon ausgehen, dass seine Bilder von anderen Benutzern wiedergefunden werden. Möglicherweise geht es vielen Benutzern darum damit bekannter zu werden, es gibt aber einen weiteren, viel wichtigeren Effekt. Bei der näheren Betrachtung der sogenannten Popular Tags stellen diese ein einheitliches, für jeden Benutzer nachvollziehbares Vokabular zur Beschreibung von Ressourcen, welches sprachübergreifend eingesetzt wird, dar. Unter Gesichtspunkten des Semantischen Webs handelt es sich hierbei um eine Art Ontologie, durch die Ressourcen sprachübergreifend in Beziehung gesetzt werden können.

Ontologien Die Partizipation wird aber nicht nur innerhalb einer Community stattfinden sondern auch zwischen den Communities selber. Die Art der Partizipation ist essentiell für die Kohärenz der in den einzelnen Communities enthaltenen Daten. Werden die einzelnen Wissensbasen bzw. das darin enthaltene Wissen anderem gegenübergestellt, so entsteht ein globales Wissen auf dem wiederum neues generiert werden kann was vorher nicht möglich war. Um solch ein globales, kohärentes Wissen aufzubauen sind Vokabulare zur Beschreibung von semantischen Beziehungen notwendig. Diese erlauben das Übersetzen von Wissen aus einer geschlossenen Welt in die andere und umgekehrt. Der Ansatz des Ontologie-Engineering befindet sich noch in einem früheren Entwicklungsstadium und ist deshalb noch nicht weit verbreitet. Ungeachtet dessen bietet diese Methode eine Möglichkeit zur Herstellung einer Kommunikation zwischen den Communities um damit den Datenaustausch Community-übergreifend zu gestalten.

Bewertung Eben wurde eine spezielle Art des Wissens bzw. Daten, die durch Partizipieren von dem Benutzer in die Anwendung hinzugefügt werden, angesprochen. Eine andere und bereits länger bekannte Art von Daten ist das Rating. Hierbei handelt es ebenfalls um Daten die von einem Benutzer erstellt werden, welche ein besonderes Merkmal aufweisen. Über Ratings kann ein Benutzer seine Meinung abgeben. Diese digitalisierte Darstellung der Bewertung kann dank der neuen Web 2.0 Anwendungen zentral gesammelt und dazu verwendet werden verschiedene Änderungen, die von Benutzern gesteuert werden, zu bewirken. Damit kann zum einen die Qualität der Daten und der Anwendung bewertet werden und damit als Wissen zur Verbesserung dieser genutzt werden. Zum anderen können diese aber auch dazu verwendet werden, um ein Profil des Benutzers zu erstellen. Wie die so erzeugten Daten gesammelt und verwendet werden können kann anhand von ([Last.fm, 2007](#)) und ([Pandora, 2006](#)) exemplarisch verdeutlicht werden.

Komposition von Wissen Bei den eben genannten Merkmalen bei der Sammlung von neuem Wissen wird das Ziel verfolgt zusammenhängende Strukturen zu erkennen. Dabei gilt in erster Linie die Division Wissen durch Sammeln von Einzelteilen zu generieren und diese zu einem Ganzen zu aggregieren.

Erweiterbarkeit (immer Beta) Eine weitere Form der Benutzerpartizipation, welche durch die neuartigen Anwendungen versucht wird zu motivieren ist die iterative Verbesserung der Anwendung selbst. Der Benutzer bekommt in dieser Art von Anwendung Möglichkeiten geboten, durch die er einen Einfluss auf diese bekommt. Damit kann er die Anwendung mitgestalten, was bisher selten und nur ansatzweise möglich war. Der Anwender kann seine Anregungen direkt formulieren und dem Dienstprovider vermitteln. Er kann diese auch indirekt beispielweise auch durch Bewertung von einzelnen Funktionen der Anwendung durchführen.

Das Neuartige an diesen Anwendungen ist, dass die Meinungen bzw. Wünsche direkt in die Anwendung einfließen und dem Benutzer bereitgestellt werden können. Diese ist deshalb möglich, da die Anwendungen webbasierend sind und wie zuvor beschrieben, werden diese durch den erneuten Zugriff automatisch aktualisiert. Natürlich gibt es heute auch andere Mechanismen, welche die Aktualisierung der Anwendung ermöglichen. Diese erfordern aber manchmal eine Deinstallation der Alten Anwendung oder andere für manche Benutzer komplizierte Tätigkeiten. Die Web 2.0 basierte Anwendungen sind in der Hinsicht vollständig wartungsfrei. Sie vermitteln den Anschein, dass diese nur für die Benutzung geschaffen sind.

Es stellt sich auch eine weitere Frage in dem Zusammenhang der benutzergetriebenen Verbesserung der Anwendung. Die Benutzer bekommen bei diesen Anwendungen die Möglichkeit bei deren Mitgestaltung teilzunehmen und Ihre Vorschläge direkt beim nächsten Zugriff

zu erkennen. Bekommt der Benutzer aber den Eindruck, dass seine Vorschläge keinen Einfluss haben, so kann dieser sich dazu entscheiden den Dienst zu wechseln. Das besondere an den Web 2.0 Anwendungen ist, dass der Dienstbereitsteller diese schnell erkennen kann. Dies führt wiederum dazu, dass der Dienstbereitsteller dazu ebenfalls motiviert wird den Wünschen der Community Folge zu leisten. Mit anderen Worten, der Benutzer kann durch sein Verhalten die Partizipation des Dienstbereitstellers beeinflussen. Dieses ist bei gewöhnlichen Desktopanwendungen im Grunde nur durch Verkaufszahlen zu erkennen, wenn überhaupt.

Komposition von Diensten (Mashups) Eine weitere Art der Partizipation, welche die Web 2.0 Anwendungen ermöglichen ist die Möglichkeit des Aufbaus von eigenen Anwendungen, die mit Mashups bezeichnet werden. Hinter Mashups verbirgt sich eine Idee neue Dienste basierend auf bereits bestehenden aufzubauen, bzw. auf den dort gesammelten Daten. Da alle Anwendungen im Web zugreifbar und dafür ausgelegt sind einfach mit neuen verbunden zu werden, wird der Benutzer in einer neuartigen Weise dazu motiviert das Web zu gestalten. Diese neue Partizipationsart fördert die Entstehung von neuen Anwendungen, die wiederum neue, bis dahin unbekannte ermöglichen. So entsteht hierbei ein neuartiger Synergie-Effekt, der das Ziel vom interaktiven Web näher rücken lässt.

Durch die Veröffentlichung von Schnittstellen ist eine neue Welle neuartiger Webanwendungen zu erwarten, welche die Prinzipien wie die Teilung von Daten und der Zusammenarbeit [Weiss \(2005\)](#).

Betrachtet man das durch Web 2.0 eingeführte Mashup Prinzip so ergeben sich auch hier Einsatzmöglichkeiten für das semantische Web. Wenn ein Dienst andere Dienste aggregiert, wird im Umfeld des Web 2.0 von einem Mashup gesprochen. Dabei befragen (leichtgewichtige) Dienste andere Informationsdienste. Die dabei übertragene Information wird üblicherweise als XML Datei übermittelt. Die DTD bzw. das XML Schema der unterschiedlichen Informationsdienste kann dabei variieren. Hierbei kann die semantische Beschreibung einen nahtlosen Übergang schaffen. In [\(Sheth u. a., 2006\)](#) wird dann in Zusammenhang mit Mashup von Semantischen Mashups (semantic mashups oder smashups) gesprochen. Diese verwenden Semantiken, um die unterschiedlichen Mashups miteinander integrieren und koordinieren zu können.

Kollaboration (Zusammenarbeit)

Mit den neuartigen Web 2.0 basierten Anwendungen ist ein neuer Trend eingeleitet worden, Menschen zu verbinden und zusammenarbeiten zu lassen. Kollaboratives Arbeiten bekommt damit eine neue Plattform, mit einfachen, benutzerfreundlichen Werkzeugen.

Auf der anderen Seite bekommt die kollaborative Arbeit einen neuen beschleunigenden Faktor. Die Web 2.0 Anwendungen basieren auf einer offenen Community. D.h. jeder kann theoretisch etwas Eigenes dazu steuern. Damit wird die kollaborative Arbeit nicht mehr auf eine kleine Gruppe von ausgewählten Experten beschränkt. Durch die Öffnung der Zusammenarbeit für jedermann können neue Wege beschrritten werden, die bisher unbekannt waren. Die dafür notwendigen Bedingungen werden hier als Merkmale einer schöpferischen Community bezeichnet.

Merkmale einer Intelligenten Community

Zu Beginn dieses Abschnittes wurde dieser mit einer Frage eingeleitet die nach den Merkmalen einer funktionierenden Community fragt. Im Rahmen dieser Arbeit ist eine gesunde Community eine, die selbstständig neues, qualitativ hochwertiges Wissen liefern kann. Mit einer ähnlichen Fragestellung hatte sich James Surowiecki auseinandergesetzt und identifizierte vier Merkmale die auch nach Meinung des Autors zum Tragen kommen.

Meinungsvielfalt Dieses Merkmal bzw. diese Forderung beschreibt den Erfahrungsschatz des Einzelnen. In der Vergangenheit zeigte sich häufig, dass homogene Gruppen von Menschen sich darauf spezialisieren bereits vorhandenes Wissen zur effektiven Lösungsfindung einsetzten. Damit werden möglicherweise neue und gegebenenfalls bessere Lösungswege unberücksichtigt bleiben. Aus diesem Grund ist meinungsvielfalt so essentiell, damit neue Wege und somit neues Wissen generiert werden kann. Durch die hier beschriebenen neuartigen, webbasierten Ansätze wird Meinungsvielfalt bei vielen neuen Anwendungsdomänen das erste mal oder noch verstärkt zum Einsatz kommen können.

Unabhängigkeit Sehr wichtig für die kollektive Intelligenz ist die Unabhängigkeit bei der Generierung von Wissen. Intelligente Ergebnisse erfordern, wie eben geschildert, eine Vielfalt von Meinungen, damit ein gültiger Konsensus gebildet werden kann. Die Betrachtung der Problemstellung darf nicht von außen beeinflusst werden und muss von jedem einzelnen unabhängig durchgeführt werden. Betrachtet man nun die vorgestellten verteilten Anwendungen, so erfüllen diese alleine durch deren konzeptuelle Realisierung dieses Merkmal, da diese alle Benutzer, die sich an verschiedenen Orten befinden, miteinander verbinden. Zumal kann eine Unabhängigkeit noch dadurch verstärkt werden, dass Benutzer ihr Wissen anonym weitergeben können.

Dezentralisierung Das für die Lösung herangezogene Wissen muss bei einer gesunden Community ebenfalls einen dezentralen Charakter besitzen. Hierbei gilt es umso dezentralisierter und damit verteilter die Community aufgebaut ist, umso mehr kann diese auf ggf. nur lokal vorhandenes Wissen zugreifen. Und gerade nur dieses spezielle Wissen bzw. lokal begrenztes Wissen kam bis heute nur selten zum Einsatz. Durch die Web 2.0 Technologie wird es erstmals möglich auch dieses zur Lösungsfindung einzusetzen und damit eine neue Qualität zu erreichen.

Aggregation Dieses unabhängige und dezentrale Wissen muss gebündelt werden, damit neues Wissen entstehen kann. Mit der Bündelung ist hier die in Beziehungsetzung der verschiedenen gesammelten Erfahrungen und deren Abwägung gegeneinander zu verstehen. Dazu erscheinen die Web 2.0 Werkzeuge gerade zu prädestiniert zu sein. Diese bieten verschiedene Lösungsansätze zur Bündelung von Wissen und deren gemeinschaftlich gestützte Nutzung zur Generierung von neuem Wissen.

Leser und Autor: Keine Abgrenzung Auf den Merkmalen von Surowiecki aufbauend müssen weitere Anforderungen erfüllt werden, damit innerhalb einer Community qualitativ hochwertiges Wissen gebildet werden kann. Einer dieser Merkmale ist die Auflösung der Unterscheidung zwischen einem Autor und einem Leser, der als Konsument hier fungiert. Es ist wichtig solche Hierarchien aufzulösen damit jedes noch so unbedeutendes Wissen aggregiert werden kann und damit als Katalysator für neues höherwertiges Wissen dienen kann. Wird diese Hierarchie aufgelöst, wird eine Plattform geschaffen auf der nicht eine subjektive Wahrheit gesammelt wird, sondern aus vielen Teilwahrheiten besteht und damit genauer die Realität abbildet.

Gemeinschaftliche Teilung von verteiltem Wissen Um intelligentes Wissen zu erstellen muss dieses mit jedem geteilt werden, damit es hinterfragt wird und zur Anwendung kommt. Es reicht nicht wenn alle Menschen bei der Wissenssammlung mithelfen, solch eine Wissensbasis muss für jeden zugänglich werden.

Selbstorganisierende Gemeinschaften Diese Eigenschaft resultiert aus den Merkmalen einer intelligenten Community. Jede Struktur die von einer übergeordneten Instanz eingeführt werden kann als hinderlich für die Weiterentwicklung sein. In der Natur sind vermehrt selbstorganisierende Systeme zu beobachten die auf dem ersten Blick chaotisch wirken, bei näherer Betrachtung allerdings selbstständig in sich organisiert sind. Diese Dynamik ermöglicht diesen Systemen eine Anpassung an die Gegebenheiten unter Berücksichtigung jeder Einzelheit, zu der eine übergeordnete Instanz nicht im Stande wäre.

Kollektive Intelligenz Eine nach diesen Merkmalen gestaltete Gemeinschaft ist im Stande ein neues Niveau des zusammen erarbeiteten Wissens zu erreichen. Der Grund liegt bei der kollektiven Betrachtung der Informationen. Es werden keine Schlüsse gezogen die nach Meinung eines Individuums als richtig erachtet werden. Um eine Schlussfolgerung als gefestigt zu betrachten muss diese vorher durch verschiedene in einer Gemeinschaft implizit enthaltene Filter durchlaufen. Erst nachdem diese keinen Widerspruch mit dem aktuellen Wissen der Gemeinschaft in sich birgt wird dieses als gegeben akzeptiert. Die hierbei repräsentierten Merkmale sichern die Qualität des neu resultierenden Wissens.

Auf der anderen Seite kann die kollektive Intelligenz als eine Antriebskraft für die Community selber betrachtet werden. Diese Antriebskraft kann auch als *Freie Entfaltung* bezeichnet werden. Dabei baut diese auf den Erfahrungen die mit der Entdeckung von neuem Wissen entstehen. Diese einmalige Erfahrung etwas Neues Geschaffen zu haben, die Möglichkeit es jedem zur Verfügung zu stellen kann neue Ausmaße in der Erarbeitung von neuem Wissen einnehmen. Hierdurch freigesetzte Kräfte können das Gestalten des Webs von Morgen in neuen Dimensionen bewirken.

Technische Aspekte

Alle bis jetzt geschilderten Technologien übergreifenden Merkmale richten sich mehr oder weniger an die Vorgaben die mit der neuen Sichtweise im Web 2.0 zu finden sind. Im Folgenden möchte der Autor die Änderungen in der Hinsicht auf die technischen Aspekte schildern.

Einfachheit Mit Einfachheit ist hier die einfache Handhabung der Anwendung gemeint. Einige der heutigen neuartigen Anwendungen zeigen eine große Übereinstimmung mit einigen, die in der Vergangenheit bereits realisiert wurden. Ein entscheidender Unterschied zwischen diesen ist, dass die heutigen Anwendungen sich sehr viel in Richtung der Benutzerführung entwickelt haben und den Benutzer nicht bei der Arbeit behindern, sondern unterstützend wirken. Damit sind Merkmale wie einfache Bedienbarkeit, ergonomische und einfache Oberflächen und offene Schnittstellen gemeint.

Wiederauffinden durch Suche Heutige Emailprogramme gelten als veraltet, wenn der Benutzer selber seine Emails klassifizieren muss. Hierfür und bei weiteren Sozialen Systemen wird heute zunehmend die Suche verwendet um entsprechende Informationseinheiten zu finden. Die vorherige Klassifizierung entfällt.

Aber auch die Suche selber ändert sich grundlegend. Vermehrt werden keine Algorithmen mehr verwendet, die das Web durchsuchen und analysieren. Die Aufgaben solcher Algorithmen übernehmen zunehmend die Menschen selber indem diese die Internetressourcen gemeinschaftlich indexieren bzw. auf eine andere Art Datenbestände zusammentragen und mit Metadaten manuell anreichern. Die Grundlage für die Suche wird nicht mehr eine automatisierte Methode sein sondern die Internetbenutzer selbst.

Versionierung Die Versionierung ist eine schon seit längerem bekannte Technik mit der man verschiedene Arbeitszustände persistent machen kann. Dies wurde bis heute hauptsächlich von Experten genutzt und stand dem gewöhnlichen Benutzer verschlossen. Mit den Web 2.0 Anwendungen wurde diese Technik dem breiten Publikum zugänglich gemacht. Sie wurde in den Prozess der Wissenserarbeitung integriert und führt dazu, dass die Teilnehmer mutiger mit Änderungen umgehen können und deswegen einfacher neue Wege begehen. Auf der anderen Seite stellt dieser Technik auch eine Methode der Sicherung des Wissens vor Vandalismus dar.

Personalisierung Weiter oben wurden einige Merkmale genannt die den Benutzer motivieren bei dieser Art von Anwendungen mitzumachen. Betrachtet man genauer diese Anwendungen wird eine weitere Eigenschaft ersichtlich, die Personalisierung der Anwendung. Durch diese kann sich der Benutzer noch mehr mit dem in dieser Anwendung realisierten Prozess identifizieren. Die weitere Fasette dieses Merkmals bezieht sich auf die Nutzung dieses Merkmals für die Wahrnehmung des Benutzerverhaltens und deren spätere Reproduzierung. Durch dieses kann sich eine Anwendung intelligenter verhalten indem die Benutzerbewegungen dazu verwendet werden den Benutzer in der Nutzung der Anwendung später besser zu leiten.

Leichtgewichtigkeit Aus der technischen Sicht hatte sich die Struktur bzw. die Offenheit von Anwendungsschnittstellen von kompliziert und nicht durchdringbar zu einfach verständlich und leicht veränderbar gewandelt. Durch diese technisch versierte Vereinfachung der Anwendung sind diese gegenüber neuen Verwendungszwecken offener und damit unterstützend wirkend bei der Entstehung neuer Anwendungen. Durch diese Leichtgewichtigkeit können Anwendungen soweit verändert werden, sodass diese nicht mehr nur der ursprünglichen Aufgabe dienen, sondern auch ganz anders eingesetzt werden können.

Proaktivität statt Passivität Die neuartigen Anwendungen können ein weiteres Merkmal aufweisen, gemeint ist die Proaktivität. Hierbei handelt es sich um den proaktiven, unterstützenden Charakter der Anwendung. Die Web 2.0 Anwendungen werden so gestaltet, dass

diese ein dynamisches Verhalten bei der Erstellung und Adaptierung von Wissen und Strukturen wiedergeben. Durch dieses wird ein Benutzer aktiv bei der Nutzung der Anwendung unterstützt. Mit anderen Worten eine proaktive Anwendung reagiert in bestimmten Situationen selbstständig, ohne dass der Benutzer sie dazu auffordert.

Hardwareübergreifend Ebenfalls ein sehr wichtiger technischer Aspekt bei den neuartigen Diensten ist die Fähigkeit diese hardwareübergreifend nutzen zu können. Die Basis für diese neuen Dienste ist das Web also eine Plattform die unabhängig von dem Betriebssystem eines Gerätes ist. Durch diese Tatsache greifen diese Dienste auf alle Geräte, die einen Zugang zum Web haben. Damit werden die bestehenden Gerätegrenzen überwunden wodurch neue Daten erzeugt werden können und neue Einsatzgebiete erschlossen werden, welche bisher zum größten Teil unberücksichtigt blieben.

Kostenlos verfügbar

Ein ganz wichtiges Merkmal aller dieser Anwendungen ist deren preisliche Gestaltung. So gut wie alle diese Dienste sind kostenlos durchgängig verfügbar. Natürlich gibt es bei einigen Diensten sog. Premiumfunktionen die nur gegen Entgelt verwendet werden können. Ungeachtet dessen wird die Kernfunktionalität, durch welche die neuen Daten erzeugt werden, immer kostenlos angeboten.

Einfach in der Verwendung - Ambient Intelligence

Ein äußerst wichtiges Merkmal der Sozialen Software ist deren Einfachheit in der Benutzung. Durch einfach aufgebaute, intuitive Schnittstellen können solche Dienste einer breiteren Masse zur Verfügung gestellt werden. Die gleiche Zielsetzung verfolgt auch das Ambient Intelligence, dort spezifizierte Merkmale der einfachen Benutzung sollten deshalb an dieser Stelle näher gebracht werden.

Im Grunde genommen ist Ambient Intelligence nichts anderes als eine Bereitstellung von Just-In-Time Diensten. Diese werden in verschiedenen erdenklichen Gegebenheiten bereitgestellt, sowohl positionsbezogen, als auch im mobilen Zustand und zu jedem Zeitpunkt. Dabei steht in Vordergrund die Intelligenz des Systems. Ein solch intelligentes System nimmt dem Benutzer verschiedene Aufgaben ab. Wenn aus dem Kontext ersichtlich wird, schlägt dieses sogar Vorschläge zur besseren Erreichung von entsprechenden Zielen vor. Im Großen und Ganzen wird der Benutzer von der Infrastruktur intelligent unterstützt und nicht von dieser über neue Informationskanäle mit Informationen überwältigt. Diesmal soll sich das Computing dem Menschen nähern und nicht umgekehrt.

Damit steht das Ambient Intelligence direkt in Beziehung zur den Community-gestützten Systemen bzw. diese können deutlich verbessert werden, wenn die Grundsätze des Ambient Intelligence in diese einfließen würden.

Leitsätze des Ambient Intelligence Ambient Intelligence ist eine europäische Forschungsinitiative die sich speziell mit der Konzeption und Entwicklung einer Infrastruktur für Dienste jeglicher Art beschäftigt. Betrachtet man Ambient Intelligence genauer, so werden mehrere Charakteristika der Informationsverarbeitung abgedeckt. In ([Shadbolt, 2003](#)) und ([Weber u. a., 2005](#)) werden folgende genannt:

Ubiquitous and Pervasive Computing: Es werden kostengünstige Geräte entwickelt, durch die das Zusammenspiel mit der zugrunde liegenden Infrastruktur möglich wird. Dabei greifen diese auf die Ressourcen der Infrastruktur Ad Hoc zu und stellen dort verfügbare Dienste zur Verfügung. Die Geräte selber werden in die Umgebung oder in Gegenständen, welche die Menschen mitführen, transparent integriert und im mobilen Zustand benutzbar.

Intelligente Systeme: Hierbei handelt es sich um Einheiten welche verschiedene Funktionalitäten in das System einführen und dieses damit intelligenter machen. Diese Einheiten verwenden hierbei Methoden wie Lernende Algorithmen, Mustererkennung und einige weitere, aus der Künstlichen Intelligenz stammende Verfahren. Dieses dient dazu das System in seiner Interaktion intelligenter zu machen. Dabei geht es in erster Linie um die Erkennung des Benutzerverhaltens sowie um die intelligente Steuerung des Systems. Außerdem spielt hier die Verteilung (Dezentralisierung) der Intelligenz selber ebenfalls eine wichtige Rolle.

Context Awareness: Um mit der Umgebung aufgabenbezogen interagieren zu können, muss das System seine Position (Kontext) wahrnehmen und entsprechend interpretieren können. Neben der Position muss das System außerdem in der Lage sein den Kontext einer Interaktion und damit den Benutzer selber ebenfalls zu erkennen. Dieses wird durch Sensoren und Annotationen wahrgenommen.

Soziale Interaktion: Dieser Bereich von Ambient Intelligence umschließt alle Aspekte der Mensch-Computer Interaktion in Hinsicht auf die sozialen Eigenschaften. Damit wird das Hauptaugenmerk auf die sozial natürliche Gestaltung der Systemschnittstellen gerichtet und die eigentliche Verarbeitung als zweitrangig betrachtet. Damit sind natürliche Schnittstellen wie Stimme und Gesten gemeint, welche zur Kommunikation mit dem System genutzt werden.

([Aarts, 2004](#)) ergänzt diese Schlüsselmerkmale um:

Personalisiert: Das System kann an den Benutzer bzw. seinen Anforderungen angepasst werden und damit diese besser erfüllen. Gemeint sind an dieser Stelle Anpassungen, welche über einen kurzen Zeitraum andauern.

Adaptiv: Basierend auf dem oben aufgeführten Merkmal "Intelligente Systeme", konkretisiert (Aarts, 2004) diese Eigenschaft. Mit Adaptiv ist gemeint, dass das System das Benutzerverhalten analysiert und sich dann gegebenenfalls den neuen Anforderungen automatisch anpasst, ohne dabei die Systemdienste in ihren Fähigkeiten zu beeinträchtigen. Dabei erstreckt sich die Dauer der Veränderung des Systemverhaltens über eine längere Zeitperiode.

Vorrausschauend: Das System ist vorausschauend und realisiert die Benutzerwünsche, ohne dass diese bewusst von dem Benutzer dem System aufgetragen wurden bzw. ohne dass der Benutzer hierzu einer expliziten Interaktion unterzogen wird.

Ein eben so wichtiger Aspekt wird in (Commission, 2001) genannt:

Verlässlichkeit und Sicher: Bei Ambient Intelligence werden verstärkt persönliche Daten zum einen gesammelt und auf deren Basis Entscheidungen getroffen. Es ist also verständlich, dass das System hohen Sicherheits-Ansprüchen genügen muss. Zum anderen muss dieses ein hohes Maß an Verlässlichkeit bieten, damit dem System verantwortungsvolle Aufgaben anvertraut werden können.

Unter Berücksichtigung der eben genannten Aspekte kann Ambient Intelligence als Plattform für Interaktionen zwischen Menschen und elektronischen Objekten basierend auf Kontextwahrnehmung, natürlichen Benutzerschnittstellen und allgegenwärtige Verfügbarkeit von Informationen verstanden werden (Aarts, 2004).

Die Herausforderung für Ambient Intelligence liegt nach Aussage von (Shadbolt, 2003) darin, ein Verständnis darüber zu bekommen, wie das Alltagsleben der Menschen aussieht, bzw. was ihren Alltag ausmacht. Zum anderen ist es wichtig zu erkennen wie die Umgebung aussieht, in der diese Menschen leben und wie diese genutzt wird, damit intelligente Systeme dieses Wissen nutzen können um den Alltag einfacher werden zu lassen.

Um den eben gestellten Herausforderungen gerecht zu werden, müssen nicht nur Fortschritte im Bereich des ubiquitären Computings erzielt werden. Alle diese Geräte müssen miteinander kommunizieren können, hierzu also vorher identifiziert und konfiguriert und nach Möglichkeiten miteinander zusammengesetzt werden. Um eine weitere Evolutionsstufe des Systems zu erreichen, also ein System welches im Stande ist den Kontext zu analysieren, diesen zu adaptieren und daraus über das Benutzerverhalten zu lernen, ist noch viel Arbeit erforderlich. Werden diese irgendwann abgeschlossen, so werden Systeme entstehen, welche die Benutzerintentionen selber erkennen werden.

Nach (Aarts u. a., 2002) versucht Ambient Intelligence eine Integration von Umgebungen, in denen sich Menschen alltäglich bewegen. Die Realisierung einer solchen Vision stützt sich

auf allgegenwärtigen Geräten, welche verborgen in die Benutzerumgebung zu integrieren sind. Dies alles wird mit dem Ziel gemacht, die Lebensqualität der Menschen zu steigern. Dieses wird nach Aussage von (Aarts, 2004) dann erreicht, wenn eine gewünschte Atmosphäre und Funktionalität in der Alltagsumgebung mit Hilfen von intelligenten und personalisierten Diensten hergestellt werden kann.

Gestellte Anforderungen Eine weitergehende Betrachtung der eben geschilderten Zielsetzung bietet (Aarts, 2004), wobei hier in zwei Gruppen differenziert wird. Die eine Gruppe der Anforderungen richtet sich an die physikalischen Aspekte der Umgebung. Die restlichen richten sich an die Intelligenz der Umgebung, bzw. das was eine Umgebung intelligent macht.

Die an die physikalische Umgebung gerichteten Anforderungen sind demnach die dezentralisierte und somit verteilte Verarbeitung von Informationen. Dabei wird diese Verarbeitung auf allgegenwärtigen Komponenten vollzogen, welche überall verfügbar sind. Gleichzeitig muss der Zugriff auf das allgegenwärtige System eine ungehinderte und transparente Interaktion erlauben.

Die Erkennung von sozialen Interaktionen eines Menschen in der Umgebung zählt zu den Anforderungen die an die Intelligenz der Umgebung gerichtet sind. Das den Menschen umgebende System muss demnach im Stande sein den Menschen als solches wahrnehmen zu können. Dafür muss es sich auf diesen anpassen können, von diesem zu lernen und nach Möglichkeit deren Emotionen erkennen.

Der zentrale Gedanke hinter Ambient Intelligence ist das Involvieren des Benutzers. Dabei muss die im Jahre 1970 im Xerox Palo Alto Research Center entwickelte Desktop Metapher aufgegeben werden. Vielmehr muss zu einer neuen Metapher übergewechselt werden, in welcher das natürliche und soziale Interagieren mit einer intelligenten Umgebung möglich werden kann.

Eine weitere essentielle Anforderung an die Ambient Intelligence ist die Akzeptanz solcher Systeme. Diese ist deshalb so wichtig, da sie den Benutzer durchgängig überwachen. Ob sich ein Benutzer damit einverstanden erklärt, hängt von Aspekten wie dem Nutzen des Systems und der natürlichen Interaktion mit dem Benutzer ab. Einige Anregungen des großen Nutzens für den Benutzer werden im folgenden Abschnitt gegeben.

Soziale Aspekte (Bezug zum Sozialen) An dieser Stelle möchte der Autor die sozialen Aspekte der Ambient Intelligence und somit Parallelen zur Sozialen Software herstellen. Ambient Intelligence umschließt menschliche Kultur durch durchgängige Wünsche, komplexe soziale Beziehungen, verschieden wertvolle Systeme, persönliche Vorlieben, Nachhaltigkeit des Ökonomischen und Natürliche Systeme sowie ethische Aspekte. Nach Aussage von

Emile Aarts von Philips Research sind das die Aspekte die den Unterschied zu den Technologien wie Pervasive oder Ubiquitous Computing auszeichnen. Nach (Bechtold, 2007) ist Ambient Intelligence eine Symbiose zwischen zwei Technologien, dem ubiquitären Computing und den menschenzentrierten Benutzerschnittstellen.

In erster Linie wird bei Ambient Intelligence unter sozialer Interaktion eine auf die Benutzer zentrierte Gestaltung von Systemschnittstellen verstanden. Hierbei werden neue Interaktionsmöglichkeiten untersucht, wie ein Mensch mit seiner Umgebung interagieren kann. Die dabei entwickelten neuen Möglichkeiten der Interaktion können dazu beitragen, dass Menschen noch offener gegenüber Computersystemen werden und damit noch mehr dazu ermutigt werden über diese mit Sozialen Systemen zu interagieren.

Nun wenn man die Umgebung selber etwas ausweitet, wird durch eine natürliche Interaktion mit den dahinter verborgenen Informationssystemen eine Verbesserung der Interaktion mit anderen Menschen (Person to Person Kommunikation) möglich. Ein solches Szenario wird auch bei Ambient Intelligence untersucht. Hierbei ist die Zielsetzung soziale Kontakte mit anderen Menschen zum einen einfacher herstellen zu können und zum anderen aufrecht zu erhalten. Damit wird die Bildung einer Gemeinschaft gefördert.

Wie im Abschnitt Soziale Software bereits erwähnt wurde, können bei den immer neuen Anwendungen Menschen durch verschiedene Gründe gespalten werden. Ein solcher Grund kann die fehlende Erfahrung mit dem Umgang solcher Systeme sein. Gerade ältere Menschen (die am stärksten wachsende Bevölkerungsgruppe) kann hierbei den Anschluss verlieren. Ambient Intelligence kann hier als Mediator-Technologie fungieren indem es eine intuitive, natürliche, sozial angepasste Interaktionsmöglichkeit mit Informationssystemen bietet. Mit so gestalteten Benutzerschnittstellen kann damit noch eine größere Anzahl an Menschen an der kooperativen Erzeugung von neuem Wissen teilnehmen. Gerade durch das Involvieren der Älteren, welche die Gruppe mit den höchsten Erfahrungen darstellen, kann Soziale Software noch größere Synergieeffekte bewirken.

Ambient Intelligence berücksichtigt drei Faktoren (auch als drei Ps genannt): den Menschen (People), den Planeten (Planet) und den Ertrag (Profit). Im weiteren Verlauf Ihrer Arbeit formuliert Aarts diese drei Aspekte aus und überführt diese auf Ambient Intelligence.

Die früheren Menschen begannen aus ihren Höhlen zu kriechen um die sie umgebende Welt zu erkunden und damit nach Möglichkeiten zu suchen ihr Leben zu verbessern. Diese Triebkraft existiert in jedem Menschen bis heute. Eine Navigation durch die nähere Umgebung würde den Menschen aufs Neue die Möglichkeit geben diese, nach Möglichkeiten der Verbesserung ihres Lebens zu suchen.

Der Planet kann durch Ambient Intelligence profitieren indem diese Technologie dazu eingesetzt wird Menschen miteinander zu verbinden und damit auf Probleme der Anderen zu sensibilisieren und diese somit weiter zu sozialisieren. Des Weiteren kann Ambient Intelligence

im alltäglichen Leben dazu beitragen, dass der Umweltschutz mehr zum Tragen kommt indem z.B. weniger Müll produziert wird, da der Haushalt optimal durch die Zuhilfenahme von Ambient Intelligence auf das Benutzerverhalten optimiert ist.

Natürlich darf die Motivation für den kommerziellen Einsatz von Ambient Intelligence nicht vernachlässigt werden. Auf der Basis dieser Technologie werden neue Dienste entstehen die leichter direkt den Benutzervorstellungen und dem Einsatzkontext entsprechen. Durch die Intelligenz der Umgebung werden kommerzielle Dienste nicht mehr als störend, sondern als hilfreich empfunden werden, da diese in Verbindung mit einer sofortigen Nutzungsmöglichkeit angeboten werden. Solch ein System wird also dazu einladen kommerzielle Dienste sinnhaft und damit wiederholt zu nutzen.

Die aus der Sozialen Software bekannten Merkmale sind auch in Ambient Intelligence zu erkennen. Die hier zur Verfügung gestellten Systeme orientieren sich in höchstem Maße an dem Benutzerverhalten, wie dies in der Soziale Software ebenfalls zu beobachten ist. Beide Bereiche versuchen eine Offenheit in der Benutzung des Systems höchstmöglich zu fördern. Außerdem ist eine weitere Parallele in dem dynamischen Charakter der Benutzerschnittstellen zu erkennen. Wie bei Soziale Software ist auch bei Ambient Intelligence eine durchgängige Verbesserung des Systemverhaltens äußerst wichtig.

Zusammenfassung

Nicht nur ausgewählte privilegierte Instanzen werden in der Zukunft für die Verbreitung der Medien zuständig sein. Durch neue Technologien wird der gewöhnliche Benutzer dazu befähigt dieses ebenfalls machen zu können. Solch eine Person kann dann über soziale Netze auf ihre Glaubwürdigkeit überprüft werden. Meinungsfreiheit erreicht neues Ausmaß, da die damit verbundenen technischen und sozialen Barrieren bis dahin durch neuartige Konzepte behoben sein werden. Soziales Bookmarking nimmt dann eine Rolle eines Stabilisators für ausgeglichene Versorgung mit Informationen ein. Durch diesen Mechanismus wird ein Weg geschaffen wie ein Mensch mit zufälligen Informationen beliefert wird und damit die Informationsversorgung selber ein viel breiteres Spektrum bekommt.

Natürlich haben die verschiedenen Anwendungen noch mit einigen Schwierigkeiten zu kämpfen. Durch das Fehlen von Verantwortlichkeiten schleichen sich Daten ein die unvollständig, ungenau und deswegen in ihrer Korrektheit wage sind. Auf der anderen Seite entsteht aber bei dem Benutzer auch ein neues Bewusstsein für diese Art von Anwendungen und die damit verbundenen Daten, sodass dieser deren Glaubwürdigkeit erstmals in Frage stellt. Werden Mechanismen wie Reputation der Daten erstmals eingeführt entfällt auch dieser Nachteil.

Die in der Vergangenheit vernachlässigten Synergieeffekte einer Gemeinschaft werden mit heutigen Technologien und neuen Leitsätzen endlich aufgefangen und weiter gefördert.

Durch die Verbindung mehrerer Web 2.0 Technologien werden in der näheren Zukunft Plattformen entwickelt, welche die kollaborative Arbeit einer Gemeinschaft zur vollen Entfaltung bringen können. Die Zusammenschaltung dieser Technologien und der sozialen Netze macht solch eine Plattform zu einer, in der jedes Wissen zurückverfolgt werden kann und damit die Verantwortlichkeitsbeziehungen zugewiesen werden können. Damit wird ein neuer Grad der Qualität von Informationen erreicht. Alles deutet darauf hin, dass die Vision von Vannevar Bush und Dr. Licklider Wirklichkeit werden.

2.3 Ambient Assisted Living

Die hier weiterentwickelte Plattform richtet sich in erster Linie an Menschen, welche in ihrer Mobilität eingeschränkt sind und damit besondere Anforderungen an die Umgebung haben. Damit ist diese Plattform eine Art unterstützendes Mittel mit dem diese Menschen einfacher ihr Leben gestalten sollen. Mit ähnlichen Zielsetzungen setzt sich das Ambient Assisted Living auseinander.

An dieser Stelle möchte der Autor eine Teilmenge der Health Care Computersysteme ansprechen, welches als Ambient Assisted Living Software bezeichnet wird. Ambient Intelligence zielt auf die Möglichkeiten des Einsatzes von Basistechnologien in der Hinsicht auf deren Nutzung in Computersystemen, die Dienste flexibel, transparent und antizipierend zur Verfügung stellen ab. Darauf basierende Systeme stellen eine Möglichkeit dar, wie ein unabhängiges, selbstbestimmendes Leben für behinderte und ältere Mitmenschen ermöglicht werden kann.

Nach (Nehmer u. a., 2006) kann eine Klassifikation der Domänen, in dem das Living Assistance eingesetzt werden kann, vorgenommen werden. Demnach unterscheidet man die Anwendungsdomänen zunächst einmal in zwei Einsatzbereiche, dem für die Gebäude und dem außerhalb dieser. Auf der anderen Seite werden die dort bereitgestellten Dienste nach deren Wichtigkeit in drei Klassen unterteilt. Die wichtigsten Dienste sind solche, die die Lebensfunktionen überwachen und bei Notwendigkeit eine schnelle ärztliche Versorgung unterstützen. Die zweite identifizierte Art von Diensten steht für solche, die ein selbständiges Leben fördern. Darüber hinaus existieren Dienste die nur dem Komfort dienen. Die verschiedenen, sich dabei ergebenden Einsatzfelder werden in Abbildung 2.6 dargestellt.

Ambient Assisted Living Systeme zielen auf die Erfüllung von Anforderungen, die ein Mensch mit Behinderung hat, um seinen Lebensstandard halten zu können, ab. Damit ist die Unabhängigkeit von externen Hilfen (anderen Menschen) in erster Linie gemeint. U.a. zielen diese Systeme ebenfalls auf die Erweiterung der Lebensspanne von älteren Menschen, in der diese selbständig in der für sie bekannten Umgebung weiter leben können, ab. Hierzu wird eine Überwachung der Körperfunktionen und der Aktivitäten der Menschen durchgeführt.

	Emergency Treatment Services	Autonomy Enhancement Services	Comfort Services
Indoor Assistance	emergency prediction emergency detection emergency prevention	cooking assistance eating assistance drinking assistance cleaning assistance dressing assistance medication assistance	logistic services services for finding things infotainment services
Outdoor Assistance	emergency prediction emergency detection emergency prevention	shopping assistance travel assistance banking assistance	transportation services orientation services

Abbildung 2.6: Systemunterteilung nach verschiedenen Domänen des Living Assistance (Nehmer u. a., 2006)

Darüber hinaus wird eine Unterstützung in den alltäglichen Tätigkeiten auch dazugezählt. Aber auch die Verbesserung des Zugriffs auf verschiedene Institutionen wie Notfalldienste, medizinische Versorgung und andere soziale Dienste sind hier gemeint. Natürlich möchte man ein solches System auch dazu verwenden den Menschen die sozialen Kontakte nach außen hin einfacher knöpfen bzw. diese aufrecht erhalten zu lassen. Zuletzt wird auch die Informationsversorgung für diese Menschen eine wichtige Rolle spielen. Deswegen soll das System im Stande sein eine kontextabhängige Informationsversorgung für sie bereitzustellen (Steg u. a., 2005). Die verschiedenen Anwendungsbereiche sind in Abbildung 2.7 (vgl. Huch, 2006) grob skizziert.

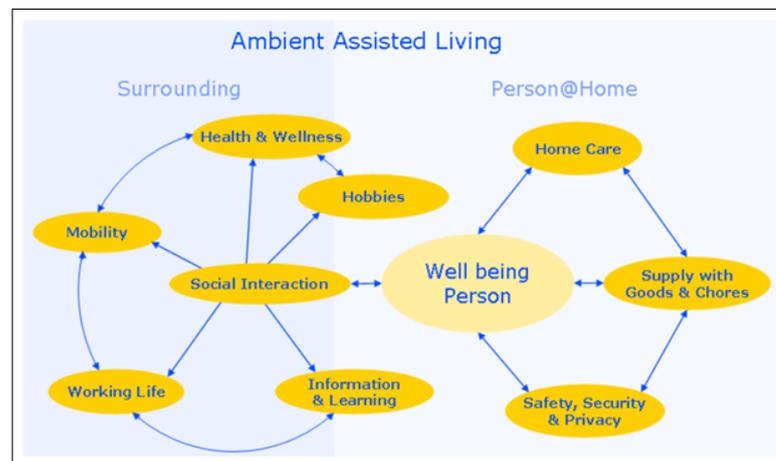


Abbildung 2.7: Aufgabenbereiche für das Ambient Assisted Living (Huch, 2006)

Betrachtet man die Ambient Assisted Living Systeme aus der technischen Sicht, so müssen hierbei verschiedene Anforderungen berücksichtigt werden. Nach (Nehmer u. a., 2006) wer-

den folgende nicht funktionale Anforderungen identifiziert, welche von einem solchen System erfüllt werden müssen:

Robustheit: Widerstandsfähigkeit gegen Fehleingaben.

Verfügbarkeit: Das System muss seiner Aufgaben beim Ausfall einiger Komponenten trotzdem noch nachgehen können.

Erweiterbarkeit: Das System muss im Stande sein jederzeit seine Funktionalität zu erweitern, indem es auf neu verfügbare Ressourcen zugreift und diese für die Erweiterung der Funktionalität nutzt.

Safety: Das System muss so realisiert sein, dass niemals unerwartete Systemzustände auftreten und das System deswegen seiner Aufgabe nicht nachgehen kann.

Sicherheit: Die Regeln für die Überwachung des Menschen müssen sorgfältig definiert werden, sodass die Privatsphäre des Menschen gewahrt werden kann.

Echtzeitverhalten: In Situationen wie Notfälle muss das System sofort reagieren können.

Effiziente Ressourcennutzung: Die knappen Ressourcen müssen zum einen effizient genutzt werden und zum anderen günstig zu beschaffen sein. Dieses impliziert auch die Dauer der Nutzung dieser Ressourcen (Abnutzung, Leistung, Speicher, Reichweite, Bandbreite).

Intuitive, vorausschauende Mensch-Computer Interaktion: Drei Gruppen von Benutzern werden beim Ambient Assisted Living identifiziert, Menschen mit Behinderungen, medizinisches Personal und Wartungspersonal. Alle diese Gruppen müssen entsprechend ihren Bedürfnissen und Fähigkeiten von dem System bei der Interaktion in natürlicher Art und Weise unterstützt werden.

Anpassungsfähigkeit: Einer der Hauptmerkmale eines Living Assistance Systems ist die Fähigkeit sich selbst an gegebene Situationen zur Laufzeit anpassen zu können. Durch die Selbstüberwachung kann das System Tätigkeiten durchführen welche dieses konfigurieren, optimieren und ebenfalls instandhalten.

Werden alle diese Anforderungen von einem Computersystem erfüllt, so wird eine Plattform entstehen, welche dem Menschen in vielerlei Hinsicht das Leben erleichtern wird. Solch ein System wird dann den bedürftigen Menschen die erforderliche Bewegungsfreiheit, Erreichbarkeit sowie die körperliche und geistige Beweglichkeit unterstützend bereitstellen ([Steg u. a., 2006](#)).

2.4 Was ist Trailblazers

Die bisherige Betrachtung gab einen Überblick über die heutigen in der Forschung und in der Industrie interessanten Ansätze zur Modernisierung des Rechnens in der Hinsicht auf den Menschen. Im Abschnitt Soziale Software wurde ein neuer Trend bei der Entwicklung von Systemen, welche die kollaborative Arbeit unterstützen deutlich. Zudem wurden in dem Abschnitt Einfach in der Verwendung - Ambient Intelligence ein Überblick über einen neuen Forschungszweig gegeben, welcher Computersysteme in der Interaktion mit dem Menschen natürlicher gestalten sollen und speziell, wie in Abschnitt Ambient Assisted Living gezeigt, auf die Verbesserung bzw. Haltung der Lebensqualität ausgerichtet sind. Bei allen den genannten Systemen und Technologien wurde die Neutralisierung von Grenzen als Zielsetzung identifiziert, diese haben aber auch neue, datenzentrische Barrieren geschaffen. Bei der Überwindung dieser, könnte die in Abschnitt Semantisches Web kurz beschriebene Technik helfen. Aus Sicht des Autors können alle diese Ansätze gemeinsam zu einer barrierefreien Welt verhelfen. Einen Beitrag hierzu ist das hier betrachtete Navigationssystem, welches aus den hier betrachteten Bereichen des neuen Rechnens viele Anregungen entnahm, um Menschen die an einen Rollstuhl gebunden sind das Leben zu vereinfachen.

2.4.1 Szenario

Bevor mit der Beschreibung des Trailblazers-Systems begonnen wird, sollte das hier zugrunde liegende Szenario beschrieben werden, welches als Motivation für die Erstellung eines solches Systems diene. In diesem Szenario ist ein Rollstuhlfahrer der Protagonist. Dieser möchte von einem Ort A zum Zielort B gelangen. Dabei überquert dieser möglicherweise ihm unbekannte Stecken.



Abbildung 2.8: Rollstuhlfahrer als Hauptnutzer des Trailblazers-Systems

Nachdem dieser den grünsten Abschnitt ohne fremde Hilfe überwunden hat (Abbildung 2.8), stößt dieser unvorhergesehen auf eine Treppe. Eine Treppe ist für Nichtrollstuhlfahrer keine Besonderheit auf die man im Alltag ständig trifft (Abbildung 2.9). Für einen Rollstuhlfahrer stellt aber eine Treppe ein unüberwindbares Hindernis dar. Natürlich kann dieser die in der Nähe sich befindenden Menschen bitten ihn bei der Überquerung des Hindernisses zu helfen, dies ist aber eher in den seltensten Fällen möglich. Damit bleibt diesem nur die Umgebung nach Möglichkeiten abzusuchen, die ihn das Wiederaufnehmen des Weges erlauben. Dieses kann sich aber je nach Lage unterschiedlich schwierig gestalten, sodass der Rollstuhlfahrer möglicherweise sein Vorhaben abbricht.



Abbildung 2.9: Treppe als Ausprägung einer unüberwindbaren Barriere für einen Rollstuhlfahrer

Dem Rollstuhlfahrer kann aber in diesem Fall geholfen werden indem die heute verfügbaren Informationstechnologien dazu verwendet werden, diesem den richtigen, rollstuhlgerechten Weg aufzuzeigen. Dieser nutzt hierzu sein SmartPhone und einen GPS-Empfänger und erfragt mit diesem nach Wegalternativen um von der jetzigen Position ans Ziel gelangen zu können. Die dort gespeicherten Weginformationen beinhalten Erfahrungen anderer Rollstuhlfahrer, welche möglicherweise auch vor diesem Problem standen und selbständig einen Weg gefunden haben. Als Teil einer Community haben diese Benutzer dann diese Weginformationen anderen Benutzern zur Verfügung gestellt, sodass die Routenberechnung auf geprüften Daten basieren kann. Nach der Berechnung wird dem Benutzer die Route gezeigt und optional mit akustischen Hinweisen durch die barrierereiche Umgebung durchgelotst (Abbildung 2.10).

Im Falle unseres Protagonisten hat dieser das Trailblazers-System nach einer Möglichkeit konsultiert wie er in ein bestimmtes Gebäude gelangen kann, ohne die Treppe benutzen zu müssen. Das System erkannte, dass bereits ein anderer Rollstuhlfahrer diesen Weg begangen hatte und die Information verfügbar gemacht hat. Darauf basierend lotste das System



Abbildung 2.10: Trailblazers-System auf einem Smartphone die der Rollstuhlfahrer konsultieren kann um Barrieren-freie Wege zu bekommen.

den Rollstuhlfahrer an die nächst gelegene, speziell für diese Menschengruppe konzipiert Rampe, über die der Protagonist an sein Ziel gelangen kann (Abbildung 2.11).



Abbildung 2.11: Speziell für die Rollstuhlfahrer konzipierte Rampe die von dem Trailblazers-System erkannt und dem Rollstuhlfahrer vorgeschlagen wurde.

2.4.2 Community

Trailblazers ist ein Navigationssystem welches basierend auf den Konzepten der Sozialer Software und den Web 2.0 Technologien entwickelt wurde. Damit ist in erster Linie das Gemeinschaftliche im Vordergrund. Dieser Aspekt greift zentral bei der Realisierung dieses Systems ein indem dieses auf Kartenmaterial basiert, welches nicht wie üblich von einer zentralen Stelle teuer eingekauft werden muss. Kartenmaterial welches beim Trailblazers-System verwendet wird, wird von dem Benutzer des Systems selber erstellt. Der Benutzer ist

wie beim Web 2.0 Ansatz nicht nur der Konsument der Information. Dieser ist gleichzeitig der Produzent und ist in der Lage die in Form von digitalisierten Karten das Trailblazers-System mitzugestalten. Dieser gestaltende Prozess wird in der Abbildung 2.12 veranschaulicht.

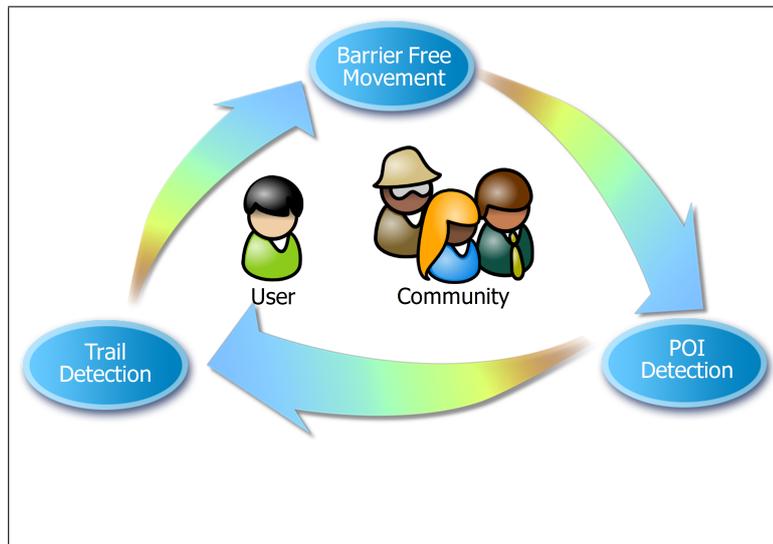


Abbildung 2.12: Trailblazers gestaltender Prozess der Karten-Erstellung und deren Nutzung

Durch diese aktiv gestaltende Rolle ergeben sich einige neue Möglichkeiten, die gewöhnliche Navigationssysteme nicht bieten können. Motivierend aus kommerzieller Sicht ist, dass ein solches System dem Nutzer kostenlos zur Verfügung gestellt werden kann, da das Kartenmaterial von ihm selbst generiert wird. Dieser Aspekt ist sehr wichtig, da hierdurch eine Benutzung durch eine breite Community erleichtert wird, wie es in dem Web 2.0 Konzept stark motiviert wird. Es existieren daneben aber weitere, sehr innovative Nebenerscheinungen, welche das Trailblazers-System zum Vorschein bringt.

2.4.3 Daten

Auf der Trailblazers-Navigationsplattform werden Geoinformationen auf zwei Wegen in die gemeinsame, zentral gehaltene Wissensbasis eingefügt. Der Benutzer nutzt sein mobiles Telefon mit der Trailblazers-Navigationssoftware um von einem Startpunkt zum Zielpunkt zu gelangen. Wurde die Route berechnet lässt der Benutzer sich von seinem mobilen Gerät zum Ziel führen. Während der Bewegung passiert aber mehr als nur die Auslesung der berechneten Navigationsdaten. Um die aktuelle Position des Benutzers zu Berechnen greift das mobile Telefon auf das GPS-Gerät zu und entnimmt von diesem die GPS-Koordinaten des Benutzers. Diese werden zum einen dazu verwendet die aktuellen Routen gegebenenfalls

der aktuellen Position anzupassen. Nach der Berechnung verwerfen die meisten Navigationsgeräte diese Daten, Trailblazers tut das nicht. Diese wertvollen Positionsdaten werden mit Zusatzinformationen angereichert und gespeichert. Bei der nächst möglichen Verbindung mit dem Navigationsserver werden diese dann übertragen und ausgewertet. Diese neuartige Auswertung berechnet aus den Daten neue Informationen, welche dazu verwendet werden die begangenen Wege auf ihre Relevanz zu annotieren. Diese Methode gleicht der Metapher eines Trampelpfades, welcher immer deutlicher wird desto mehr Leute ihn benutzen. Dies ist beim Trailblazers-System realisiert. Das System zeichnet dadurch zum einen die Verkehrsdichte und zum anderen die Aktualität des Weges auf. Wird ein Weg eine lange Zeit nicht begangen, so kann dieser nach einer bestimmten Zeit als nichtrelevant gekennzeichnet werden, wodurch er von der Navigationskarte entfernt wird.

Um aber neue Wegdaten erzeugen zu können, muss man sich nicht notwendigerweise auf bereits bekannten Wegen bewegen. Das eben kurz eingeleitete Verfahren funktioniert auch mit Positionsdaten die auf neuen unbekanntem Pfaden gesammelt werden. Ein Benutzer kann jeden ihm persönlich bekannten Weg beschreiten und die dazugehörigen Daten sammeln. Das später angewendete Verfahren erkennt dann einen neuen Weg und trägt diesen in die Datenbasis ein. Mit dieser Methode wird der Benutzer quasi zu einem Kartographen gemacht.

Wie zu Beginn angedeutet stellt die Trailblazers-Plattform einen weiteren Weg zur Verfügung um Navigationsdaten in die Wissensbasis einfügen zu können und damit die Umgebung zu kartographieren. Jeder Benutzer hat die Möglichkeit von zu Hause aus jederzeit über ein entsprechendes Werkzeug, welches in Form eines Webclients zur Verfügung gestellt wird, auf alle Daten zuzugreifen. Dieser, dem Prinzipien der Sozialen Software entsprechender, auf Web 2.0 Technologien aufbauender Client, erlaubt das Arbeiten mit den gesammelten Positionsdaten sowie dessen Korrektur. Zudem kann der Benutzer durch ihn auch neue Pfade in die Datenbasis eintragen. Mit diesem Werkzeug wird der Nutzer in die Lage versetzt alle erforderlichen Feinarbeiten zur Korrektur von Daten durchführen zu können.

2.4.4 Aktualität

Die Trailblazers-Plattform bietet nicht nur eine neuartige Möglichkeit des Kartographierens von unbekanntem Landflächen. Eine ebenfalls wichtige Eigenschaft der gesammelten Daten ist deren Aktualität. Gerade für Benutzer mit hohen Ansprüchen an die Navigation sind auf aktuelle Daten angewiesen. Daten, deren Gültigkeit nicht mehr zutrifft, können zu einer Irreführung der Nutzer führen. Diese Gefahr, die den meisten Navigationssystemen innewohnt, wird von dem Trailblazers-Ansatz erkannt und behandelt. Wie im Abschnitt Daten angedeutet, werden vorhandene Wege durch Wiederbegehen sozusagen in ihrer Eigenschaft der

Aktualität auf den neuesten Stand gebracht. Hierdurch wird nicht nur die Relevanz des wegen, sondern auch seine Aktualität zum Ausdruck gebracht. Jeder Benutzer bestätigt durch Wiederbegehen die Gültigkeit eines Weges.

2.4.5 Benutzergerechte Navigation

Die Anreicherung der Positionsdaten mit neuen Daten anhand deren die Gültigkeit und Relevanz der resultierenden Pfade ersichtlich wird, führt zur neuartigen Navigationsdaten. Das Trailblazers-System geht einen Schritt weitere und erweitert die Positionsdaten um ein weiteres Merkmal, den Benutzerkontext. Dahinter verbirgt sich die abstrahierende Markierung der Positionsdaten mit einzelnen Benutzerklassen bzw. deren Bewegungskontext. Ein Rollstuhlfahrer hat ganz andere Anforderungen an eine Route als ein gewöhnlicher Fußgänger. Um eine rollstuhlfahrgerechte Route berechnen zu können wird hierbei zwischen Wegen, welche von Rollstuhlfahrern bereits durchquert wurden und Wegen wo dies nicht der Fall ist unterschieden. Letztere sind dabei bereits von Nichtrollstuhlfahrern durchquert worden. Gewöhnliche Navigationssysteme machen hierbei keinen Unterschied und schlagen deswegen Wege vor, die möglicherweise für einen Rollstuhlfahrer unmöglich zu nehmen wären.

2.4.6 Neuartige Daten

Zur Unterscheidung der berechneten Pfade nach den verschiedenen Benutzerklassen kommt eine weitere Eigenschaft hinzu, die Benutzerbewertung (engl. Rating) eines Pfades. Jeder Benutzer kann nach dem Durchqueren eines berechneten Weges seine Meinung zur Brauchbarkeit dieser Route für ihn abgeben. Hiermit wird eine Art Klassifizierung von Pfaden in das System eingeführt, welches von dem Ansatz hier mit dem in Web 2.0 bekannten Rating vergleichbar ist. Hinzu kommt noch eine Möglichkeit jede Strecke beliebig mit Deskriptoren zu annotieren sodass hierdurch eine vergleichbare Folksonomie zur den begangenen Wegen entsteht. Wie die heutigen Web 2.0 Anwendungen zeigen, ist eine solche soziale Indexierung weit mächtiger als jeglicher Algorithmus, sodass vermehrt diese Daten als Grundlage für eine Suche verwendet werden. Das Trailblazers-System macht auch hier von diesem Prinzip der Sozialen Software gebrauch und bietet die Möglichkeit begangene Wege zu klassifizieren. Diese, von dem Benutzer generierte Bewertung, wird dann zum einen für die Wiederfindung konkreter Wege verwendet. Eine hierbei entstehende wichtige Innovation ist, dass die hierbei entstehenden Daten auch zur Berechnung von Routen verwendet werden. Die hierbei abgebildete Meinung der Benutzer, welche zu bestimmten Benutzerklassen gehören, kann Rückschlüsse auf die Qualität der Wege für diese Benutzerklasse erlauben. Also wenn mehrere Rollstuhlfahrer einen Weg als höchst befahrbar klassifiziert haben, ist es

folglich sinnvoller diesen Pfad bei der nächsten Berechnung der Routen wiederzunehmen, anstatt einen der vielleicht nur erschwert befahrbar ist.

Als neuartige Daten werden von dem Trailblazers-System auch sog. Points of Interests gesehen. Diese stellen eine Möglichkeit dar Strecken durch zusätzliche multimediale Daten anzureichern. Es ist dann möglich an jedem Ort einen solchen POI anzulegen um eine gesonderte Stelle auszuzeichnen und mit verschiedenen Mitteln zu beschreiben. Diese Daten können dann Zuhause bei der Berechnung der Route berücksichtigt werden oder vor Ort über das mobile Telefon abgerufen werden. Dabei kann es sich um Beschreibungen von Rollstuhlrampen, Öffnungszeiten, aber auch um Echtzeitdaten wie Busfahrpläne handeln. Alle diese Daten sollen eine weitere Hilfe sein, mit der es dem Benutzer ermöglicht wird, sich einfacher in einer unbekanntem Umgebung fortzubewegen.

Alle hier gesammelten Daten weisen die gleiche Eigenschaft wie bei Sozialer Software auf. Mit steigender Benutzerzahl steigt auch der Wert der Daten. Dieses ist hierbei damit begründet, da mit der steigenden Benutzerzahl die Gültigkeit, Aktualität und Reichhaltigkeit der Daten steigt. Durch diese Tatsache wird die Qualität der Navigation gesteigert und führt natürlich zur besseren Zufriedenheit der Benutzer. Durch die Abbildung des Benutzerverhaltens in anonymisierter Form können sich aber auch neue Einsatzgebiete eröffnen die später im Verlauf der Arbeit erläutert werden.

2.4.7 Selbstbestimmend Leben

Trailblazers greift gleichzeitig auch im Umfeld des Ambient Assisted Living ein, indem es das selbständige Leben von Rollstuhlfahrern steigert. Bisher waren die Rollstuhlfahrer in ihrer Bewegung zum einen von einer weiteren Person abhängig, die unterstützend bei der Bewegung wirkte. Zum anderen hatten sich diese Menschen einfach nicht getraut selbständig in die so barriere reiche Welt hinauszugehen. Das Trailblazers-System stellt hierbei einen möglichen Ausweg aus dieser Gegebenheit indem es einem Rollstuhlfahrer die Möglichkeit gibt Informationen von anderen Rollstuhlfahrern dazu zu verwenden selbständig die Welt zu erkunden. Erkennt ein Rollstuhlfahrer dass bereits ein anderer diesen Weg begangen hat, wird er mutiger und motivierter diesen Weg selber zu befahren und damit weiteren Rollstuhlfahrern die Sicherheit über die Gültigkeit dieses Weges vermitteln. Zudem können mehrere Rollstuhlfahrer über die Trailblazers-Plattform neue soziale Kontakte mit anderen Rollstuhlfahrern knüpfen und mit diesen gemeinsam versuchen die Welt zu erkunden. Damit ermöglicht die Trailblazers-Plattform in vielerlei Hinsicht die Steigerung der Lebensqualität von Rollstuhlfahrern indem dessen Soziale Interaktion durch verbesserte Mobilität gesteigert wird.

Eine Navigation durch die nähere Umgebung würde den Menschen aufs Neue die Möglichkeit geben, diese nach Möglichkeiten zur Verbesserung ihres Lebens abzusuchen. Zudem werden, durch die hier eingesetzten Technologien, Menschen miteinander verbunden

und damit auf Probleme der Anderen sensibilisiert, folglich also sozialisiert. Die Trailblazers-Navigationsplattform ist gleichzeitig auch ein Kommunikationskanal durch den positionsabhängige und damit dem Einsatzkontext entsprechende Dienste angeboten werden können, welche hierdurch besser den Benutzervorstellungen entsprechen. Analog zur Ambient Intelligence, werden durch die Wahrnehmung der Kontextinformationen dem Nutzer hilfreiche Zusatzdienste zur Verfügung gestellt. Die Trailblazers-Plattform mit für Rollstuhlfahrer geeigneten Geräten kann deswegen zu Recht als Lebensunterstützendes System bezeichnet werden.

2.4.8 Trailblazers - Collaborative Mapping

Balram & Dragicevic setzen sich in (Balram und Dragicevic, 2006) verstärkt mit den Eigenschaften von Systemen auseinander, die auf die Bereitstellung von Kartenmaterial spezialisiert wurden. Diese Systeme werden unter anderen auch in der Hinsicht auf den Ansatz der Community-Unterstützung näher betrachtet. Nach Aussage der Autoren wird aktuell verstärkt in dieser Richtung geforscht. Dabei steht die Zusammenführung der Vorteile der Community-gestützten und der geographischen Informationssystemen im Vordergrund. In diesem Zusammenhang wird von „*Collaborative Mapping*“ gesprochen. Nach Balram & Dragicevic setzt sich das Collaborative Mapping aus vier Eigenschaften zusammen, die in der Abbildung 2.13 verdeutlicht werden.

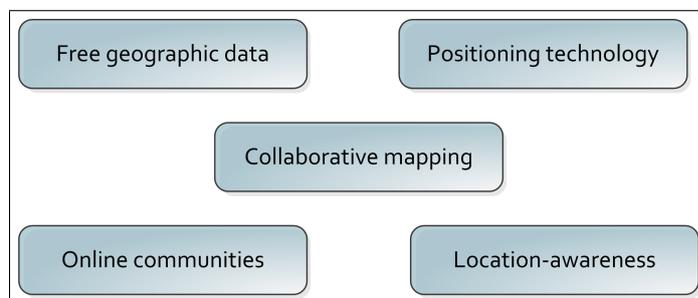


Abbildung 2.13: Collaborative Mapping und seine Eigenschaften nach nach (Balram und Dragicevic, 2006)

Betrachtet man das Trailblazers-System, erfüllt dieses die in der Abbildung 2.13 dargestellten Eigenschaften. Dieses basiert auf freien, durch Benutzer des Systems gesammelten geographischen Daten. Diese Positionsdaten werden über das GPS-Modul gesammelt. Diese Positionierungstechnologie dient auch gleichzeitig der Bestimmung der Position des Betreffenden. Zudem werden im Trailblazers-System gleichzeitig Ortsbezogenen Dienste abhängig von der Position angeboten. Und zuletzt bildet die Community die eigentliche Grundlage für das System. Demnach ist Trailblazers eine für Collaborative Mapping geeignete Plattform.

2.5 Zusammenfassung

In diesem Abschnitt wurde zunächst eine chronologisch Betrachtung der Entstehung der Systeme durchgeführt, die mit dem Merkmal Community-gestützt ausgezeichnet sind. Von der Entstehung bis zur Gegenwart wurden hierbei die beiden Ansätze Soziale Software und Web 2.0 vorgestellt. Dabei wurde gleichzeitig eine Begriffsklärung durchgeführt. Als nächstes wurde eine technisch versierte Betrachtung der dabei zugrunde liegenden Technologien durchgeführt. Die in dieser Untersuchung ermittelten Techniken wurden auf ihre technischen Merkmale und auf ihre Funktionsweise untersucht. Hierbei wurden die neuen Ansätze sowie die hiermit verbundenen technischen Änderungen und die aus diesen Änderungen resultierenden technischen Konsequenzen verdeutlicht und diskutiert. In diesem Zusammenhang wurden verschiedene Ausprägungen der Systeme und deren Anwendungsgebiete besprochen.

Der wichtigste Abschnitt enthält die Identifizierung und die Analyse der Leitsätze und Merkmale, die von den Community-gestützten Systemen verfolgt werden. Diese wurden aus den vorangegangenen Abschnitten extrahiert und hier diskutiert. Die durch diese zentrale Diskussion hervorgehobenen Prinzipien sind wichtig für die spätere Realisierung des Karteneditors. Dabei wurden die Einsatzmöglichkeiten diskutiert und die daraus resultierenden Verbesserungen beschrieben, welche durch die neuen Ansätze zu erwarten sind. Als Ergebnis wurde hier eine Liste von Eigenschaften erarbeitet, welche bei der folgenden Umsetzung des Karteneditors helfen sollen, den größten Nutzen für die Benutzer und die gemeinschaftliche Arbeit zu bringen. Diese Qualitätsmerkmale kommen dann in Verlauf der Arbeit zum Tragen kommen.

Abschließend erfolgte von Ambient Assisted Living und den dahinter stehenden Prinzipien eine Überleitung zu der Trailblazers-Plattform. Dabei wurde genau anhand eines Szenarios verdeutlicht, was diese Plattform ist und wie diese einem in seiner Mobilität eingeschränkten Menschen, einem Rollstuhlfahrer, das Leben vereinfachen kann.

3 Analyse

3.1 Konkretisierung des Anwendungsszenarios

Bevor mit der konkreten Analyse begonnen wird, möchte der Autor exemplarisch ein Anwendungsszenario vorstellen. Verdeutlicht werden soll, wie sich das Trailblazers-System in Bezug auf die Erstellung von Kartenmaterial verhält. Dabei sollen nur die wesentlichen Vorgänge verdeutlicht werden. Abbildung 3.1 skizziert ein solches Szenario.

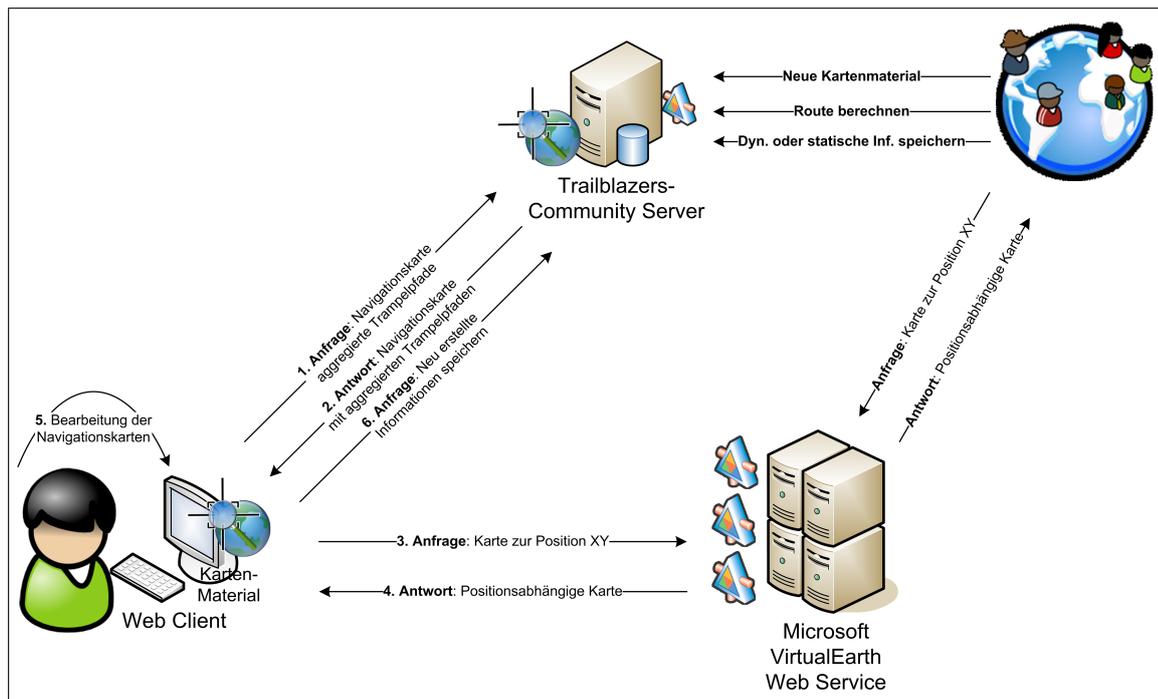


Abbildung 3.1: Skizzierter Verlauf des Anwendungsszenarios

Ein potentieller Benutzer möchte für die Trailblazers-Plattform neues Kartenmaterial erstellen. Mit Kartenmaterial sind in diesem Fall entweder neue Trampelpfade oder Annotationen gemeint. Im ersteren Fall müssen die gesammelten Daten vor der Verarbeitung auf den Server übertragen werden. Dort werden durch den Trampelpfad-Algorithmus neue Wege

erstellt. Nach diesem Schritt stehen die Daten für Modifikationen durch den Benutzer zur Verfügung.

Zu Beginn muss der Benutzer sich über einen Webbrowser mit dem Trailblazers-Community Server verbinden und den dort residierenden Webclient aufrufen. Über diesen kann der Betreffende alle notwendigen Informationen und damit auch den eigentlichen Trail-Editor auf seinen Rechner übertragen. Hierzu muss der Benutzer keine Software installieren.

Jetzt beginnt das in der Abbildung 3.1 dargestellte Szenario. Mit dem Aufruf des Trail-Editors kann der Benutzer nun eine geographische Position auswählen, die von ihm bearbeitet werden soll. Durch die Auswahl der Position werden Anfragen von dem auf dem Benutzerrechner geladenen Trail-Editor gestartet, welche an den Trailblazers-Community Server und an dem Microsoft Virtual Earth Web Service gesendet werden. Diese werden im Folgenden aufgezählt:

1. Zuerst wird eine Anfrage an den Trailblazers-Community Server gesendet, in der dieser aufgefordert wird, Trampelpfade für die vorgegebene Position zu senden.
2. Der Trailblazers-Community Server berechnet für die Fläche verfügbare Trampelpfade und schickt diese an den Client.
3. Als nächstes sendet der Trail-Editor eine Anfrage an den Microsoft Virtual Earth Web Service mit den Trampelpfaden und der Position.
4. Der Virtual Earth Server berechnet nun die resultierende Karte für die vorgegebene Position. Gleichzeitig projiziert dieser die zuvor übergebenen Trampelpfade auf diese Karte. Ist die Karte vollständig berechnet worden, wird diese an den Trail-Editor auf dem Benutzerrechner übertragen.
5. Nun folgt die eigentliche Bearbeitungsphase. In dieser kann der Benutzer die geladenen Trampelpfade nun graphisch auf dem Trail-Editor darstellen und verändern. Was in der Abbildung 3.1 nicht entnommen werden kann ist der Ablauf der Kommunikation während der Bearbeitungsphase. Dieser Sachverhalt würde diese Darstellung des Szenarios nur unnötig unübersichtlich machen, weswegen auch dieses hier nicht explizit aufgeführt ist.
6. Nach dem Beenden der mit der Bearbeitung der Navigationskarte relevanten Arbeiten, kommuniziert der Benutzer über den Trail-Editor die Aufforderung zum Speichern der Daten. Damit werden diese an den Trailblazers-Community Server übermittelt und dort persistent gemacht. Damit ist auch die Bearbeitungsphase abgeschlossen.

Was in Abbildung 3.1 auch dargestellt wird, ist die parallele Nutzung von Daten durch weitere Nutzer. Diese können verschiedene Arbeiten ausüben. In der Skizze dargestellt sind einige Aufgaben verdeutlicht wie das Erstellen von neuem Kartenmaterial, das Annotieren von Kartenmaterial oder einfach die Nutzung der auf dem Trailblazers-Community Server

gespeicherten Daten. Was in der Skizze nicht deutlich wird ist die gegenseitige Bereitstellung von Trampelpfaden. Ebenfalls nicht aus der Skizze zu entnehmen ist die Bereitstellung von Echtzeitinformationen. Diese Aspekte sind aber für die nun folgende Analyse nicht relevant.

3.2 Funktionale Analyse (Fachanalyse)

Abstraktion hilft komplexe Dinge besser erfassen zu können und die dort verborgene Problemstellung zu begreifen. Diese wiederum werden wieder abstrahiert und somit für Lösungen zugänglich gemacht. Diese Vorgehensweise wird auch in dieser Arbeit verfolgt und an dieser Stelle angewendet. Anhand dieser hier durchgeführten fachbezogenen Analyse sollte die erste Systemidee abstrakt erfasst und die dahinterstehende Zielsetzung erkannt werden. Die so entstehenden Vorgaben werden dann bei dem späteren Entwurf berücksichtigt.

3.2.1 Die Systemidee

Sehr wichtig ist zu wissen wie ein zukünftiges System grob aussehen sollte. Eine Systemidee spezifiziert die groben Vorgaben und hilft bei der Identifizierung der Hauptanforderungen. Die groben Eigenschaften sind:

- Systemvoraussetzungen (Rahmenbedingungen)
- Wichtigste Eigenschaften

Betrachtet man unter diesen Gesichtspunkten das Trailblazers-System können folgende Systemeigenschaften (Anwendungsfälle) identifiziert werden:

- Navigation
 - Rollstuhlfahrgerechte Wege
 - Berücksichtigung von auf dem Weg liegenden POIs
 - Online/Offline Betrieb möglich
 - Echtzeit-Navigation im Online-Modus
 - Spontane Wegänderungen
 - Wahrnehmung der Umgebung in Echtzeit (Informationen über Bus, Bahn, Verkehrslage)
- Kollaborative Kartenerstellung
 - Erstellung von Kartenmaterial anhand gesammelter Daten (Semi-Automatisch)

- Annotierung (Anreicherung) des Kartenmaterials durch Zusatzinformationen (POIs)
- Integration von bereits bestehendem Kartenmaterial (Ergänzung)
- Nachbearbeitung von Kartenmaterial
- Teilung von gesammelten Informationen (Community-gestützt)
- Sich kontinuierlich verbesserndes Kartenmaterial (Community-gestützt)

Wie bereits aus der Unterteilung der spezifizierten Funktionalitäten des Systems eindeutig ersichtlich wurde, setzt sich die Trailbazers-Plattform aus zwei wesentlichen Teilen zusammen. Es gibt eine zentrale und eine mobile Komponente. Die mobile Systemkomponente umfasst die Tätigkeiten, welche bei der Fortbewegung durch den Benutzer abgefragt wird. Sie zielt hauptsächlich auf die Nutzung der Navigationskarten vor Ort.

Die zweite Systemkomponente spezialisiert sich auf die konstruktiven Tätigkeiten im Sinne der mit der Produktion der Weginformationen und deren Weiterbearbeitung und Anpassung. Wie aus dem Buch von [die Weisheit der Mengen] zu entnehmen ist, muss ein dezentralisiertes, spezialisiertes Wissen an einer Stelle gebündelt werden. Dies ist erforderlich damit sich die Quintessenz des gesuchten Wissens bilden kann. Diese Bündelung von Informationen wird an dieser zentralen Komponente durchgeführt.

Die für die Navigation wesentlichen Gesichtspunkte werden hier näher analysiert aber nicht realisiert. Die Realisierung wird in der parallel fortgeführten Arbeit durchgeführt (Thome). Ungeachtet dessen ist es für das Verständnis des hier analysierten Konzeptes der Community-gestützten kollaborativen Erarbeitung von Navigationskarten essentiell, diese analytisch näher zu bringen. Die unter dem Punkt der kollaborative Kartenerstellung gesammelten Aspekte sollen nicht nur im Folgenden betrachtet werden. Nachdem der hinter jedem Punkt stehende Anwendungsfall auf seine Merkmale aufgelöst wird, sollte dieser im späteren Verlauf der Arbeit realisiert werden.

Die Analyse dieser Anwendungsfälle und die spätere Realisierung wird nach und nach den Prinzipien des Web 2.0 genügen. Ungeachtet dessen wird aber das System soweit konzipiert sein, dass eine Weiterentwicklung durch die darauf aufsetzende Community vollzogen werden kann und folglich aus Sicht des Web 2.0 als sich kontinuierlich weiter entwickelndes System betrachtet wird. Damit werden die nächsten Schritte in der Evolution dieser Community-gestützten Software getan. Die kontinuierliche, Community-gestützte Weiterentwicklung erlaubt die vom Autor außer acht gelassenen Anforderungen und die damit entstandenen Lücken durch die Gemeinschaft zu füllen. Ungeachtet dessen versucht der Autor an dieser Stelle eine solide Basis für eine solche Plattform zu erarbeiten, auf der dann durch kollektive Nutzung und Erweiterung dieser Plattform, ihr vollständiges Potential entwickelt wird.

3.2.2 Anforderungen eines geographischen Informationssystems

Das Trailblazers-System kann sicherlich ohne Bedenken als ein geographisches Informationssystem bezeichnet werden. Dies spiegelt alle Eigenschaften eines solchen Systems wieder bzw. sollte diese erfüllen. Bevor mit der fachbezogenen Analyse begonnen wird, sollten diese Eigenschaften jedoch identifiziert werden. Diese sollten bei der folgenden Analyse berücksichtigt werden. Bill Kropla setzt sich mit diesem Thema in seinem Buch [Kropla \(2005\)](#) auseinander. Demnach besitzt ein GIS¹ folgenden Eigenschaften.

Methoden der Datenmanipulation

In einer Datenbank eines GIS werden viele verschiedene Daten zusammengetragen. Das Grundgerüst stellen die Vektordaten auf deren Basis weitere Daten zur Annotierung der kartographierten Umgebung hinzukommen. Um diesen Vorgang durchführen zu können muss das GIS verschiedene Funktionen bereitstellen. Die einfachste ist eine Möglichkeit zur Abfrage der gespeicherten Informationen. Diese wird nicht nur von dem Endbenutzer genutzt sondern auch bei dem Sicherungsprozess der Integrität der Daten. Werden Daten ausfindig gemacht, die den definierten Anforderungen nicht entsprechen, müssen diese über weitere Methoden entfernt werden. Zu dem Prozess der Instandhaltung zählen auch weitere Aufgaben, wie das Importieren sowie Exportieren von Daten. Diese beiden Funktionen erlauben einen Austausch und somit eine Synchronisierung von Daten mit anderen Datenquellen. Bei diesen werden unter Umständen weitere Methoden zur Vereinheitlichung von Daten erforderlich sein. Mit deren Hilfe sollen verschiedene Formate² in die Datenbasis importiert, sowie im umgekehrten Weg in andere Formate überführt werden.

Analytische Methoden der Datenabfrage

In der vorherigen Eigenschaft wurde bereits auf die Wichtigkeit der Abfrage-Funktion eines GIS in Hinsicht auf die Instandhaltung der Datenbasis hingewiesen. Ebenso wichtig ist diese Funktion für den Endanwender der mit deren Hilfe verschiedene Fragestellungen beantwortet haben möchte. Diese kommuniziert hauptsächlich über diese Schnittstelle mit dem System. Eine solche Funktion muss in ihrem Umfang verschiedene Anfragemöglichkeiten anbieten um möglicherweise nicht nur einfache einzelne Informationen herauszulesen. Diese muss vielmehr auch darauf ausgelegt sein Anfragen zuzulassen, die auch Zusammenhänge zwischen verschiedenen Informationsartefakten zulässt. Als Beispiel kann hier die Datenbankabfragesprache genannt werden. Diese besitzt eine ausreichende Mächtigkeit um Zusammenhänge aus der riesigen Datenmenge herausfiltern zu können.

¹Geographic Information System

²Unter ([Mitchell und McKenna, 2005](#)) werden die gängigen Formate zusammengetragen und beschrieben.

Räumlich bezogene Annotierung

Vektordaten stellen das Fundament beim Kartographieren dar. Sie bieten die wichtigsten Positionsinformationen um von einem Punkt zum anderen durchgelotst zu werden. In unserem, auf multimediale Ressourcen versierten Zeitalter wäre eine solche Datenbasis sicherlich nicht ausreichend. Deshalb sollten die bereits vorhandenen vielseitigen Multimedien in eine solche Datenbasis integriert werden um den Informationsgehalt zu steigern. Hierzu sollten diese Ressourcen mit dem geographischen Informationen in Beziehung gesetzt werden. Am Beispiel einer einfachen, in digitalisierten Form vorhandenen Karte, welche eine spezielle Information aggregiert wird jedoch ein Problem ersichtlich. Digitalisierte Karten können in verschiedenen Datenformaten und Größen vorhanden sein. Außerdem müssen diese möglicherweise einer Anpassung wie Rotation unterzogen werden, bevor diese korrekterweise mit den Positionsdaten in Beziehung gesetzt werden können. Hierzu muss ein GIS Möglichkeiten anbieten diese erforderlichen Arbeiten an Ressourcen durchzuführen und damit die Annotierung der geographischen Informationen zu unterstützen.

Graphische Integrierung sowie Generierung

Zuvor wurden bereits die spezialisierten Karten angesprochen. Diese bieten unter Umständen eine andere Sichtweise auf eine geographische Umgebung. Mit deren Hilfe könnten unübersichtliche Regionen bzw. Regionen, in welchen die Orientierung durch örtliche Gegebenheiten erschwert wird mit zusätzlichen Informationen angereichert werden, welche diesen Makel abschwächt. Hierzu muss das GIS eine Integrationsmöglichkeit solcher Karten bereitstellen und deren geeignete Darstellung unterstützen. Auf der anderen Seite muss das in dem System enthaltene Kartenmaterial dem Benutzer entsprechend seiner Anforderungen bereitgestellt werden. Bei der Generierung dieser Daten werden zwei Arten der Bereitstellung identifiziert, eine quantitative und eine qualitative. Bei den ersten handelt es sich um eine Bereitstellung von Daten die miteinander in bestimmter Relation stehen. Dabei werden hauptsächlich Annotationen herausgefiltert und zurückgegeben. Die zweite Art der Bereitstellung zielt auf die Generierung der eigentlichen, auf das Verständnis optimierten, räumlichen Darstellung.

Weitere Anforderungen

Neben den genannten Anforderungen können in der Literatur weitere wichtige Eigenschaften, welche ein GIS besitzen soll, identifiziert werden. Demnach sind nach (ESRI, 2007) Funktionen zur Mehrbenutzerverarbeitung von Informationen sowie zur Archivierung von verschiedenen zeitlich abhängigen Zuständen der Karten notwendig. Diese Funktionen des GIS

erlauben das Erzeugen von neuen geographischen Daten durch mehrere Benutzer gleichzeitig und die Untersuchung von Änderungen der kartographierten Umgebung.

3.2.3 Identifizierung der Anwendungsfälle

Aus den hier identifizierten Anwendungsfällen sollen die Vorgaben für das System ersichtlich werden. In der Publikation (Simoes u. a., 2006) wird unter Berücksichtigung der Bedürfnisse eines Menschen, welcher in seiner Mobilität eingeschränkt ist, gezielt auf die dabei möglichen Anwendungsfälle exemplarisch eingegangen. Die hier folgende Betrachtung exploriert die möglichen abstrakten Anwendungsfälle und geht genauer auf deren Aufbau ein.

Abstrakte Anwendungsfälle

Unter abstrakten Anwendungsfällen wird an dieser Stelle die Sicht der Benutzer auf das System unter Abstrahierung der technischen Aspekte verstanden. In dieser Analyse sind die Anwender des Trailblazers-Systems, speziell die an einen Rollstuhl gebundenen Menschen, gemeint. In dieser Betrachtung wird davon ausgegangen, dass diese kein technisches Verständnis für das System haben. Allerdings verfügen sie über fundiertes Wissen zur Nutzung einer solchen Plattform. Durch diese technologieunabhängige Betrachtung sollen die essentiellen Merkmale der mit der Nutzung des Systems verbundenen Absichten extrahiert werden. Damit wird ersichtlich was das Trailblazers-System macht aber nicht wie es dies bewerkstelligt.

Abbildung 3.2 zeigt ein Anwendungsfalldiagramm in dem alle wesentlichen abstrakten Anwendungsfälle graphisch veranschaulicht werden, welche bei der Verwendung des Trailblazers-Systems zu erwarten sind. Zu erkennen sind zwei fundamentale, aggregierende Anwendungsfälle, die Navigation und kollektive Kartenerstellung. Diese beiden fundamentalen Anwendungsfälle repräsentieren gleichzeitig die beiden Kernfunktionalitäten der Trailblazers-Plattform. Weitere abstrakte Anwendungsfälle verfeinern diese beiden übergeordneten, abstrakten Anwendungsfälle und zeigen die groben hierbei anfallenden Teilaufgaben. Diese werden im weiteren Verlauf der Analyse weiter verfeinert werden müssen, sodass im Einzelnen die technischen Aspekte deutlicher werden. Bis dahin sollten aber die einzelnen untergeordneten abstrakten Anwendungsfälle näher spezifiziert werden.

Wie aus der Abbildung 3.2 zu entnehmen ist, werden dort zwei Arten von Benutzern identifiziert, der Benutzer der das System produktiv nutzt und der Administrator der das System betreut. Die Darstellung der, in diesem System identifizierten Akteure ist an dieser Stelle nicht vollständig und bedarf weiterer Auseinandersetzung. Ein Benutzer stellt die Oberklasse aller möglichen Benutzer dar. Dieser repräsentiert hier die Anwendergruppen wie Rollstuhlfahrer, Fußgänger, Mütter mit Kinderwagen und einige andere Gruppen. Natürlich ist auch der

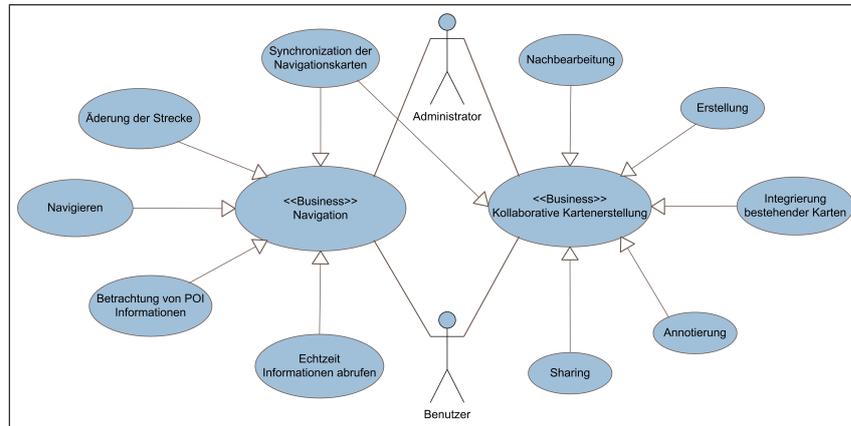


Abbildung 3.2: Identifizierte Anwendungsfälle im Trailblazers-System

Administrator eine Art Anwender. Dieser wird in erster Linie sich mit der Validierung des Kartenmaterials beschäftigen und alle damit verbundenen Tätigkeiten auseinandersetzen. Aus der Abbildung kann man darüber hinaus entnehmen, dass der Administrator die Navigationsaufgaben ebenfalls nutzt. Darunter wird die nachträgliche stichprobenartige Überprüfung der Navigationskarten verstanden.

Navigation Betrachtet man das System genauer, so wird ersichtlich dass die Navigation einen, von den systemtechnischen Aspekten unabhängigen, Anwendungsfall darstellt. Dieser ist in sich höchst abstrakt und aggregiert die darunter liegenden Anwendungsfälle. Bei dem Navigationsanwendungsfall handelt es sich um alle Tätigkeiten die dazu notwendig sind, die einem Benutzer erforderlichen Informationen zu liefern, damit dieser einen geeigneten Weg von Start- bis zum Endpunkt finden kann. Tabelle 3.1 veranschaulicht abstrakt diesen Anwendungsfall.

Bei näherer Betrachtung wird an dieser Stelle ersichtlich, dass der hier beschriebene Anwendungsfall sehr abstrakt ist und damit viele Teilaufgaben in sich aggregiert. Bei letzteren sind wiederum viele Ausnahmen zugelassen. Betrachtet man genauer den Punkt Motivation aus Tabelle 3.1, wird man bereits hier eine Abweichung erkennen. Der Benutzer, der die Navigation in Anspruch nehmen möchte, muss nicht zwangsweise die erforderlichen Eingaben tätigen, wenn dieser sich bereits auf der Straße befindet. Speziell bei den Rollstuhlfahrern wird dieser Fall eher selten auftreten. Aus den Anforderungen dieser speziellen Benutzergruppe wird es für diese sinnvoller sein, den Weg vor dem Antreten der Reise zu planen.

Hierfür wird der Rollstuhlfahrer den Webclient zur Hilfe nehmen und die entsprechenden Eingaben dort tätigen (diese ist speziell für die Bedürfnisse der Rollstuhlfahrer konzipiert und erleichtert damit das Interagieren mit dem System). Nachdem die notwendige Weginformation errechnet wurde, wird diese auf das mobile Gerät übertragen (mit bestehenden

Name	Navigation
Art	Abstrakter Anwendungsfall
Kurzbeschreibung	Ein Benutzer startet die Navigation.
Auslöser, ggf. Motivation	Der Benutzer steht auf der Straße und möchte von einem Punkt A zu einem Punkt B geführt werden. Dazu stellt er seine Anforderung an das System.
Ergebnis	Der Benutzer bekommt die Angeforderte Karteninformation und wird zu seinem Ziel geführt.
Akteure	Keine
Eingehende Informationen	Die Positionsdaten des Start- und Endpunktes. Des Weiteren werden erforderliche Trackinginformationen an das System übertragen.
Vorbedingungen	Karte wurde übertragen und das GPS-Gerät ist bereit. Eine gültige Route wurde berechnet.
Nachbedingungen	Benutzer kommt zum Ziel. Die Tracking Information stimmt mit der Zielposition überein.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Position ermitteln 2. Zielposition ermitteln 3. Route Berechnen 4. Karte Laden

Tabelle 3.1: Abstrakter Anwendungsfall - Navigation

Informationen synchronisiert). Diese Ausnahme hat natürlich Einfluss auf die weiteren Eigenschaften dieses Anwendungsfalles, sollte aber an dieser Stelle nicht weiter verdeutlicht werden. Die hierzu dazugehörenden Einzelheiten werden zu einem späteren Zeitpunkt bei der Konkretisierung der Anwendungsfälle verdeutlicht.

Kollaborative Kartenerstellung Die Erstellung von Kartenmaterial stellt einen weiteren fundamentalen, technisch neutralen Anwendungsfall dar. Dieser beschreibt die grundsätzlichen Tätigkeiten, welche bei der Erstellung von Navigationskarten durchgeführt werden. Bereits bei der Betrachtung der Kurzbeschreibung dieses Anwendungsfalles, welche in der Tabelle 3.2 spezifiziert ist, wird die Diskrepanz zum Namen dieses Anwendungsfalles ersichtlich. Wie bei der Navigation ist diese Abweichung gleich zu motivieren. Diese wird durch die hohe Abstraktionsstufe hervorgerufen.

Unter diesem abstrakten Anwendungsfall werden alle Tätigkeiten zusammengefasst, die bei der Interaktion mit den Karteninformationen anfallen können. Damit sind Tätigkeiten wie Erstellung, Nachbearbeitung, Annotierung, Integration von bestehendem Kartenmaterial und das spätere Sharing gemeint. Wie ersichtlich wird stellen diese ebenfalls abstrakte Anwen-

Name	Kollaborative Kartenerstellung
Art	Abstrakter Anwendungsfall
Kurzbeschreibung	Ein Benutzer möchte eine Navigationskarte bearbeiten
Auslöser, ggf. Motivation	Der Benutzer möchte die neuen Trampelpfade einfügen beziehungsweise bestehende nachbearbeiten.
Ergebnis	Der Benutzer fügt der Karte neue Trampelpfade hinzu bzw. bearbeitet die bestehende nach.
Akteure	Keine
Eingehende Informationen	Benutzerbezogene Daten, Änderungsdaten, Positionsdaten
Vorbedingungen	Gültige Schreibrechte, Gültige Berechtigungen.
Nachbedingungen	Änderungen sind valide
Essentielle Schritte	<ol style="list-style-type: none"> 1. Voraussetzungen für die Bearbeitung der Karte prüfen 2. Karte Editieren (Mehrere einzelne Schritte) 3. Änderungen überprüfen und übernehmen 4. Mit der Datenbank synchronisieren

Tabelle 3.2: Abstrakter Anwendungsfall - Kollaborative Kartenerstellung

dungsfälle dar. Diese und die aus dem abstrakten Anwendungsfall Navigation resultierenden Anwendungsfälle sollen im Folgenden extrahiert und näher betrachtet werden.

Navigieren Unter dem abstrakten Anwendungsfall *Navigieren* (siehe Tabelle 3.3) werden Tätigkeiten verstanden, die erfolgen müssen, damit ein Benutzer vom Start- bis zum Zielpunkt von dem System geführt wird. Um diesen Anwendungsfall zu ermöglichen muss der Benutzer vor dem Antreten der Reise die notwendigen Informationen auf das mobile Gerät laden, damit diese unterwegs verfügbar sind. Zusätzlich hierzu muss der Benutzer den GPS-Empfänger einschalten damit seine aktuelle Position erfasst werden kann und diese Information an das mobile Gerät übertragen wird. Auf den Weginformationen basierend konsultiert das mobile Gerät den GPS-Empfänger nach den Positionsdaten und ermittelt daraus die nächsten Bewegungsvorschläge, welche vom Benutzer wahrgenommen werden muss. Diese werden auf dem Display des mobilen Gerätes graphisch dargestellt.

In Zusammenhang mit der graphischen Darstellung der Information auf dem mobilen Gerät, müssen natürlich die energierelevanten Aspekte mit betrachtet werden. Möglich sind zwei Formen der Bereitstellung der Information, die Dauerhafte und die bei Aufforderung. Die erste durchgängige Bereitstellung der Information ist aus heutiger Sicht möglicherweise nicht praktikabel, da die Laufzeit der Akkus dafür noch nicht ausgelegt ist. Die zweite Variante

Name	Navigation
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Den Benutzer auf der ausgewählten Strecke führen.
Auslöser, ggf. Motivation	Benutzer steht am Startpunkt und möchte mit Hilfe des Systems bis zum Zielpunkt geführt werden.
Ergebnis	Benutzer befindet sich am gewünschten Zielpunkt.
Akteure	Keine
Eingehende Informationen	Positionsdaten (Tracking)
Vorbedingungen	Eine gültige Navigationskarte mit der Wegbeschreibung wurde auf das mobile Gerät geladen.
Nachbedingungen	Trackinginformationen stimmen mit den Positionsdaten des Zielpunktes überein.
Essentielle Schritte	1. Tracking starten

Tabelle 3.3: Abstrakter Anwendungsfall - Navigieren

bietet sich an dieser Stelle eher an. So werden die notwendigen Informationen beim Einschalten des mobilen Gerätes angefordert und berechnet. In der dazwischen liegenden Zeit sollte das Gerät nicht unnötig Energie verbrauchen und wird somit ausgeschaltet bleiben.

Dieser Anwendungsfall benutzt weitere zum Teil optionale Anwendungsfälle. So ist es notwendig, dass der Benutzer vor dem Beginn der Navigationsphase die notwendigen Daten auf das mobile Gerät lädt, also die *Synchronisation der Navigationskarten* durchführt (hierbei ist eine weitere analytische Betrachtung notwendig, die eine Unterscheidung der Betriebsart in einem Online/Offline Szenario durchführt). Auf dem Weg liegende Orte können optional durch die Betrachtung von Informationen näher untersucht werden (*Betrachtung der POI Informationen*). Natürlich kann auch eine Wegänderung während der Bewegung erforderlich sein (*Änderung der Strecke*). Ein weiterer optionaler Anwendungsfall ist die *Echtzeitinformation abrufen*. Die Beziehungen unter den Anwendungsfällen werden im weiteren Verlauf dieser Analyse spezifiziert.

Änderung der Strecke Bei der Ausführung der Navigation kann es gelegentlich dazu kommen, dass die berechnete Route Informationen in die Berechnung mit einbezieht die nicht mehr aktuell sind und somit der Realität nicht entsprechen. Ein solcher Fall tritt z. B. dann auf, wenn auf der barrierefreien Route ein unerwartetes Hindernis vorgefunden wird. Hierbei handelt es sich nicht um die einzige Ausnahme bei der ein solcher Fall auftreten kann. Man stelle sich vor man ist unterwegs und muss auf einmal seine Route wechseln und sich zu einem

ganz anderen Ziel begeben als ursprünglich geplant. Die beiden Sonderfälle verdeutlichen, dass eine spontane Änderung der Route gelegentlich sinnvoll erscheint. Die geschilderten Sonderfälle muss das System erfassen und deren Durchführung ermöglichen können.

Bei den eben beschriebenen Situationen werden also Eingaben vom Benutzer erforderlich, durch welche dieser den Start- und Endpunkt der Route neu spezifizieren kann. Anhand der Eingaben und auf dem Gerät vorliegenden Navigationskarten wird eine neue Route berechnet und dem Benutzer auf dem Display präsentiert. Anhand dieser kann der Benutzer seine Navigation fortführen indem dieser entweder die Barriere überwindet, eine alternative Wegstrecke berechnet oder ein neues Ziel spezifiziert.

Name	Änderung der Strecke
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Eine alternative Route während der Navigation zu einem Ziel berechnen.
Auslöser, ggf. Motivation	Eine unvorhergesehene Barriere taucht auf dem Weg oder eine Zieländerung wird als notwendig erachtet.
Ergebnis	Dem Benutzer wird eine neue Route berechnet.
Akteure	Keine
Eingehende Informationen	Startposition, Endposition, (optional: die Aufnahme der Barriere in das System)
Vorbedingungen	Eine gültige Navigationskarte mit der Wegbeschreibung wurde auf das mobile Gerät geladen. Zudem müssen auf dem mobilen Gerät Trampelpfade gespeichert sein, damit eine Routenberechnung durchgeführt werden kann (sind implizit in der Navigationskarte enthalten).
Nachbedingungen	Eine alternative Route wurde von dem Start- bis zum Endpunkt berechnet.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Eingabe des Startpunktes (optional: GPS kann diese Information liefern) 2. Eingabe des Endpunktes 3. Berechnung der Route 4. Navigationskarte synchronisieren (beziehungsweise den berechneten Weg auf dieser neu darstellen lassen).

Tabelle 3.4: Abstrakter Anwendungsfall - Änderung der Strecke

Auch bei diesem abstrakten Anwendungsfall (siehe Tabelle 3.4) sind die Ausnahmen bzw. Abhängigkeiten zu den technischen Rahmenbedingungen zu diskutieren. Diese sollten hier kurz angedeutet werden, im Abschnitt *Systemtechnische Anwendungsfälle* aber weiter verfeinert werden. Als erster Kritikpunkt sollte hier die Speicherung von vollständigen Navigationskarten auf dem mobilen Gerät erachtet werden. Ursprünglich wird auf dem Gerät nur die Karte geladen die notwendig ist um die Wegstrecke erfolgreich abzugehen (abzufahren). Das

Gerät sollte auf keinen Fall mit einer zu großen Menge von Trampelpfaden überlastet werden. Hierzu müssen geeignete Strategien ausgearbeitet werden, die z.B. die Speicherung und damit die Identifizierung von Daten erlauben, welche am wahrscheinlichsten wiederverwendet werden müssen.

Aber auch wenn der Speicherplatz kein Problem mehr darstellen sollte tritt noch ein zweiter, viel wichtigerer Punkt in den Vordergrund. Wird von der Annahme ausgegangen, dass sich alle notwendigen Weginformationen auf dem mobilen Gerät befinden, so könnte die wiederkehrende Ausführung des Routingalgorithmus die Energieproblematik wieder aufleben lassen. Der Autor sieht hierbei zwei folgende Probleme auftauchen: Die Route basiert auf Daten die veraltet sein könnten. Zum anderen wird das Gerät möglicherweise die Route berechnen können, aber durch hohen Energieverlust diese nicht mehr den Benutzer anzeigen können.

Bei diesem Anwendungsfall muss klar zwischen einem Online und Offline Szenario unterschieden werden. Diese beiden Szenarien bieten unterschiedliche Möglichkeiten an, wie man an die hier betrachtete Problemstellung herangehen kann. Das Onlineszenario bietet bei den eben geschilderten Problemfällen einen Ausweg. Unter der Annahme, dass eine Wegänderung eher eine Ausnahme darstellt und somit eher selten auftritt kann der Benutzer durch Zusatzkosten eine Anfrage an den Server über das verfügbare Funknetz senden. Davon abgesehen ist davon auszugehen, dass die Verfügbarkeit des Mobilens Internets in der näheren Zukunft auf hohen Zuspruch und somit schneller Verbreitung stößt. Durch einen Online (oder zum Teil Online) Betrieb würden diese beiden Problemfälle dann nivelliert werden.

Die detaillierte Auseinandersetzung mit diesen Problemstellungen sollte jedoch in dieser Arbeit nicht weiter fortgeführt werden. Hierzu wird auf ([Thome](#)) wo diese Problemstellung einer genaueren Untersuchung unterzogen wird und hierfür geeignete Strategien zur deren Nivellierung entwickelt werden.

Synchronisation der Navigationskarten Basierend auf dem Anwendungsfall Route neu Berechnen (*Änderung der Strecke*) ist dieser auf einen in diesem Anwendungsfall nicht dargestellten Anwendungsfall *Route zusammenstellen* zurückzuführen. Diese beiden abstrakten Anwendungsfälle ähneln sich vollständig (zumindest aus der abstrakten Sicht) und werden hier nicht weiter differenziert betrachtet. Der einzige semantische Unterschied ist der Zeitpunkt in dem diese durchlaufen werden. Bei beiden wird es aber notwendig die dabei durchgeführten Tätigkeiten durch eine weitere zu ergänzen. Diese verbirgt sich hinter dem Anwendungsfall *Synchronisation der Navigationskarten*.

Tabelle 3.5 verdeutlicht die Einzelheiten die hinter diesem Anwendungsfall stehen. Der Benutzer wird nach der Zusammenstellung der notwendigen Informationen, in den meisten Fällen also eine Navigationskarte, diese auf sein mobiles Gerät übertragen wollen. Dazu muss

dieser sich mit der Trailblazers-Plattform verbinden und die bereits verfügbare Navigationskarte (optional auch andere notwendige Informationen) auswählen, die zwischen den beiden Systemkomponenten ausgetauscht werden soll.

Name	Synchronisation der Navigationskarten
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Hierbei handelt es sich um Tätigkeiten, die notwendig sind, um eine berechnete Navigationskarte mit der entsprechenden Route von der Trailblazers-Plattform auf das mobile Gerät zu übertragen.
Auslöser, ggf. Motivation	Eine neue Route wurde berechnet und sollte auf dem mobilen Geräte verfügbar gemacht werden und somit während der Navigation.
Ergebnis	Das Gerät ist mit aktuellen Daten und Weginformationen versorgt.
Akteure	Keine
Eingehende Informationen	Navigationskarten (damit inbegriffen: Verschiedene andere Karten, auf dem Weg liegende POIs).
Vorbedingungen	Eine gültige Navigationskarte wurde ermittelt und die Verbindung zwischen den Systemkomponenten wurde hergestellt.
Nachbedingungen	Mobiles Gerät enthält alle notwendigen Informationen.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Beide Systemkomponenten verbinden 2. Navigationskarte auswählen 3. (Optional: Andere Informationen zusammenstellen) 4. Karte übertragen

Tabelle 3.5: Abstrakter Anwendungsfall - Synchronisation der Navigationskarten

Auch bei diesem Anwendungsfall haben die technischen Aspekte Online/Offline einen Einfluss auf deren Betrachtung, sodass hieraus zwei verschiedene systemtechnische Anwendungsfälle resultieren, wie ersichtlich wird. Zum Einen kann die Synchronisation vor dem Antritt der Reise durchgeführt werden. Dort kann eine Verbindung mit dem System problemlos aufgebaut werden. Zum anderen kann eine Synchronisation während der Fortbewegung gewünscht sein. In diesem Fall müssen andere Anforderungen an die eigentliche Kommunikation gestellt werden.

Um die beiden Komponenten, die Trailblazers-Plattform und das mobile Gerät, miteinander kommunizieren lassen zu können, muss zwischen Ihnen eine Schnittstelle definiert sein. Dabei stehen in erster Linie die technischen, gerätespezifischen Herausforderungen im Vordergrund, welche später genauer betrachtet werden sollen.

Bei dem Austausch von Informationen müssen zwei weitere Aspekte diskutiert werden. Wichtig ist dabei darauf hinzuweisen, dass es sich hinter der Synchronisation um einen Aus-

tausch von Daten in zwei Richtungen handelt. Es werden nicht nur Navigationskarten auf das Gerät geladen, es werden auf der anderen Seite Trackinginformationen auf die Plattform übertragen. Dabei wird ein weiterer Aspekt deutlich, die Berücksichtigung der Auswahlkriterien, durch die klar ersichtlich wird, welche Daten übertragen werden sollen und welche nicht. Ein solcher Auswahlmechanismus muss im Stande sein, die Verschiedenartigkeit der Informationen, welche zwischen den beiden Systemkomponenten übertragen werden, unter Kontrolle zu bekommen. Dabei kann es sich nicht nur um Trackinginformationen und Navigationskarten handeln. Der Benutzer kann z.B. bestimmte Annotierungsinformationen als wichtig erachten und diese deshalb verfügbar auf dem mobilen Gerät haben. Auf der anderen Seite möchte dieser eine, während der Navigation aufgenommene Annotierungsinformation auf die Trailblazers-Plattform laden um diese der Gemeinschaft zur Verfügung zu stellen. Ein Synchronisationsmechanismus muss verschiedene Aspekte, die hierbei zum Vorschein kommen, berücksichtigen.

Um die Routenberechnung auf dem mobilen Gerät ermöglichen zu können, müssen auf diesem vollständige Informationen, auf welchen die Routingberechnung durchgeführt wird, vorhanden sein. Hierzu müssen möglicherweise Sektoren von bereits kartographierten Arealen ausgewählt werden die auf das mobile Gerät mit übertragen werden sollen.

Betrachtung von POI Informationen Bei der Fortbewegung auf der, durch das Trailblazers-System ermittelten kürzesten und gleichzeitig barrierefreien Route werden auf dieser zusätzliche Informationen für den Benutzer bereitgestellt. Diese so genannten *Point of Interests* stellen Informationenbehälter für jegliche, von der Sicht des Benutzers abhängige, Information dar. Für die hier speziell betrachtete Zielgruppe der Rollstuhlfahrer könnte eine Information z.B. die Steigung der auf dem Weg liegenden Rampe sein. Hinter den POIs kann sich jegliche Art von Information verbergen, die der Rollstuhlfahrer als notwendig erachtet. In besonderen Fällen wird es möglicherweise erforderlich sein diese bei der Navigation vor Ort zu betrachten.

Auf der Navigationskarte werden zur Navigationszeit alle POIs angezeigt³. Der Benutzer kann einen auswählen um die notwendige Information entnehmen zu können. Durch die Auswahl wird eine Darstellung gewählt die für den Benutzer am besten geeignet ist, sodass die Information vollständig aufgenommen werden kann. In der Tabelle 3.6 ist die Spezifikation des hier betrachteten Anwendungsfalles zusammengefasst.

Einige Problemfälle sind in Zusammenhang mit diesem Anwendungsfall zu betrachten. Was ist eine benutzergerechte Darstellung dieser Information? Auf der anderen Seite muss hierbei überlegt werden in wieweit eine Navigationskarte ihre Lesbarkeit behält und wann diese,

³Auch hier mag es sinnvoller sein eine Filtermethode für die Darstellung von Point Of Interests zu haben, durch die der Benutzer entscheiden kann, welche Art wie häufig in welcher Darstellung angezeigt werden sollen.

Name	Betrachtung von POI Informationen
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Beim diesen Anwendungsfall wird die Betrachtung eines Point of Interests, bzw. die dahinter liegenden Informationen verstanden.
Auslöser, ggf. Motivation	(Hier muß noch eine klare Motivation extrahiert werden) Auf dem Weg könnte vielleicht die Information über die Öffnungszeiten wichtig sein.
Ergebnis	Das Gerät zeigt die angeforderte Information an.
Akteure	Keine
Eingehende Informationen	Auswahl des POI (bzw. die Übertragung der POI ID).
Vorbedingungen	-
Nachbedingungen	Darstellung des POIs
Essentielle Schritte	<ol style="list-style-type: none"> 1. Beide Systemkomponenten verbinden 2. Navigationskarte auswählen 3. (Optional: Andere Informationen zusammenstellen) 4. Karte übertragen

Tabelle 3.6: Abstrakter Anwendungsfall - Betrachtung von POI Informationen

durch die Überladung mit einer großen Menge von POIs, diese Eigenschaft wieder verliert. Wie weit reicht es aus diese Informationseinheit auf dem mobilen Gerät zu speichern und somit diese nicht mehr aktuelle Information dem Benutzer bereitzustellen? Man wird hierbei sicherlich zwei Arten von POIs identifizieren, zeitlich kritische und nicht kritische. Die Betrachtung dieser Aspekte findet in dem späteren Verlauf dieser Arbeit statt.

Echtzeit Informationen abrufen Hinter diesem Anwendungsfall (siehe Tabelle 3.7) verbirgt sich die Forderung die Umgebung, in der man sich bewegt, mehr wahrzunehmen. Man stelle sich vor, der Zugriff auf die Bus- und Bahninformationen ist möglich. Ein Rollstuhlfahrer möchte z.B. in seiner bekannten Umgebung nicht unbedingt auf die Navigation sondern auf Informationsquellen (vielleicht spezialisierte POIs) zugreifen. Eine solche Information könnte der nächste, für die Beförderung der Rollstuhlfahrer konzipierte Bus oder die näher liegende Bushaltestelle sein. Mit anderen Worten handelt es sich hierbei um Echtzeitinformationen über die Gegebenheiten in der näheren Umgebung.

Hierzu übergibt er seine Wünsche an das mobile Gerät. Es handelt sich bei den Eingaben um Positionsdaten über einen Ort⁴, über dem man Informationen erfragen möchte. Dann

⁴Die Position des Rollstuhlfahrers kann auch über das GPS-Gerät automatisch ermittelt werden.

spezifiziert er die geforderte Art der Information und einige weiteren Eingaben, die sich von Dienst zu Dienst unterscheiden. Nachdem die Eingaben in das mobile Geräte eingegeben wurden, versucht die Trailblazers-Software die erforderlichen Dienste aufzufinden und bei erfolgreicher Suche dem Benutzer gewünschte Information zu präsentieren.

Name	Echtzeit Informationen abrufen
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Während der Fortbewegung Umgebungsinformationen abfragen.
Auslöser, ggf. Motivation	Gewünschte Nutzung von rollstuhlgerechten Nahverkehrsmitteln.
Ergebnis	Das Gerät zeigt die angeforderte Information an.
Akteure	Entsprechende Informationsquellen
Eingehende Informationen	Positions- und Zeitdaten.
Vorbedingungen	Informationsquellen sind verfügbar.
Nachbedingungen	Gewünschte Information wird angezeigt.
Essentielle Schritte	1. Eingabe notwendiger Informationen

Tabelle 3.7: Abstrakter Anwendungsfall - Echtzeit Informationen abrufen

Die Spezifizierung des Informationsdienstes bringt hierbei neue Herausforderungen zum Vorschein. Wie soll ein Informationsdienst (auch wenn es sich nur um einen Point of Interest handelt) korrekt spezifiziert werden, damit dieser wiederauffindbar ist? Die zweite Frage, die sich hierbei stellt, ist die benutzergerechte Darstellung einer solchen Information auf dem mobilen Gerät. Und die vielleicht interessantere Frage ist, wie kann ein semi- oder vollautomatisiertes Routing hinter einem solchen Informationsdienst sich verbergende Informationen bei der Berechnung der Route mit berücksichtigen?

Erstellung der Navigationskarten Der Kerngedanke, hinter der Trailblazers-Plattform, ist die Nutzung der aus der Community hervorgehenden Vorteile zur Erstellung von qualitativ hochwertigen Navigationskarten. Diese Navigationskarten bestehen aus verschiedenen Komponenten wie Trampelpfaden, Points of Interests und spezialisierten Zusatzkarten. Auf der Basis dieser kooperativ gesammelten und auf einer Navigationskarte aggregierten Informationen finden dann alle mit der Navigation verbundenen Berechnungen statt. Aber bevor überhaupt eine Aggregation und Routenberechnung durchgeführt werden kann, muss eine Navigationskarte erstellt werden.

Durch den Anwendungsfall Erstellen der Navigationskarten (siehe Tabelle 3.8) werden die damit verbundenen Aktivitäten spezifiziert. Der Benutzer muss sich zuerst an der

Trailblazers-Plattform über den WebClient anmelden. Dabei werden die entsprechenden Berechtigungen geprüft. Ist der Benutzer berechtigt eine Karte anzulegen wird dieser eine Karte erstellen. Nachträglich kann er wahlweise für seine neu erstellte Navigationskarte Leserechte/Schreiberechtigungen festlegen. Durch diese kann der Benutzer anderen (Gruppen oder einzelnen Benutzern) seine Karten zum Lesen (durch das Leserecht können die anderen Benutzer diese Karten für ihre Navigation nutzen) bereitstellen. Die Vergabe des Schreiberechts an andere Benutzer führt dazu, daß diese die Navigationskarte mitgestalten können. Ist erstmals eine dieser Karten erstellt, kann der Benutzer seine gesammelten Informationen mit seiner Navigationskarte synchronisieren.

Name	Erstellen der Navigationskarten
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Bevor die gesammelten Navigationsdaten ausgewertet werden, sollten diese in irgendeiner Art in Beziehung gesetzt werden. Hierzu wird eine Karte als Aggregierungs- zum anderen als Teilungsmedium hierfür verwendet. Auf dieser Karte werden dann die erforderlichen Informationen gesammelt.
Auslöser, ggf. Motivation	Ein Benutzer möchte für sich eine Karte erstellen.
Ergebnis	Eine Navigationskarte wird für den Benutzer erstellt auf der er dann seine Daten sammeln kann.
Akteure	keine
Eingehende Informationen	Positions- und Zeitdaten, Points Of Interest, Spezialisierte Zusatzkarten
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Der Benutzer bekommt eine Karte zur Verfügung gestellt.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den WebClient 2. Überprüfung der Berechtigungen 3. Erstellung einer Kartenentität 4. (optional: Schreib/Leserechte setzen) 5. Datenübergabe (Trackinginformationen, POIs, spezialisierte Karten).

Tabelle 3.8: Abstrakter Anwendungsfall - Erstellen von Navigationskarten

Die Bezeichnung dieses abstrakten Anwendungsfalles ist etwas undeutlich formuliert und läßt keine eindeutige Zuordnung zu. Hinter der Erstellung der Navigationskarten verbergen sich weitere abstrakte Anwendungsfälle. Zum einen basiert die Erstellung darauf, dass die gesammelten Weginformationen (welche sich auf dem Gerät befinden) mit der Plattform synchronisiert werden. Dadurch stehen diese Daten der automatischen Verarbeitung zur Verfügung und können von dem Trailblazers-System entsprechend ausgewertet werden. Die

durch diese Auswertung entstehenden Trampelpfade werden dann auf der benutzereigenen Karte angezeigt und der manuellen (kollaborativen) Verarbeitung zugänglich gemacht. Diese nachträgliche Verarbeitung wird durch den abstrakten Anwendungsfall Nachbearbeitung beschrieben.

Ein bei der Erstellung von Navigationskarten ebenso wichtiger Aspekt ist das Hinzufügen von Zusatzinformationen, welche helfen könnten die Strecke erfolgreich zu passieren und damit ans Ziel zu gelangen. Mit anderen Worten eine Karte mit Weginformationen ist noch lange nicht fertig und somit der Erstellungsprozess noch nicht beendet. Um diese entsprechend zu vervollständigen sind weitere Tätigkeiten erforderlich. Diese werden durch die beiden abstrakten Anwendungsfälle *Annotierung und Integration bestehender Karten* abgedeckt.

Hinter der Erstellung verbirgt sich noch eine weitere für die Navigation wesentliche Tätigkeit. Bevor man die Navigation startet ist es erforderlich eine Strecke mit den erforderlichen Informationen zusammenzustellen. Bei diesem Prozess verändert der Benutzer das Kartenmaterial nicht. Stattdessen stellt dieser für sich ein Darstellungsprofil und die Wegstrecke für die Navigationskarten zusammen. Diese wird nach dieser Erstellungsphase zusammen mit den zusätzlichen Informationen auf das mobile Gerät geladen.

Zu diskutieren an diese Stelle ist, ob die an dem Benutzer delegierte Vergabe des Leserechts das System bzw. der hinter diesem stehende Idee der Trampelpfade nicht schadet. Angenommen alle Benutzer vergeben keine Leserechte an Dritte, so kann die Bildung von Trampelpfaden unterbunden werden. Der in dem System integrierte Algorithmus bewertet das Verhalten mehrerer Benutzer und erst daraus kann dieser eine Annahme über die Aktualität des Weges aufstellen.

Nachbearbeitung der Navigationskarten Es ist zu erwarten, dass durch technische und räumliche Gegebenheiten die Aggregation von Informationen zum Teil nicht das vom Benutzer gewünschte Ergebnis liefern. Mit anderen Worten werden bei der Erstellung von Navigationskarten ungeeignete Informationen bei der Synchronisierung in diese einfließen. Da ist es selbstverständlich, dass ein Werkzeug dem Benutzer zur Verfügung gestellt werden muss, mit dem dieser die erforderlichen Änderungen an der Navigationskarte vornehmen kann. Durch die damit verbundenen Tätigkeiten werden die Informationen wieder an die Realität und an die Benutzervorstellungen nachträglich angepasst werden können.

Bei der Nachbearbeitung muss sich der Benutzer zuerst über den Webclient an das Trailblazers-System anmelden. Es ist wichtig zu überprüfen ob der Benutzer auch gültige Berechtigungen hat, damit dieser nur die Informationen verändert für die er Schreibrechte besitzt. Ist die Anmeldung und die Prüfung der Privilegien erfolgreich durchgeführt worden, beginnt der Benutzer die Modifikation der Navigationskarte. Nach der erfolgreichen Veränderung der gewünschten Informationen wird eine Prüfung dieser Veränderungen nach deren Validität mit bestehenden Vorgaben durchgeführt. Nachdem diese erfolgreich durchlaufen

wurde, werden die Änderungen persistent in dem Trailblazers-System gespeichert. In der Tabelle 3.9 ist die Spezifikation des hier betrachteten Anwendungsfalles zusammengefasst.

Name	Nachbearbeitung der Navigationskarten
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Durch das Fehlverhalten und falsche Berechnungen werden verfälschte Daten produziert. Diese müssen nachträglich wieder korrigierbar sein. Die damit verbundenen Tätigkeiten werden durch diesen Anwendungsfall zusammengefasst.
Auslöser, ggf. Motivation	Ein Benutzer möchte seine Karte nachträglich verändern.
Ergebnis	Die gewünschte Änderung der Navigationskarten wurde erfolgreich durchgeführt.
Akteure	keine
Eingehende Informationen	Änderungsdaten und somit zum Teil: Positions- und Zeitdaten, Point Of Interest, Spezialisierte Zusatzkarten
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Die Veränderung ist gültig und wurde von dem System übernommen.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Kartenentität Veränderungen unterziehen 4. Prüfung der durchgeführten Änderungen 5. Übernahme der Änderungen

Tabelle 3.9: Abstrakter Anwendungsfall - Nachbearbeitung der Navigationskarten

Unter dem fünften Punkt bei den essentiellen Schritten wird von der Veränderung der Navigationskarte gesprochen. Durch genauere Betrachtung wird es ersichtlich, dass sich hinter dem Verändern auch das Neuerstellen verbirgt. Genauer gesagt, eine Navigationskarte wird nicht nur durch die Änderung der bestehenden Daten verändert. Diese wird gleichzeitig durch das Hinzufügen neuer Informationenartenfakte wie POIs, spezialisierte Karten letztendlich ebenfalls neu gestaltet.

Die Veränderung bestehender Daten ist aber nicht nur für den einzelnen Benutzer interessant. Verleiht man in der Community Aufsichtsrollen, durch welche die Benutzer quasi zu Administratoren von Informationen werden, brauchen diese noch dringender eine solche Möglichkeit Veränderungen durchzuführen. Dieser Anwendungsfall beschreibt gleichzeitig eine Möglichkeit, welche ein System (Community) benötigt um sich selbst organisieren zu können und damit die qualitativ besten Informationen extrahieren zu können. Die hier grob

beschriebene fundamentale Aktivität erfordert sicherlich eine noch differenzierte Betrachtung.

Integration bestehender Karten Es ist ersichtlich, dass vor der Zeit der Einführung des Trailblazers-Systems viele, mit vergleichbarem Verwendungszweck, also der Hilfestellung bei der Navigation, Karten entstanden sind. Das Trailblazers-System stellt eine verbesserte Alternative hierzu und bietet neue Möglichkeiten an, um die Navigation effizienter zu gestalten. Das bestehende Kartenmaterial unberücksichtigt beiseite liegen zu lassen wäre aber sicherlich eine zu große Verschwendung. Zum anderen würden die Menschen die auf ihr bestehendes Kartenmaterial vertrauen zu dem Trailblazers-System überwechseln, wenn dieses Ihre Informationen berücksichtigen würde. Der abstrakte Anwendungsfall *Integration bestehender Karten* umschließt alle Aktivitäten, welche für diese Aufgabe notwendig sind.

Der Benutzer hat auf seinem Rechner spezialisierte Karten in digitalisierter Form vorliegen. Für die Integration dieser Karten in die bestehende Navigationskarte muss sich der Benutzer wiederum über den Webclient an das System anmelden, wo er als erstes auf seine Privilegien überprüft wird. Nach der Auswahl einer seiner Navigationskarten wird diese Karte in das System übertragen. Hierdurch wird das System in die Lage versetzt, beide Karten übereinander zu legen. Natürlich sind auch noch weitere Parameter notwendig, welche für die Nachjustierung der Karte erforderlich sind. Danach werden die Benutzereingaben validiert. Nach der erfolgreichen Validierung wird der Algorithmus, der die Verschmelzung der Karten durchführt, gestartet. Danach wird das Ergebnis dem Benutzer zur Überprüfung präsentiert. Entspricht das Ergebnis der Benutzervorstellung bestätigt er dieses, wodurch die Informationen in dem System persistent gespeichert werden. In der Tabelle 3.10 ist die Spezifikation des hier betrachteten Anwendungsfalles zusammengefasst.

Die hier angedeutete Beschreibung ist in sich unvollständig und wird durch systemtechnische Anwendungsfälle später näher beleuchtet. Zu erwarten ist, dass hierbei durch die Verwendung externer Werkzeuge neue Anwendungsfälle entstehen werden.

Annotierung der Navigationskarten Des Öfteren wurde von Points Of Interest gesprochen. Diese stellen in dem Trailblazers-System eine Möglichkeit dar, die dort zusammengestellten Navigationskarten durch weitere Informationen zu annotieren. Ein Point Of Interest steht für eine, für den Benutzer dieser Karten interessante, Informationseinheit. Dieser ist als Referenzpunkt für die eigentlichen, hinter diesem liegenden Informationen gedacht, durch den diese Informationen wiederauffindbar und in einer normierten Form zusammengefasst werden können. Die hinter dem Referenzpunkt liegende Menge an Informationen ist nicht durch bestimmte Arten begrenzt. Damit soll der Prozess der Annotierung weitgehend offen gestaltet werden können.

Name	Integration bestehender Karten
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Es werden spezialisierte Karten zu den bestehenden hinzugefügt. Diese können dann übereinander gelegt werden und damit eine neue Sichtweise (und somit Information) liefern. Der Benutzer wählt ein Areal aus indem er eine Karte integrieren möchte, spezifiziert weitere zusätzliche Informationen und fügt diese hinzu.
Auslöser, ggf. Motivation	Ein Benutzer möchte seine Karte durch weitere spezialisierte Karten erweitern.
Ergebnis	Die bestehende Navigationskarte wird durch die hinzugefügte Karte ergänzt.
Akteure	keine
Eingehende Informationen	Spezialisierte Zusatzkarten und für die Integration erforderlichen Daten.
Vorbedingungen	Navigationskarte muss existieren. (Bestimmte Vorgaben muss die spezialisierte Karte erfüllen).
Nachbedingungen	Die Karte wurde in die bestehende Navigationskarte integriert.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Uploaden der Karten 4. Eingabe der erforderlichen Daten 5. Validierung der Information 6. Prüfung der automatischen Integration 7. Speicherung der Daten

Tabelle 3.10: Abstrakter Anwendungsfall - Integration bestehender Karten

Nachdem der Benutzer über den Webclient seine Navigationskarte abgerufen und somit die notwendigen Sicherheitskriterien erfüllt hat, wird diesem die Möglichkeit geboten seine Navigationskarte durch POIs zu ergänzen. Hierzu sucht dieser eine dafür vorgesehene Position und legt dann den POI dort an. Nachdem dieser angelegt wurde, kann der Benutzer diesen durch die eigentlichen Informationen anreichern. Dazu kann der Benutzer Textbeschreibungen, Verweise auf andere Ressourcen wie Webseiten und Bilder hinzufügen. Des Weiteren kann für diesen POI ein Bewertungswert (Ratingwert) abgelegt werden, welcher diesen nach seinem Informationsgehalt bewertet. Ist die Bearbeitung abgeschlossen wird das System nach entsprechenden Regeln diese Informationen prüfen und danach im Trailblazers-System speichern. In der Tabelle 3.11 ist die Spezifikation des hier betrachteten Anwendungsfalles zusammengefasst.

Name	Annotierung der Navigationskarten
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Die hier zusammengefassten Tätigkeiten werden bei der Anreicherung der Navigationskarten durch zusätzliche Informationen ausgeführt. Dabei wählt ein Benutzer eine Navigationskarte und fügt dieser einen Point of Interest hinzu und erweitert diesen um neue Informationen.
Auslöser, ggf. Motivation	Benutzer möchte eine für ihn relevante Information zu der Navigationskarte hinzufügen.
Ergebnis	Die bestehende Navigationskarte wird durch den hinzugefügten POI ergänzt.
Akteure	keine
Eingehende Informationen	Hinter einem Point of Interest können sich verschiedenartige Informationen verbergen. (Text, Links, Bilder, Rating?).
Vorbedingungen	Navigationskarte muss existieren, damit dieser POI persistent gespeichert werden kann.
Nachbedingungen	Der POI würde in die bestehenden Karten integriert.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Erstellung eines POIs an der gewählten Position 4. Erweiterung des POIs um Informationen 5. Validierungsphase 6. Speicherung in dem System

Tabelle 3.11: Abstrakter Anwendungsfall - Annotierung der Navigation

Dieser abstrakte Anwendungsfall beschreibt grob die mit der Annotierung verbundenen Aufgaben. Durch die Abstraktionsstufe ist deren Granularität sehr grob gewählt worden. Bei der Sichtung dieser Spezifikation wird aber sehr schnell klar, dass hier eine weitere Verfeinerung notwendig ist. Dies scheint gerade für den vierten essentiellen Schritt *Erweiterung des POIs um Informationen* dringend notwendig zu sein. Dort sammeln sich mehrere mögliche Anwendungsfälle, welche den Prozess des Editierens der verschiedenen zugrundeliegenden Ressourcen spezifizieren.

Für die spätere maschinelle Verarbeitung ebenfalls unabdingbar ist die Klärung der Struktur eines POIs. Auf der anderen Seite muss betrachtet werden, ob Vorgaben von dem System die Freiheit eines Benutzers (somit der gesamten Community) zu sehr einschränken und somit die freie Entfaltung des Systems hindern können. Hier bedarf es einer genaueren Analyse der Vor- und Nachteile.

Eine zusätzliche Beschreibungskomponente ist der Rating Mechanismus. Dieser kann auf mehreren Wegen in diesem System zum Einsatz kommen. Dieser muss nicht unbedingt nur die Qualität der Informationseinheiten widerspiegeln. Durch die Einführung weiterer Ratingmarkierungen (Bewertungsdeskriptoren) kann der Benutzer z. B. eine Bewertung durchführen, aus der hervorgeht wie gut hatte dieser POI einer bestimmten Profilgruppe bei der Navigation geholfen. Hieraus können gruppenrelevante POIs extrahiert werden. Durch diese Information kann eine qualitative Verbesserung bei der Berechnung der Routen die Folge sein.

Sharing der Navigationskarten Unter dem Sharing von Ressourcen wird bei der Trailblazers Plattform das Austauschen von Navigationskarten und der damit verbundenen Ressourcen verstanden. Dieser Anwendungsfall (siehe Tabelle 3.12) ist für die Entfaltung und den Fortbestand der Trailblazers-Plattform essentiell. Die sich dahinter verbergende Idee ist die Teilung von bereits gesammeltem Wissen über kartographierte Landflächen. Durch das Sharingkonzept soll die Community im Stande sein jedes Gebiet kartographisch zu erschließen. Gerade weil die Navigationskarten mit jedem Benutzer geteilt werden, auf dem von jedem Benutzer in erster Linie nur seine kleine Umgebung kartographiert wird, erschließt sich bei der Bündelung aller Navigationskarten früher oder später jedes wesentliche Gebiet. Die Trailblazers-Plattform baut auf zwei Merkmalen auf, der hohen Verteilung der Benutzer und die Bündelung der Information durch das System. Diese beiden Merkmale sind eine gute Voraussetzung zur Erstellung neuartiger Navigationskarten. Um diese Vision realisieren zu können ist es dringend erforderlich, dass der Benutzer durch das System Unterstützung bei der Offenlegung des von ihm gesammelten Wissens bekommt.

Um eine Navigationskarte mit der Community teilen zu können ist eine Anmeldung über den Webclient notwendig. Nach der Anmeldung und Prüfung der Privilegien wählt der Benutzer die Navigationskarte, die dieser mit den anderen Benutzern teilen möchte. Als nächstes müssen die Benutzer ausgewählt werden, welche die Erlaubnis zur Teilung der Ressourcen bekommen sollen. Hierbei kann es sich um einzelne Benutzer, Benutzergruppen oder schlicht um alle Benutzer handeln. Nachdem die Navigationskarte den Benutzern zugeordnet wurde, müssen die Privilegien spezifiziert werden, die den Benutzern bei der Teilung der Ressource zugeteilt werden sollen. Anschließend werden die Benutzervorgaben geprüft und in dem System gespeichert.

Außer Schreib/Leserecht existieren möglicherweise noch weitere für den einzelnen Benutzer wichtige Privilegien, durch welche dieser sein Wissen auf der Trailblazers-Plattform richtig verwalten kann. Vorstellbar sind verschiedene Stufen der Sichtbarkeit einzuführen, damit die Anonymität des Benutzers gewahrt bleiben kann. Dadurch kann der Benutzer bestimmte Bereiche der Lesbarkeit entziehen, damit seine Privatsphäre gewahrt bleiben kann. Dieses kann z. B. das, um sein Haus oder Arbeitsstelle, umliegende Areal sein. Die Einführung von

Name	Sharing der Navigationskarten
Art	Untergeordneter abstrakter Anwendungsfall
Kurzbeschreibung	Dabei werden alle erforderlichen Tätigkeiten zusammengefasst, die ein Benutzer ausüben muss, damit seine Navigationskarte mit der Community (oder einzelnen Benutzern) geteilt werden kann.
Auslöser, ggf. Motivation	Benutzer möchte seine Karte der Community (einzelnen Benutzern) bereitstellen.
Ergebnis	Die Community (einzelne Benutzer) haben auf die Navigationskarte Zugriff.
Akteure	keine
Eingehende Informationen	Benutzerinformationen, für die Rechtevergabe relevanten Informationen.
Vorbedingungen	Die ausgewählten Benutzerentitäten und die für die Teilung vorgesehene Navigationskarte existieren. Der Benutzer selbst besitzt die erforderlichen Rechte.
Nachbedingungen	Die Benutzer bekommen die richtigen Zugriffsrechte auf die Navigationskarte.
Essentielle Schritte	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Auswahl der Gruppe und der Ressource 4. Zuweisung der Privilegien 5. Speicherung dieser Information im System

Tabelle 3.12: Abstrakter Anwendungsfall - Sharing der Navigationskarten

sogenannten *Protected Areas* kann hier eine Möglichkeit darstellen dieser Herausforderung entgegenzukommen.

Bei der Rechtevergabe für eine Ressource muss möglicherweise ein weiteres Recht eingeführt werden, durch welches der Benutzer die Weitergabe der Rechteverteilung an weitere Benutzer delegieren kann. Zu klären ist ob ein Benutzer überhaupt möchte, dass weitere Benutzer auf seiner Navigationskarte Rechte bekommen sollen, die wiederum dieses Recht von anderen bekamen. Aus Sicht der Community ist das eher nicht gewünscht.

3.2.4 Resultierende Sicht auf die abstrakten Anwendungsfälle

Bis hierher wurden die wesentlichsten abstrakten Anwendungsfälle identifiziert, die in Zusammenhang mit der Trailblazers-Plattform auftauchen. Bei der durchgeführten Betrachtung

sind einige wenige, nicht eindeutige Merkmale der verschiedenen Anwendungsfälle kurz angesprochen worden. Einige davon verdeutlichten, dass die klare Trennung zwischen diesen abstrakten Anwendungsfällen noch nicht bestand. Hierdurch wurde in einigen Fällen die gegenseitige Abhängigkeit (Beziehung) nicht eindeutig ersichtlich. Die Klärung dieser Unstimmigkeiten ist aber für die spätere Spezifizierung der systemtechnischen Anwendungsfälle unabdingbar. Dieses soll nun an dieser Stelle folgen.

Aus der durchgeführten Analyse wurde ersichtlich, dass der abstrakte Anwendungsfall Navigation sehr stark auf anderen Anwendungsfällen aufbaut. Um eine erfolgreiche Navigation zu bieten sind vier Anwendungsfälle notwendig: *Die Synchronisation der Navigationskarten*, die *Änderung der Strecken*, die *Betrachtung von POI Informationen* und das Abrufen von *Echtzeitinformationen*. Der erste abstrakte Anwendungsfall wird hauptsächlich vor und nach der Navigationsphase durchgeführt. Natürlich wäre auch eine Synchronisation während der Navigation möglich, ist aber aus verschiedenen Gründen nicht praktikabel (Kosten, kein mobiles Internet). Die drei anderen Anwendungsfälle werden während der Navigationsphase durchlaufen. Bei der näheren Betrachtung dieser drei Anwendungsfälle können diese in deren Wichtigkeit für eine erfolgreiche Navigation unterteilt werden. Bei einer relativ neuen Umgebung kann es durchaus vermehrt dazu kommen, dass auf der vorher berechneten Route nichtkartographierte Barrieren auftauchen. Ein ähnliches Verhalten kann auch in wenig besuchten Umgebungen erscheinen. In solchen Fällen ist es ersichtlich, dass eine alternative Route gesucht, also die *Änderung der Strecke* durchgeführt werden muss. Die beiden Anwendungsfälle *Betrachtung von POI Informationen* und *Echtzeitinformationen abrufen* kapseln Tätigkeiten, die die Navigation vervollständigen würden. Diese sind aber für die eigentliche Navigation nicht dringend erforderlich und können somit als variable Zusatzdienste erachtet werden.

Betrachtet man die mit der Änderung der Strecke verbundenen abstrakten Anwendungsfälle, so werden zwei weitere identifiziert: Die Nachbearbeitung und Synchronisation der Navigationskarten. Der Anwendungsfall Synchronisation der Navigationskarten steht an dieser Stelle für zwei, sich sehr ähnelnden Prozesse. Zum Einen ist hier die Synchronisation mit der zentralen Wissensbasis gemeint. Diese wird aber wie zuvor aus kosten- und technischen Gründen zur Zeit als nicht relevant erachtet. Auf der anderen Seite wird darunter die Synchronisation der bestehenden Karten mit den, auf dem mobilen Gerät, neu berechneten Routeninformationen verstanden. Dabei wird ersichtlich, dass hierbei die lokale Übernahme von Änderungen gemeint ist. Die eben geschilderten Zusammenhänge zwischen den einzelnen Anwendungsfällen werden in [Abbildung 3.3](#) verdeutlicht.

Durch diese Analyse zeigt sich, dass die Navigation von sehr vielen anderen untergeordneten Anwendungsfällen abhängt und verdeutlicht somit die Komplexität dieses Anwendungsfalles.

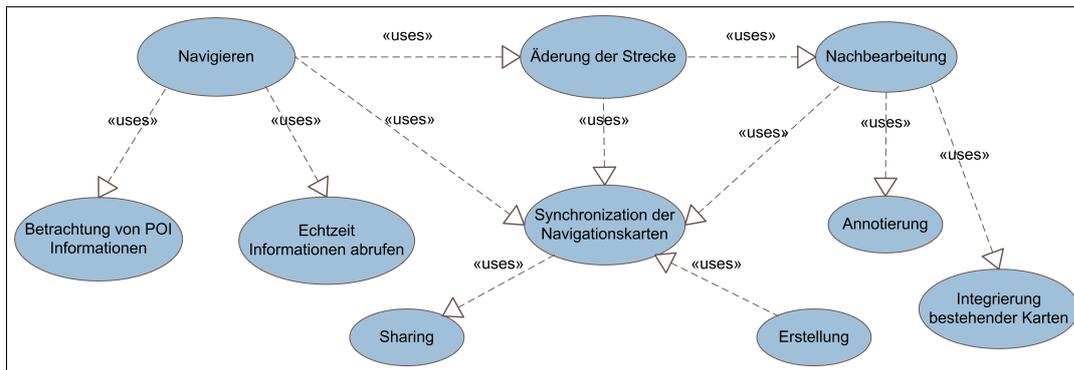


Abbildung 3.3: Beziehungen zwischen den einzelnen abstrakten Anwendungsfällen

3.2.5 Nichtfunktionale Anforderungen

Bei den nichtfunktionalen Anforderungen handelt es sich um Vorgaben an das System, welche nicht an deren eigentliche Funktion gestellt werden. Diese beschreiben vielmehr die allgemeinen Aspekte des Systems, die entscheidend für das Bestehen des Systems sind. Dabei wird in der Literatur von Leistungsbedingungen und besonderen Qualitäten und Randbedingungen, welche eingehalten werden müssen, gesprochen. Dieser Abschnitt basiert zum einen an den Vorgaben, welche in (Tanenbaum und van Steen, 2002) an ein verteiltes System gestellt werden, nimmt aber auch Anregungen aus dem ISO/IEC 9126 (Wikipedia, 2006c) Standard auf. Darüberhinaus beschäftigt sich mit diesem Thema die Software Technik wie in (Kahlbrandt, 2001). Daneben werden die in diesem Kapitel identifizierten Anforderungen zum Teil mit berücksichtigt.

Portierbarkeit

Hinter der Anforderung der Portierbarkeit verbergen sich Aspekte die eine Anwendung erfüllen muss, damit diese auf verschiedene Plattformen eingesetzt werden kann. Dabei spielen die Anpassbarkeit der Anwendung, die Austauschbarkeit und das Koexistieren in einer Umgebung mit anderen Anwendungen eine wichtige Rolle. Natürlich spielt bei den gewöhnlichen Desktopanwendungen auch deren Installation eine wichtige Rolle, bei der hier betrachteten webbasierten Anwendung ist eine Betrachtung dieses Merkmals nicht erforderlich.

Aktuell ist der Internet Explorer der meist genutzte Webbrowser. Daneben existieren aber nichtsdestotrotz noch einige andere, die von der Trailblazers-Plattform ebenfalls unterstützt werden sollen, damit jeder an dem Kartographieren teilnehmen kann. Hierbei muss die Web-Anwendung soweit entwickelt werden um den Kontext, indem diese ausgeführt wird, erkennen zu können. Dieses ist notwendig um die geeignete Realisierung für den einzelnen Webbrowser wählen zu können und damit sich auf seine Einzelheiten angepasst werden kann.

Wird eine Kontexterkennung erfolgreich durchgeführt kann jeder Browser als Ausführungsumgebung für diese Anwendung benutzt werden und damit den höchsten Grad der Portierbarkeit erreichen.

Performance

Das hier zu entwickelnde Portal muss gewisse Anforderungen in der Leistungsfähigkeit erfüllen, damit dieses auf eine große Akzeptanz stößt und damit seinen Fortbestand sichern kann. Es ist ersichtlich, dass die spätere Lösung im Stande sein muss eine Antwort innerhalb einer akzeptablen Zeit zu liefern. Zu große Wartezeiten machen das System unhandlich und bewirken beim Benutzer Unzufriedenheit.

Auf der anderen Seite muss darauf geachtet werden, dass die Performance nicht auf Kosten anderer Prozesse gewonnen wird. Die Basis für die Sozialen Anwendungen sind die Web 2.0 Technologien. Diese sind in dem Browser integriert, sodass die Programme, zumindest die Darstellungsschicht, auf dem Client-Browser ausgeführt werden. Da ist es wichtig darauf zu achten, dass diese Programme dem Client nicht die gesamte Rechenleistung entziehen und damit die anderen Prozesse benachteiligen.

Verfügbarkeit

Unter Verfügbarkeit wird die Zeit, in welcher der Dienst für die Nutzung verfügbar ist, verstanden. Aus Sicht eines Benutzers muss die Verfügbarkeit eines solches Werkzeuges durchgängig gegeben sein, damit dieser neue Routen berechnen kann. Wird der Dienst für die Zusammenstellung der Route häufig nicht verfügbar, wird dieser in Endeffekt sehr schnell nicht mehr verwendet. Aber auch aus Sicht der Trailblazers-Plattform ist es notwendig eine 99 prozentige Verfügbarkeit zu gewährleisten, damit die Benutzer nicht beim Kartographieren gehindert werden. Damit ist die Verfügbarkeit der Plattform essentiell für ihr Bestehen zu sehen und sollte gegeben sein.

Zuverlässigkeit

Unter Zuverlässigkeit wird die Reife der hier zugrunde liegenden Software als Anforderung verstanden. Natürlich entwickelt sich eine Web 2.0 basierte Software kontinuierlich weiter. Ungeachtet dessen sollte diese ein hohes Maß an Reife aufweisen und damit eher selten in Fehlerzustände laufen, da hierdurch die Benutzerakzeptanz rapide gemindert würde.

Ebenfalls unter der Zuverlässigkeit ist die Fehlertoleranz angesiedelt. In verteilten Systemen können viele verschiedene Fehler auftreten. Das System soll bei möglichen Soft- bzw. Hardwarefehlern diese entsprechend verarbeiten können und weiter verfügbar bleiben (Tanenbaum und van Steen, 2002). In diesem Fall handelt es sich um die Fähigkeit eines Systems ein gewisses Leistungsniveau unter auftretenden Systemfehlern halten zu können. Dieses Mindestmaß ist bei dem Webclient schwer zu definieren. Zum Einen ist natürlich aus Benutzersicht eine voll funktionierende Anwendung wünschenswert. Auf der anderen Seite ist die Kernfunktionalität, welche durch den Web-Client angeboten wird, unter der Klassifizierung des Dienstes nach der Wichtigkeit aus Sicht des Ambient Assisted Living Konzeptes als Autonomy Enhancement Service zu betrachten. Diese ist zwar notwendig um die Autonomie der Rollstuhlfahrer aufrecht zu erhalten, ist aber als kein kritischer Dienst zu betrachten. Folglich muss unter Berücksichtigung dieser Tatsache der Dienst keine hohe Fehlertoleranz gewährleisten.

Ein weiteres wichtiges Merkmal eines Softwaresystems aus softwaretechnischer sowie Ambient Assisted Living Sicht ist die Robustheit gegenüber Fehleingaben, mit anderen Worten: Die Beständigkeit gegenüber Fehlnutzung. Das hier entwickelte webbasierte Werkzeug wird einer häufigen Nutzung ausgesetzt, dabei werden viele verschiedene Benutzergruppen mit unterschiedlichen Kenntnissen existieren. Die hier entwickelte Anwendung sollte nach außen hin als ein stabiles System wirken.

Die vorher genannte Vorgabe ist wünschenswert, nichtsdestotrotz sind Systemausfälle nicht ausgeschlossen. Tritt eine solche Situation auf müssen Gegenmaßnahmen vom System gestartet werden, durch die der aktuelle Ausführungskontext mit den dazugehörigen Daten wiederhergestellt werden kann. Wie gut ein Softwaresystem dieses bewerkstelligt sollte hängt von dem Wert der Daten und den damit verbundenem Arbeitsaufwand ab. Bei der Erstellung von annotierten Karten werden sicherlich neue Daten erstellt, diese können aber jederzeit reproduziert werden. Ungeachtet dessen ist es trotzdem wünschenswert, dass das hier entwickelte Werkzeug bei einem Fehlverhalten die bisher getätigten Aktionen für den Benutzer einfach reproduzieren kann.

Pflege und Wartbarkeit (Änderbarkeit)

Die Pflege und Wartung beginnt bei jedem Softwaresystem mit dem Tag an dem diese das erste Mal in Produktion geht. Diese Phase beinhaltet alle Aktivitäten welche dazu dienen das Softwaresystem weiter zu stabilisieren, also die zu diesem Zeitpunkt nicht gefundenen Fehler zu korrigieren. Damit wird eine Möglichkeit geschaffen die später zum Vorschein kommenden Schwächen des Entwurfs mit relativ wenig Arbeitsaufwand zu beseitigen, ohne dabei das gesamte System verändern zu müssen. Hinzu kommt das jedes System sich anders in Produktion verhält als in der Entwicklung. Durch die Pflege und Wartung wird es möglich

dieses System dann im Nachhinein zu optimieren und damit eine Leistungsverbesserung zu erreichen. Hierzu sollte die hier entwickelte Anwendung analysierbar, modifizierbar und prüfbar sein. Sind diese Anforderungen erfüllt wird der Aufwand der Diagnostizierung von Fehlern und ihre Beseitigung verkürzt was auch aus Sicht des Web 2.0 wünschenswert ist.

Das hier zugrunde liegende System muss demnach Methoden anbieten durch die relativ einfach das auftretende Systemfehlverhalten protokollieren. Damit muss der Benutzer (bzw. die Community) ohne tiefgreifendes Wissen in die Lage versetzt werden Mängel oder Ursachen von Versagen des Systems festzustellen.

Indem die Fehler erkannt oder neue Anforderungen identifiziert wurden beginnt nun die Umsetzung der spezifizierten Vorgaben. Der hier entwickelte Webclient muss durch seine Beschaffenheit so gestaltet sein, damit diese Änderung mit relativ wenig Aufwand in das System integriert werden kann. Nachdem diese Arbeiten durchgeführt wurden ist es wichtig ebenfalls die Korrektheit des Systems zu prüfen. Dazu bedingt es geeignete Prüfmethode, welche die Korrektheit des Systems zu einem gewissen Maß widerspiegeln. Der Web-Client wird aber in erster Linie von der Community selber auf seine Richtigkeit überprüft.

Wiederverwendbarkeit

Hinter der Wiederverwendbarkeit verbirgt sich nach ([Kahlbrandt, 2001](#)) die erneute Benutzung eines Systems oder einer Systemkomponente in einem anderen Einsatzkontext. Die Vorteile die hierbei entstehen sind zum einen die signifikante Verkürzung der Entwicklungsphase in einem anderen Projekt. Zum anderen fließt hier aber auch die Erhöhung der Stabilität des Systems mit ein, da dieses auf bewährten Komponenten basiert. Im Kontext des Web 2.0 spielt die Wiederverwendung ebenfalls eine wichtige Rolle und ist u.a. als Beispiel in dem Mashup-Prinzip, wie es in Abschnitt *Komposition von Diensten (Mashups)* eingeleitet wird, wiederzufinden. Hierzu sollte das Trailblazers-System offene Schnittstellen anbieten, welche von jedem ohne fundiertes Wissen verstanden und genutzt werden können um dort zur Verfügung gestellte Funktionalitäten und Daten in einem neuen Kontext verwenden zu können.

Skalierbarkeit

Unter Skalierbarkeit wird in ([Tanenbaum und van Steen, 2002](#)) die Erweiterbarkeit eines Systems um neue Hardware verstanden, wobei das ursprüngliche System hierbei sein Verhalten nicht ändern darf. Diese Erweiterung verfolgt das Ziel der Verbesserung der Performance des Systems. Um dieses zu ermöglichen muss in der Entwurfsphase dieses mitberücksichtigt werden, damit einzelne Systemkomponenten redundant auf der zusätzlichen Ressourcen installiert werden können und damit eine Leistungssteigerung bewirken. Dieses ist im Falle

der Trailblazers-Plattform sehr erwünscht, da deren Fortbestand vom fortwährenden Wachsen der Benutzer gesichert wird. Damit muss die Plattform auf Spitzenzeiten vorbereitet sein, in denen eine viel größere Benutzeranzahl diese nutzen möchte.

Erweiterbarkeit (Offenheit)

Nach (Tanenbaum und van Steen, 2002) sollten verteilte Systeme erweiterbar sein. Dieses geschieht indem neue Dienste oder gemeinsam nutzbare Ressourcen eingefügt werden. Hierbei ist es wichtig das die bereits bestehenden Dienste nicht beeinträchtigt werden. Durch die Offenheit, welche durch das Offenlegen der Spezifikationen und Dokumentationen der wichtigsten Softwareschnittstellen erreicht wird, kann eine Erweiterung des verteilten Systems begünstigt werden. Dieses Merkmal, welches auch in dem Web 2.0 Universum bekannt ist, sollte auch von der Trailblazers-Plattform beachtet werden. Dieser Community-getriebene Ansatz muss auch in seiner Erweiterung offen sein, damit die Wünsche der Community in Hinsicht auf die Verbesserung der Werkzeuge erfüllt werden können. Durch diese Offenheit soll eine zeitnahe Erweiterung dieser Plattform um neue Funktionen, sowie eine Verbesserung der bereits verfügbaren ermöglicht werden.

Benutzbarkeit (Qualität)

Ambient Assisted Living setzt sehr viel Aufwand in die Entwicklung neuartiger, intuitiver und vorausschauender Mensch-Computer Interaktionen. Dieses zielt darauf jedem Benutzer abhängig von seinen Fähigkeiten und Anforderungen eine Möglichkeit zu geben mit dem entsprechenden System mit wenig Aufwand zu interagieren. Aus der systemtechnischen Sicht spielen hierbei Aspekte wie Verständlichkeit, Erlernbarkeit, Bedienbarkeit, Attraktivität und Zugänglichkeit eine wichtige Rolle.

Das Maß der Verständlichkeit eines Systems hängt vom Aufwand ab, welchen ein Benutzer investieren muss um das System bedienen zu können. In dieser Hinsicht, auch im Sinne der Sozialen Software, muss das Werkzeug zur Erstellung von annotierten Karten seine Aufgabe und seine Funktionalität leicht dem Benutzer vermitteln. Zudem muss dieses auch Möglichkeiten anbieten die Funktionsweise leicht zu Erlernen bzw. durch Benutzerführung diesen Lernvorgang zu unterstützen. Hierbei sollten auch die Erkenntnisse aus der Softwareergonomie mit einfließen. Betrachtet man die heutige Soziale Software erscheint ein weiterer nicht zu unterschätzender Aspekt: Die Attraktivität der Anwendung. Beobachtbar ist, dass eine ansprechbare, zeitgemäße Gestaltung der Anwendung parallel mit dem Erfolg dieser läuft. Somit ist dieser Aspekt bei der Realisierung des Werkzeuges mit zu berücksichtigen.

Ein sehr wichtiger Aspekt in diesem Zusammenhang ist die Zugänglichkeit. Unter diesem Aspekt wird im Kontext dieser Arbeit die Schaffung einer geeigneten Arbeitsplattform verstanden, dessen Benutzer in ihrer physikalischen Motorik eingeschränkt sind. Diese Gruppen der Benutzer mit besonderen Anforderungen benötigen neue Methoden um bei der Erstellung der Navigationskarten teilnehmen zu können.

Gleichzeitigkeit

Trailblazers-Plattform ist eine Umgebung, welche das Zusammenarbeiten fördern soll. Dabei werden vermehrt Ressourcen wie Navigationskarten zur gleichen Zeit erarbeitet. Nach [Tanenbaum und van Steen \(2002\)](#) ist unter der Gleichzeitigkeit die Eigenschaft des Systems gemeint, durch die mehrere Prozesse zur gleichen Zeit ausgeführt werden können, ohne sich gegenseitig zu beeinflussen. Ein solches verteiltes System muss sicherstellen, dass bei mehreren gleichzeitigen Zugriffen auf eine Ressource, diese dafür ausgelegt ist und damit in einer nebenläufigen Umgebung sicher ist. Die Trailblazers-Plattform und die damit darauf zur Verfügung gestellten Werkzeuge, müssen diese Anforderung in hohem Maße genügen. Es müssen Mechanismen eingesetzt werden, welche eine Teilung der Ressourcen ermöglichen.

Einfachheit

Die Soziale Software zeichnet sich dadurch aus, dass diese sehr einfach gestaltet und damit für jeden einfach verständlich ist. Zudem ist durch ihre Einfachheit deren Erweiterung begünstigt. Damit versucht dieser Ansatz das besser zu machen, was die zu komplexen Desktopanwendungen aus den Augen verloren haben, nämlich dem Benutzer unterstützend bei der Arbeit zur Seite zu stehen. Diese Anforderung wird auch an die Trailblazers-Plattform gestellt. Diese Plattform soll so einfach wie möglich für den Benutzer gestaltet werden. Gleichzeitig soll die erforderliche Komplexität für diesen transparent gemacht werden.

Modularität

Softwaretechnisch betrachtet muss jedes System modular aufgebaut werden. Damit wird die Unterteilung des Systems verstanden, bei der fachlich gleichartige Komponenten zusammengefasst werden. Wird eine solche Unterteilung durchgeführt, führt dies zur Vereinfachung der Änderbarkeit des Systems sowie zu einem einfacheren Austausch der jeweiligen Module, wobei die hierbei entstehenden Auswirkungen auf das gesamte System minimal gehalten werden können. Die Trailblazers-Plattform wird als eine sich sehr schnell änderbare

Plattform konzipiert. Damit muss bei dessen Entwurf dieser Aspekt durchgehend berücksichtigt werden.

Effizienz

Bei dem hier entwickelten Werkzeug werden keine zeitkritischen Tätigkeiten durchgeführt. Auf der anderen Seite muss diese Soziale Software ein angemessenes Zeitverhalten aufweisen. Die Interaktivität ist für die Attraktivität und Benutzerfreundlichkeit dieser Anwendung unerlässlich sodass diese eine angemessene Antwort- und Verarbeitungszeit bieten muss. Dem Benutzer dieser Anwendung darf nicht zugemutet werden, dass der Betreffende eine deutlich längere Bearbeitung seiner Anfragen duldet und folglich also ein anderes Verhalten in Kauf nimmt, als diesem von den Desktopanwendungen bekannt ist. Demnach muss das hier entwickelte Werkzeug den Vorgaben einer Desktop-Anwendung genügen. Auf der anderen Seite muss darauf geachtet werden, dass die effiziente Bearbeitung nicht auf Kosten anderer Anwendungen durchgeführt wird. Die im Webbrowser geladene Anwendung darf den ausführenden Rechner nicht vollständig auslasten. Mit anderen Worten sollte bei der Realisierung darauf geachtet werden, dass die knappen Betriebsmittel vernünftig genutzt werden.

Sicherheit

Auf der Trailblazers-Plattform werden personenbezogene Daten gesammelt. Dabei handelt es sich um ein Bewegungs-Profil sowie dazu gehörende andere hinzugefügte Informationen. Diese könnten von Dritten missbraucht werden bzw. der Benutzer selber möchte seine Privatsphäre nicht aufgeben. Um diesen entgegenzuwirken muss die hier betrachtete Plattform die Daten in einer anonymisierten Form speichern sodass andere keine Rückschlüsse über den Benutzer ziehen können.

3.3 Nichtfunktionale Analyse

3.3.1 Trailblazers - Herausforderungen und Risiken

An dieser Stelle sollen die Problemfälle besprochen werden, welche das Trailblazers-System angehen möchte. Damit wird die Grundlage für die problembezogene Analyse gegeben. Diese zielt speziell auf die Merkmale, welche aus Sicht eines Benutzers zur Geltung kommen, ab. Der Benutzer, von dem hier die Rede ist, ist ein Mensch, der durch seine Behinderung in seiner Bewegungsfreiheit eingeschränkt ist und somit an einem Rollstuhl gebunden ist.

Seine Fortbewegung erfordert spezielle Wege, welche die Eigenschaft der Barrierefreiheit besitzen.

Es existieren viele Navigationssysteme die speziell auf Straßenkartenmaterial aufbauen. Diese navigieren Fahrzeuge durch ein Netz von Straßen. Ein Problem entsteht dann, wenn ein solches Navigationssystem von einem Benutzer dazu verwendet wird, diesem den kürzesten Fußweg zu berechnen und anhand dieser Weginformation zu navigieren. Das gewöhnliche System besitzt keine Weginformation, somit kann dieses auch nicht den kürzesten Fußweg anzeigen. Diese Betrachtung geht aber noch einen Schritt weiter und zielt auf die Vorgaben eines Rollstuhlfahrers ab. Dabei ist klar zu unterscheiden, dass Wege die für einen Fußgänger geeignet sein können noch lange nicht für einen Rollstuhlfahrer geeignet sein müssen. Mit anderen Worten hat solch ein Navigationssystem noch mehr Vorgaben, als es bei konventionellen Autonavigationssystemen oder der Fußgängernavigation der Fall ist.

Das Trailblazers-System besitzt die hierfür notwendigen Informationen und kann deswegen den besten Weg berechnen und den Benutzer, welcher ohne Fahrzeug unterwegs ist, zum Ziel zu führen. Aber woher kommen diese Daten und wie genau sind diese? Die Antwort auf diese Frage ist sehr einfach; jeder Benutzer besitzt die notwendigen Informationen und durch die zentrale Speicherung wird diese jedem anderen Benutzer zugänglich gemacht. Das Trailblazers-System ist eine Plattform auf der eine Community gemeinsam die notwendigen Daten erarbeitet und der Community zur Verfügung stellt. Natürlich ist in der Startphase die Informationsmenge sehr gering. Das System und somit die Informationsmenge wächst aber mit jedem Benutzer. Mit anderen Worten das Potential des Systems ist abhängig von der Benutzeranzahl und ist somit nach oben offen. Welche Probleme sind aber hier zu erwarten, welche das System an seiner Entfaltung hindern könnten?

Der Kerngedanke des Trailblazers-System ist also die gemeinsame Erarbeitung von Kartenmaterial. Um diese Herausforderung bewältigen zu können ist also eine Plattform notwendig, auf der sich eine Community bilden kann. Hierzu müssen gewisse Voraussetzungen erfüllt werden. Eine solche Plattform muss die Kommunikation zwischen den Community-Teilnehmern ermöglichen. Darauf basierend sollen Benutzer untereinander über die gestellte Problematik diskutieren können. Die zweite Voraussetzung, die erfüllt werden muss, ist die Ermöglichung der Zusammenarbeit zwischen mehreren Benutzern und damit deren Interaktion miteinander. Hierzu müssen entsprechende Werkzeuge zur Verfügung gestellt werden. Sind diese beiden Voraussetzungen erfüllt, können die Teilnehmer dieser Community an der Problemstellung gemeinsam arbeiten.

Trailblazers erfüllt diese Voraussetzungen. Damit ist die Trailblazers-Community dazu befähigt durch die Zusammenarbeit alle notwendigen Informationen zu erarbeiten. Dabei entstehen aber neue Probleme die im Folgenden betrachtet werden sollen.

3.3.2 Zu kleiner Markt

Wieso soll ein System wie Trailblazers genutzt werden? Diese Frage soll anhand folgender Argumentationskette beantwortet werden: Rollstuhlfahrer, Skater, Mütter mit Buggies, das alles sind kleine Gruppen und somit aus kommerzieller Sicht ein kleiner Markt. Durch diesen Umstand wird ersichtlich, dass eine zentralisierte Haltung und Erstellung von Kartenmaterial zu teuer sein würde und somit nicht realisierbar ist. An dieser Stelle tritt das Trailblazers-System in den Vordergrund. Durch seinen dezentralisierten Ansatz der Erstellung von Navigationskarten werden diese Umstände wieder relativiert. Eine Community erstellt ihr Kartenmaterial selbst, wodurch diese Ressource so gut wie kostenfrei sein wird. Zum anderen wird eine so gesehene benutzerzentrische Sichtweise durch Trailblazers stark motiviert. Jeder Benutzer hat die nötigen Werkzeuge um das Kartenmaterial auf seine Wünsche anzupassen. Das System kann natürlich eine Auswertung von Informationen durchführen und von der Darstellung abstrahieren - hierdurch kann jede Information einem anderen Benutzer auf andere Weise präsentiert werden. Dennoch sollte eine Vereinheitlichung der Navigationskarten motiviert werden. Durch den Einsatz der Web 2.0 Technologien bewegt sich aber die Sichtweise des *User Centered Design* noch eine Stufe weiter. Der Benutzer bekommt nicht nur die Möglichkeit das notwendige Kartenmaterial zu erstellen, er kann die Plattform mitgestalten und somit bei ihrer evolutionären Weiterentwicklung mithelfen. Die Weisheit der Mengen tritt hier zum Vorschein.

3.3.3 Barrieren

Jedes Kartenmaterial dient dazu eine Barriere, also den unbekanntem Pfad zum Ziel schnellstmöglich durchqueren zu können, zu beseitigen. Diese Barriere wurde seit längerer Zeit geschlossen indem Kartenmaterial in verschiedenen Formen verfügbar gemacht wurde. Heute kann man verschiedene Möglichkeiten der Navigation finden, eine davon ist z.B. ein Navigationssystem. Durch dieses werden alle befahrbaren Wege aufgezeigt die befahrbar sind und letztendlich zum ausgewählten Ziel führen. Betrachtet man ein Navigationssystem für Rollstuhlfahrer so erkennt man dort einen entscheidenden Unterschied. Die Vorgehensweise bzw. Strategie, wie bei dem Straßennetz, ist etwas anders. Ein Rollstuhlfahrer ist sicherlich an Wegen interessiert, die von ihm befahren werden dürfen. Ungeachtet dessen kann aber ein Weg nicht einfach als befahrbar markiert werden, da dieser unter Umständen für manche Rollstuhlfahrer nicht befahrbar sein kann (Grad der Behinderung, Elektro-Rollstuhl). Mit anderen Worten ein Weg hängt von der Sicht des Betrachters ab und kann zu einer (unüberwindbaren/überwindbaren) Barriere mutieren. Aus diesem Grund muss das Kartenmaterial verglichen mit herkömmlichen Navigationssystemen anders gestaltet werden. Rollstuhlfahrer-gerechte Karten enthalten möglichst viele Informationen über einen Weg und deren Barrieren. Das Hauptaugenmerk ist das Hindernis selber und liegt

somit in dem zentralen Interesse eines Rollstuhlfahrers. Dabei muss die Betrachtung auf die Barrieren weiter differenziert werden. So sind hier zwei Gruppen von Barrieren zu unterscheiden. Es existieren Barrieren die einfach nicht überwindbar sind. Auf der anderen Seite existieren Barrieren die abhängig von der Situation überwindbar sein könnten (wenn z.B. Begleitung da ist, die bei der Überwindung der Barriere hilft).

3.3.4 Aggregation von Weginformationen

Die Trailblazers-Plattform basiert auf dem Konzept von Wikipedia. Dieser Ansatz wurde aus zwei Gründen gewählt. Von jedem Benutzer werden Weginformationen gesammelt. Diese müssen dem Benutzer wieder zugänglich gemacht werden. Hierzu werden diese Daten beim Verbindungsaufbau mit dem Server auf diesen übertragen. Damit werden diese persistent gemacht und mit den bereits gespeicherten Daten synchronisiert. Jede Änderung der Umgebung wird so wahrgenommen und für die Berechnung von Routen zugänglich gemacht.

Das zweite und noch wichtigere Argument für den Einsatz einer Community war das Teilen (Sharing) von Ressourcen. Jeder Benutzer kennt seine Umgebung am besten. Diese wird er schnell explorieren, jede Änderung wahrnehmen und dafür geeignete Umwege finden können. Durch den Abgleich und somit Speicherung auf dem Server stellt er aber sein Wissen über seine Umgebung der gesamten Community bereit. Denn jedem Benutzer ist es freigestellt auf die gesamte Datenbasis zuzugreifen. Betrachtet man diesen Effekt aus Sicht der gesamten Community, so bildet sich innerhalb einer kurzen Zeit eine Wissensbasis, welche qualitativ sehr genauere Umgebungsinformationen enthält. Durch die Verteilung der Sammlung von Weginformationen auf einzelne Benutzer wurde der hierbei anfallende Aufwand deutlich minimiert. Dieser ruht auf der Tatsache, dass jeder Benutzer zur Beginn Weginformationen über seine eigene und somit sehr gut bekannte Umgebung in das System einspeist. Mit Hilfe des hier entwickelten webbasierten Werkzeugs soll es dann zusätzlich noch möglich sein nachträglich die gesammelten Daten nachzubereiten und zu verfeinern.

3.3.5 Handhabbarkeit

Eine weitere Herausforderung ist die Handhabbarkeit. Wikipedia verdankt ihren Erfolg sicherlich auch z.T. der einfachen Benutzung. So ist hierbei der Verarbeitungsprozess äußerst einfach und damit für jeden verständlich. Das Hilfswerkzeug, welches den Benutzer dabei hilft diesen gewählten Verarbeitungsprozess zu durchlaufen ist ebenfalls sehr einfach konzipiert. Diese Aspekte müssen immer erfüllt werden damit das System sich breiter Akzeptanz erfreut. Das Trailblazers-System kann nur bestehen wenn dieses schnell eine Community bildet. Mit anderen Worten spielt hierbei die Akzeptanz ebenso eine entscheidende Rolle. Daraus folgt, dass die Trailblazers-Plattform eine einfache Abfolge von Tätigkeiten in sich

enthalten soll, welche bei der Nutzung dieses Systems notwendig sind. Diese müssen in sich einfach und verständlich sein. Daneben, wie bei Wikipedia, muss Trailblazers auch Werkzeuge bereitstellen, die dem Benutzer bei der Nutzung des Systems unterstützen. Und zu aller letzt ist es wichtig, dass Trailblazers an sich einfach in der Benutzung ist.

3.3.6 Privacy

Wie bereits erwähnt werden die Benutzer ihre nähere Umgebung erkunden und die darüber generierten Informationen der Community bereitstellen. Hierbei werden somit Informationen über ihre Umgebung, ihre Bewegungsmuster sowie höchstwahrscheinlich auch die zeitliche Beziehung jedermann zugänglich gemacht. Dieses ist sicherlich nicht im Sinne der Benutzer. Diese wollen nicht, dass ihre Privatsphäre eingeschränkt sondern von dem System gewahrt wird. Sehr wichtig ist es deswegen an dieser Stelle die Gesichtspunkte, die sich hinter Privacy verbergen, zu betrachten.

Das System muss für den Benutzer transparent sein und damit ihm die Möglichkeit geben seine Sichtbarkeit in dem System steuern zu können. Dabei sollten verschiedene Stufen der Sichtbarkeit eingeführt werden, da verschiedene Benutzer dem System beitreten werden. Zum Einen werden Benutzer erwartet die jegliche Sichtbarkeit scheuen und somit ihre Transparenz erzwingen werden. Auf der anderen Seite existieren Benutzer welche der Problematik der Privacy nicht so kritisch gegenüberstehen. Diese Gruppe wird versuchen möglicherweise mit verschiedenen Abstufungen ihr Bewegungsverhalten der Community offen zu legen. Möglicherweise werden diese personenbezogenen Informationen nur Freunden oder der Familie zur Verfügung gestellt. Hier müssen die verschiedenen Sichtweisen bzw. Benutzerprofile identifiziert werden. Darauf basierend muss ein Modell erarbeitet werden durch das die Sichtbarkeit flexibel gesteuert werden kann.

3.3.7 Tracking

Aufbauend auf der Problematik der Privacy kommt eine weitere interessante Möglichkeit, gleichzeitig aber auch neue Gefahr zum Vorschein. Bis jetzt haben wir von der Bereitstellung der Weginformationen und den damit verbundenen diversen, impliziten Informationen wie das Verhalten der Benutzer gesprochen. Es ist in der näheren Zukunft zu erwarten, dass das mobile Internet sich sehr schnell verbreiten wird. Hierdurch entstehen die Möglichkeiten die einzelnen Benutzer auf ihre Position zu prüfen (Tracking). Man stelle sich Benutzer wie Eltern vor, die von überall überwachen und somit feststellen können, ob das Kind auch wirklich da ist wo es sein sollte. Ein neues Gefühl der Sicherheit würde somit den Eltern und anderen Aufsichtspflichtigen gegeben werden.

Die zweite Seite der Medaille, welche von den Kritikern des Trackingansatzes angesprochen wird, ist die hierdurch entstandene Überwachungsmöglichkeit. Wer sollte das Recht bekommen und wie weit sollte dieses gehen die Bewegungen eines Menschen zu überwachen. Wird möglicherweise hierdurch eine Möglichkeit geschaffen den Menschen wieder ein Stück seiner Freiheit zu rauben? Welche Risiken sind damit verbunden und wie werden solche Ansätze dem Menschen mehr zum Verhängnis? Hier muss stark auf der Privacyproblematik aufbauend ebenfalls ein Modell entwickelt werden, wodurch sichergestellt werden kann, dass die Menschen nicht benachteiligt werden, wenn Sie Ihre Echtzeitpositionsdaten nach außen verfügbar machen.

3.3.8 Kontraproduktive Verwendung

Der Community-gestützte Ansatz, der durch das Trailblazers-System umgesetzt und gefördert werden soll, bietet der kollaborativen Erstellung von Navigationskarten einige Vor- aber auch Nachteile. Einer dieser Nachteile ist die kontraproduktive Verwendung. Dabei kann diese in zwei verschiedene Ausprägungen unterteilt werden: Der Beabsichtigten und der Unbeabsichtigten. Die unbeabsichtigte wird im Trailblazers-System mehr oder weniger durch den Automatismus dieser Plattform vermieden. Zum anderen wird die nachträgliche Überprüfung und Beseitigung von Fehlinformationen durch das webbasierte Werkzeug ermöglicht. Der Community-gestützte Ansatz beruht auf der Tatsache, dass jeder zu jeder Zeit jede Informationen verändern darf. Dieses ermöglicht ebenfalls einem Benutzer, dass dieser eine falsche Information mit Absicht zum System hinzufügen kann. Wird dies einmalig gemacht wird diese Änderung durch den nächsten Benutzer, der diesen Trampelpfad verwendet, festgestellt und korrigiert. Wie soll sich aber das System verhalten wenn Benutzer ständig absichtlich falsche Informationen hinzufügen und das System somit unbrauchbar für die Navigation machen? Wie kann man so ein Verhalten feststellen und sich davor gegebenenfalls schützen? Wenn aber eine Schutzmaßnahme sich als zu stark erweisen sollte, kann diese die Wirkung des Community-gestützten Ansatzes schwächen. Ein Schutzmechanismus muss erarbeitet werden, welcher zum einen die Benutzer nicht einschränkt zum anderen aber diese vor böswilligen Benutzern schützt.

3.3.9 Kartographieren und Lokation

Das Trailblazers-System braucht eine Technologie, durch die es möglich wird, die Lokation des Benutzers festzustellen. Hierzu wurde darauf geachtet, dass die Verfügbarkeit und Preishöhe optimal für den breiten Markt ist, sodass möglichst viele Menschen das System nutzen können. GPS stellt hier die beste Wahl dar. Durch den Ansatz des Differential GPS kann die Genauigkeit der Lokation unter einen Zentimeter fallen. Mit anderen Worten ist die

notwendige Technik verfügbar um ein solches Szenario zu ermöglichen. Genauer betrachtet werden aber auch Problemfälle sichtbar.

Ein Benutzer der sich durch ein dicht bewohntes Gelände bewegt, welches sich dazu noch durch sehr hochstehende Gebäuden auszeichnet, muss damit rechnen, dass sein GPS-Gerät einer hohen Reflektion des GPS-Signals durch diese Gebäude ausgesetzt wird. Dieses führt dazu, dass der GPS Empfänger fälschlicherweise eine falsche Ortung berechnet. Weitere Schwierigkeiten entstehen, wenn der Benutzer sich im Untergrund bewegt wie z.B. einem Tunnel. Dort ist überhaupt keine Möglichkeit mehr gegeben auf die GPS Daten zurückzugreifen.

Aus den geschilderten Problemfällen wird ersichtlich, dass das automatisch produzierte Kartenmaterial in bestimmten Fällen einfach falsch sein wird. Um dieses Problem in dem Griff zu bekommen muss eine nachträgliche Bearbeitung der erstellten Karten möglich sein. Mit dieser sollen solche Fehler erkannt und entsprechend verbessert werden können. Zusätzlich hierzu sollte eine Möglichkeit geboten werden bestimmte Trampelpfade bei wiederkehrenden falschen Daten einzufrieren. Hierdurch wird der Trampelpfad nicht mehr durch falsche Daten verfälscht. (Die Eigenschaft der Trampelpfade kann weiter bestehen bleiben, da nur die Gestalt des Trampelpfades eingefroren wird und nicht die Auswertung der Verwendungshäufigkeit. Mit anderen Worten wird diese weiterhin berücksichtigt, der Trampelpfad aber selber kartographisch nicht mehr ausgewertet).

Eine Karte ist zweidimensional, unsere Welt nicht. Diese Tatsache führt zu einem weiteren Problem, welches bei dem Kartographieren der Trampelpfade beachtet werden muss. Ein Benutzer kann sich die meiste Zeit auf einer Ebene bewegen und so sein Ziel erreichen. Ist dieser aber in einer Stadt unterwegs kann es leicht passieren, dass dieser auf Wege stößt die sich zwar auf einer Karte kreuzen oder überlappen, in der realen Welt ist das aber nicht der Fall. So z.B. Wege die über und unter die Oberfläche führen. Eine Unterführung kann eine Erleichterung bei der Überquerung eines Hindernisses sein, auf der anderen Seite kann diese auf zwei dimensional Karten zum Fehlrouen führen. Das Trailblazers-System muss zum einen Möglichkeiten finden wie eine solche Information gerecht für den Benutzer dargestellt werden kann. Zum anderen muss die Routenberechnung darauf Rücksicht nehmen und das Kartenmaterial als dreidimensional und nicht zweidimensional auswerten. Dieses ist deswegen notwendig da sonst Kreuzpunkte auf den Trampelpfaden entstehen, die in Wirklichkeit nicht existieren. Durch solche nicht real existierenden Kreuzungspunkte können dann unüberwindbare Routen entstehen.

3.3.10 Point Of Interest

Unter den Points Of Interests werden aus Sicht des Trailblazers-Systems alle Arten von Informationsartefakten verstanden, die dazu helfen können eine barrierefreie Fortbewegung

zu ermöglichen. Mit dieser kurzen Beschreibung offenbart sich gleichzeitig die hierbei liegende Schwierigkeit. Wie soll ein Point Of Interest beschrieben werden? Dabei muss eine Darstellung der Information gewählt werden, bei der möglichst viel hilfreiche Information umschlossen werden kann. Zum anderen muss aber die Komplexität des Hinzufügens eines Points Of Interest minimal gehalten werden. Wie kann man hier die beiden Extreme am Besten einander annähern?

Eine weitere Schwierigkeit die hier zum Vorschein kommt betrifft die Darstellung des Points Of Interest auf den Navigationskarten. Zum Einem darf eine Karte nicht durch Unmengen von Points Of Interest überfüllt und somit unlesbar gemacht werden. Zum anderen müssen Darstellungsformen für die POIs gewählt werden, welche die Wiedererkennung fördern. Mit anderen Worten müssen mehr oder weniger Möglichkeiten der Standardisierung dieser eingeführt werden. Dabei muss u.a. darauf geachtet werden, dass durch diese Maßnahme auch Doppeldeutigkeiten und Duplizierungen von Semantiken unterbunden werden.

Die automatisierte Auswertung von POIs bei der Routenberechnung stellt eine weitere Herausforderung dar. Damit eine Route den entsprechenden Vorgaben eines Benutzers entspricht, wird die in den POIs enthaltene Information bei der Routenberechnung mit berücksichtigt. Um diese Information der automatisierten Verarbeitung zur Verfügung zu stellen muss diese entsprechend in einer geeigneten Darstellung abgelegt werden. Wichtig ist auch noch herauszufinden welche Informationen der POIs von Nutzen sein könnten um die beste Route zu berechnen. Oder soll ein offenes Format gewählt werden, welches eine beliebige Annotierung der POIs bzw. deren Erweiterung zulässt? Wenn ja, wie soll dann der Auswertungsmechanismus (Strategie) aussehen?

3.3.11 Gemeinsame Erarbeitung von Material

Der hier zugrunde liegende Ansatz der kollaborativen Erarbeitung von Navigationskarten basiert darauf, dass das Kartenmaterial für eine gemeinsame Verarbeitung bereitgestellt wird. Paralleles Arbeiten an einer Ressource entfacht neue Probleme wie der gleichzeitige Schreib/Lesezugriff. Dabei muss irgendwie unterschieden werden zwischen einer sich gerade in der Bearbeitung befindenden Informationseinheit und einer zum Lesen freigegebenen. Auf der anderen Seite treten bei einer gemeinsamen Verarbeitung zunehmend Schwierigkeiten auf z.B. die der oszillierenden Veränderung⁵. Die durch gemeinsame Verarbeitung entstehenden Fehler werden zum einen durch die Community beseitigt. Zum anderen sollte diese möglicherweise durch Versionierungswerkzeuge wieder in den richtigen Zustand gebracht werden können.

⁵Unter einer oszillierenden Änderung wird das ständige entfernen und wieder hinzufügen eine Information verstanden.

Indem ein Benutzer für die gemeinsame Erarbeitung seine privaten Weginformationen zur Verfügung stellt, teilt er diese mit den anderen. Es kann der Fall entstehen, dass eine Karte nach seiner Vorstellung erstellt wurde (POIs), diese entspricht aber gleichzeitig der Vorstellung der anderen Benutzer nicht. Wie kann ein solcher Fall gehandhabt werden? Sollen mehrere parallele Kartenversionen eines Gebiets existieren? Muss vielleicht eine Abstraktionsschicht in der Datenhaltung hinzugefügt werden, durch die mehrere Repräsentationen eines Gebietes möglich werden? Oder reicht eine Karte für die gemeinsame Verarbeitung und Benutzung aus? Wenn ja, muss dann eine Möglichkeit geschaffen werden, durch die überwacht wird, dass nur ein glaubwürdiger Benutzer eine Informationseinheit persistent speichern und wiederum entfernen darf? Mit anderen Worten muss eine Rechtevergabe auf bestimmte (private) Bereiche eingefügt werden.

Diese und weitere Probleme wurden in dem Distributed Collaborative Work bereits genauer untersucht. Aus diesem Bereich stammende, für das Trailblazers-System geeignete Vorgehensweisen sollten hier zum Tragen kommen.

3.3.12 Verteilung der Sichtweise

Eine weitere Herausforderung kann die Verteilung des Verständnisses über die in dem Trailblazers-System gesammelten Ressourcen sein. Jeder Benutzer interpretiert eine Informationseinheit anders, da diese möglicherweise aus unterschiedlich motiviertem Kontext diese in das System eingefügt haben. Zudem kann eine variierende Priorisierung von Informationen entstehen die ebenfalls auf die Zusammenführung von verschiedenen Benutzern und somit verschiedenen Betrachtungen zurückzuführen ist.

3.3.13 Echtzeit

Die Qualität jedes Kartenmaterials hängt immer sehr stark von deren Aktualität ab. Entspricht die Karteninformationen nicht der Gegebenheit so sind diese wertlos bzw. führen zu erheblicher Erschwernis bei der Navigation. Diese Problematik muss auch in dem Trailblazers-System berücksichtigt werden. Dabei sind zwei Szenarien möglich. Bei dem Ersten kann der Benutzer die Aktualisierung der Karten nur vor dem Beginn der Reise durchführen. In dem zweiten Szenario verfügt der Benutzer über eine durchgängige Verbindung mit dem System und kann jederzeit die Änderungen von Weginformationen wahrnehmen.

An dieser Stelle soll das zweite Szenario genauer betrachtet werden (das erste hat durch ihren Offline Charakter keine Möglichkeit diese Problematik zu umgehen). Hierbei werden bei der Fortbewegung laufend Änderungen in das System eingefügt. Was soll geschehen wenn Trampelpfade entfernt werden (was nichts anderes bedeutet, dass diese Strecke nicht mehr

begehrbar ist)? Eine andere Situation kann genauso entstehen in der neue Trampelpfade eingefügt wurden, wodurch der z.Z. verfolgte Weg deutlich verkürzt werden kann. Soll die Route automatisch neu berechnet und dem Benutzer zur Verfügung gestellt werden? Was soll passieren wenn noch unentdeckte Barrieren auf dem Weg entdeckt werden obwohl das System keine Barrieren meldet? Das System muss die Route neu berechnen können. Dabei sollte dieses die vom Benutzer erkannte Barriere mit berücksichtigen. Wie teilt der Benutzer eine solche Barriere in Echtzeit mit (dieser kann möglicherweise nicht wirklich Benutzereingaben machen, sei es durch Einschränkungen des Gerätes oder durch den Behinderungsgrad)?

3.4 Anforderungen an die Benutzerschnittstelle

Die Trailblazers-Plattform setzt sich aus verschiedenen Dialogen zusammen, welche das Informationsportal mit den dazugehörigen Werkzeugen verbinden. Im Mittelpunkt dieser Betrachtungen steht das für die Erstellung der annotierten Karten notwendige Werkzeug. Bei dem Entwurf dieses Werkzeuges sollten die bereits, aus dem Entwurf von desktopbasierten Anwendungen gewonnen Erkenntnisse und geltenden Richtlinien hierbei zum Tragen kommen. Zitiert in (Balzert, 2005) sollten die in der ISO 9241-10:1996 spezifizierten Prinzipien bei der Gestaltung der Trailblazers-Werkzeuge mit einfließen. Hierzu werden diese in Folgende einleitend besprochen.

3.4.1 Angemessenheit der Aufgaben

Das Werkzeug zur Entwicklung der Navigationskarten ist eine sog. *Single Document Interface* Anwendung. Diese sich nach dem Modell des *Modalen Dialogs* richtende Anwendung erlaubt das Arbeiten nur an einer Navigationskarte zum gleichen Zeitpunkt. Alle dort zur Verfügung gestellten Funktionen sollen den Benutzer effektiv und effizient bei der Fertigstellung der Navigationskarten unterstützen. Hierzu wird die Anwendung zwei Bedienungsarten unterstützen, die objekt- und funktionsorientierte Bedienung. Die direkte Manipulation soll vermieden werden damit die Komplexität der Bedienung vermindert werden kann. Damit wird hauptsächlich über Popup-Menüs gearbeitet.

3.4.2 Selbstbeschreibungsfähigkeit

Die heutigen Desktopanwendungen unterstützen den Benutzer mehr oder weniger bei dem Verständnis der Funktionsweise der Anwendung indem diese interaktiv Dialoge mit Hinweisen aufzeigen. Neben der Funktionalität kann dem Benutzer auch eine Besonderheit oder kritische Stelle der Verarbeitung deutlich gemacht werden. Damit wird diese zum Einem

selbstbewusster in der Bedienung der Anwendung, zum Andere werden aber auch mögliche Fehler vermieden, da das System auf diese hinweist. Auch das hier entwickelte Werkzeug kann bei falscher Verwendung zum Datenverlust führen. Um dieses unerwünschte Verhalten zu vermeiden, sollte das hier entwickelte Werkzeug dieser Richtlinie nachgehen.

3.4.3 Steuerbarkeit

Gerade für Menschen mit eingeschränkten Möglichkeiten der Motorik, welche sich auch auf die Nutzung des Computers auswirken kann, müssen bei der Gestaltung dieses Werkzeugs berücksichtigt werden. Die Erfüllung der Anforderung aus Sicht der Steuerbarkeit erlaubt dem Benutzer die Anwendung frei nach seinem Verständnis zu verwenden. Damit inbegriffen ist der Wechsel der Richtung der Verarbeitungskette bzw. das Aufsetzen an beliebiger Stelle.

3.4.4 Erwartungskonformität

Hinter der Erwartungskonformität steckt eine einfache Motivation, der Benutzer soll bei der Nutzung der Anwendung durch konformes Verhalten intuitiv erkennen können was zu tun ist. Dieses Wissen wird neben der Anwendung selbst, aus dem Erfahrungsschatz des Benutzers stammen. Um so erwartungskonformer eine Anwendung ist und damit das Verhalten auf dem Alltagswissen des Benutzers basiert, umso vertrauter wirkt die Anwendung auf den Benutzer. Die Trailblazers-Anwendung soll den gängigen Desktopanwendungen konform, aber auch den ähnlichen bereits bestehenden Webanwendungen gleichen.

3.4.5 Fehlertoleranz

Die Fehlertoleranz wurde allgemein mit Systemanforderungen bereits diskutiert. Im Zusammenhang mit der Entwicklung der GUI handelt es sich hierbei um unterstützende Aufgaben die diese übernehmen müssen, wenn z.B. falsche Eingaben getätigt wurden. Damit eine schnelle Benachrichtigung des Fehlers getätigt werden kann, wird die GUI z.B. den Benutzer auf diesen hinweisen.

3.4.6 Individualisierbarkeit

Ein sehr wichtiger Aspekt, auch aus Sicht des Web 2.0, ist der Grad der Individualisierung der GUI als wichtige benutzerunterstützende Eigenschaft. Zum Einem kann sich der Benutzer besser mit der Anwendung identifizieren. Wichtiger ist aber, dass der Benutzer die GUI

soweit seiner gewohnten Arbeitsweise anpassen kann, dass dieser dadurch eine Aufgabe effizienter erledigen kann.

3.4.7 Lernförderlichkeit

Dieser Aspekt ist wichtig, wie bereits oben angesprochen, und wird durch die Einfachheit dieser Anwendung begünstigt. Diese muss in sich verständlich gestaltet werden damit die Lernphase in ihrer Dauer maximal verkürzt werden kann.

3.4.8 W3C Web Content Accessibility Guidelines

Von der W3C wurden ebenfalls Empfehlung zur Gestaltung barrierefreier Web-Seiten ausgesprochen ([W3C, 1999](#)). Diese sind wie folgt:

Provide equivalent alternatives to auditory and visual content: Diese Richtlinie ist wichtig für Menschen die gehörgeschädigt sind oder visuelle (z.B. Farbenblind) eingeschränkte Möglichkeiten haben. Hierzu soll der auditive und visuelle Inhalt gleichzeitig auch als Text verfügbar gemacht werden.

Don not rely on color alone: Diese Richtlinie richtet sich speziell auf die Menschen die eine Schwäche bei der Unterscheidung von Farben haben. Eine Web-Seite sollte so entwickelt werden, dass diese ihre Lesbarkeit nicht verliert, wenn diese Schwarz-Weiß gestaltet wäre.

Use markup and style sheets and do so properly: Um ein einheitliches Format und damit eine bessere Lesbarkeit zu unterstützen, sollten Strukturelemente genutzt werden um somit die Darstellung des Inhalts zu vereinheitlichen. Neben der Lesbarkeit wird auch die Portierbarkeit auf andere Geräte einfacher.

Clarify natural language usage: Werden im Text Abkürzungen, Fremdwörter oder andere Inhalte verwendet, welche die allgemeine Sprache nicht enthält, sollten diese markiert und verdeutlicht werden. Damit soll ihm das Folgen des Textes ermöglicht werden.

Create tables that transform gracefully: Bei der Spezifizierung von Tabellen sollten entsprechende Formate genutzt werden, welche die Umwandlung zwischen verschiedenen Browsern erlauben und dabei ihre Übersichtlichkeit nicht einbüßen.

Ensure that pages featuring new technologies transform gracefully: Bei der Entwicklung einer barrierefreien Web-Seite, bei der verstärkt neue Technologie eingesetzt wird, sollte darauf geachtet werden, dass der Inhalt auch für ältere Browser zugänglich ist, wenn diese die entsprechende Technologie nicht unterstützen.

Ensure user control of time-sensitive content changes: Bei der Verwendung von Darstellungen, die sich zeitlich verändern, sollte der Benutzer die Möglichkeit haben diese Änderungen zu pausieren und damit den Inhalt entnehmen können.

Ensure direct accessibility of embedded user interfaces: Für die Verbesserung der Zugänglichkeit von Webinhalten existieren spezielle Benutzerschnittstellen, welche dieses unterstützen. Diese oder Alternativen sollten nach Möglichkeit unterstützt werden.

Design for device-independence: Die Webseite sollte so gestaltet werden, dass der Benutzer auf diese mit verschiedenen Geräten zugreifen kann. Mit anderen Worten sollte die Webseite ihren Inhalt geräteunabhängig zugänglich machen.

Use interim solutions: Die letzte Empfehlung sagt aus, dass vorrangig Technologien verwendet werden sollen, welche ihre Funktionsweise im Falle des Aufrufes durch einen alten Webbrowser nicht verändern, bzw. den Benutzer nicht irreführen.

Use W3C technologies and guidelines: Es sollte nach Möglichkeit versucht werden bestehende Richtlinien des W3C bzw. alternative zu verwenden. Ist dieses nicht möglich, sollten Alternativseiten realisiert werden, welche barrierefrei sind.

Provide context and orientation information: Die Inhalte einer Webseite sollten durch zusätzliche Informationen verdeutlicht werden. Darüber hinaus sollten die Orientierung bei komplex aufgebauten Webseiten durch entsprechende Methoden entschärft werden.

Provide clear navigation mechanisms: Hierbei soll die Webseite Möglichkeiten anbieten durch die der Benutzer einfach gesuchte Informationen finden kann.

Ensure that documents are clear and simple: Diese Forderung zielt auf die Grundprinzipien der Gestaltung von Dokumenten ab und empfiehlt das Zusammenfassen sowie die Bewahrung der Einfachheit.

Diese eben genannten Richtlinien zielen auf die Gestaltung von barrierefreien Webseiten ab. Natürlich sind einige davon in Konflikt mit dem neuen Technologien, bzw. deren Möglichkeiten zur Darstellung von Inhalten. Trotzdem sollten diese nach Möglichkeit berücksichtigt werden und ggf. die technischen Innovationen der Lesbarkeit den Vortritt lassen. Alternativ sollte wie bereits durch die Richtlinie selber vorgeschrieben wird, eine alternative Webseite erstellt werden die den Benutzervorgaben entspricht. Dafür wäre z.B.: die Spezifikation von Benutzerarten und die dazu gehörenden Darstellungen eine Möglichkeit, welche dann mit den Benutzerprofilen in Beziehung gesetzt werden. Die einzige Hürde wäre die Anmeldung selber, danach wäre der Inhalt speziell auf den Benutzer zugeschnitten.

Die Einhaltung dieses Standards ermöglicht aber nicht nur den Zugang für die Benutzer, die ohne diese Richtlinie erschwert oder überhaupt nicht mit dem Inhalt umgehen würden. Der hier betrachtete Webclient wird früher oder später auf dem mobilen Geräten verfügbar

gemacht. Durch die Einhaltung dieser Richtlinien kann dieser Übergang in die mobile Welt sich einfach gestaltet werden.

Durch die Einhaltung der hier definierten Richtlinien, kann der Inhalt verschiedenen weiteren Benutzergruppen zugänglich gemacht werden um damit ein breiteres Publikum anzusprechen. Wie bereits erwähnt, lebt dieser Ansatz von den Benutzern. Je größer die Benutzergruppe desto wahrscheinlicher ist es, dass neuer Inhalt von dieser erstellt wird. Dies impliziert also, dass die Einhaltung dieser Richtlinien auch die Chancen dieses Systems steigern. Aus diesem Grund ist die genauere Untersuchung dieser Richtlinien erforderlich. Eine ausführliche Betrachtung hierzu bietet ([Thatcher u. a., 2006](#)).

3.4.9 Rollstuhlfahrgerechte Informationsdarstellung

In Rahmen der ASK-IT Konferenz wurden Arbeiten vorgestellt die eine vergleichbare Zielsetzung, wie diese Arbeit sie verfolgt, haben. Einer der interessanten Arbeiten wurde von Ringbauer & Schlegel ([Ringbauer und Schlegel, 2006](#)) vorgestellt. Diese setzt sich mit der Problemstellung auseinander, Rollstuhlfahrern die Zusammenstellung und Darstellung von Navigationsinformationen zu vereinfachen. Dabei wurde das dort erarbeitete Konzept bereits mit zehn Betroffenen auf seine Tauglichkeit erfolgreich geprüft und damit die praxisreife des erarbeiteten Konzeptes bestätigt. Damit sind die dort gewonnenen Erkenntnisse auch für diese Arbeit sehr wertvoll und sollten hier berücksichtigt werden.

Die dort vorgeschlagene Anwendung versucht einen einfachen Zugriff auf die dort verfügbaren Funktionen anzubieten. Hierzu wird diese Anwendung in vier Bereiche unterteilt. Diese sind:

Navigationsleiste: Diese hat die Aufgabe alle verfügbaren Funktionen sauber zu strukturieren. Hinter den Aufgabengruppen verbergen sich einzelne Aufgaben, welche menüartig erscheinen, wenn die Aufgabengruppe ausgewählt wird.

Unterstützung der Navigation: Dieser Bereich umschließt zwei wichtige Informationsarten, einen Fortschrittsbalken und eine Sammlung von Möglichkeiten eines schnellen Zugriffes auf Funktionen. Der Fortschrittsbalken soll dem Betreffenden die Information liefern, wie weit die Bearbeitung fortgeschritten ist. Der Schnellzugriff soll dem Benutzer, angepasst auf sein Profil, die wichtigsten und am häufigsten verwendenden Funktionen bereitstellen.

Statusleiste: Die Statusleiste bietet allgemeine Informationen, welche bei der Arbeit mit dem Webclient hilfreich sein könnten, an.

Hauptbearbeitungsfenster: In diesem findet die eigentliche Bearbeitung der entsprechenden Informationen statt.

Die grobe Struktur des Prototypen wird in Abbildung 3.4 dargestellt.

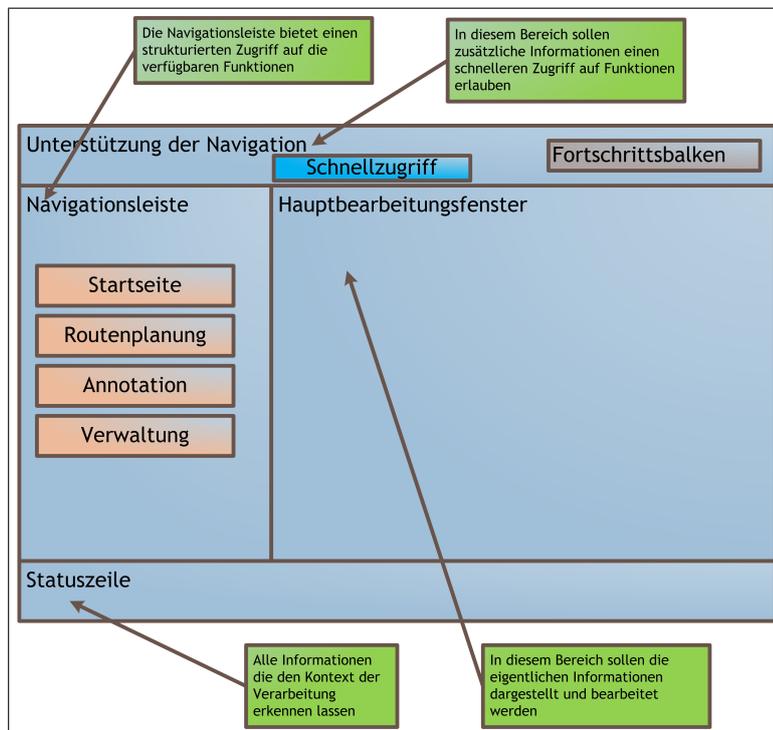


Abbildung 3.4: Gestaltungsstruktur des Prototypen angelehnt an (Ringbauer und Schlegel, 2006).

Der von Ringbauer & Schlegel vorgestellte Prototyp unterstützt nach Aussagen der Testpersonen eine übersichtliche und unterstützende Erarbeitung von Routenempfehlungen. Die Aspekte betreffend der Partizipation mit der Community wurden hierbei nicht betrachtet, spielen aber für den hier betrachteten Ansatz eine wichtige Rolle. Ungeachtet dessen sollten die vorgeschlagene und bereits geprüfte Struktur der Anwendung bei dem Entwurf zum größten Teil berücksichtigt werden.

3.5 Wahrnehmung der Umgebung durch Rollstuhlfahrer

Bevor mit dem Entwurf begonnen wird, sollte die Wahrnehmung der Rollstuhlfahrer untersucht werden. Hierbei soll betrachtet werden, wie diese Menschengruppe die Umgebung wahrnimmt, bzw. welche Informationen für sie sehr großen Stellenwert besitzen. Die hier gesammelten Erkenntnisse sind essentiell für die Gestaltung des Systems, da die identifizierten Informationsarten unterstützt werden müssen.

Als Grundlage für diese Betrachtung wird ein von der Hamburger Landesarbeitsgemeinschaft für behinderte Menschen e.V. erarbeiteter Stadtführer für Rollstuhlfahrer genutzt (Lagh-e.V., 2004). In dieser Broschüre wird versucht die barrierefreie Fortbewegung in Orten wie Freizeit-, Kultur-, Sport- und Gastronomiestätten durch eine einheitliche, auf Rollstuhlfahrer abzielende Beschreibung zu gewährleisten. Die dort identifizierten wichtigsten Informationen sollen nun im Folgenden zusammengetragen werden.

3.5.1 Zugänglichkeit

Alle Orte müssen durch Informationen über die Gegebenheiten des Zuganges näher beschrieben werden. Dabei werden drei Stufen der Zugänglichkeit identifiziert, von *zugänglich* über *eingeschränkt zugänglich* bis *nicht zugänglich*. Bei den betrachteten Orten müssen verschiedene zusätzliche Informationen gegeben werden. Sehr wichtig sind die Abmessungen über die dort vorgefundenen Eingänge, also Türbreite, Fahrstuhltiefe aber auch die Steigung von entsprechenden Rampen. Für jede dieser Eigenschaften gibt es Grenzwerte, nach denen die oben genannte Klassifizierung durchgeführt wird. Von einem Nichtrollstuhlfahrer wird eine Treppenstufe so gut wie nicht wahrgenommen, von einem Rollstuhlfahrer schon. Aus diesem Grund wird auch eine einzelne Stufe in die Beschreibung mit aufgenommen.

3.5.2 Parkplätze

Die oben genannte Zugänglichkeit zielt nicht nur auf die Betrachtung der Befahrbarkeit von Wegen mit einem Rollstuhl ab. Ein Teil der Betroffenen bewegt sich u.a. mit einem Auto fort. In solchen Fällen ist für den Betroffenen wichtig, auch einen rollstuhlgerechten Parkplatz zu finden. Neben dem Vorhandensein von Parkplätzen ist es wichtig deren Breite zu erfahren. Zudem werden hierzu Informationen über die Lage der Bordsteine, also wie weit diese ein Hindernis darstellen können, zusammengetragen.

3.5.3 Beschaffenheit der Eingänge

Neben den Haupteingängen existieren auch Nebeneingänge die speziell für den Zutritt von Rollstuhlfahrern konzipiert wurden. Solch eine Möglichkeit hat allerdings nur wenig Nutzen, wenn diese nicht auffindbar ist. Diese muss explizit beschrieben werden. Dabei sind Merkmale wie: Klingel, Sprechanlage, verschlossene Tür oder notwendige Voranmeldung wichtig mit aufgeführt zu sein. Die Differenzierung der Beschaffenheit des Einganges fängt schon bei der Tür selber an. Dabei ist schon der Hinweis ob eine Tür existiert und wenn ja, wie diese geöffnet wird, sehr wichtig. Der Betroffene muss wissen ob die Tür nach Innen oder

nach Außen aufgeht bzw. ob es sich möglicherweise um eine Drehtür handelt. Jede noch so kleine Anmerkung kann für die Rollstuhlfahrer eine deutliche Erleichterung bei der Planung der Reise darstellen.

3.5.4 Aufzüge

Ähnlich wie bei den Türen ist die Beschaffenheit der Fahrstühle für einen Rollstuhlfahrer von Interesse. Neben der Beschaffenheit der Fahrstuhltür selber, welche auch den Anforderungen für Rollstuhlfahrer entsprechen muss, ist es wichtig für ihn die Eigenschaften des Bedienungsfeldes zu kennen. Aus der Beschreibung muss hervorgehen, ob dieses auch auf einer geeigneten Höhe in waagerechter Position verfügbar ist. Wenn nicht, muss geklärt werden ob der Fahrstuhl nur mit Personal benutzbar ist. Ebenfalls ist darauf hinzuweisen, ob das oberste Stockwerk nicht mehr mit einem Fahrstuhl erreichbar ist.

3.5.5 Sanitäre Orte

Die Aspekte der Zugänglichkeit sind aber nicht nur wichtig für Orte, die der Betreffende vor hat zu besuchen. Neben diesen Orten existieren auch solche, die nur bei einer Notwendigkeit besucht werden müssen, wie das bei einer Toilette der Fall ist. Sehr wichtig ist bei der Planung der Reise ebenfalls das Vorhandensein von entsprechenden Orten mit zu berücksichtigen. Dabei sind neben den bereits erwähnten Eigenschaften weitere für den Betreffenden wichtig, wie z.B. von welcher Seite dieser die Toilette anfahren soll. Jedes noch so kleine Detail sollte in die Beschreibung mit aufgenommen werden.

3.5.6 Zusammenfassung

Die hier kurz verdeutlichte Sichtweise bzw. die Aufzählung einiger Merkmale, auf die ein Rollstuhlfahrer achtet, verdeutlicht, dass das Ausbleiben einige noch so kleine Informationen über den Erfolg einer Reise entscheiden kann. Möchte man dieser Menschengruppe ein Navigationssystem bereitstellen, so sollte dieses Möglichkeiten bereitstellen, die neben den oben genannten Informationen eine Menge weitere in sich aufzunehmen. Des Weiteren sollte bei der Berechnung einer geeigneten Route, diese Information durch den Algorithmus berücksichtigt werden. Ist dieses zu aufwendig zu realisieren, sollte zumindest eine benutzergerechte Bereitstellung dieser Informationen von dem System angeboten werden.

3.6 Zusammenfassung

Dieses Kapitel diente der näheren Untersuchung der Sachverhalte des hier zugrunde liegenden Systems. Hierzu wurden besonders die Prinzipien der objektorientierten Analyse eingesetzt. Damit wurden fachliche Aspekte näher beleuchtet, die als Vorgaben in die Realisierung des Systems eingegangen sind. Zur Beginn wurde die Funktionsweise des Systems einer gründlichen Untersuchung unterzogen. Dabei extrahierte abstrakte Anwendungsfälle verdeutlichten die Sichtweise eines Anwenders auf das System. Zusätzlich wurde die Wahrnehmung des Systems durch den Benutzer in Hinsicht auf Aspekte untersucht, die indirekt mit der durch das System bereitgestellte Funktionalität zusammenhängen.

Neben der funktionalen Analyse wurde eine Untersuchung von Sachverhalten durchgeführt, die essentiell für die Akzeptanz und das Fortbestehen des Systems wichtig sind. Hierzu wurden einzelne Herausforderungen und Risiken, aber auch Chancen identifiziert und diskutiert, welche durch das System offenbart werden.

Das Trailblazers-System zielt auf viele Benutzergruppen ab, die sehr unterschiedliche Vorstellungen von der Benutzung bzw. sehr unterschiedliche Einsatzmöglichkeiten haben. Es existieren Menschen, die Sehschwächen aufweise, motorisch beeinträchtigt sind oder aber auch Menschen, die mit der Benutzung webbasierter Anwendungen schlichtweg nicht vertraut sind. Hierzu wurden existierenden Möglichkeiten und Einschränkungen betrachtet, um das System für alle Gruppen mit einer geeigneten Schnittstelle auszustatten.

In weiteren Verlauf der Analyse stand die Betrachtung von Rollstuhlfahrern im Mittelpunkt. Hierbei wurde die Sichtweise eines Rollstuhlfahrers bzw. seine Anforderungen an eine Navigationskarte betrachtet. Für sie sollte dadurch die Detailtreue bzw. der Informationsgehalt einer Navigationskarte angepasst werden.

4 Entwurf

Nachdem oben die funktionalen und nichtfunktionalen Anforderungen geklärt wurden sollen nun die erarbeiteten Anwendungsfälle systemtechnisch konkretisiert werden, sodass Rahmenbedingungen für das hier zugrunde liegende System genauer spezifiziert werden. Durch diese Betrachtung werden aber weitere Ungereimtheiten und Unstimmigkeiten ersichtlich, weshalb eine Fortführung der analytischen Tätigkeiten entsprechend den zusätzlichen Anforderungen erfolgen wird. Als Substanz wird am Ende dieses Abschnittes eine Systemskizze entwickelt, die alle Vorgaben des Lösungskonzeptes beschreibt, welche beachtet und umgesetzt werden müssen.

Dabei wird der Entwurf anhand des Schichtenmodells verdeutlicht, wie die systemintegralen Elemente und deren Verantwortlichkeiten unter Berücksichtigung des Drei-Schichtenmodells aufgebaut sind. Um eine Vorstellung davon zu bekommen, wie das System und seine einzelnen Komponenten sich physisch verteilen, wird hierfür anschließend das Verteilungsmodell skizziert.

Anhand der bisher identifizierten Fachklassen und Verteilungsaspekte werden dann im folgenden Abschnitt die Systemkomponenten implementiert. Um die Abläufe, die hinter den Anwendungsfällen stehen, entsprechend abbilden zu können, werden auch dort die Anwendungsfallsteuerungsklassen spezifiziert und den entsprechenden Systemkomponenten zugeordnet (Bildung von fachlichen Subsystemen).

Um ein ausgereiftes Systemmodell erarbeiten zu können, müssen die in der Analyse identifizierten abstrakten Anwendungsfälle durch Ergänzung von systemtechnischen Aspekten verfeinert werden. Die dabei entstehende feine Granularität hilft dabei, die Komplexität des Gesamtsystems soweit zu verringern, sodass es handhabbar wird. Damit verbundene Tätigkeiten werden zu Beginn dieses Abschnittes durchgeführt.

Im Abschnitt Identifizierung der Fachklassen werden die Objekte auf der Grundlage des fachlichen und systemtechnischen Modells entworfen. Diese stehen für die ersten in dem System verwendeten Fachklassen. Zudem werden die Beziehungen und die Wechselwirkungen der Entitäten skizziert. Die zur Beginn dieses Abschnittes erarbeiteten Anwendungsfälle geben bereits eine Übersicht über die Anforderungen, die an die Funktionsweise des Systems gestellt werden. Die dahinter konkreten Abläufe sind aber für eine Implementation noch zu

informell beschrieben. Mit der Methode der Aktivitätsdiagramme wird daher das Systemablaufmodell für das hier betrachtete System spezifiziert.

4.1 Schichtenmodell

Das Trailblazers-System soll wie ein klassisches Schichtenmodell aufgebaut werden. Dabei ist zwischen der Hardware sowie Software zu differenzieren. Es wird zunächst die Verteilung der Systemkomponenten in der Software und Hardware sowie deren Beziehungen zueinander spezifiziert. Die Systemkomponenten der verschiedenen Schichten werden dann näher definiert. Hierbei werden die Eigenschaften der Systembestandteile aufgeführt, wodurch ihre Aufgaben und Zuständigkeiten deutlich werden. Anschließend wird das Zusammenspiel der Komponenten beschrieben. Die Verwendung von Entwurfs- und Architekturmustern wird hierbei ebenfalls angestrebt.

Durch die Schichtenarchitektur wird eine einfache aber sehr wichtige Anforderung gestellt - die oberen Schichten greifen auf die darunter liegenden zu, aber nicht umgekehrt. Dagegen werden für Beziehungen innerhalb einer Schicht keine Vorgaben festgelegt. Wichtig ist noch zu erwähnen, dass die Präsentationsschicht einer Drei-Schichten Architektur¹ in Falle von Trailblazers-System ausgelassen werden kann und Anfragen direkt an die Dienste (natürlich über die Dienst-Fassade) gestellt werden können. Dieses ist bei dem Trailblazers-System gewollt, da es auch dem Mashup-Prinzip entsprechen soll. Über diesen Weg kann ein externer Dienst Anfragen direkt stellen. Die verschiedenen Zugriffswege werden in der Abbildung 4.1 dargestellt.

Softwarearchitekturen bestehen aus zwei Arten von Komponenten: Die, die spezifisch für eine Anwendung ist, und die, die bei allen Systemen vorkommen, also sogenannten Infrastrukturkomponenten. Unter Beachtung der eben spezifizierten Vorgaben soll an dieser Stelle die Beschreibung der drei Schichten und darin enthaltenen Infrastrukturkomponenten folgen. Eine Skizze der Trailblazers-Architektur ist in der Abbildung 4.2 verdeutlicht.

Die Abbildung 4.2 veranschaulicht den grundlegenden Aufbau des Systems. Dieser teilt sich grob in zwei voneinander unabhängigen Hardwarekomponenten ein - es handelt sich um ein Client-/Server-Prinzip. Auf dem Server befindet sich die Anwendungslogik, die die fachlichen Anwendungsfälle wiedergibt. Daneben residiert auf dem Server eine Datenbank zur Speicherung der Anwendungsdaten. Hinzu kommen noch die dort verfügbaren externen Sys-

¹Die Drei-Schichten Architektur ist hier als grobe Vorgabe zu sehen um die Verbindung zur Präsentationsschicht zu verdeutlichen.

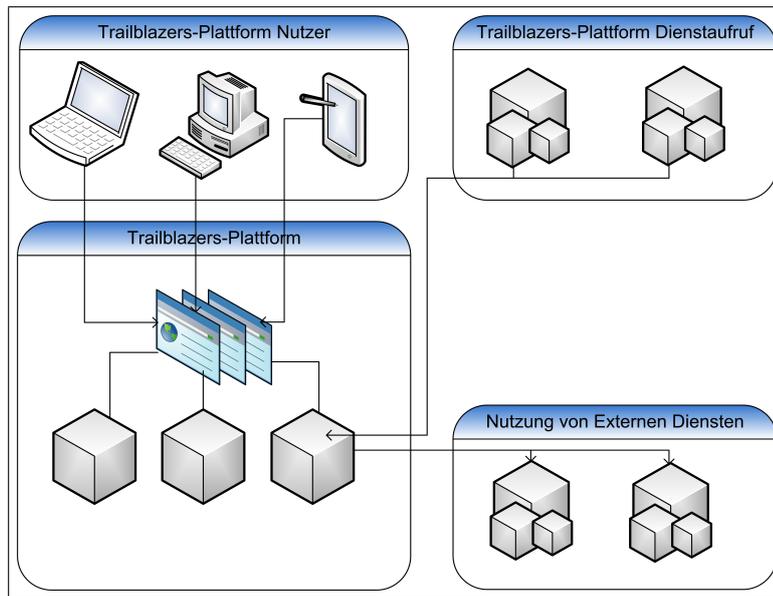


Abbildung 4.1: Zugriffswege auf beziehungsweise von der Trailblazers-Plattform

teme². In diesem Schichtenmodell wurden bereits enthaltenen groben Komponenten und verschiedene logische Schichten, die die Komponenten nach fachlichen Gesichtspunkten zusammenfassen, eingeführt. Deren Aufgaben und Verantwortlichkeiten sollen nun einer ausführlicher Untersuchung unterzogen werden.

Nach (Balzert, 2005) kann ein System in einer Mehrschichtarchitektur Abgebildet werden. Untersucht man das oben vorgegebene Schichtenmodell unter Berücksichtigung der in diesem Buch spezifizierten Vorgaben entstehen folgende Schichten in folgender Reihenfolge:

- einer Präsentationsschicht,
- einer Anwendungsschicht,
- einer Fachkonzept-Zugriffsschicht,
- einer Fachkonzeptschicht,
- einer Datenhaltungs-Zugriffsschicht und
- einer Datenhaltungsschicht.

Im Folgenden werden diese Schichten kurz erläutert.

²Falls diese technisch realisierbar sind. Die physikalische Entkopplung bietet viele Vorteile und ermöglicht einen sofortigen Austausch der Systeme. Zur Gunsten der Veranschaulichung wurde an dieser Stelle auf die Darstellung verzichtet.

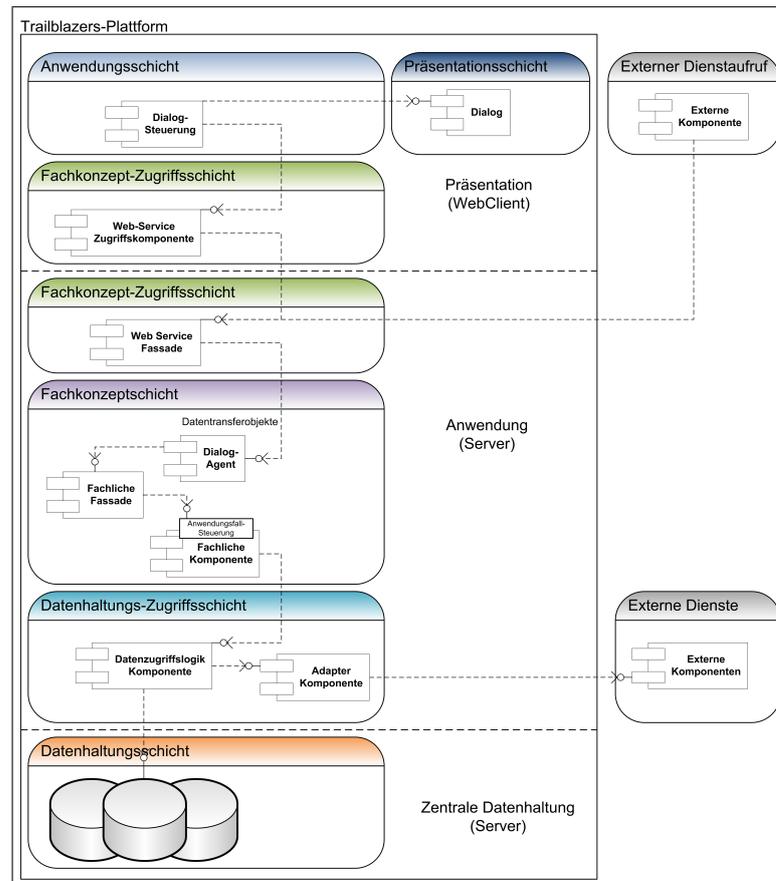


Abbildung 4.2: Schichtenmodell des webbasierten, Server-seitigen Trailblazers Werkzeug zur Bearbeitung der Navigationskarten.

4.1.1 Dialog (Präsentationsschicht)

Dialoge dienen dem Benutzer als Schnittstelle zu dem System in beide Richtungen, der Informationsbereitstellung (Präsentation) und der Informationsübernahme (Benutzereingaben). Sie ist von der Dialogsteuerung abhängig, die die Anwendungslogik zur Formalisierung der Interaktion enthält. Jeder Dialog enthält ebenfalls einen Controller, der die Formatierung der Daten entsprechend der Dialogvorgaben durchführt. Die Steuerung der im Dialog enthaltenen Elemente, damit also deren Interaktionsverhalten, erfolgt ebenfalls an dieser Stelle.

4.1.2 Dialogsteuerung (Anwendungsschicht)

Die Dialogsteuerung ist die Anwendungslogik auf der Clientseite, in der die gesamten Interaktionsabläufe abgebildet sind. Diese enthält Vorgaben über den Ablauf der Interaktionen

zwischen den Benutzern und dem System³. Darüber hinaus behandelt sie die von dem Modell empfangenen Daten soweit, dass diese in dem Dialog dargestellt werden können. Somit ist diese für die Transformation der Daten entsprechend der Vorgaben des Dialogs zuständig.

4.1.3 Web Service Zugriffskomponente (Fachkonzept-Zugriffsschicht)

Um mit dem System kommunizieren zu können, müssen Nachrichten über den zur Verfügung gestellten Web-Service abgesetzt werden. Bei dieser Kommunikation entstehende Aufgaben wie Koordinierung, Auffindung, Bindung etc. werden in dieser Komponente behandelt. Beim Einsatz einer Pufferstrategie wird das in dieser Komponente untergebracht.

4.1.4 Web-Service-Fassade (Fachkonzept-Zugriffsschicht)

Über die Web-Service-Fassade⁴ wird die auf dem Server realisierte Anwendung nach außen für die Clients verfügbar gemacht. Jede Clientanwendung greift bei der Kommunikation mit der Serveranwendung auf diese Schnittstelle zu. Neben der Spezifikation der Schnittstelle der dahinter verborgenen Dienste ist die Komponente für Belange wie die Sicherung des Protokolls, des Datenformats und der nachrichtenorientierten Kommunikationsart. Nicht zu vergessen sind die bei der Kommunikation hervorgerufenen Ausnahmen, die an dieser Stelle ebenfalls behandelt werden.

Wie aus der Architektur-Skizze in der Abbildung 4.2 entnommen werden kann, können über die Web-Service-Fassade auch externe Anwendungen auf die bereitgestellten Dienste zugreifen. Deshalb soll an dieser Stelle die Zugriffskontrolle erfolgen, die auch für die Webclient Anwendung durchgeführt werden muss.

4.1.5 Dialog-Agent (Fachkonzeptschicht)

Zur Koordinierung der einzelnen Client-Dialoge residiert für jeden Dialog explizit ein Dialog-Agent auf dem Server (dieser fungiert als Fassade zwischen dem Client und dem Server). Damit soll die Kommunikation zwischen dem Client und dem Server optimal genutzt werden. Die hierbei realisierten Koordinierungsaufgaben betreffen die Weiterleitung der Nachrichten von dem Client an die richtige fachliche Komponente. In umgekehrte Richtung optimiert der

³Prozessfluss-Steuerung die den Ablauf von Benutzer-relevantem Abläufen sichert.

⁴Für nähere Informationen sehe ([Gamma u. a., 2004](#))

Dialog-Agent die von den fachlichen Komponenten übermittelten Nachrichten, sodass diese effizienter an den Client gelangen können. Hierbei betrachtete Vorgaben sind die Steigerung des Datendurchsatzes und Verringerung der unnötigen ausgetauschten Nachrichten ([Oestereich, 2006](#)). Diese Komponente ist eine Verwaltungskomponente.

Zudem hat der Dialog-Agent die Aufgabe, die Fachkonzept-Komplexität vor dem eigentlichen Dialog zu verbergen. Diese sehr wichtige Aufgabe erlaubt einfache Dialoge. Dazu werden die in der Fachkonzeptschicht enthaltenen Datentypen auf die in dem Dialog angepasst. Damit wird unter Zuhilfenahme der Datentransferobjekte eine fachlich versierte Trennung bewirkt.

4.1.6 Fachliche Fassade (Fachkonzeptschicht)

In vielen Systemen existieren mehr als nur einfache atomare Abläufe. Diese werden in der Literatur häufig als langlaufende Transaktionen bezeichnet. Mit dieser wird ein Prozessfluss auf der fachlichen Ebene sichergestellt. Dies ist notwendig, wenn zum einen eine Transaktionen aus mehreren Schritten besteht, zum anderen wenn andere Dienste als Zwischenschritte in Anspruch genommen werden ([Microsoft, 2002](#)).

4.1.7 Fachliche Komponente (Fachkonzeptschicht)

Die fachlichen Komponenten umschließen die in den Anwendungsfällen spezifizierten fachlichen Zusammenhänge. Mit deren Hilfe wird der eigentliche fachliche Anwendungsbereich wiedergegeben bzw. der Problembereich modelliert. Hier findet die eigentliche Verarbeitung der Anwendungs-bezogenen Daten statt. Alle hierzu notwendigen Funktionen werden nach fachlichen Aspekten durch einzelne Komponenten gekapselt. Diese übernehmen selber die Verantwortung für die Koordination untereinander sowie für die Sicherung ihrer Konsistenz. Auf der anderen Seite enthalten diese kein Wissen über den Ausführungskontext und ebenso wenig über die Belange der Darstellung der von ihnen verarbeitenden Daten.

In den fachlichen Komponenten findet bei der Verarbeitung der Daten eine Interaktion zwischen den fachlichen Klassen statt. Jede fachliche Komponente enthält eine Steuerungseinheit, welche die Abläufe, die in den Systemanwendungsfällen spezifiziert und mit Aktivitätsmodellen verfeinert wurden, im idealen Fall widerspiegelt. Durch die Steuerungsklasse wiedergegebene Möglichkeiten der Interaktion können ein oder mehrere Systemanwendungsfälle beschreiben. Um die Steueraufgaben durchführen zu können, entnimmt die Anwendungsfallsteuerung den Kontext der Bearbeitung aus den antreffenden Daten und leitet die Nachrichten dann weiter an die entsprechenden fachlichen Komponenten. Die eigentliche fachliche Bearbeitung der Anfrage übernimmt dann die fachliche Komponente (Fachklasse).

Durch die Kapselung der Koordinierungsaufgaben in der Anwendungssteuerung wird eine Entkopplung der fachlichen Komponenten untereinander bewirkt.

4.1.8 Adapter-Komponente (Datenhaltungs-Zugriffsschicht)

Manchmal sind zur erfolgreichen Bearbeitung der Benutzeranfragen Zugriffe auf externe Systeme notwendig. Damit die für den Zugriff verantwortliche Logik nicht in die Anwendungslogik mit aufgenommen wird und damit diese schlecht wartbar macht, werden sogenannte Adapter verwendet. Diese Adapter Komponenten kapseln die Schnittstellen der externen Dienste, sowie dafür notwendige Semantik der Kommunikation zwischen diesen. Darin inbegriffen sind die Eigenheiten der Protokolle. Für näheres sehe ([Gamma u. a., 2004](#)).

4.1.9 Datenzugriffs-Komponente (Datenhaltungs-Zugriffsschicht)

Die Datenzugriffs-Schicht ist für den Zugriff auf die Datenhaltungsschicht zuständig. Im Detail werden hierbei die entsprechenden Datentypen, die in der Fachkonzeptschicht verwendet werden, aufbereitet. Der Zweck dieser Aufbereitung ist das Verbergen der datenhaltungsspezifischen Logik vor der Fachkonzeptschicht und damit die Entlastung dieser von möglicher Komplexität. Im genaueren werden an dieser Stelle alle Datenbankoperationen auf vier fundamentale fachliche abgebildet, also das Erstellen, Lesen, Verändern und Entfernen.

4.1.10 Datenhaltungsschicht

Als Basis jeder Anwendung wird die persistente Speicherung von Daten gesehen. Um die technisch versierten Belange sehr schnell von der eigentlichen Anwendungslogik zu entkoppeln, wird die Datenzugriffsschicht eingeführt. In dieser wird die Datenzugriffslogik abstrahiert und aus der Datenhaltungs-Zugriffsschicht ausgelagert. Damit wird diese einfach zu warten bzw. die Anwendungslogik nicht mit spezifischen Belangen eines Datenbankherstellers konfrontiert.

4.1.11 Externe Komponenten

In der Trailblazers-Architektur werden externe Dienste eine wichtige Rolle spielen, sodass hierzu notwendige Kommunikationsmechanismen berücksichtigt wurden. Dabei unterscheidet die Architektur zwei Interaktionsarten mit diesen externen Diensten. Wie aus dem Web 2.0-Ansatz bekannt ist, werden zunehmend notwendige Dienste nicht mehr neu entwickelt,

sondern aus bestehenden zusammengesetzt. Um dies zu ermöglichen, muss ein Dienst nach außen eine Schnittstelle anbieten. In der Trailblazers-Systemarchitektur ist solch ein direkter Zugriff durch externe Dienste über die Web-Service-Fassade möglich.

Die zweite Interaktionsart mit externen Diensten umfasst deren Nutzung. Auch die Trailblazers-Plattform soll auf die sich durchs Mashup-Prinzip eröffnenden Möglichkeiten nicht verzichten und sich die bestehenden, bereits in der Praxis geprüften externen Dienste zu Nutze machen. Hierdurch soll die Komplexität der speziellen Tätigkeiten aus dem hier betrachteten System ausgelagert werden. Diese Maßnahme soll die Spezialisierung der Trailblazers-Plattform auf die Kernaufgabe bewirken.

4.2 Verteilungsmodell

Hierbei wird die Verteilung des Systems, also der Hardware- sowie Softwarekomponenten, aus physischer Sicht untersucht und festgelegt. Dabei werden nicht nur deren Beziehungen untereinander sichtbar, sondern wie sie miteinander in Wechselwirkung stehen. In speziellen werden die zur Laufzeit existierenden Hardwareeinheiten spezifiziert, auf den dann Systemeinheiten ausgeführt werden. Welche es sein sollen und wo sich diese befinden wird in weiterer Ausführung dieser Betrachtung näher gebracht. Die bei solchen Verteilung entstehenden Entfernungen müssen dann mit geeigneten Kommunikationswegen überbrückt werden. Diese werden ebenfalls an dieser Stelle identifiziert. Abbildung 4.3 skizziert diesen Sachverhalt.

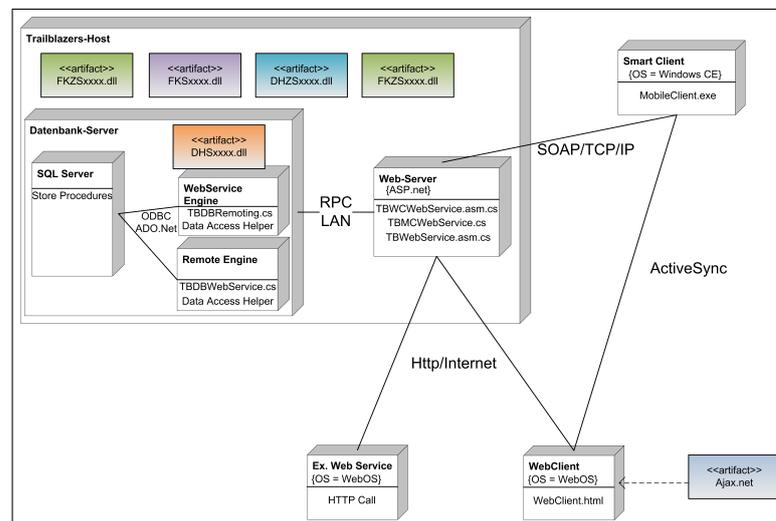


Abbildung 4.3: Einsatz- und Verteilungsdiagramm für Trailblazers

4.2.1 Hardwarekomponenten

In Abbildung 4.3 sind zwei Infrastruktureinheiten zu erkennen: der Datenbankserver und der Web-Server. Sie sind logisch zu dem Trailblazers-Host zusammengefasst. Die physische Trennung wurde eingeführt, damit eine nachträgliche Skalierung der einzelnen Komponenten möglich ist. Zwei weitere Hardwareeinheiten ermöglichen dem Benutzer mit dem System zu interagieren. Zum einen ein mobiles Gerät, welches vom Benutzer vorwiegend während der Navigation genutzt wird. Die zweite Einheit ist wahlweise eine stationäre oder portable Einheit mit deutlich höherer Bandbreite an Interaktionsmöglichkeiten und der dafür notwendigen Leistungsfähigkeit.

Aus dem Verteilungsdiagramm in Abbildung 4.3 können ebenfalls die verschiedenen Arten der Kommunikationswege entnommen werden. Zwischen den beiden Infrastruktureinheiten, dem Datenbankserver und dem Webserver, besteht eine LAN-Verbindung⁵ über die beide Systemeinheiten miteinander kommunizieren können. Die beiden Clientkomponenten greifen nicht direkt auf den Datenbankserver zu und unterscheiden sich in der Kommunikationstechnik voneinander.

Die leistungsstärkere Einheit kommuniziert auf der Ebene der Anwendungsschicht, dem HTTP. Diese einfache Kommunikationsart erlaubt nicht nur synchrone Interaktion mit dem Trailblazers-System, die bei der Bearbeitung hauptsächlich erforderlich ist. Auf diesem Kommunikationsweg kann auch eine asynchrone Kommunikation stattfinden, sodass neuartige Interaktionsformen entsprechend denen im Web 2.0 möglich werden.

Der mobile Client andererseits interagiert selten mit dem Webserver über HTTP. Diese Interaktion wird stattdessen auf der Basis der Web-Service-Schnittstellentechnik durchgeführt. In dem derzeitigen Offline-Szenario agiert die mobile Einheit hauptsächlich mit der stationären Einheit zusammen. Eine Verbindung wird vor allem dann aufgebaut, wenn eine Synchronisation mit der stationären Einheit durchgeführt werden soll. Das hierfür geeignete Protokoll ist *ActiveSync*. Der mobile Client kann ebenfalls mit einer externen GPS-Einheit verbunden werden. Diese werden aber inzwischen vermehrt in die mobile Einheit integriert, sodass ein zusätzliches GPS-Gerät nicht mehr gebraucht wird.

4.2.2 Softwarekomponenten

Eine .Net-Anwendung setzt sich aus sogenannten Assemblies zusammen, in denen Teile der Anwendung als Programmbibliotheken abgelegt werden. Um die Wartbarkeit und Verteilung

⁵Die zwischen dem Datenbankserver und dem Webserver bestehende Kommunikation kann entweder über einen Web-Service oder RMI ablaufen. Beide Kommunikationsarten haben Vor- und Nachteile. Die RMI-basierte Kommunikation würde deutlich schneller ablaufen, ein Web-Service würde dafür aber eine Entkopplung der beiden Einheiten begünstigen.

der Anwendung zu vereinfachen, werden fachlich voneinander unabhängigen Programm-Einheiten in unterschiedliche Assemblies verteilt. Auf der anderen Seite sollte ebenfalls vermieden werden, dass gleiche Assemblies an mehreren physikalischen Stellen residieren. Sie sollten nach Möglichkeit nicht dupliziert werden.

Daraus folgt: Alle Kernkomponenten befinden sich auf einem einzigen Server. Dies ist jedoch nur für den Prototyp sinnvoll, da später der Rechenbedarf für einen einzelnen Rechner zu hoch werden wird. Deshalb wird intern eine logische Trennung durchgeführt, sodass die Komponenten in einer späteren Produktionsumgebung auf unterschiedliche physikalische Server ausgelagert werden können. Auf dem Trailblazers-Host (oder einer Server-Farm) werden die jeweiligen fachlichen Artefakte (Assemblies) verteilt, die die fachliche Anwendung abbilden. Desweiteren wird dort die Datenhaltung mit den entsprechenden Abstraktions⁶- und Zugriffskomponenten⁷ enthalten sein. Der ebenfalls enthaltene Webserver bietet hier die notwendigen ASP-Seiten an, über die auf die Dienste der Trailblazers-Plattform zugegriffen werden kann.

Es existieren drei Arten von Clients, die auf den Trailblazers-Server zugreifen werden: Der Smart-Client greift auf diesen indirekt über einen Desktop-PC - auf dem auch der Webclient ausgeführt wird - über die ActiveSync-Verbindung zu. Hierzu muss die notwendige Software auf beiden Rechnern installiert sein. In der Abbildung 4.3 ist eine theoretische SOAP-Verbindung angedeutet, die im Falle einer kontinuierlichen Internet-Verbindung mit dem Server aufgebaut werden kann. Über HTTP ruft der Webclient den Trailblazers-Dienst auf, der in Form von ASP Seiten zur Verfügung gestellt wird. Im genaueren kommuniziert dieser über eine Ajax.Net Engine mit dem Server, durch die die asynchrone Kommunikation ermöglicht wird. Daneben existieren natürlich auch externe Dienste, die direkt über HTTP den Trailblazers-Dienst nutzen können.

4.3 Systemtechnische Anwendungsfälle

Abstrakte Anwendungsfälle beschreiben die möglichen Abläufe, die in Zusammenhang mit der Trailblazers-Plattform möglich sind. Die systemtechnischen Anwendungsfälle stellen die tatsächlich in dem Softwaresystem umgesetzten Anwendungsfälle dar. Das heißt, es existieren möglicherweise weitere abstrakte Anwendungsfälle (Geschäftsanwendungsfälle) die zwar sinnhaft sind, aber systemtechnisch nicht umgesetzt werden (müssen).

⁶Nach Möglichkeit sollten alle Anfrage- und Änderungsoperationen über Store Procedures realisiert werden. Diese Maßnahme führt zu einer Erhöhung des Durchsatzes und gleichzeitig zur der Verbesserung der Wartung.

⁷Durch den Einsatz vom sogenannten Data Access Helper wird der datenbankspezifische Programmcode an einer Stelle zentralisiert.

Aus der Analyse resultierende abstrakte Anwendungsfälle beschreiben die fachliche Komplexität der Abläufe, um das System für den Benutzer zu veranschaulichen. Wegen der hohen Abstraktion sind diese als Richtlinie für die Gestaltung des Systems allein nicht geeignet bzw. zu ungenau. Es ist mühselig, die dahinter stehender Komplexität zu erfassen und, unter Berücksichtigung aller Details, in ein reales Softwaresystem zu überführen. Um diese Schwierigkeit umgehen zu können und damit verbundene Risiken⁸ zu vermeiden, werden im Folgenden die abstrakten Anwendungsfälle weiter in ihre Bestandteile zerlegt.

Bei dieser Zerlegung stehen die systemtechnischen Aspekte in Vordergrund, sodass die abstrakten Anwendungsfälle an dieser Stelle durch Umsetzungsvarianten konkretisiert werden. Diese systemtechnischen Aspekte berücksichtigen die technische Architektur des hier zugrunde liegenden Systems. Durch diese Zerlegung sollen Eigenschaften der dahinter stehenden Abläufe, sowie deren Anforderungen ersichtlich werden. An dieser Stelle werden jedoch nur Anwendungsfälle betrachtet, die mit der Benutzung von Navigationskarten zusammenhängen. Die restlichen abstrakten Anwendungsfälle, die der Navigation zugeordnet sind, werden in (Thome) näher beleuchtet. Die Abbildung 4.4 zeigt die Anwendungsfälle, die für die kollaborative Erstellung von Kartenmaterial relevant sind.

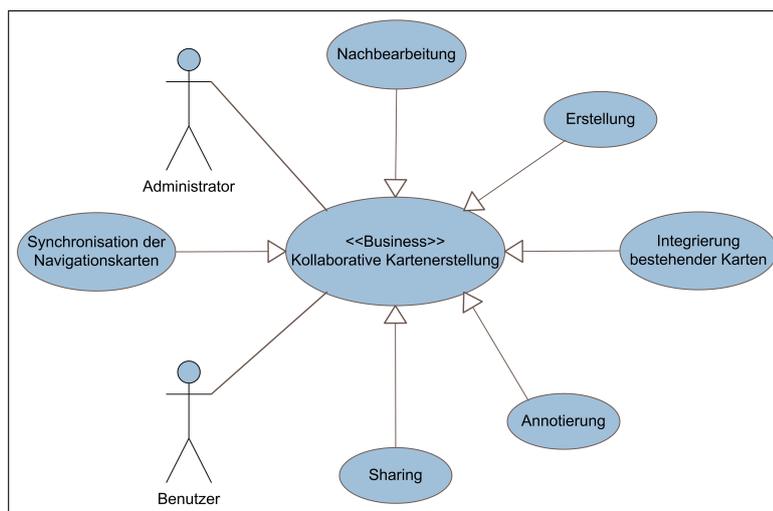


Abbildung 4.4: Für die kollaborative Erstellung relevanten abstrakten Anwendungsfälle

Im Folgenden werden alle, in der Abbildung 4.4 dargestellte abstrakte Anwendungsfälle nacheinander konkretisiert. Dabei ist zu erwarten, dass aus einem abstrakten mehrere systemtechnische Anwendungsfälle entstehen. Dieses dient der Vereinfachung und hilft bei der Gestaltung der Systemarchitektur. Jeder dieser möglichen systemtechnischen Anwendungsfälle hat eigene Anforderungen. Dabei sind die dort vorkommenden Abläufe ebenfalls zu berücksichtigen.

⁸Damit ist die höhere Wahrscheinlichkeit für das Übersehen von wichtigen Details, sowie die verfälschte Wahrnehmung des Systems gemeint.

4.3.1 Erstellung von Navigationskarten (Erstellung)

Beginnen wir als erstes, die Erstellungsabläufe genauer zu definieren, um diesen Begriff für diese Betrachtung festzulegen. Unter der Erstellung werden folgende Abläufe verstanden:

- Erstellung einer neuen Navigationskarte
- Erstellung einer neuen Route (und somit einer neuen Darstellungsschicht der Navigationskarte, auf der die Route dargestellt werden soll)

Die Erstellung von Navigationskarten kann durch einen Benutzer und einen höher privilegierten administrierenden Benutzer ausgelöst werden. Durch das System werden mehrere Benutzergruppen unterstützt, die unterschiedliche Vorgaben an eine Navigationskarte haben werden. Da es nicht sinnvoll ist gesamte Informationen auf einer Navigationskarte erscheinen zu lassen, werden hierzu bei der Erstellung bestimmte Eigenschaften abgefragt, die die Navigationskarte besitzen soll. Für die Betrachtung werden hier insgesamt drei Benutzergruppen berücksichtigt, wie die folgende Abbildung 4.5 veranschaulicht.

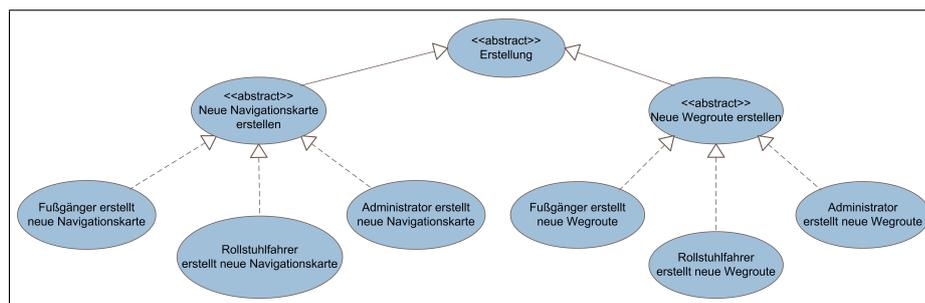


Abbildung 4.5: Mögliche systemtechnische Anwendungsfälle für die Erstellung von Navigationskarten

In dem Abschnitt Analyse wurde unter *Erstellung* ebenfalls das Hinzufügen von Neuen Karten und Points of Interest als solches betrachtet. Daneben würde auch das Hinzufügen von Tracking-Informationen sowie spezialisierten Navigationskarten als Teil der *Erstellung* identifiziert. Dieses ist aus semantischer Sicht korrekt. Da der Vorgang aber mit der Annotierung⁹ von vorhandenen Navigationskarten gleichzusetzen ist, wird aus systemtechnischer Sicht¹⁰ diese hier nicht mit aufgenommen. Die genannten Anwendungsfälle werden im folgenden Abschnitt *Annotierung* betrachtet.

Zwischen den einzelnen Benutzergruppen werden die Erstellungsabläufe sich nur in kleinen Details unterscheiden, weswegen an dieser Stelle die weitere Differenzierung dieser nicht durchgeführt wird. Die Erstellung einer Navigationskarte und einer Route erfordert aber genauere Untersuchung, da diese Anwendungsfälle sich weitgehend unterscheiden.

⁹Diese steht für einen kontinuierlichen Veränderungsprozess.

¹⁰Bildung von Komponenten und somit Bündelung von verwandten Fachklassen.

Neue Navigationskarte erstellen

Der im Abschnitt Analyse definierte abstrakter Anwendungsfall *Erstellen von Navigationskarten* stellt bereits eine gute Beschreibungsgrundlage für den gleichnamigen systemtechnischen Anwendungsfall dar. Dennoch sind bei diesem einige Änderungen durchgeführt worden. So werden weniger eingehenden Informationen erwartet als ursprünglich angenommen. Für die Erstellung einer Navigationskarte werden einmal die benutzerbezogenen Informationen sowie Daten, die die Eigenschaft der Navigationskarte beschreiben, benötigt. Beide Arten von Informationen werden über Eingabegeräte in das System übermittelt. Durch die Verschiebung einiger Abläufe zum Anwendungsfall *Annotierung* werden bei der Erstellung von Navigationskarten nur vier anstelle der ursprünglichen fünf essentiellen Schritte notwendig. Die Tabelle 4.1 fasst die Spezifikation dieses Anwendungsfalls zusammen.

Name	Neue Navigationskarte erstellen
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Bevor die gesammelten Navigationsdaten ausgewertet werden, sollten diese in irgendeiner Art in Beziehung gesetzt werden. Hierzu wird eine Karte als Aggregierungs- zum anderen als Teilungsmedium hierfür verwendet. Auf dieser Karte werden dann die erforderlichen Informationen gesammelt.
Auslöser, ggf. Motivation	Ein Benutzer möchte eine Navigationskarte erstellen.
Ergebnis	Eine Navigationskarte wird für den Benutzer erstellt.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Eigenschaften der Navigationskarte.
Vorbedingungen	Der Benutzer besitzt die erforderlichen Berechtigungen.
Nachbedingungen	Der Benutzer bekommt eine Karte zur Verfügung gestellt.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient Der Benutzer gibt seinen Benutzernamen und sein Kennwort in die Webclient-Maske ein. Hierdurch werden seine Einstellungen geladen. 2. Überprüfung der Berechtigungen 3. Eingabe der Eigenschaften der Navigationskarte 4. (optional: Schreib/Leserechte setzen)

Tabelle 4.1: Systemtechnischer Anwendungsfall - Neue Navigationskarte erstellen

Neue Route erstellen

In der Tabelle 4.2 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst. Diese würde zuvor im Abschnitt Analyse nur kurz angedeu-

tet und nicht weiter ausformuliert. Bei diesem Anwendungsfall wird angenommen, dass die Erstellung von Routen zusammen mit deren Darstellung auf der zugrunde liegenden Navigationskarte zusammenhängt, da die Route anhand der darin enthaltenen Daten berechnet wird. Für die Berechnungen werden zumindest die Trampelpfade benötigt. Wie aber zuvor erwähnt, kann sich die Navigation erheblich qualitativ steigern, wenn für die Berechnung weitere Annotierungsdaten¹¹ berücksichtigt werden. Nachdem die Route durch den Server berechnet wurde, wird diese auf die betrachtete Navigationskarte projiziert und somit zur ihre Bestandteil. Eine so erweiterte Karte stellt dann die Grundlage für die Navigation dar und kann auf ein mobiles Gerät geladen werden.

Die größte Komplexität verbirgt sich bei diesem Anwendungsfall in dem dritten essentiellen Schritt, der Spezifikation der Routeneigenschaften, was auch gleichzeitig den hier betrachteten Anwendungsfall widerspiegelt. Um diese Komplexität weiter zu verringern, werden im folgenden Abschnitt Aktivitätsdiagramme eingesetzt.

Name	Neue Route erstellen
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Die Navigationskarten beinhalten Trampelpfade, die begehbare Wege darstellen. Basierend auf diesen werden Routen berechnet. Eine solche berechnete Route muss dann einer Navigationskarte hinzugefügt werden, damit diese als Navigationsgrundlage fungiert. Dieser Anwendungsfall umschließt die Berechnung dieser Route und Darstellung derselben auf der Navigationskarte.
Auslöser, ggf. Motivation	Ein Benutzer möchte für sich eine Route erstellen.
Ergebnis	Eine Route wird für den Benutzer erstellt.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Eigenschaften der Route (also auch den Start und Endpunkt), Navigationskarte
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen, Navigationskarte besitzt gültige Informationen (Minimale Menge an Trampelpfaden).
Nachbedingungen	Der Benutzer bekommt eine Route zur Verfügung gestellt. (Diese wird auf der Navigationskarte wiedergegeben)
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Spezifikation der Eigenschaften der Route (siehe Beschreibung) 4. Berechnung der resultierenden Navigationskarte

Tabelle 4.2: Systemtechnischer Anwendungsfall - Neue Route erstellen

¹¹Zu der Routenberechnung können auch die Annotierungsinformationen in Form von Points-of-Interest oder/und Bewertungsdaten hinzugenommen werden.

4.3.2 Nachbearbeitung der Navigationskarten (Nachbearbeitung)

Hinter dem abstrakten Anwendungsfall *Nachbearbeitung der Navigationskarten* verbergen sich alle Tätigkeiten, welche das Verändern der Navigationskarte nach dem Erstellen ermöglichen sollen. Bei diesem Vorgang könnten folgende Arten der Veränderungen stattfinden:

- Änderungen der Eigenschaften der Navigationskarte,
- Änderung der berechneten Route (was ein Bestandteil der Navigationskarte darstellt),
- Änderung der Points-of-Interest,
 - POI Eigenschaften,
 - Bewertungswerte (Rating),
 - Die Anzahl der Eigenschaften (falls diese variabel gehalten werden sollten),
- Änderung der integrierten spezialisierten Karten und
- Änderung der Trampelpfade.

Diese soeben genannten Punkte nennen alle systemtechnischen Anwendungsfällen, die unter Nachbearbeitung fallen. Wichtig ist an dieser Stelle die Bedeutung von Nachbearbeitung genauer zu spezifizieren. Die bei der Nachbearbeitung vorgenommenen Änderungen betreffen in erster Linie die Änderung der bestehenden Daten. Diese Grenze wird in dieser Betrachtung etwas gedehnt, indem die Elimination von Informationsartefakten ebenfalls als eine Tätigkeit verstanden wird. Folglich wird diese auch als Bestandteil der Nachbearbeitung verstanden.

Natürlich unterscheiden sich auch diese Anwendungsfälle in Abhängigkeit von dem Benutzer, der diese ausführen möchte. Es ist ersichtlich, dass einem Benutzer sehr wahrscheinlich weniger Änderungsmöglichkeiten zur Verfügung stehen, als einem Administrator. Die Abbildung 4.6 verdeutlicht die entstehenden Anwendungsfälle, wobei von der Rolle des Benutzers ausgegangen wird.

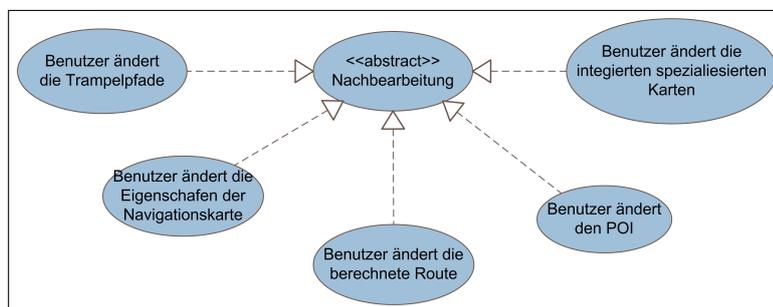


Abbildung 4.6: Mögliche systemtechnische Anwendungsfälle für die Nachbearbeitung

Ein sehr kritischer Punkt ist unter der Änderung den Point of Interest zu finden. An dieser Stelle wird die Änderung der Beschaffung der Strukturen, die die semantische Beschreibung enthalten, als Nachbearbeitung verstanden. Darunter fallen aber auch Tätigkeiten wie das Erstellen von neuen und das Entfernen von alten Strukturen.

Im Folgenden sollen die oben genannten resultierenden systemtechnischen Anwendungsfälle genauer betrachtet werden. Da diese in den meisten Fällen nur eine kleine Abweichung zu dem Anwendungsfall *Erstellung sowie Annotierung* besitzen, werden diese hier nur in tabellarischer Form spezifiziert. Eine der wesentlichen Unterschiede bei allen hier betrachteten systemtechnischen Anwendungsfällen ist die übernahmen der bereits bestehenden Daten.

Ändern der Eigenschaften der Navigationskarte

In der Tabelle 4.3 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Name	Ändern der Eigenschaften der Navigationskarten
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Jede Navigationskarte wird bei der Erstellung durch Eigenschaften genauer spezifiziert. Diese müssen möglicherweise nachträglich verändert werden. Dabei ist möglich, dass die gesamte Karte verändert wird.
Auslöser, ggf. Motivation	Ein Benutzer möchte entweder falsche Eingaben oder neue Gegebenheiten nachträglich hinzufügen.
Ergebnis	Die Navigationskarte wurde den neuen Eigenschaften angepasst.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Eigenschaften der Navigationskarte, Navigationskarte
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Der Benutzer bekommt eine Navigationskarte, die seinen Vorgaben entspricht.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Spezifikation der Eigenschaften der Navigationskarte 4. Berechnung der resultierenden Navigationskarte

Tabelle 4.3: Systemtechnischer Anwendungsfall - Ändern der Eigenschaften der Navigationskarte

Ändern der berechneten Route

In der Tabelle 4.4 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Name	Ändern der berechneten Route
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Eine berechnete Route muss anders gestaltet werden. Hierzu müssen neue Eingaben erfolgen, alte übernommen werden. Anschließend wird die Route neu berechnet und in die Karte intergiert.
Auslöser, ggf. Motivation	Ein Benutzer möchte entweder falsche Eingaben oder neue Gegebenheiten nachträglich hinzufügen.
Ergebnis	Die Route wurde den neuen Eigenschaften angepasst.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Eigenschaften der Route, Navigationskarte
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Der Benutzer bekommt eine Route, die seinen Vorgaben entspricht.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Spezifikation der Eigenschaften der Route 4. Berechnung der resultierenden Navigationskarte

Tabelle 4.4: Systemtechnischer Anwendungsfall - Ändern der berechneten Route

Ändern des POI

Dieser systemtechnische Anwendungsfall (siehe Tabelle 4.5) stellt eine Besonderheit bei der Nachbearbeitung dar. Diese wird bereits durch das auslösende Ereignis ersichtlich. Zum einen kann der Benutzer nur die Eigenschaften des POI ändern wollen, wodurch diese z. B. sich von einer Barriere zu einer überwindbaren Barriere wandeln kann. Daneben kann der Benutzer aber auch die Änderung der mit einem POI assoziierten Informationen wünschen, wie die Aktualisierung eines Fotos. In dritten und letzten Fall kann aber der Benutzer die Struktur der semantischen Beschreibung (Annotierung) verändern wollen. Somit kann eine fehlende Eigenschaft immer durch den Benutzer hinzugenommen werden, sodass dieser damit die Annotierung an seine Wünsche anpassen kann.

Name	Ändern des POI
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Die hinter einem POI verborgene Information muss möglicherweise verändert werden. Dieser Anwendungsfall beschreibt die damit verbundenen Abläufe.
Auslöser, ggf. Motivation	<ol style="list-style-type: none"> 1. Benutzer möchte die Eigenschaften des POI müssen verändert. 2. Benutzer möchte die dahinter hängende Informationen verändern. 3. Der Benutzer möchte die Struktur der semantischen Beschreibung verändern.
Ergebnis	Der POI enthält die gewünschten Änderungen.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Navigationskarte <ol style="list-style-type: none"> 1. Eigenschaften des POI 2. Beschreibungsinformationen 3. Strukturinformationen
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Der POI entspricht der Benutzervorgaben.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Spezifikation der Veränderung <ol style="list-style-type: none"> a) Spezifizierung Eigenschaften des POI (oder/und) b) Spezifizierung der Beschreibungsinformationen (oder/und) c) Spezifizierung der neuen Strukturinformationen

Tabelle 4.5: Systemtechnischer Anwendungsfall - Ändern des POI

Ändern der integrierten spezialisierten Karte

In der Tabelle 4.6 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Ändern des Trampelpfades

In der Tabelle 4.7 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Name	Ändern der integrierten spezialisierten Karte
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Falls eine, in der Navigationskarte integrierte, Karte durch eine neue ersetzt werden soll oder schlicht entfernt werden muss.
Auslöser, ggf. Motivation	Änderung der integrierten spezialisierten Karte.
Ergebnis	Die Navigationskarte wurde den neuen Eigenschaften angepasst.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, spezialisierte Karte (möglicherweise deren Eigenschaften), Navigationskarte
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Der Benutzer bekommt eine Navigationskarte, die seinen Vorgaben entspricht.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Veränderung der spezialisierten Karte (Entfernen, neu Hinzufügen, Eingabe von Eigenschaften) 4. Berechnung der resultierenden Navigationskarte

Tabelle 4.6: Systemtechnischer Anwendungsfall - Ändern der integrierten spezialisierten Karte

4.3.3 Integrierung bestehender Karten

Die Abläufe, die hinter diesem Anwendungsfall stehen, dienen dem Erweitern der Navigationskarten um für bestimmte Bereiche spezialisierte Karten. An dieser Stelle werden nur die Abläufe betrachtet, welche notwendig sind, um eine neue Karte in das System zu integrieren. Die Änderung von Karten ist Bestandteil der Nachbearbeitung und wird hier nicht weiter betrachtet. Zu den hier relevanten Abläufen gehören:

- Vorbereitungsphase,
- Vorbereitete Karte auf das System laden,
- Navigationskarte mit der digitalisierten Karte vereinbaren,
 - Spezifikation von Eigenschaften,
 - Nachjustierung.

Durch die genauere Analyse der Abläufe wird eine deutliche Trennung zwischen den beiden ersten und dem letzten ersichtlich, da der Ablauf „Navigationskarte mit der digitalisierten

Name	Ändern des Trampelpfades
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Eine Navigationskarte besteht hauptsächlich aus Trampelpfaden. Diese können durch verschiedene Umstände den realen Gegebenheiten nicht entsprechen, sodass diese entsprechend angepasst werden müssen.
Auslöser, ggf. Motivation	Benutzer hat einen verfälschten Trampelpfad erkannt und möchte diesen verbessern.
Ergebnis	Der Trampelpfad wurde angepasst.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Trampelpfad-Informationen, Navigationskarte
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Der Trampelpfad wurde den Vorgaben des Benutzers angepasst.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Spezifikation der Eigenschaften der Navigationskarte 4. Berechnung der resultierenden Navigationskarte

Tabelle 4.7: Systemtechnischer Anwendungsfall - Ändern des Trampelpfades

Karte vereinbaren“ durch eine externe Komponente vollzogen wird. Alle mit der Funktionalität der externen Komponente verbundenen Tätigkeiten sind aus systemtechnischer Sicht in einem separaten Anwendungsfall zu kapseln.

Der erste Punkt in der Ablaufliste kann aus Sicht des Systems vernachlässigt werden. Dieses wird möglich, indem das System Forderungen an den Benutzer stellt, die eine digitalisierte Karte erfüllen muss, wenn diese in das System integriert werden soll. Ist das nicht der Fall, wird diese von der Integration ausgenommen. Daraus ergeben sich also letztendlich zwei wesentlich, systemtechnische Anwendungsfälle, die in der Abbildung 4.7 dargestellt werden.

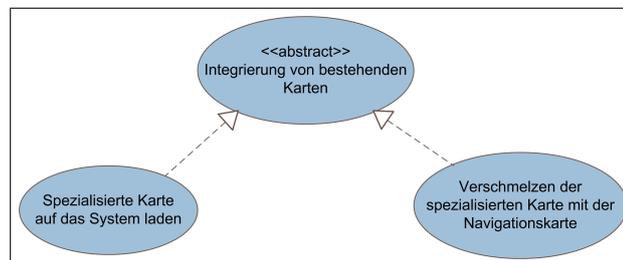


Abbildung 4.7: Mögliche systemtechnische Anwendungsfälle für die Integrierung von bestehenden Karten

Bei den beiden resultierenden, den abstrakten Anwendungsfall „Integration der bestehenden Karten“ realisierenden, systemtechnischen Anwendungsfällen wird zwischen einem gewöhnlichen- und einen administrierenden Benutzer unterschiede. Dieser wird in der Abbildung 4.7 nicht explizit dargestellt. Die weiteren Details der Anwendungsfälle werden in der folgenden beiden Unterabschnitten beschrieben.

Spezialisierte Karte auf das System laden

In der Tabelle 4.8 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Name	Spezialisierte Karte auf das System laden
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Um eine spezialisierte Karte in einer Navigationskarte integrieren zu können muss diese vorher in das System geladen werden.
Auslöser, ggf. Motivation	Benutzer möchte eine Karte in das System laden.
Ergebnis	Die spezialisierte Karte befindet sich in dem System.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, spezialisierte Karte
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen, die digitalisierte Karte entspricht den Systemvorgaben.
Nachbedingungen	Spezialisierte Karte ist in dem System verfügbar.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Auswahl der Ressource Hierbei handelt es sich um alle erforderlichen Tätigkeiten mit denen eine Ressource, die hochgeladen werden soll, spezifiziert werden kann. 4. Hochladen der Ressource

Tabelle 4.8: Systemtechnischer Anwendungsfall - Spezialisierte Karte auf das System laden

Verschmelzen der spezialisierten Karten mit der Navigationskarte

In der Tabelle 4.9 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Unter dem dritten Punkt beim Ablauf *Bearbeitung der beiden Karten* ist eine Reihe von weiteren Abläufen, die von dem Werkzeug abhängen, gebündelt worden. Diese sind an das

Name	Verschmelzen der spezialisierten Karte mit der Navigationskarte
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Hierbei wird die Bearbeitung der beiden Karten durchgeführt, sodass diese schließlich zusammenverschmelzen können.
Auslöser, ggf. Motivation	Benutzer möchte Navigationskarte und spezialisierte Karte verbinden.
Ergebnis	Navigationskarte enthält die spezialisierte Karte.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Eigenschaften der Navigationskarte, spezialisierten Karte und deren Eigenschaften.
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Der Benutzer sieht die verschmolzenen Karten.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Bearbeitung der beiden Karten Hierbei handelt es sich um alle Bearbeitungsaufgaben die erforderlich sind um, die beiden Karten genau der Anforderungen zu spezifizieren und genau zu justieren. 4. (optional: Schreib/Leserechte setzen) Für die hier hinzugefügte Resource müssen möglicherweise der Schreib/Leserechte vergeben werden. 5. Speicherung der Daten

Tabelle 4.9: Systemtechnischer Anwendungsfall - Verschmelzen der spezialisierten Karte mit der Navigationskarte

jeweilige externe Werkzeug gebunden, sind also unveränderbar, und erfordern deswegen keine weitere Betrachtung.

4.3.4 Annotierung der Navigationskarten (Annotierung)

Unter der Annotierung wird aus Sicht der Trailblazers-Plattform die Anreicherung von Karten durch jegliche Art von Informationen, welche zweckmäßig für eine erfolgreiche Navigation sind, verstanden. Wenn man der Vorstellung freien Lauf lässt, kommen hierbei viele Arten von Informationen in Frage. Zurzeit sollten aber nur die folgenden Informationsartefakte berücksichtigt werden:

- Trampelpfade (als essenzieller Bestandteil einer Navigationskarte),
- Point of Interest (als Ankerpunkt für weitere Informationen) und
- Spezialisierte Navigationskarten (als Träger von Domänen-spezifischer Informationen).

Der letztere Punkt wird einer expliziten Betrachtung in dem Abschnitt „*Integrierung bestehender Karten*“ unterzogen und wird an dieser Stelle von der Analyse ausgenommen. Es sollte jedoch an dieser Stelle unmissverständlich darauf hingewiesen werden, dass das Hinzufügen von anderen Karten als Annotierung der Kernkarte angesehen wird. Da hierbei aber externe Systemkomponenten zum Einsatz kommen, gehört dieser Aspekt aus der systemtechnischen Sicht nicht an diese Stelle.

Wie bereits angedeutet, steht der *Point of Interest* als Referenzpunkt für weitere Informationen. Deren Funktion liegt aber nicht nur in der Bündelung von vielen Informationen über einen physischen Ort, welcher kartographisch zugeordnet ist. Sie hat noch einen Klassifizierungscharakter, wodurch eine Differenzierung der verschiedenartigen Informationspunkte vorgenommen werden kann. Diese Klassifizierung wiederum kann in zwei weitere Arten unterteilt werden: Die eine Information beschreibt seine statische Eigenschaften, welche selten oder nie geändert werden. Die zweite Art der Klassifizierung besitzt einen dynamischen Charakter. Diese unterliegt der ständigen Änderung und spiegelt die häufige Änderung der realen Gegebenheiten der Umgebung wieder. Dieses kann durch einen Rating Mechanismus realisiert werden. Es ergeben sich also folgende Arten der Annotierung eines Point of Interest für das kollaborative Erstellen von Navigationskarten:

Statische Information (fundamentale Eigenschaften): Als Metapher kann an dieser Stelle ein Verkehrszeichen angesehen werden. Dieses besitzt - abhängig von seinem Aufstellungsort - bestimmte Eigenschaften, die sein Verwendungszweck beziehungsweise die Informationsart sofort verdeutlichen. Zum einem ist durch deren Form und Größe deren Wichtigkeit unterstrichen. Diese geben auch die ersten sehr groben Informationen wieder, die die Aufmerksamkeit des Betrachters unterschiedlich an sich lenken. Damit werden die wichtigsten essentiellen Informationen wiedergegeben.

Zusatzinformationen: Bleibt man bei dem Metapher der Verkehrszeichen so wird bei näherer Betrachtung deutlich, dass diese eigentlich nur ein Hinweis auf die gesamte, an den Ort verbundene Information gegeben. Aus praktischen Gründen können diese dem Autofahrer nicht die vollständige Information wiedergeben (in den meisten Fällen ist dies auch nicht erforderlich). Doch aus Sicht eines Rollstuhlfahrers, sowie eines Fußgängers (diese haben auch die benötigte Zeit die gesamte Information zu betrachten), können Zusatzinformationen die Fortbewegung noch weiter komfortabel gestalten. Mit zusätzlichen Informationen sind hier Photos, Links auf Ressource gemeint, die die Gegebenheit näher beschreiben oder schlicht etwas ausführliche Beschreibung.

Dynamischen Informationen: Unter diesem Punkt werden Informationen verstanden der ständigen Änderung unterliegen. Als Beispiel hierfür kann das Rating an diese Stelle näher gelegt werden. Ein Anwender des Navigationssystems bekommt dadurch eine Möglichkeit nach entnehme der Informationen aus einem POI, die Qualität und deren Nützlichkeit für eine bestimmte Gruppe der Anwender zu bewerten. So kann ein

E-Rollstuhlfahrer einen sandigen Weg als mittelgut überwindbar bewerten. Hierdurch kann die spätere Routenberechnung die dynamische Bewertung des POIs als Hilfinformation dazu nutzen, die Route entsprechend zu gestalten. Dieser schaut zum einem wie geeignet die Information für den Anwender und somit seiner Gruppe, die er selber dazugehört, ist, und lässt diese entsprechend der Gewichtung in die Berechnung einfließen. Durch diese höchst dynamische Art der Navigationskarten eröffnen sich weitere neue Möglichkeiten, die hier nicht weiter ausgeführt werden.

Aus den soeben gegebenen Ausführungen wird deutlich, wie wichtig und essentiell die Annotierung für die hier betrachtete Herausforderung erachtet wird. In dem Abschnitt "Erstellung" wurde darauf hingewiesen, dass die Erstellung von Annotierungen an dieser Stelle systemtechnisch besser angesiedelt ist. Aus der Erstellung von Annotierung resultierende systemtechnische Anwendungsfälle verdeutlicht die Abbildung 4.8.

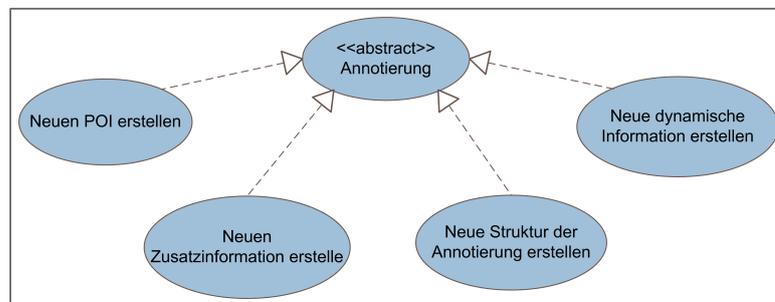


Abbildung 4.8: Mögliche systemtechnische Anwendungsfälle für die Annotierung

In der Abbildung 4.8 dargestellte systemtechnische Anwendungsfälle beziehen sich auf alle Anwendergruppen. Natürlich sind hier die höher privilegierten Anwender möglicherweise mit zusätzlichen Möglichkeiten (Funktionalitäten) ausgestattet. Diese spielen aber bei der Annotierung und damit auch in dieser Betrachtung eine untergeordnete Rolle. Im Folgenden werden die einzelnen Anwendungsfälle für sich einzeln betrachtet.

Neuen POI erstellen

In der Tabelle 4.10 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Bevor man einen POI anlegen kann, muss sich ein Benutzer über den Webclient an das System anmelden und seine Privilegien prüfen lassen. Ist diese Phase erfolgreich durchlaufen worden, kann erst dann zu den eigentlichen Abläufen, also der Erstellung von POIs übergegangen werden.

Name	Neuen POI erstellen
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Um eine Navigationskarte annotieren zu können, werden hierzu POI erstellt. Diese werden als Referenzpunkte auf der Karte sichtbar.
Auslöser, ggf. Motivation	Benutzer möchte einen POI in die Navigationskarte hinzufügen.
Ergebnis	Navigationskarte mit dem spezifizierten POI.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Navigationskarte, (statische) Eigenschaften des POI
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Der Benutzer sieht die POI auf der Navigationskarte.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Auswahl der Navigationskarte 4. Erstellung des POI Hierbei handelt es sich um alle Bearbeitungsaufgaben, die erforderlich sind, um eine POI auf der Karte zu erstellen und deren Eigenschaften zu spezifizieren. 5. (optional: Schreib/Leserechte setzen) Den hier hinzugefügten POI müssen möglicherweise Schreib/Leserechte vergeben werden. 6. Speicherung der Daten

Tabelle 4.10: Systemtechnischer Anwendungsfall - Neuen POI erstellen

Bei dem Anlegen von neuen „*Point of Interest*“ ist es wichtig diesen bereits der vorgegebenen Klassifizierung zu unterziehen. Dieses ist deshalb erforderlich damit die grundlegende Darstellung des POI, auf der Navigationskarte angepasst, den Benutzerwünschen entspricht. Durch diese Klassifizierung kann eine Navigationskarte durch entsprechend einfache Filtermethoden, mit POIs, in ihrer Darstellung erweitert werden. Daneben basiert die einfache Routenberechnung auf den so vorgegebenen statischen Merkmalen eines POIs.

Nachdem alle mit der Beschreibung des POI relevanten Aufgaben erfolgreich beendet wurden, müssen für diesen die entsprechenden Rechte vergeben werden. Es ist möglich, dass der Benutzer POI angelegt hatte, die nur für ihn bestimmt sind und für andere nicht sichtbar sein sollten (private POI). Ebenso kann es wünschenswert sein, ein POI für die Navigation den Anderen verfügbar zu machen. Dieser soll aber der Änderung durch andere Benutzer entzogen sein. Dieses ist z. B. bei öfter alternierender Verarbeitung¹² der spezifizierten Informationen der Fall. Sind alle für die Rechtevergabe relevanten Fragen geklärt, wird die Speicherung der Daten auf der Navigationskarte durchgeführt.

¹²Eine Informationseinheit, die ständig zwischen zwei Zuständen wechselt.

Neue Zusatzinformation erstellen

In der Tabelle 4.11 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Name	Neue Zusatzinformationen erstellen
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Hinter jedem POI kann eine weitere zusätzliche Information angehängt werden. Diese dienen der genaueren Beschreibung der POI.
Auslöser, ggf. Motivation	Benutzer möchte einen POI neue Zusatzinformationen hinzufügen.
Ergebnis	POI enthält die hinzugefügte Information.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Navigationskarte, Zusatzinformationen.
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Die hinter dem POI stehenden Informationen sind für den Benutzer sichtbar.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Auswahl des POI Unter diesem Punkt sind alle Aktivitäten gekapselt, die das Auffinden und Auswählen des gewünschten POIs ermöglichen. 4. Spezifizierung der Zusatzinformationen Dabei handelt es sich um Tätigkeiten, bei dem die verschiedenen Ressourcen näher beschrieben und den POI übergeben werden. 5. (optional: Schreib/Leserechte setzen) An dieser Stelle werden die Rechte für die Zusatzinformationen spezifiziert. 6. Speicherung der Daten

Tabelle 4.11: Systemtechnischer Anwendungsfall - Neue Zusatzinformationen erstellen

Für die Erstellung von neuen Informationen ist als voraussetzend die vorherige Erstellung von POI zu sehen. Erst, nachdem so ein Referenzpunkt existiert, können Zusatzinformationen in das System hinzugefügt werden. Die Bezeichnung „*Neue Zusatzinformationen erstellen*“ ist an dieser Stelle nicht ganz richtig. Es handelt sich hierbei eher um das Hinzufügen von bereits bestehender Information. Dieses ist für die eindeutige Zuordnung dieses Anwendungsfalles wichtig. Natürlich wird aus Systemsicht hier eine neue Information erstellt. Diese wird aber von dem Benutzer nicht erstellt sonder zusammengetragen.

Eine solche Information ist als Beispiel ein Foto, welches für den Benutzer wichtige Informationen über den POI liefern kann. Dieses wird in Internet verfügbar gemacht und ist über eine URL referenzierbar. Diese URL kann dann als Beschreibung an den POI übergeben werden.

Natürlich können Fotos auch in dem System selber abgespeichert werden, um eine bessere Verfügbarkeit und Stabilität der Daten zu gewährleisten. In erster Linie sollte jedoch auch die vollständige Unterstützung der Prinzipien der „*Social Software*“ gegeben sein. Zumal erfreut sich flickr eines großen Erfolges und somit großer Community, sodass seine Benutzung der breiten Masse vertraut ist. Diese Gegebenheiten können den hier zugrunde liegenden Prozess fördern.

Ein weitere wichtiger Aspekt ist die Vergabe der Schreib/Leserechte für die verschiedenen Informationsartefakte. Hierdurch können einige als sichtbar, einige als bearbeitbar markiert werden. Durch diese Differenzierung können private Informationen nicht sichtbar gemacht werden, dabei hat diese Maßnahme aber keinen Einfluss auf den eigentlichen POI. Dieses würde dazu führen, dass POI trotzdem sichtbar sein würden, obwohl die dahinter hängende Information nicht frei gegeben wäre. Zudem würde sich eine Art zweistufige Erarbeitung von Kartenmaterial bilden, indem jeder Benutzer an einen öffentlichen POI seine spezialisierte private und gleichzeitig öffentliche Information anhängen würde. Der Benutzer könnte damit in die Lage versetzt zuerst private Bemerkungen an den POI anzuhängen, bevor dieser erst diese als öffentlichen Informationen publizieren würde.

Ein sehr wichtiger Aspekt bei diesem Anwendungsfall ist die Betrachtung der systemtechnischen Stabilität. Als Gesamtheit betrachtet, stellt dieser Anwendungsfall eine genaue Wiedergabe der Abläufe, welche bei der Erstellung von POIs durchlaufen werden. Betrachtet man aber den dritten Punkt der Erstellung des POIs gesondert, so ist dieser für sich eher unstabil. Dieses kommt durch die Freiheit bei der Bereitstellung der Information. Über eine einfache Schnittstelle wird es dem Benutzer auf mehreren Wegen möglich sein, die gewünschte Information in das System zu integrieren. Dieses ist aus Sicht der Grundsätze der „*Social Software*“ und somit auch aus der Sicht der Trailblazers-Plattform höchst erwünscht. Solange die Instabilität nur diesen einzelnen Schritt betrifft, ist der gesamte Anwendungsfall soweit als stabil zu bewerten.

Neue dynamische Information erstellen

In der Tabelle 4.12 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Die Bezeichnung „*Neue dynamische Informationen Erstellen*“ kann hierbei zu einigen Missverständnissen führen. Es handelt sich bei diesem Anwendungsfall um eine Art Erstellung, aber in etwas anderer Art, als man das gewohnt ist. Durch die Abgabe seiner Bewertung hat man eigentlich nichts erstellt. Aus Sicht des Systems ist aber eine zusätzliche Bewertung hinzugekommen. Wenn also die Daten-zentrische Sicht angenommen wird, fand bei diesem Anwendungsfall eine Art Erschaffung statt. Durch diese Information gewinnt das System,

Name	Neue dynamische Informationen erstellen
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Ein POI besteht aus statischen und dynamischen Informationen. Diese werden über diesen auf einer Navigationskarte verfügbar gemacht. Dabei wird der Benutzer durch seine Interaktion mit dem POI neue dynamische Informationen erstellen.
Auslöser, ggf. Motivation	Benutzer möchte einem POI zusätzliche dynamische Information hinzufügen.
Ergebnis	POI enthält die hinzugefügte Information.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Navigationskarte, dynamische Information.
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Die hinter dem POI stehenden Informationen sind für den Benutzer sichtbar.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Auswahl des POI 4. Auswahl der Art der dynamischen Information An diesen Schritt wählt der Benutzer die Art der dynamischen Information (Art des Ratings), die dieser dem System hinzufügen möchte. 5. Eingabe der dynamischen Information Hierbei wird die Eingabe der dynamischen Information durchgeführt (möglicherweise eine Auswahl von Bewertungswerten). 6. Speicherung der Daten

Tabelle 4.12: Systemtechnischer Anwendungsfall - Neue dynamische Informationen erstellen

also gleichzeitig auch die Benutzer, zusätzliche Möglichkeit die physische Umgebung wahrzunehmen.

Das Hinzufügen von dynamischen Informationen wird durch vorherige Auswahl eines POIs, den Charakter der dynamischen Information und die Auswahl des hierfür geeigneten Wertes durchgeführt. Die Werte der einzelnen dynamischen Informationen hängen von deren Art ab. Für die Bewertung von POIs können Prozentwerte infrage kommen, welche deren Qualität beschreiben. Es könnte aber auch die Brauchbarkeit des POIs mittels einfacher ja/nein Abfrage durchgeführt werden, wie das Pandora¹³ vormacht. Eine weitere Art der dynamischen Information kann eine Folksonomie sein, durch die eine Benutzer-freie Wahrnehmung des POIs wiedergegeben werden kann. Hierbei wird der Benutzer dazu aufgefordert ein Symbol

¹³(vgl. [Pandora, 2006](#)), (vgl. [Last.fm, 2007](#))

aus der Menge der bereits bestehenden auszuwählen, oder selber einen neuen zu spezifizieren (Tagging).

Durch die Benutzung der berechneten Routen werden neuen Positionsdaten gesammelt, die für die Erstellung von neuen Trampelpfaden genutzt werden. Es ist ebenfalls möglich, die Information über begangene Wege zu sammeln und diese entsprechend den Vorgaben der dynamischen Informationen der semiautomatischen Auswertung zu unterziehen. Durch diese könnte so z. B. die Bewertung der auf der Strecke liegenden POIs ohne Benutzerinteraktion durchgeführt werden. Dieses kann aber nicht für alle Arten der dynamischen Information angewendet werden. Ungeachtet dessen kann diese Informationseinheit trotzdem für das System Relevanz haben.

Aus systemtechnischer Sicht ist hier die Performance und Verfügbarkeit des so aufgebauten Systems einmal zu diskutieren. Es ist zu erwarten, dass die externen System-Komponenten, die hier einbezogen werden, zum einen ausfallen zum anderen zu Spitzenzeiten überlastet und somit schlechte Performance liefern werden. Dieses kann durch das Gesamtsystem nur durch die Einführung von Proxys ausbalanciert werden. Konsequenterweise verschlechtert man dadurch die Echtzeit-Eigenschaften des Systems (Einbezug von nicht aktuellen Daten). Auf der anderen Seite aber sichert man das reibungslose und performante Systemverhalten.

Neue Struktur der Annotierung erstellen

In der Tabelle 4.13 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

In dem Anwendungsfall „*Neue dynamische Informationen Erstellen*“ wird beschrieben, wie eine Annotierung mit dynamischer Information von einem Benutzer durchgeführt wird. Aber bevor dieser dieses durchführen kann, muss die Art der Annotierung für einen POI erstellt werden. Die kollaborative Kartenerstellung unterstützt den Benutzer bei diesen Tätigkeiten, soll ihn in keiner Weise daran hindern. Aus diesem Grund und der Berücksichtigung der Prinzipien der „*Sozialen Software*“ wird dem Benutzer hiermit die Möglichkeit gegeben, die Annotierung frei gestalten zu können. Aus systemtechnischer Sicht ist diese Variabilität in einem Punkt dieses Anwendungsfalles zugelassen, wodurch der gesamte Anwendungsfall als hoch stabil erachtet wird.

4.3.5 Sharing der Navigationskarten (Sharing)

Einer der zentralen Aspekte der kollektiven Erarbeitung von Navigationskarten ist die Zusammenarbeit. Diese ermöglicht erst die Bildung von solch einem kollektiven kartographi-

Name	Neue Struktur der Annotierung erstellen
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Jedes POI besitzt dynamische Informationen. Es muss aber spezifiziert werden welche Art diese sein sollen und wie diese der kollaborativen Verarbeitung bereitgestellt werden.
Auslöser, ggf. Motivation	Benutzer möchte einem POI eine neue Art der Annotierung (dynamische Information) hinzufügen.
Ergebnis	POI kann durch neue Art dynamischer Information annotiert werden.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Navigationskarte, Eigenschaften der Annotierungsart.
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Die hinter dem POI stehende neue dynamische Informationsart ist für den Benutzer sichtbar.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Auswahl des POI 4. Spezifizierung und Erstellung der Annotierungsart Hierbei werden Aktivitäten durchlaufen, die die Art der Annotierung spezifizieren und anschließend diese erstellen. 5. (optional: Schreib/Leserechte setzte) An dieser Stelle werden die Rechte für die Annotierungsart spezifiziert. 6. Speicherung der Daten

Tabelle 4.13: Systemtechnischer Anwendungsfall - Neue Struktur der Annotierung erstellen

schen Wissen und macht die Realisierung dieser Idee möglich. Mit anderen Worten ist dieser abstrakte Anwendungsfall eine der Kernfunktionalitäten der hier entwickelten Plattform und muss damit mit größter Sorgfalt betrachtet werden.

Auf den ersten Blick erscheint das Teilen von Ressourcen keine so schwierige Angelegenheit zu sein. Betrachtet man das Konzept, auf dem Wikipedia aufgebaut ist, muss nur eine Zugriffsmöglichkeit gegeben werden, durch welche dann alle Benutzer auf diese Ressource zugreifen können. Dieses Prinzip könnte leicht auf die hier gestellte Problemstellung angewendet werden, es wird aber hier sehr wahrscheinlich nicht funktionieren. Es legt an der Offenheit dieses Prinzips. Jeder Benutzer würde (zumindest in der Anfangsphase) seine Position und andere private Informationen nicht wirklich ohne Bedenken freigeben wollen. Aus diesem Grund müssen neben den einfachen Zugriff, überwachende Mechanismen eingeführt werden. Mit diesen soll der Benutzer die Möglichkeit bekommen seine Privatsphäre

zu schützen und nur freiwillig Informationen an die Gemeinschaft weiter zu geben¹⁴. Durch diese Maßnahme würde man zwei Benutzergruppen diese Plattform attraktiv machen. Der einen Gruppe gehören Benutzer an, die hohen Wert auf den Schutz ihrer Privatsphäre liegen. Benutzer der anderen Gruppe betrachten möglicherweise diese Aspekte nicht so streng und können nach Belieben diese Funktionalität benutzen oder nach Möglichkeit sogar abschalten. Um dieses anbieten zu können, sind folgende Aufgaben zu erfüllen:

- Rechtevergabe auf Navigationskarten
- Rechtevergabe auf POI
- Rechtevergabe auf Zusatzinformationen
- Rechtevergabe auf spezialisierte Karten
- Rechtevergabe auf die Art der Annotierung (vielleicht nicht notwendig)
- Spezifizierung von Benutzergruppen mit vorgegebenem Recht

Bei den genannten Aufgaben kann man zwei von sich deutlich getrennten Aufgabengruppen erkennen, die Rechtevergabe und die Gruppenbildung. Bei der Gruppenbildung handelt es sich um einen gesonderten Anwendungsfall, bei dem alle Abläufe spezifiziert sind, die der Benutzer durchlaufen muss, um eine eigene Gruppe bestehend aus Benutzern zu definieren. Diese Gruppen fungieren als Entitäten die dann bestimmte Rechte bekommen. Es wäre für dieses Szenario nicht praktikabel jedem einzelnen Benutzer ein Recht vergeben zu müssen. Die Rechtevergabe selber ist hier nach jeder Art der Ressourcen unterteilt, bei dem diese Tätigkeiten durchgeführt werden sollen. Eine solche Differenzierung wäre sicherlich bei Rechtevergabe sinnvoll, bei der viele Arten von Rechten verteilt werden müssten. In diesem speziellen Fall werden nur drei Arten des Rechts vergeben dem:

- Leserecht
- Schreibrecht
- Schreib/Leserecht

Bei solcher Simplifizierung der Rechtevergabe ist die feine Unterscheidung nach Ressourcen nicht mehr notwendig. Aus diesem Grund kann diese auch zur Rechtevergabe auf einer Ressource vereinfacht werden. Unter Berücksichtigung dieser Schlussfolgerung ergeben sich folgende systemtechnische Anwendungsfälle, die in der Abbildung 4.9 dargestellt werden.

Der Anwendungsfall „*Eine Benutzergruppe Bilden*“ wird als unabhängiger Anwendungsfall betrachtet. Der zweite Anwendungsfall „*Recht an eine Ressource vergeben*“ stellt einen Anwendungsfall dar, der als Bestandteil von anderen übergeordneten systemtechnischen Anwendungsfällen zu sehen ist. Die Unterscheidung zwischen einen administrierenden Benut-

¹⁴Auch Wikipedia überlegt die Einführung von Einschränkung durch die Hinzunahmen von Benutzerhierarchien

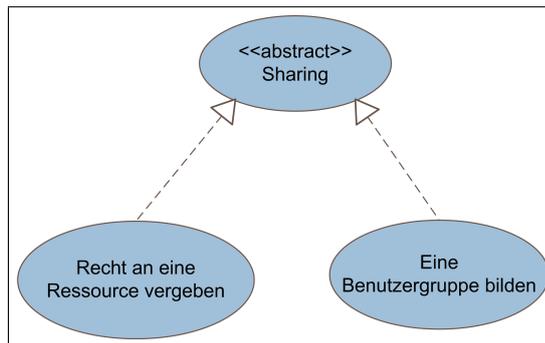


Abbildung 4.9: Mögliche systemtechnische Anwendungsfälle für das Sharing

zer und einen gewöhnlichen Benutzer wird hier nicht durchgeführt.¹⁵ Die genauere Betrachtung wird in den folgenden Abschnitten durchgeführt.

Eine Benutzergruppe bilden

In der Tabelle 4.14 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Bei diesem Anwendungsfall handelt es sich um die Definition von Benutzergruppen. Unter dem Punkt Ablauf findet man unter den vierten Punkt zwei getrennte Tätigkeiten. Diese besitzen aber für sich eine geringe Komplexität, sodass einer Trennung dieser nicht erforderlich ist.

Recht an eine Ressource vergeben

In der Tabelle 4.15 ist die Spezifikation des hier betrachteten systemtechnischen Anwendungsfalles zusammengefasst.

Bei dieser Rechtevergabe werden einzelne Ressourcen mit Benutzergruppen in Beziehung gesetzt und diese mit entsprechendem Recht versehen. Bei einem einzelnen Benutzer, der nur seine wenigen Ressourcen verwaltet, ist dieses in der Anfangszeit durchaus realisierbar. Dieses wird aber mit der Zunahme von der Anzahl der Ressourcen immer schwieriger in der Durchführung. Aus diesem Grund ist möglicherweise einer Rechtevergabe zur Zeit der Erstellung sinnvoller. Wenn also eine Ressource erstellt wird, wird bei diesem Vorgang dem Benutzer angeboten diese mit einer Gruppe in Beziehung zu setzen und das gewünschte

¹⁵Der administrierende Benutzer bekommt von dem System selber die Rechte und hat mit der fachlichen Funktionsweise des Systems erstmals nichts zu tun. Aus diesem Grund kann die Unterscheidung hier vernachlässigt werden.

Name	Eine Benutzergruppe bilden
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Hierbei definiert ein Benutzer eine Benutzergruppe. Eine Benutzergruppe fungiert dann als Aggregat für eine Menge von Benutzern.
Auslöser, ggf. Motivation	Benutzer möchte neue Benutzergruppe anlegen.
Ergebnis	Eine Benutzergruppe wurde erstellt.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Der Benutzer bekommt eine Navigationskarte, die seinen Vorgaben entspricht.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Spezifizierung der Benutzer Mit dieser Tätigkeit wählt der Benutzer für ihn zur Auswahl stehenden Benutzer, die zu einer Gruppe hinzugezogen werden sollen. 4. Spezifizierung bzw. Erstellung der Benutzergruppe Bei dieser Tätigkeit werden die ausgewählten Benutzer einer Gruppe hinzugefügt. Dabei wird die gewünschte Gruppe ausgewählt, oder eine neue angelegt. 5. (optional: Schreib/Leserechte setzte) Die spezifizierten Benutzergruppen können mit Rechten versehen werden, durch welche diese nach Außen der Community verfügbar gemacht werden können. 6. Speicherung der Daten

Tabelle 4.14: Systemtechnischer Anwendungsfall - Eine Benutzergruppe bilden

Recht zu vergeben. Damit würde der Benutzer eine nachträgliche Durchführung der Rechtevergabe nicht machen müssen, was sehr wahrscheinlich nicht vollständig aus Zeitgründen durchgeführt wäre.

Die eben geschilderte Möglichkeit der Rechtevergabe hilft sicherlich einem gewöhnlichen Benutzer aber keinen der administrative Arbeiten durchführen muss. Man stelle sich vor dieser muss einen Benutzer das Schreibrecht auf alle Point of Interests entziehen. Dabei müsste dieser jeden einzelnen auswählen und das Recht verändern. Eine andere Möglichkeit wäre Ressourcengruppen einzuführen. Die Gestaltung solcher Aggregate könnte gleich mit dem Oben genannte Ressourcenarten gesetzt werden. Hierdurch wurde der Administrator gleich alle POIs auswählen und das Recht vergeben können.

Um die Rechteverwaltung dem Benutzer etwas zu vereinfachen wäre es sinnvoll eine Art vordefinierte Gruppen zu definieren. Diese Gruppen könnten in der Anfangsphase der Res-

Name	Recht an eine Ressource vergeben
Art	Systemtechnischer Anwendungsfall
Kurzbeschreibung	Hierbei wird die entsprechende Rechte an Ressource vergeben. Dabei wird einer Ressource einer Gruppe zugewiesen in Verbindung mit dem entsprechenden Recht.
Auslöser, ggf. Motivation	Benutzer möchte eine Ressource einer Gruppe mit entsprechendem Recht ausstatten.
Ergebnis	Eine Gruppe hat entsprechendes Recht bekommen.
Akteure	Benutzer
Eingehende Informationen	Benutzerbezogene Informationen, Gruppeninformationen, Ressourceninformation.
Vorbedingungen	Der Benutzer besitzt die richtigen Berechtigungen.
Nachbedingungen	Benutzergruppe hat das entsprechende Recht auf eine Ressource erhalten.
Ablauf	<ol style="list-style-type: none"> 1. Anmeldung an der Plattform über den Webclient 2. Überprüfung der Berechtigungen 3. Auswahl der Ressource Hierbei wird der Benutzer mit Hilfe der von System gestellter Werkzeugen die gewünschte Ressource suchen und anschließend auswählen. 4. Auswahl der Gruppe Der Benutzer wird mit der Unterstützung des Systems gewünschte Benutzergruppe suchen und auswählen. 5. Rechtevergabe Die beiden Entitäten, welche vorher ausgewählt wurden, werden an dieser Stelle in Beziehung gesetzt. Anschließend bekommt die Beziehung das entsprechende Recht. 6. Speicherung der Daten

Tabelle 4.15: Systemtechnischer Anwendungsfall - Recht an eine Ressource vergeben

sourcenerstellung sofort benutzt werden, ohne dass der Benutzer sich Gedanken machen muss erstmals diese definieren zu müssen. Neben dieser Vereinfachung sollte das System trotzdem die Option der Definierung von Benutzergruppen bereitstellen. Diese könnte von etwas mehr erfahrenen Benutzern genutzt werden, um möglicherweise ungenau definierte Benutzergruppen entsprechend ihren Vorgaben zu definieren.

4.4 Anwendungsfallmodell

In der Analyse der einzelnen Anwendungsfälle wurden vermehrt sich wiederholenden Abläufen identifiziert, die in diesen integriert sind. An dieser Stelle sollte nun eine funktionale Zerlegung der primären Anwendungsfälle folgen, durch die die sekundären Anwendungsfälle identifiziert und als eigenständige Anwendungsfälle definiert werden. Daneben werden

die zwischen diesen beiden Klassen von Anwendungsfällen herrschenden Beziehungen verdeutlicht. Hierzu eignet sich am besten das Anwendungsfallmodell.

Um diese Aufgabe bewältigen zu können, werden nun bisher erarbeitete Systemanwendungsfälle zusammengetragen. Danach werden die einzelnen Anwendungsfälle nach gleichartigen Abläufen untersucht. Diese werden aus dem übergeordneten Anwendungsfall extrahiert und zu einem sekundären Anwendungsfall zusammengefasst. Hierdurch wird eine Reduzierung der internen Redundanzen innerhalb der primären Anwendungsfälle bewirkt.

Abschließend ist es notwendig, zwischen den primären und sekundären Anwendungsfällen bestehenden Beziehungen zu definieren. Zum einen werden die neuen identifizierten systemtechnischen Anwendungsfälle zu den abstrakten Anwendungsfällen mit der Spezialisierungsbeziehung verbunden. Die primären Anwendungsfälle werden außerdem mit einer Nutzungsbeziehung mit den sekundären Anwendungsfällen in Beziehung gesetzt. Zuletzt sollten aber auch die Abhängigkeitsbeziehungen zwischen den einzelnen Anwendungsfällen definiert werden.

In der Abbildung 4.10 wird das Anwendungsfallmodell skizziert, in dem alle primären und sekundären Systemanwendungsfälle veranschaulicht werden. Daneben werden zwei Beziehungen zwischen den Anwendungsfällen skizziert, die Spezialisierungs- und Benutzungsbeziehung. Aus Übersichtlichkeitsgründen wurden die Abhängigkeitsbeziehungen zwischen den primären Anwendungsfällen in der Abbildung 4.10 nicht beigefügt. Die Veranschaulichung dieser folgt nun in der Abbildung 4.11 und verdeutlicht. Dabei wird auf die oben dargestellten Beziehungen und die abgeleitete systemtechnische Anwendungsfälle verzichtet. Die einzelnen impliziten Abhängigkeitsbeziehungen sollen in textueller Form erläutert werden.

Zwischen Anwendungsfall „*Nachbearbeitung*“ und „*Integrierung bestehender Karten*“ ist eine semantische Abhängigkeit zu erkennen. Es werden bei der Nachbearbeitung sicherlich auch neue spezialisierte Karten in die Navigationskarte integriert. Aus fachlicher Sicht wurde deswegen eine Trennung zwischen Nachbearbeitung und Erstellung gezogen und das Verändern der Navigationskarte durch Hinzufügen von neuen spezialisierten Karten nicht unter Nachbearbeitung positioniert.

Ungeachtet dessen wird eine Navigationskarte durch das Hinzufügen anderer Komponenten verändert und somit als nachbearbeitet erachtet. Dabei spielen alle Anwendungsfälle, die unter der Annotierung aggregiert sind, eine wesentliche Rolle. Bei jeder Nachbearbeitung kann also ein neuer POI erstellt werden. Zu diesem kann dann auch eine Zusatzinformation und/oder eine neue dynamische Information hinzugefügt werden. Darüber hinaus kann auch dieser POI um eine neue Annotierungsart erweitert werden. Diese drei Anwendungsfälle können also Bestandteil der Nachbearbeitungsphase sein.

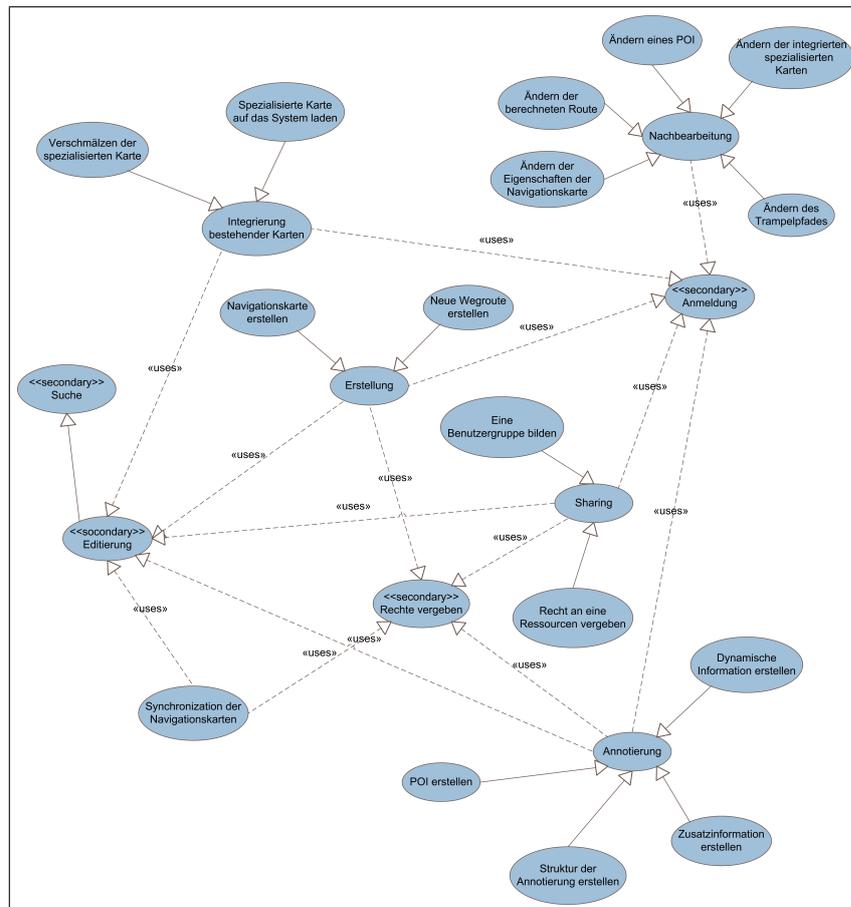


Abbildung 4.10: Anwendungsfallmodell mit primären und resultierenden sekundären Anwendungsfällen

Jede Art der Änderung, also auch das erstellen von neuen Elementen, muss sicherlich auch persistent dem System übergeben werden, damit diese nicht verloren gehen. Somit müssen die Erstellung, die Nachbearbeitung, die Nachbearbeitung, die Annotierung und die Integrierung bestehender Karten am Ende auf den Anwendungsfall „*Synchronisation der Navigationskarten*“ zurückgreifen. Dieser Anwendungsfall muss wieder den Anwendungsfall Sharing integrieren, durch welchen die Tätigkeiten die Gruppenrechte, und somit der einzelner Benutzer, in Bezug zu allen zu speicherten Ressourcen setzen kann.

4.5 Identifizierung der Fachklassen

In den zuvor definierten Anwendungsfällen würden neben den Abläufen auch zum Teil dort verwendete fachliche Gegenstände identifiziert. Dabei wurde eher auf deren zeitliche Be-

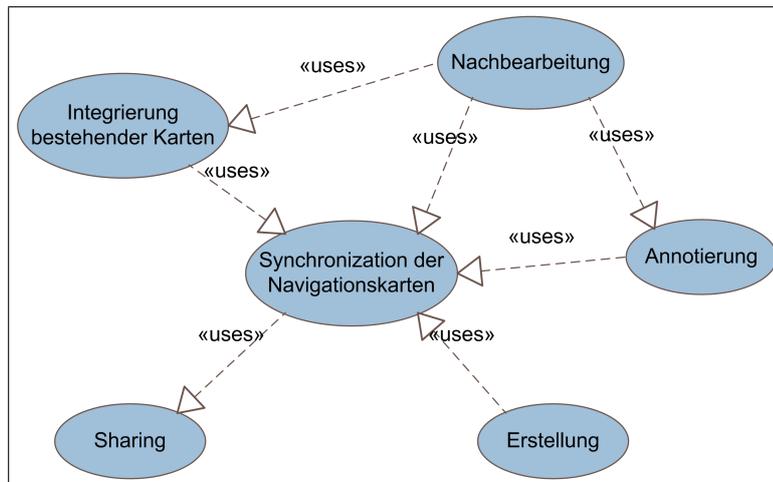


Abbildung 4.11: Abhängigkeitsbeziehung zwischen den übergeordneten Anwendungsfällen

nutzung eingegangen und weniger auf deren Eigenschaften. An dieser Stelle sollten diese identifizierten fachlichen Klassen einmal zusammengetragen und deren strukturelle Zusammenhänge verdeutlicht werden. Außerdem werden die identifizierten Fachklassen, welche das System repräsentieren sollen, näher auf fachliche Eigenschaften untersucht. Die Einzelheiten der Realisierung der identifizierten Klassen spielen an dieser Stelle eine untergeordnete Rolle und werden in späteren Verlauf der Arbeit näher beschrieben.

Betrachte man die spezifizierten Anwendungsfälle genauer, so werden folgende grobkörnige System-Entitäten dort verwendet:

- Anwendungsspezifische Fachklassen
 - Navigationskarte
 - Spezialisierte Karte
 - Route
 - Trampelpfad
 - Point of Interest
 - Spezialisierte Information
 - Dynamische Information
- Verwaltungsspezifische Fachklassen
 - Benutzergruppe
 - Benutzer
 - Rollstuhlfahrer
 - Administrator

- Benutzerrecht
- Ressource

Unter dem Punkt „Anwendungsspezifische Fachklassen“ werden alle Klassen aggregiert, welche die zentrale Funktionalität des Systems betreffen, also die Navigation selbst. Die zweite Gruppe die Verwaltungsspezifische Fachklassen fasst alle Klassen zusammen, welche hauptsächlich bei Koordinierungstätigkeiten zum Einsatz kommen, die in Zusammenhang mit der kollaborativen Arbeit anfallen. Betrachtet alle genannten Fachklassen zusammen und untersucht diese nach deren Beziehungen so entsteht folgendes, in der Abbildung 4.12 dargestellte, Klassenmodell.

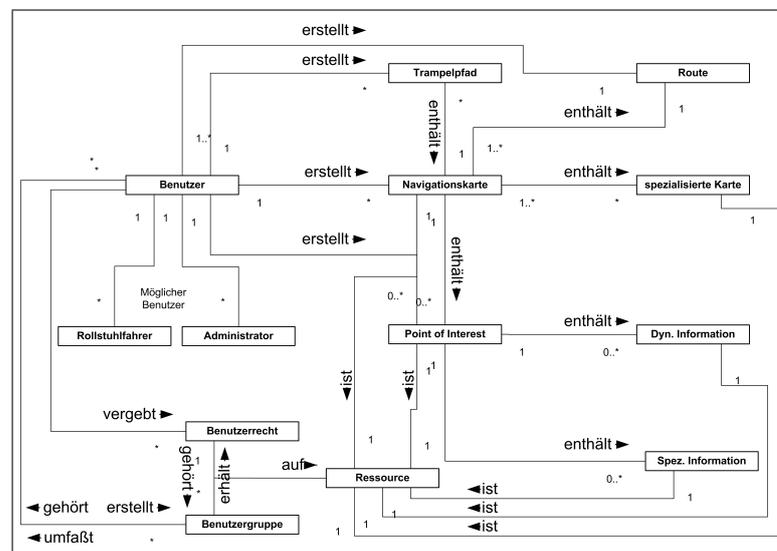


Abbildung 4.12: Zentrale fachliche Klassen und ihre Beziehungen

Dieses Klassenmodell veranschaulicht die grobe Struktur des Systems. Die Fachklassen stellen zum Teil echte Anwendungsklassen, zum Teil werden aber aus Ihnen noch weitere resultierende Klassen folgen. Im Folgenden sollen die einzelnen Fachklassen nach deren fachlichen Eigenschaften einzeln untersucht werden, wodurch ihre Beziehung zu den anderen Fachklassen erkennbar wird. Aus dieser Beschreibung werden in späteren Abschnitten die systemtechnischen Klassen erarbeitet.

4.5.1 Navigationskarte

Die Klasse Navigationskarte repräsentiert fachlich die Navigationskarte auf der jegliche, für die Navigation relevante, Informationen aggregiert werden. Dabei aggregierten Informationen werden durch weitere Fachklassen abstrahiert. Diese Klassen sind „Spezialisierte Kar-

te“, „Route“, „Trampelpfad“ und „Point of Interest“. Auf deren Basis wird Information entsprechend strukturiert.

Semantisch betrachtet ist eine Navigationskarte ohne Information wenig sinnvoll deswegen muss diese mindestens die Entitäten Trampelpfad und Route enthalten. Die verbliebenen Klassen gelten als Zusatzklassen und tragen zu der Aussagekraft der Navigationskarte bei.

Der Lebenszyklus ist abhängig von dem Benutzerverhalten und kann bis zur Vollendung der Navigationsphase, aber auch auf eine unbestimmte Zeit weiter bestehen. Der Normalfall sieht jedoch vor, dass ein Benutzer seine Navigationskarten behält und dieser immer wiederverwendet.

Wie bereits erwähnt aggregiert die Klassen Navigation vier weiteren Fachklassen. Die Beziehung zur der Fachklasse „Spezialisierte Karte“ sieht vor, dass eine Karte nicht unbedingt als Bestandteil einer Navigationskarte sein muss. Auf der anderen Seite kann eine Navigationskarte beliebig viele „Spezialisierten Karten“ enthalten. Analog hierzu ist die Beziehung zu der Fachklasse „Point of Interest“ definiert.

Wie bereits erwähnt werden zwei weitere Fachklassen von der Navigationsklasse referenziert, die immer als ein Bestandteil einer Navigationskarte betrachtet werden. Die Fachklasse „Route“ genau einmal von einer Navigationskarte instanziiert. Die Beziehung zur der Klasse „Trampelpfad“ sieht vor, dass diese mindestens einmal referenziert wird.

Auf jede Navigationskarte kann zugegriffen werden, wenn die erforderlichen Berechtigungen vorhanden sind. Eine Navigationskarte enthält selber keine Informationen darüber.

4.5.2 Spezialisierte Karte

Die fachliche Klasse „Spezialisierte Karten“ kapselt notwendige Funktionalität um die digitalisierten Karten zum einen dem System bekanntzumachen (Referenz). Auf der anderen Seite bemüht sich diese Klasse Eigenschaften der gespeicherten Karte, entsprechend den Vorgaben der Navigationskarte anzupassen. Aus fachlicher Sicht benötigt diese Klasse keine weiteren fachlichen Klassen, fungiert aber als Bestandteil einer Navigationskarte.

Eine Instanz dieser Klasse wird mit dem Einfügen einer neuer spezialisierten Karte erzeugt. Diese bleibt bestehen, bis eine explizite Aufforderung zur ihren Löschung folgt. Gleichzeitig kann es sinnvoll sein bei der Löschung alle Navigationskarten, die eine spezialisierte Karte enthalten, synchron auch die im System nicht referenzierten spezialisierten Karten zu entfernen. Diese Anforderung muss aber noch auf ihre Vor- und Nachteile beurteilt werden, da das Bestehen dieser Karten nach der Löschung der Navigationskarten von dem Benutzer gleichwohl gewünscht sein kann.

Klasse *Spezialisierte Karte* referenziert keine weiteren fachlichen Komponente. Diese steht für sich alleine und wird als eine Aggregationskomponente von der Klasse *Navigationskarte*, zu deren Vervollständigung, genutzt. Daneben besteht eine Referenzbeziehung zu der Klasse *Ressourcen*. Die Klasse selber trägt aber in sich kein Wissen über die aufgebauten Beziehungen. Dieses verhält sich analog bei der Betrachtung der Aspekte des geregelten Zugriffes. Diese werden an eine andere Klasse delegiert.

4.5.3 Route

Die Klasse „*Route*“ steht für die durch das System berechnete, aus den Vorgaben des Benutzers resultierende Route. Dabei werden alle notwendigen Informationen (Referenzen) in dieser referenziert, damit eine nachfolgende Darstellung (möglicherweise Verbesserung) der Route möglich wird. Fachlich gesehen steht diese Klasse alleine für sich da, ist aber gleichzeitig eine Aggregationskomponente der Klasse *Navigationskarte*.

Eine Route wird bei der Berechnung einer neuen Route instanziiert. Diese bleibt durch die gesamte Navigationsphase bestehen und wird erst bei der Beendigung dieser für den Benutzer nicht mehr relevant. Aus fachlicher Sicht ist eine persistente Haltung einer Route mit keinem Mehrwert verbunden¹⁶, sodass diese nach deren Nutzung aus dem System entfernt wird.

Eine Route wird von einem Benutzer erstellt und somit auch diesen zugeordnet. Hierzu besteht zwischen den beiden Klassen eine Beziehung die diese Zuordnung erkennen lässt. Eine weitere Beziehung besteht zu der Klasse *Navigationskarte*. Dabei wird eine Route einer *Navigationskarte* zugeordnet. Zur den beiden genannten Klassen aufgebauten Beziehung werden in der Klasse *Route* nicht explizit gemacht. Da eine Route nur jeweils einem Benutzer gehört ist die Betrachtung der Rechteverwaltung für diese fachliche Klasse irrelevant.

4.5.4 Trampelpfad

Hinter der fachlichen Klasse „*Trampelpfad*“ verbirgt sich die erforderliche Funktionalität, mit der man einen Trampelpfad in dem System abbilden kann. Ein Trampelpfad entsteht aus den in dem System gespeisten Bewegungs-Informationen und steht aus der Graphen theoretischer Sicht für einen neuen Verbindungspfad zwischen zwei Punkten. Diese Klasse hat als Aufgabe einen solchen Pfad systemtechnisch abzubilden. Damit wird diese Klasse zur Berechnung einer neuen Route sowie der Darstellung der begangenen Wege genutzt.

¹⁶Natürlich kann die Route für weitere Auswertung zur Verfügung gestellt werden. Welche dass sein kann, wird aus dieser Betrachtung ausgeklammert.

Ein Trampelpfad entsteht bei dem Zeitpunkt der Speisung der Positionsdaten in das System und darauf folgenden Auswertungsphase. Die Lebensdauer eines Trampelpfades wird abhängig sein von dem Hinzufügen neuer Positionsdaten. Werden in dieser Datenmenge Positionsdaten enthalten sein die den Trampelpfad fortwährend bestätigen, wird dieser nie aus dem System entfernt. Bekommt das System aus den Positionsdaten keine Bestätigung für die Relevanz dieses Trampelpfades, so wird dieser nach Ablauf einer bestimmten Zeit aus dem System entfernt.

Eine Navigationskarte aggregiert alle Trampelpfade und macht aus diesem einen Netz von Wegen auf dem die eigentliche Navigation aufbaut. Somit referenziert diese alle relevante Trampelpfade. Der Trampelpfad kann jeweils in unterschiedlicher Navigationskarte verwendet werden. Aus fachlicher Sicht ist somit der Aufbau der Referenzinformationen bei den Navigationskarten sinnvoll. Ein Trampelpfad wird aus den Daten eines Benutzers erstellt und weiter am Leben gehalten. Somit besteht eine Beziehung zwischen den beiden fachlichen Entitäten. Für einige fachliche Belange kann es sinnvoll sein diese Beziehung persistent zu halten (Bewegungsverhalten des Benutzers auswerten). Durch die Abstrahierung der Positionsdaten und die Aggregation dieser in eine Navigationskarte werden die Belange der Benutzer-Berechtigungen an die übergeordnete Klasse weiter delegiert.

4.5.5 Point of Interest

Die fachliche Klasse Point of Interest dient zur Abstrahierung positionsrelevanter Informationen. Hinter dieser Klasse verbergen sich fundamentalen Eigenschaften der Art der hier abstrahierten Interessendomäne sowie der Verarbeitung relevanten Funktionalität. Außerdem aggregiert diese Klasse weitere fachliche Klassen die eine spezifizierte Beschreibung liefern. Die zwei aggregierenden Klassen sind „*Spezialisierte Information*“ und „*Dynamische Information*“.

Ein Point of Interest entsteht, wenn der Benutzer Informationen über eine Gegebenheit in das System speichern möchte. Diese besteht, solange die aggregierende Navigationskarte existiert. Daneben kann dieser natürlich auch explizit durch den Benutzer gelöscht werden. Genauer betrachtet steht ein POI in einer Beziehung mit einer Navigationskarte und einer geographischer Position. Über die geographischen Koordinaten kann dann die Information durch andere Benutzer in ihre Karten integriert werden.

Wie bereits erwähnt aggregiert diese fachliche Klasse zwei weitere Klassen, die „*Spezialisierte Information*“ und die „*Dynamische Information*“. Dabei kann ein POI beliebig viele Instanzen diese beiden Klassen oder überhaupt keine referenzieren. Das Wissen über die Aggregations-Beziehung hält Point of Interest Klasse selber.

Die Problematik der Berechtigungen wird bei dieser Klasse ebenfalls an eine übergeordnete Komponente weiter delegiert. Aus fachlicher Sicht muss hier eine Strategie für die Zugriffskontrolle eingeführt werden, damit persönliche Daten vor Dritten geschützt werden können. Wie aber dieses von statten geht wird dieser Klasse vorenthalten.

4.5.6 Spezialisierte Information

Die fachliche Klasse „*Spezialisierte Informationen*“ repräsentiert Informations-Artefakte, welche in verschiedenen Formen existieren, um eine genauere Beschreibung zu gewährleisten. Diese fachliche Klasse kapselt somit alle notwendigen Methoden zur Speicherung, Verarbeitung und Darstellung der Informationsartefakte. Fachlich gesehen steht diese Klasse allein für sich da.

Um eine Instanz dieser Klasse konstruieren zu können, muss vorher ein geeignetes Point of Interest existieren, durch den diese Instanz wiederauffindbar gemacht wird. Ist diese Bedingung erfüllt, kann ein Informations-Artefakt, gekapselt durch diese Klasse, jederzeit von Benutzer erstellt werden. Dieser kann gleichzeitig jederzeit von dem Benutzer explizit oder implizit durch die Entfernung der übergeordneten Komponenten aus dem System entfernt werden.

Diese Klasse enthält keine aggregierende oder komponierende Assoziationen zur anderen Klassen. Demgegenüber wird diese aber von der Klasse Point Of Interest referenziert. Auf einen Informations-Artefakt sollte nicht Jeder zugreifen können. Hierzu müssen entsprechende Berechtigungen vergeben werden. Die damit verbundenen Aspekte werden durch eine übergeordnete Komponente behandelt.

4.5.7 Dynamische Information

Die Klasse „*Dynamische Information*“ stellt einen Adapter für weitere Informations-Artefakte dar, die eine gesonderte Betrachtung und einer anderen funktionalen Semantik unterlegen, als die Spezialisierte Informationen. Diese fachliche Klasse assoziiert keine weiteren Klassen.

Die Erstellung dieser Klasse wird von der Klasse Point of Interest abhängig gemacht. Nur über diese kann eine Instanz dieser Klasse erzeugt werden. Dieses ist deshalb erforderlich, da die Existenz eines POIs den Bezugspunkt zu der eigentlichen Information darstellt. Dynamische Information wäre ohne diese Referenzklasse nicht interpretierbar. Die Entfernung der Instanz dieser Klasse erfolgt entweder implizit durch die Entfernung der aggregierenden übergeordneten Komponente oder durch explizite manuelle Entfernung. Diese Klasse besitzt

selber kein Wissen über Beziehungen zur anderen Klassen, wird aber von der Klasse „*Point of Interest*“ referenziert.

Es mag auch bei dieser Klasse sinnvoll sein ein Mechanismus zur Regelung der Berechtigung einzuführen, damit die Sichtbarkeit der dort abgelegter Informationen gesteuert werden kann. Auf der anderen sind die Dynamischen Information zum einen einer sehr großen Änderung durch die Abhängigkeit zu der realen Umgebung ausgesetzt. Dazu kommen noch die Charaktereigenschaften dieser Informations-Artefakte, welche auf den Prinzipien des kollektiven Wissens aufbauen und somit jedem zur Verfügung gestellt werden sollen. Durch diese Tatsache macht die Einführung von Rechten hier wenig Sinn.

4.5.8 Benutzergruppe

Die Klasse „*Benutzergruppe*“ stellt alle Funktionen bereit um eine Gruppe von Benutzer bilden zu können. Dort werden nicht die eigentlichen Benutzer gespeichert sondern vielmehr die Beziehungen zur der Klasse Benutzer selber. Außer der Klasse Benutzer besitzt diese Klasse ebenfalls eine Assoziation zu der Klasse „*Benutzerrecht*“. Somit stellt diese eine Beziehungsklasse zwischen einen Benutzer, einer Benutzergruppe und einem Benutzerrecht. Eine Benutzergruppe wird explizit durch einen Benutzer angelegt. Dieser kann eine oder mehrere Instanzen dieser fachlichen Klasse erstellen. Das Entfernen wird ebenfalls durch den Benutzer explizit herbeigeführt.

Die Klasse Benutzer steht in einer aggregierenden Beziehung zur der Klasse Benutzerrecht. Eine Benutzergruppe kann nur ein Benutzerrecht aggregieren. Ausgesehen von dieser Klasse besitzt diese noch eine weitere Beziehung zur der Klasse Benutzer. Funktional und semantisch steht diese Beziehung zum einen für die Aggregation von Benutzern. Zum anderen verdeutlicht diese aber auch den Besitzer der Benutzergruppe.

Eine Benutzergruppe stellt eine Art der Realisierung (Lösung) der Problematik der Vergabe der Berechtigungen. Hierdurch ist diese speziell für jeden Benutzer gesondert zu instanzieren und nur für diesen sichtbar gemacht. Benutzergruppen können nicht weitergegeben werden , sodass hierbei keine Berechtigungen gesetzt werden müssen.

4.5.9 Benutzer

Die fachliche Klasse „*Benutzer*“ repräsentiert einen echten Benutzer. Genauer gesagt fungiert diese als Beschreibungs- und somit Identifizierungs-Komponente. Hierdurch kann ein Benutzer in System auffindbar gemacht werden. Diese Klasse bietet somit die Möglichkeit an, einem existierenden Benutzer seine in System gespeicherten Informationen zu finden. Diese sind zum einen die für die Navigation relevanten Informationen. Auf der anderen Seite

handelt es sich hierbei ebenfalls um Verwaltungs-Informationen, die eine Zusammenarbeit zwischen mehreren Benutzern und Ressourcen erlauben. Betrachtet man die Benutzerklasse genauer, wird es ersichtlich, dass diese eine eher abstrakte Klasse darstellt. Um die Klasse „*Benutzer*“ zu konkretisieren, besitzt diese zwei Assoziationen der Art Spezialisierung auf zwei weitere Klassen.

Eine Instanz der Klasse Benutzer wird bei dem Eintreten eines existierenden Benutzers in das System erstellt. Diese wird über die gesamte Dauer der Benutzung persistent gehalten. Verlässt der Benutzer das System, wird die Instanz gelöscht, die damit verbundenen Daten (nicht die Benutzerinformationen) sollen damit auch gelöscht werden.

Wie bereits erwähnt hat die Klasse Benutzer aus fachlicher Sicht eine Referenz- und Identifizierungs-Funktion. Um die Referenzfunktion zu genügen, muss diese Klasse Wissen über viele andere fachliche Klassen explizit besitzen. So aggregiert diese Klasse die Klassen „*Route*“, „*Trampelpfad*“, „*Benutzergruppe*“, „*Point of Interest*“, „*Spezialisierte Information*“ und „*Navigationskarte*“. Hierdurch ist der Zugriff auf alle Ressourcen sowie die Darstellung der Erstellungsbeziehung gesichert. Eine weitere Beziehungsart besteht zwischen dieser Klasse und der Klasse „*Benutzerrecht*“. Diese verdeutlicht die fachliche Richtung der Rechtevergabe. Die dritte Beziehungsart ist die Spezialisierungs-Assoziation, welche auf die konkretisierenden Klassen zeigt. Durch die Art dieser fachlichen Klasse ist eine Vergabe von Berechtigungen für diese nicht erforderlich.

4.5.10 Rollstuhlfahrer

Die Klasse „*Rollstuhlfahrer*“ hat einen konkretisierenden Charakter. Diese spezialisiert die abgeleitete fachliche Klasse Benutzer. Dabei werden alle von der Klasse „*Benutzer*“ vordefinierten Anforderungen beachtet und um die entsprechenden, von der Art der Interessen abhängigen, Funktionen erweitert.

Wie die übergeordnete Klasse „*Benutzer*“ sogleich auch diese wird bei der Eingliederung eines Benutzers in das System instanziiert. Die Lebensdauer und die Entfernung der Instanz gestaltet sich ebenfalls analog zur der Klasse „*Benutzer*“. Diese Klasse hat nur eine ableitende Assoziation zur der Klasse „*Benutzer*“. Wie auch die Klasse „*Benutzer*“ besitzt diese keine expliziten Berechtigungen.

4.5.11 Administrator

Diese Klasse beschreibt einen Benutzer mit administrativen Privilegien und repräsentiert ebenfalls eine Konkretisierungs-Variante der Klasse „*Benutzer*“. Die Instanz dieser Klasse

wird bei der Weitergabe von entsprechenden Rechten an einen „*Benutzer*“ erstellt und beschreibt somit seine neue fachliche Rolle aus der Systemsicht. Darüber hinaus besitzt diese keine nennenswerten neuen Unterscheidungsmerkmale zu der abgeleiteten Klasse.

4.5.12 Benutzerrecht

Die Klasse „*Benutzerrecht*“ ist für alle Belange der Rechtevergabe zuständig. Dabei beschreibt die Klasse alle möglichen Rechte, die an eine Gruppe und eine Ressource vergeben werden können. Bei dieser Beschreibung sind drei Klassen beteiligt, die Klassen „*Benutzergruppe*“, „*Ressource*“ und „*Benutzer*“.

Ein Benutzerrecht wird von einem Benutzer vergeben. Vergibt der Benutzer ein Recht, wird eine Instanz dieser Klasse erzeugt, welche diese Gegebenheit beschreibt. Das Recht kann jederzeit einem Benutzer entzogen werden. Dieses ist gleichzeitig der Zeitpunkt, an dem die Instanz dieser Klasse aus dem System entfernt wird.

Die Klasse „*Benutzerrecht*“ besitzt drei Beziehungen. Diese sind fachlicher Natur und beschreiben die Interaktionen zwischen den Klassen „*Benutzerrecht*“, „*Benutzergruppe*“, „*Benutzer*“ und „*Ressource*“. Ausgehend von der Klasse „*Benutzerrecht*“ gehört diese jeder Gruppe an. Damit wird angedeutet, dass ein Recht zu einer Benutzergruppe zugeordnet werden muss. Daneben zeigt die Klasse „*Benutzerrecht*“ auf die Klasse „*Ressource*“ und beschreibt damit, dass das Benutzerrecht in Verbindung mit einer Gruppe auf eine konkrete Ressource vergeben werden kann.

4.5.13 Ressource

Die Klasse „*Ressource*“ stellt eine Abstraktions-Möglichkeit dar, um die im System verschiedenartigen Ressourcen generalisieren zu können. Hierdurch ist es möglich jede Ressource einer Benutzergruppe mit entsprechendem Benutzerrecht in Beziehung zu setzen und somit eine einfache Art der Rechtevergabe zu realisieren.

Eine Instanz der Klasse „*Ressource*“ wird bei dem Einfügen von Navigationskarten, Spezialisierten Karten, Dynamischen Ressourcen und Point of Interest erzeugt. Diese Instanz bleibt solange die eben genannten Ressourcen nicht entfernt werden, in System bestehen. Wird eine dieser Ressourcen entfernt, wird auch die Instanz entfernt. Diese Klasse besitzt selber keine Beziehungen zu anderen Klassen.

4.6 Fachliches Subsystemmodell

Die zuvor gründlich durchgeführte Analyse der Abstrakten Anwendungsfälle fungiert hier als Grundlage. Anhand dieser wird die Verteilung der einzelnen Tätigkeiten beziehungsweise deren Bündelung durch Bildung von Systemkomponenten durchgeführt. Ersichtlich wird hierbei die Zusammensetzung des Gesamtsystems. An dieser Stelle werden Komponenten definiert, die fachlich verwandte Klassen in sich nach Außen kapseln und die über eine festdefinierte Schnittstelle zugreifbar machen. Jedes externe System wird in dem hier definierten Anwendungsfallmodell als ein Akteur dargestellt. Um die Integration dieser Systeme in die Architektur zu vereinfachen, empfiehlt sich diese in einer Komponente zu kapseln. Damit wird diese als Komponente mit definierter Schnittstelle einfacher in das System integrierbar sein.

Desweiteren sollten aus der fachlichen Sicht gleiche Klassen zu fachlichen Komponenten zusammengefasst werden. Als Ergebnis wird ein Entwurf einer Komponenten-basierten Anwendungs-Architektur erwartet. Diese stellt gleichzeitig eine Konkretisierung des zuvor vorgestellten Schichtenmodells und darin eingeleiteten Infrastruktur-Komponenten dar.

Es wird wie folgt vorgegangen. Zur Beginn sollen das in Abschnitt „*Identifizierung der Fachklassen*“ entstandene Klassenmodell nach fachbezogenen Zusammenhängen untersucht und folglich modularisiert werden. Als Nächstes werden die zum Einsatz kommenden externen Systeme als Komponenten aufgefasst. Entsprechend den aus den Anwendungsfällen spezifizierten Vorgaben wird dann jeweils eine Anwendungsfall-Steuerung definiert. Nach dieser Komponenten-Bildungsphase werden diese dann in einem Komponenten-Modell miteinander in Beziehung gesetzt. Darauf basiert sollen dann Subsysteme gebildet werden.

4.6.1 Fachliche Komponenten

Das zuvor erarbeitete Klassenmodell soll nun an dieser Stelle anhand der fachbezogenen Zusammenhänge in fachliche Komponenten unterteilt werden. Angefangen bei der Klasse Benutzer wird diese durch weitere Klassen spezialisiert. In diesem Fall sind es nur zwei Klasse, die Klasse Rollstuhlfahrer und Administrator. Aus fachlicher Sicht sind diese unbestritten sehr vergleichbar und unterscheiden sich nur in sehr wenige Details. Natürlich können weitere Benutzerklassen hinzugefügt werden, dieses wird aber die Kohärenz der Klassen nicht beeinflussen. Damit kann die Benutzerklasse, sowie weitere Konkretisierungen dieser, in einer gleichnamigen fachlichen Komponente zusammengefasst werden.

Die Klasse Navigationskarte stellt fachlich gesehen die Kernklasse des Systems. Diese aggregiert primär alle Informationen über die Umgebung in Form von Trampelpfaden. Diese Trampelpfade werden wiederum durch gleichnamige Klasse repräsentiert. Beide genannten

Klassen und ihre Abhängigkeiten sind fachlich zusammengehörend und werden in durch die Komponente „*Navigationskarte*“ zusammengefasst.

Eine Navigationskarte allein stellt ein Minimum an geographische Information, anhand deren man eine Route berechnen kann. Um eine Steigerung der Qualität der berechneten Routen und Quantität der zur Verfügung gestellten Informationen zu bewirken, wurden Konzepte der Annotierung zu dem System hinzugefügt. Repräsentiert werden die genannten Konzepte durch die fachliche Klassen „*Point of Interests*“ und „*Spezialisierte Karte*“. Die „*Point of Interest*“ Klasse aggregiert wiederum zwei weitere fachliche Klassen durch die zwei Arten von Informationen repräsentiert werden. Hierbei handelt es sich um die Klassen „*Spezialisierte Information*“ und „*Dynamische Information*“. Die vier genannten fachlich gleichartigen Klassen werden zu einer Komponente *Annotierung* zusammengefasst.

Die Navigationskarten und deren Annotierung bilden die Grundlage für die Navigation. Dabei wird auf deren Basis unter Beachtung der Benutzervorgaben eine Route berechnet. Eine Weg-Route wird durch die Klasse *Route* repräsentiert. Weitere konkrete Klassen, die bei der Berechnung der Route beteiligt sind, gehören aus fachlicher Sicht ebenfalls dazu. Diese werden in der Komponente „*Navigation*“ zusammengefasst.

Um eine kollektive Arbeit ermöglichen zu können, sind weitere Klassen notwendig, die eine einfache, für das Konzept ausreichende Rechteverwaltung abbilden. Hierzu würden drei Klassen „*Benutzerrecht*“, „*Benutzergruppe*“ und „*Ressource*“ identifiziert. Diese drei Klassen werden zu einer Komponente „*Rechteverwaltung*“ zusammengefasst.

Die beiden Komponenten Benutzer und Rechteverwaltung dienen der Verwaltung von Benutzern, ihren Ressourcen und damit verbundener Zugriffe. Diese beiden Komponenten sind notwendig um das Konzept der Kooperativen Zusammenarbeit bei der Erarbeitung von Navigationskarten zu ermöglichen. Somit können diese zu einem Subsystem *Kollaboration* zusammengefasst werden. Navigationskarte, Annotierung und Navigation aggregieren alle Klassen, die zu der Kernfunktionalität gehören, der Navigation. Damit stellen die drei Komponenten aus fachlicher Sicht ein Subsystem, welches die Navigation als solches ermöglicht. Die erläuterten Module sind in der Abbildung 4.13 bildlich verdeutlicht.

Außer den bisher identifizierten Komponenten existieren weitere. Diese sind:

Berechnung des Trampelpfades: Hierbei handelt es sich um eine Komponente, in der Algorithmen zur Berechnung von Trampelpfaden enthalten sind, die anhand der in das System gespeisten Bewegungsinformationen, berechnet werden. Diese gehört fachlich zur der Komponente *Navigationskarte*, durch ihre großen Umfang wird diese aber als einer Unterkomponente *Trampelpfad-Erkennung* gekapselt. Hierbei ist es wichtig durch diese Komponente eine Abstraktionsstufe einzuführen, sodass verschiedene Strategien der Berechnung das System unterstützen kann, die zeitlich versetzt

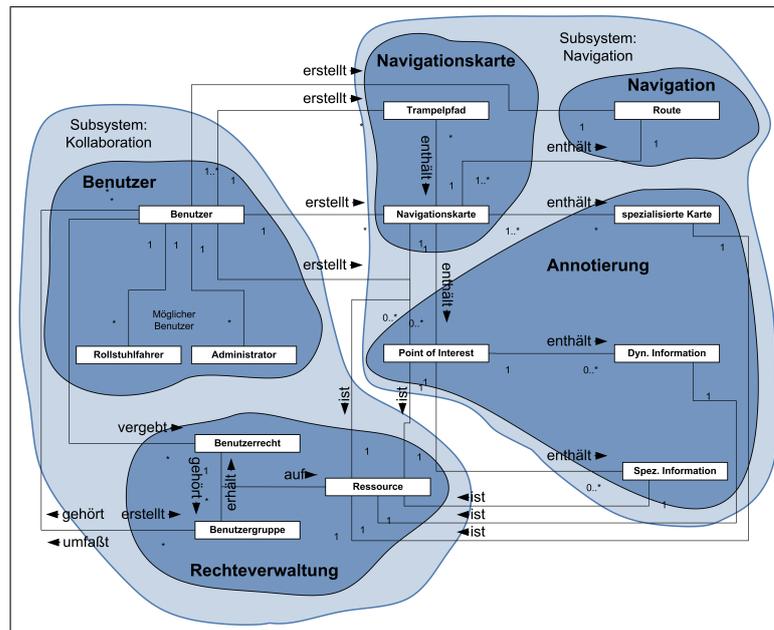


Abbildung 4.13: Analyse-Klassmodell welches die Aufteilung der Klassen in Komponenten darstellt

realisiert werden, ohne dass diese Maßnahme die Grenzen der Komponenten durchbricht.

Identifizierung des Bewegungsmusters: Hierbei handelt es sich um eine Komponente die Implementierung von bestimmten Algorithmen kapselt, welche dazu dienen sollen aus den Beschleunigungssensoren stammenden Daten Muster der Bewegung des Benutzers zu identifizieren. Aus fachlicher Sicht gehört dieser Unterkomponente *Bewegungskontext* zur der Komponente Annotierung. Wie bei der vorherigen Komponente wird auch hier eine Abstrahierung der hier zum Einsatz kommenden Strategien durchgeführt.

Bewertungsmechanismus: Wie bereits verdeutlicht, soll die Trailblazers-Plattform nicht nur auf rein statischer Information in Form von Graphen, die durch die Trampelpfad-Erkennung aus den Bewegungs-Informationen gewonnen werden, basieren. Die durch Trailblazers eingeleitete neue Generation der Navigation, sollte auch auf das Benutzerverhalten angepasst werden. In der Forschung spricht man dabei von *Social Navigation*. Eine Ausprägung hierzu ist ein Bewertungssystem welches dem Benutzer erlaubt, seine Meinung über die begangenen Wege in das System einzuspeisen. Routenberechnung die solche, höchst dynamische Information berücksichtigen würde stellt eine große Innovation in Bereich der Navigationssoftware dar. Alle hierfür notwendige Funktionalität soll in dieser Komponente konzentriert werden.

Incentivierungs-System: Incentivierungs, oder auch Belohnungssystem ist für jede web-basierte datenzentrische Anwendung essentiell. Durch dieses werden die Teilnehmer verschiedenen Reizen ausgesetzt. Diese wirken stimulierend für beide, für den Benutzer sowie für das System selber. Die verschiedenen Belohnungsstrategien, damit verbundene Funktionalität sollte in dieser Komponente angesiedelt werden.

4.6.2 Externe Systeme

Wie zuvor bereits angedeutet kommen in diesem System externe Systeme zu Einsatz. Der Mapcruncher ([Microsoft, 2006i](#)) dient als Basis für die Integration von externen, anders gear- teten Karten in die systembezogenen Navigationskarten. Um die Interaktion dieser externen Systeme mit den hier entworfenen Systemkomponenten modularisiert zu gestalten und so- mit eine Entkopplung an die Interna des externen Systems zu bewirken, wird dieser in dem Systementwurf als eine Komponente repräsentiert. Diese Abstraktionsstufe erlaubt außer- dem die Einführung von Anwendungs-relevanter Implementierung der Abläufe.

Optional können weiter externe Systeme zum Einsatz kommen wie:

Flickr: Dieses Web-Portal kann in diesem System bei der Annotierung der Point of Inte- rest genutzten Bilder unterstützend wirken. So könnte die Lagerung der verwendeten Bilder auf diese Plattform ausgelagert werden und damit verbundenen Problemstel- lungen. Hierdurch könnten die bereits vorhandenen Flickr-Funktionen genutzt werden, wodurch die Trailblazers-Plattform diese Aspekte vernachlässigen würde. Eine wichti- ge Flickr-Funktion ist die Annotierung von Bildern. Trailblazers-Plattform könnte dann auf dieses externe Wissen zugreifen und dadurch auf neue Fakten zurückgreifen, die dort impliziert sind.

Rechteverwaltung: Um das System zu vereinfachen, kann möglicherweise eine externe Rechteverwaltung genutzt werden.

Externe Navigationssysteme: Ebenfalls kann es sinnvoll sein, Dienste zu integrieren, die speziell auf eine Art von Routing konzipiert sind. Dazu gehören Bus- und Bahnaus- kunft, wobei hier nicht nur die Route sondern auch die Art der Fortbewegung abgefragt werden kann.

Externe kollaborative Arbeitsumgebungen: Hierbei handelt es sich um Systeme die ei- ne Kollektive Arbeit an beliebigen Ressourcen ermöglichen. Damit kann das Subsys- tem "Kollaboration" vollständig durch ein externes System ersetzt werden.

Externe Karten: Um eine bessere Veranschaulichung der Daten mit zu ermöglichen, be- ziehungsweise diese in Bezug auf existierende Karten setzen zu können, wird auf ein Webservice zugegriffen, der Karten in digitaler Form zur Verfügung stellt. Dabei handelt es sich um einen Web Service, über den man die Karten auf das System

lädt und damit die Trampelpfade auf diesen projiziert. Diese Komponente soll die Abstraktionsaufgaben übernehmen, sodass der Kartenprovider beliebig ausgewechselt werden kann. Die beiden gängigen WeBServices sind Microsoft MapPoint ([Microsoft, 2006f](#)) bzw. Virtual Earth ([Microsoft, 2006h](#)) und Google Maps ([Google, 2006](#)).

4.6.3 Spezifizierung der Anwendungsfallsteuerung

Um das System weiter modular zu gestalten, werden die zuvor identifizierten Anwendungsfälle beziehungsweise die damit verbundenen Abläufe als Steuerungs-Komponenten entworfen und den jeweiligen Komponenten zugeordnet unter Beachtung der fachlichen Zusammenhänge. So ergibt sich hierbei folgende Komponenten/Anwendungsfall und damit analog Anwendungsfall-Steuerung Zuordnung, welche nach den jeweiligen Komponenten unterteilt ist:

Navigationskarte

- Neue Navigationskarte Erstellen
- Änderungen der Eigenschaften der Navigationskarte
- Ändern des Trampelpfades

Annotierung

- Ändern des POI
- Ändern der intergrierten spezialisierten Karte
- Spezialisierte Karte auf das System laden
- Verschmelzen der spezialisierten Karte mit der Navigationskarte
- Neuen POI erstellen
- Neue Zusatzinformationen erstellen
- Neue dynamische Information erstellen
- Neue Struktur der Annotierung erstellen

Navigation

- Neue Route erstellen
- Ändern der berechneten Route

Benutzer

- Eine Benutzergruppe bilden

Rechteverwaltung

- Rechte an eine Ressource vergeben

4.6.4 Spezifizierung der Beziehungen zwischen den Komponenten

Die in vorherigen Abschnitten näher spezifizierten Komponenten müssen miteinander kommunizieren, um die identifizierten Anwendungsfälle abbilden zu können. Die Interaktion zwischen diesen geschieht mittels definierten Schnittstellen die diese Interaktion ermöglichen. So greift die Komponente *Benutzerverwaltung* auf die Komponente Benutzer zu. Hierdurch wird eine Beziehung zwischen einem konkreten Benutzer und einer Ressource hergestellt. Hierzu greift die gleiche Komponente auf die Komponente *Navigationskarte* zu. Über diese genannten Interaktionswege kann somit die beschriebene Beziehung zu Stande kommen.

Die Komponente *Navigationskarte* greift auf zwei Komponenten zu, die Annotierung und die Benutzerverwaltung. Der Zugriff auf die Komponente Annotierung ermöglicht eine Kapselung der Annotierungs-Informationen durch die eine Navigationskarte. Hierdurch wird der Zugriff auf diese Informationseinheiten durch den Bezug auf die Trampelpfade in Verbindung gesetzt. Diese Maßnahme ist aus fachlicher Sicht motiviert, da eine Annotierung ohne Bezug auf eine Navigationskarte nicht existieren darf. Benutzerverwaltung ist die zweite Komponente mit der die Navigationskarte interagiert. Damit kann eine Navigationskarte eigenständig den dazugehörigen (abhängig von dem Benutzerrecht) Benutzer identifizieren und somit eine Filterung der dargestellten, bei der Berechnung genutzten Informationen durchführen.

Die Komponenten *Navigation* interagiert nur mit einer Komponente *Navigationskarte*. Durch diesen Kommunikationskanal werden Informationen ausgetauscht durch die eine Berechnung der Benutzervorgaben-gerechte Route möglich wird. Navigationskarte stellt eine vollständige Schnittstelle hierzu um dieses zu ermöglichen, was fachlich motiviert ist.

Annotierung ist eine Komponente, welche mit drei anderen Komponenten gleichzeitig interagiert. Durch den Zugriff auf die Benutzerverwaltung können diese Referenzen auf Ressourcen hinterlegen, welche zu dem System neu hinzugekommen sind. Diese Referenzinformation ist die Grundlage für das Sharing von Ressourcen und somit für die kollektive Erstellung von Navigationskarten. Die direkte Beziehung zu dieser Komponente macht es möglich, dass eine Annotierungs-Information einen Benutzer gehören kann, dem die Navigationskarte nicht gehört. Durch den Zugriff auf die Komponente *Navigationskarte* wird eine Beziehung zwischen einer Navigationskarte und den geographischen Informationen hergestellt. Der Zugriff auf das externe System *MapCruncher* wird durch die dritte Beziehung verdeutlicht. Durch die hierbei verdeutlichte Schnittstelle wird die Funktionalität des MapCrunchers fachlich auf die Bedürfnisse des Systems zusammengefasst, sodass eine Integrierung einer Externen Karte möglich wird. In Abbildung 4.14 werden die Beziehungen zwischen den fachlichen Komponenten skizziert.

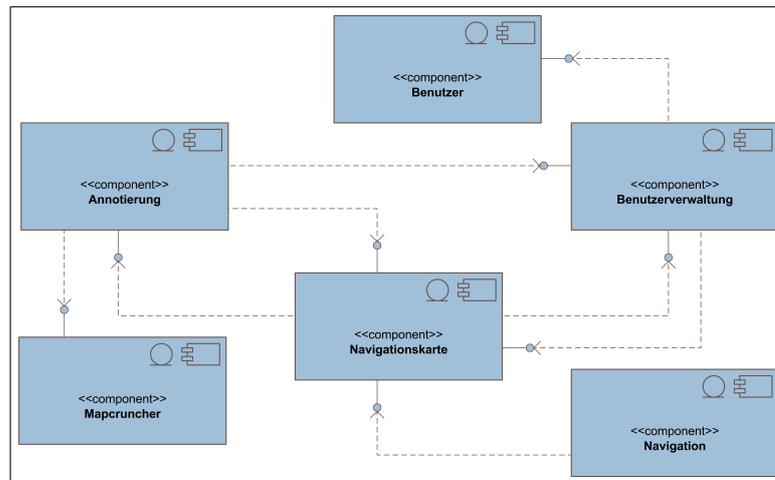


Abbildung 4.14: Veranschaulichung der Beziehungen anhand eines Komponentenmodells für fachliche Komponenten

4.6.5 Identifizierung der Subsysteme

Die nun identifizierten System-Komponenten müssen weiter auf ihre Interna konkretisiert werden. Die Konkretisierungsarbeiten sollen anhand einer Komponente *Annotierung* verdeutlicht werden. Der Aufbau der Komponente unter Berücksichtigung aller Klassen gestaltet sich wie folgt. In dem Subsystem *Annotierung* werden fünf Komponenten angesiedelt, die fachlich zusammenhängen. Hierbei handelt es sich um Komponenten „*PointOfInterest*“, „*Bewegungskontext*“, „*SpezialisierteKarte*“, „*Annotierungs-Struktur*“ und „*Annotierung*“.

Die Komponente *Annotierung* delegiert alle, an das Subsystem gestellten, Anfragen an die entsprechenden Komponenten weiter. Hierzu fungiert diese Komponente als ein Vermittler, der anhand des Kontextes der Anfrage entscheidet, wo die Anfrage weitergeleitet werden soll. Zum anderen realisiert diese eine Art Schnittstelle, über welche auf das gesamte Subsystem zugegriffen werden kann.

Die Komponente „*PointOfInterest*“ enthält die gesamte Anwendungslogik, um eine Navigationskarte mit Point of Interest annotieren zu können. Die hierbei zu durchlaufende Abläufe sind jeweils durch die einzelnen Steuerungsklassen abgebildet, die wiederum mit den entsprechenden Entitäten interagieren. Um für einen Point of Interest bei Bedarf den Bewegungskontext zu bestimmen, greift die Komponente auf eine Komponente „*Bewegungskontext*“. Diese enthält die hierzu notwendige Logik und gibt der *PointOfInterest* Komponente das berechnete Ergebnis zurück.

Die Komponente „*SpezialisierteKarte*“ baut sehr stark auf einem externen System, dem MapCruncher auf. Um in der Zukunft einen schnellen Austausch dieses Systems zu ermöglichen, wird eine Abstraktionsstufe eingeführt. Diese soll die beiden interagierenden Komponenten

voneinander entkoppelt. Hierzu wird eine Art stellvertretende Unterkomponente in die Komponente „SpezialisierteKarte“ integriert, welche die Zugriffe auf das externe System kapselt. Ein Modell der Annotierungs-Komponente wird in der Abbildung 4.15 wiedergegeben.

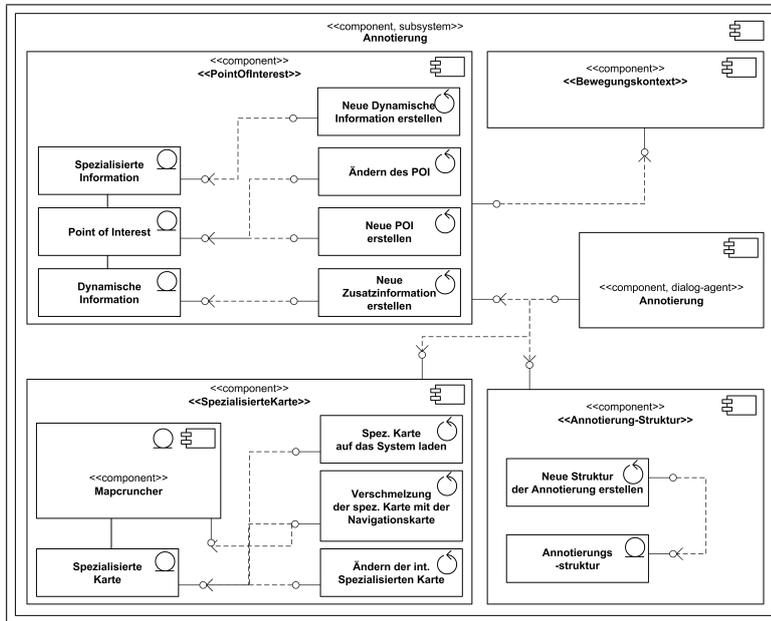


Abbildung 4.15: Modell des Subsystems „Annotierung“

Der prinzipielle Aufbau dieses Subsystems ist auf die verbliebenen Subsysteme analog angewendet worden. Natürlich unterscheiden sich diese in Details, die Diskussion dieser wurde aber sicherlich den Umfang dieser Arbeit sprengen. Aus diesem Grund wird nun im Folgenden mit dem Entwurf weiter fortgeführt.

4.7 Konkretisierung der Interaktionen

Im Folgenden soll eine Variante der Interaktion der Objekte innerhalb der entworfenen Komponente „Annotierung“ exemplarisch verdeutlicht werden. Hierbei soll ein wesentlicher Erkenntnisgewinn entstehen, bei dem mögliche Schwächen des Entwurfs sichtbar werden sollen. Dabei werden bisher außer Acht gelassen Details sichtbar wie fehlende Interaktionen und hierbei verwendeten Entitäten. Gezielt wird hier das Subsystem „Annotierung“ betrachtet beziehungsweise einige damit zusammenhängende Anwendungsfälle.

In der Abbildung 4.16 werden zwischen den jeweiligen Komponenten bestehende Zusammenhänge in Bezug auf den Anwendungsfall „Neuen POI erstellen“ untersucht.

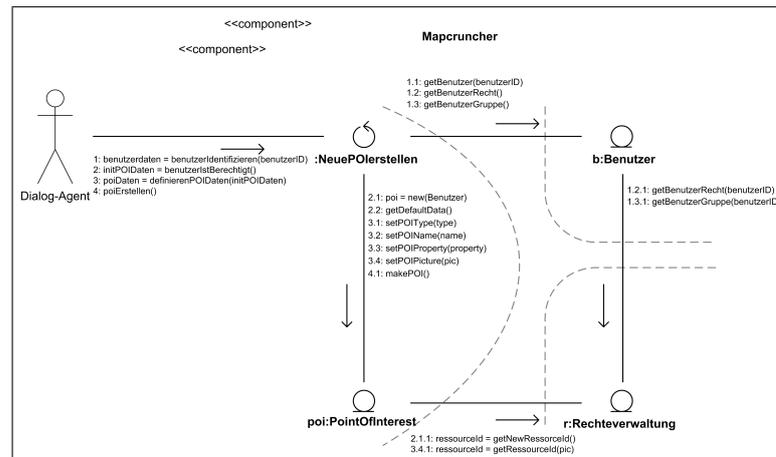


Abbildung 4.16: Veranschaulichung der Erstellung eines Point of Interest in einem Kommunikationsdiagramm

Die in der Abbildung 28 skizzierte Interaktion wird von dem Dialog-Agenten angestoßen. Dieses fungiert in dieser Betrachtung als Akteur für diesen Anwendungsfall. Aus dem Kommunikationsdiagramm wird ersichtlich, dass die Interaktion mit der Nachricht *benutzerIdentifizieren()* beginnt. Da der Dialog-Agent vorher den Verarbeitungskontext bereits erkannt hat, weiß dieser, dass diese Nachricht an die Komponente „Annotierung“ gesendet werden soll, die wiederum diese an die Komponente „PointOfInterest“ sendet. Für den richtigen Ablauf der Erstellung eines neuen Point of Interest ist die Steuerungsklasse „Neue POI erstellen“ zuständig, sodass diese auch die gesendeten Nachrichten empfängt und fachlich koordiniert.

Um also einen Benutzer identifizieren zu können, muss sich die Anwendungsfallsteuerung weiter an entsprechende Komponenten wenden. In diesem Fall ist das die Komponente Benutzer, die mit der Nachricht *getBenutzer()* kontaktiert wird, woraufhin der gesuchte Benutzer aufgefunden wird. Um den hier in Mittelpunkt stehenden Anwendungsfall abzuarbeiten zu können sind aber darüber hinaus weitere Benutzerinformationen notwendig. Hierzu befragt die Anwendungsfallsteuerung die Komponente Benutzer auf die dem identifizierten Benutzer zugewiesenen Privilegien ab, indem diese die Nachricht *getBenutzerRecht()* sendet. Die Komponente Benutzer hält die allgemeinen Benutzerinformationen, die Aspekte der Rechtevergabe wurden aber an eine weitere gleichnamige Komponente weiterdelegiert. Durch die Weiterleitung bekommt die Benutzerkomponente die geforderte Information und gibt diese der Anwendungsfallsteuerung zurück. Analog dazu verhält sich die Verarbeitung der Nachricht *getBenutzerGruppe()*. Nachdem alle notwendigen Informationen über den Benutzer geladen wurden kapselt die Anwendungsfallsteuerung diese in der Datenstruktur (DTO) *benutzerdaten* und übergibt diese an den Dialog-Agenten zurück.

Nachdem der Benutzer identifiziert wurde und seine Berechtigungen geprüft wurden, beginnt die Anwendungsfallsteuerung weiter die Abläufe des Anwendungsfalles abzuarbeiten. Die Nachricht *benutzerIstBerechtigt()*, die an die Anwendungsfallsteuerung gesendet wird, fordert diese auf einen neuen Point of Interest anzulegen. Dieses geschieht durch das Senden der Nachricht *new()* an die „*PointOfInterest*“ Klasse, wodurch eine neue Instanz dieser erstellt wird. Um die Beziehung zwischen dem Benutzer und einem Point of Interest abbilden zu können, werden bei der Instanziierung dem Objekt die Informationen des Benutzers übergeben. Desweiteren wird bei der Instanziierung der *PointOfInterest* Klasse eine Nachricht *getNewRessourceld()* an die Komponente Rechteverwaltung gesendet. Durch diese Nachricht wird eine neue „*ressourceld*“ ermittelt die dann als Referenz auf den neuen Point of Interest fungiert und diese der Komponente Rechteverwaltung zugänglich gemacht wird. Diese Maßnahme ist erforderlich um neuen Ressource der kollektiven Verarbeitung unterziehen zu können. Nach dieser Verwaltungs-relevanten Tätigkeit befragt die Anwendungsfall-Steuerung die Klasse „*PointOfInterst*“ nach den initialen Daten eines Point of Interest und baut daraus die Datenstruktur „*initPOIDaten*“, die dann an den Dialog-Agenten zurückgesendet wird.

Als nächstes fordert der Dialog-Agent die Anwendungsfallsteuerung den neu erstellten POI mit geeigneten Daten zu füllen, indem dieser die Nachricht *definierenPOIDaten()* absetzt. Damit startet die Anwendungsfallsteuerung die Abarbeitung der damit verbundenen Tätigkeiten, indem diese die Nachrichten *setPOIType()*, *setPOIName()*, *setPOIProperty()* und *setPOIPicture()* an die Klasse „*PointOfInterest*“ sendet. Bei den ersten drei Nachrichten werden primitive Eigenschaften eines Point of Interest spezifiziert. Die letzte Nachricht spezifiziert ebenfalls einer der minimalen Eigenschaften, die fügt aber der Ressource eine neue Ressource hinzu, die eine bildliche Beschreibung darstellt. Das Hinzufügen einer bildlichen Darstellung zu dem System wurde wegen der Anschaulichkeit weggelassen. Um den Anwendungsfall erfolgreich zu Ende zu führen, wird die Existenz dieser Ressource in dem System als gegeben betrachtet. Für den Aufbau einer Beziehung zwischen den beiden Ressourcen Point of Interest und einem Bild herstellen zu können, wird nur von der Klasse „*PointOfInterest*“ eine Nachricht *getRessourceld()* an die Komponente Rechteverwaltung gesendet. Dabei wird das übergebene Bild dem System übergeben und hierfür zugewiesene „*ressourcenId*“ zurückgegeben. Gleichzeitig wird aber in der Komponente die Beziehung zwischen dem neuen Point of Interest und dem Bild aufgebaut. Alle zu diesem neuen Point of Interest gehörenden Daten werden nun von der Anwendungsfallsteuerung zurück an den Dialog-Agenten in der Datenstruktur *poiDaten()* zurückgegeben.

Durch die Darstellung der Point of Interest Daten wird dessen Überprüfung möglich. Eine erfolgreiche Überprüfung wird mit der Nachricht *poiErstelle()* anschließend quittiert. Diese Nachricht fordert die Anwendungsfallsteuerung nun auf, den neuen Point of Interest nun als persistent in das System abzuspeichern. Dazu senden diese die Nachricht *makePOI()* an die

Klasse *PointOfInterest*. Damit ist nun die Erstellung eines neuen Point of Interest erfolgreich zu Ende geführt .

4.8 Zugrunde liegendes Datenbankmodell

Zuvor wurden die wesentlichen, in dem System zum Einsatz kommenden Klassen (Entitäten) identifiziert und miteinander in Beziehung gesetzt. Anhand dieser Klassen erzeugten Instanzen enthalten wichtige Zustände (Daten), die auch nach dem Programmende verfügbar sein müssen. Hierzu sollten diese in einen nichtflüchtigen Speicher strukturiert abgelegt werden, folglich in einer Datenbank. Um eine effiziente Wiedergewinnung der Daten aus einer Datenbank zu gewährleisten, ist ein Entwurf eines Datenbankmodells unabdingbar. Dabei sollte ein Spagat zwischen Performance und der Normalisierung der Daten erreicht werden.

4.8.1 Modularisierung der Datenbank

Bei dem Entwurf dieses Datenbankmodells ist neben den beiden Eigenschaften eine weitere zu berücksichtigen. Das hier entwickelte Modell soll speziell auf die persistente Speicherung geographischer Informationen spezialisiert werden. Hierzu sollten die in der Literatur empfohlenen Eigenschaften als Grundlage fungieren. Dennoch ist hierbei zu erwähnen, dass aus der logischen Sicht untersuchten Entitäten in verschiedene logische, voneinander unabhängige Datenbanken unterteilt werden können. Demnach können in einem Community-gestützten System, bei dem Navigationskarten in Partizipation mit anderen Benutzern erstellt werden, folgende logische Datenbanken identifiziert werden:

Kollaboration regelnde Datenbank: Diese ist zuständig für die Verwaltung aller Daten die die Partizipation unterstützen.

Geographische Datenbank: In dieser werden Daten gespeichert die speziell auf die räumlichen Gegebenheiten widerspiegeln.

Annotationsdatenbank: Alle Ressourcen der Annotierung werden in dieser gespeichert.

Benutzerdatenbank: Hierbei werden die Benutzer selber abgebildet.

Bei der exemplarischen Betrachtung des Datenbankschemas wurde diese Teilung außen vor gelassen. Ungeachtet dessen wird dieses in der späteren Realisierung des Schemas und damit verbundener Modularisierung der Datenbank berücksichtigt.

4.8.2 Darstellung der geographischen Informationen

Bei den gängigen geographischen Informationssystemen können zwei verschiedene Formen der Repräsentation der gespeicherten geographischen Informationen identifiziert werden. In dem ersten Fall wird Umgebung in einer Vektordarstellung abgespeichert. Damit wird eine Abbildung der Umgebung in Form von Vektornetzen bzw. Punkten möglich. Auf der anderen Seite existieren Systeme die geographische Information in Objektraster Darstellung abspeichern. Damit werden hauptsächlich graphische Informationen abgespeichert und durch entsprechende Attribute an eine geographische Position gebunden. Die Vektordarstellung bietet einen sehr wichtigen Vorteil. Durch die Nutzung dieser Darstellung und damit darin enthaltene Struktur, wird den darin abgespeicherten Daten automatisch eine Bindung an eine geographische Position verlieht. Dieser wesentliche Vorteil motiviert den Einsatz dieser Darstellung in dem Trailblazers-System. Ungeachtet dessen sollte auf die Rasterdarstellung nicht verzichtet werden, da dieses optional für spezialisierte Karten verwendet werden kann.

4.8.3 Eigenschaften eines GIS

Verschiedene in einer GIS-Datenbank enthaltene Informations-Artefakte werden in (ESRI, 2007) genannt. Folglich sind dass:

Annotation: Diese beschreiben alle Objekte mit zusätzlicher Information.

Dimension: Hierbei handelt es sich um Informationen über die Maße der Objekte, die in der GIS enthalten sind.

Eigenschaftsklasse: Diese Informationseinheit soll eine Art Klassifizierung für in der GIS enthaltene Objekte sein.

Eigenschaftssatz: Dieser speichert räumlich miteinander in Beziehung stehende Eigenschaftsklassen.

Geometrisches Netz: Hierbei handelt es sich um gerichtete Netze von Graphen, die räumliche Zusammenhänge zwischen den Objekten herstellen.

Netzwerkdatensatz: Hierbei handelt es sich um nicht gerichtete Graphen, die aus den geometrischen Netzen resultieren.

Rasterkatalog: Hierbei werden verschiedene Rasterformate definiert.

Rasterdatensatz: Dieser aggregiert vergleichbare, in Rasterformat repräsentierte Objekte.

Relationsklasse: Eine Relationsklasse beschreibt die Beziehungen zwischen verschiedenen, in der GIS enthaltenen Objekten.

Schemadatensatz: Wird für die visuelle Darstellung des schematischen Aufbaus des Graphs und der darin abgebildeten Beziehungen benutzt.

Erhebungsdatensatz: Hierbei handelt es sich Erhebungseigenschaften und damit verbundenen Maße.

Tabelle: Hierbei werden die Datenelemente der verschiedenen Tabellen definiert.

Gelände: Spezielle Darstellungen durch *Triangulated irregular networks* mit den man verschiedene Auflösungen der kartographierten Oberfläche durchführen kann.

Werkzeuge: Mit den hier verfügbar gemachten Werkzeugen soll die GIS analysiert und optimiert werden können.

Topologie: Durch hier definierten Vorgaben soll die Integrität der Daten und ihre topologischen Zusammenhänge gesichert werden.

Viele der hierbei spezifizierten Eigenschaften eines GIS können und sollen nicht in das hier betrachtete System einfließen. Ein GIS arbeitet auf einem anderen viel genaueren Genauigkeitsgrad als das hier betrachtete System. Trailblazers nutzt nur ein Bruchteil der in einem vollwertigen GIS enthaltenen Informationen. Diese geographischen Informationen werden bei dem Trailblazers-System nur für die Verdeutlichung der gesammelten Daten genutzt. Ungeachtet der bei einem GIS erkannten Komplexität sollten die dort gesammelten Erkenntnisse nicht unberücksichtigt bleiben.

Dennoch werden einige allgemeine Elemente spezifiziert, die ein Datenmodell unterstützen soll. Bei den, für den Entwurf eines solchen Modells wichtigen Richtlinien, die gleichzeitig für den Entwurf des Trailblazers-Datenbankmodells von Interesse sind, handelt es sich um:

Thematische Sichten: Ein GIS sollte mehrere Darstellungsschichten besitzen, die thematische und geographische Informationen aggregieren.

Räumliche Repräsentation: Es sollte spezifiziert werden wie sich die Datenrepräsentation gestaltet, also wie wird dieses in einer Datenbank abgebildet.

Art der Datentypklassen minimieren: In einem GIS enthaltene Elemente können klassifiziert werden. Die Anzahl der verschiedenen Klassen soll minimal gehalten werden.

Bevor mit dem Entwurf begonnen wird, sollte darauf hingewiesen werden, dass bei GIS Systemen verschiedene Datenbanken für jeweilige Domäne entwickelt werden. Trailblazers-System soll aber Domänen übergreifend sein. Aus diesem Grund ist an dieser Stelle darauf hinzuweisen, dass die Trailblazers-Datenbank gleichzeitig verschiedenen Domänen abdecken, aber auch übersichtlich bleiben muss.

Im Folgenden wird anhand der bisher identifizierten Klassen ein Klassenmodell entworfen. Dabei werden die einzelnen Klassen genauer auf die Existenz von weiteren Entitäten untersucht. In erste Linie wird jedoch die Klasse vollständig auf eine Tabelle übertragen und mit anderen Tabellen in Beziehung gesetzt.

4.8.4 Trampelpfad

Der Trampelpfad wird aus einer Menge von Positions-Punkten durch Aggregation erzeugt. Aus Sicht der Graphentheorie stellt dieser einen nicht gerichteten Graphen dar. Ein Graph besteht aus Knoten welche jeweils durch Kanten verbunden werden. Die beiden Ecken werden durch zwei geographische Positionen beschrieben, die wiederum jeweils durch den Längen- und Breitengrad dargestellt wird.

In den ersten Überlegungen besitzt jeder Vektor einen Zustand. Dieser deutet auf die Merkmale wie die Qualität und Fortschritt der Erkennung des Trampelpfades. Darüber hinaus sollten dort weitere Daten untergebracht werden die als Grundlage der Bewertung des Trampelpfades dienen sollen. In einfachsten Fall wird dort für den Pfad eine Wertung abgelegt, die wiederum mit einer Klasse von Benutzern in Beziehung gesetzt wird. Diese Art der Annotierung des Vektors gehört in eine weitere Tabelle, da diese Daten voneinander funktional abhängen. Die Hauptmerkmale stehen auch in einer funktionalen Abhängigkeit zu jedem Vektor, sollten aber aus Sicht der Performance nicht aus der Tabelle normalisiert werden.

Funktionale Abhängigkeit besteht ebenfalls zwischen dem Vektor selber und dem geographischen Punkten. Diese können sich in einigen Fällen, z. B. bei einer Kreuzung von Wegen, von mehreren Vektoren genutzt werden. Aus Sicht der Performance können die beiden Punkte in die Vektor Tabelle mit aufgenommen werden und die entstehende Redundanz in Kauf genommen. Auf der anderen Seite kommt eine Kreuzung von Vektoren nicht wirklich selten vor. Dieses könnte in dicht begangenen Bereichen zur nicht mehr akzeptablen Redundanz führen. Aus diesem Grund werden die Punkte in einer zusätzlichen Tabelle abgelegt¹⁷.

Ein Trampelpfad wird von einer oder mehreren Benutzertypen möglicherweise als nutzbar angesehen. Hierzu muss markiert werden, welche Benutzertypen das sind. Dabei kann die bereits genannte Wertung zur Darstellung von Graden der Nutzbarkeit hierbei mit aufgenommen werden. Hierzu müssen die Benutzertypen horizontal als Attribute in die Tabelle aufgenommen werden. Damit wird gleichzeitig eine nachträgliche Erweiterung ermöglicht. Die beschriebenen Entitäten und ihre Beziehungen werden in der folgenden Abbildung 4.17 dargestellt.

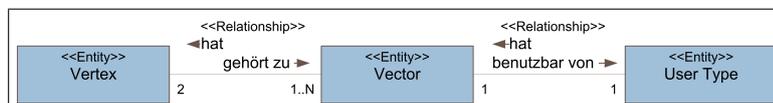


Abbildung 4.17: ER-Diagramm zur Trampelpfad und davon abhängigen Entitäten

¹⁷in der Praxis geprüft werden, wie sich das System bei einer großen Datenmenge verhält.

4.8.5 Spezialisierte Karte

Wie bereits motiviert, kann ein geographischer Bereich durch zusätzliche Darstellungsschichten um weitere Informationen angereichert werden. Hierzu muss für jede dieser Schichten (integrierten Karten) ein Bereich definiert werden, an den diese gelten sollen. Damit steht diese Tabelle in Beziehung zu der Tabelle Vertex. Dabei wird ein geographischer Punkt als Anker definiert. Darüber hinaus müssen weitere Merkmale der Karte sowie diese selbst gespeichert werden (siehe Abbildung 4.18).



Abbildung 4.18: ER-Diagramm zur Spezialisierte Karte und ihre Abhängigkeiten

4.8.6 Benutzer

Die Entität Benutzer dient der Benutzer-Verwaltung. Hiermit sollen Zusammenhänge zwischen Benutzern und anderen in System enthaltenen Ressourcen hergestellt werden. Darüber hinaus steht diese Tabelle ebenfalls in Beziehung zu einer weiteren Tabelle Rights, welche die Rechte der Benutzer beschreibt.

Neben den genannten Informationen soll über den Benutzer sein Benutzertyp ermittelt werden können. Dieses kann bei der Wahl der Benutzer-angepassten Schnittstelle genutzt werden. Daneben referenziert die Tabelle die User Group Tabelle durch. Damit wird die Bildung von Benutzergruppen, welche einzelnen Benutzer zu Gruppen aggregieren, dargestellt (siehe Abbildung 4.19).

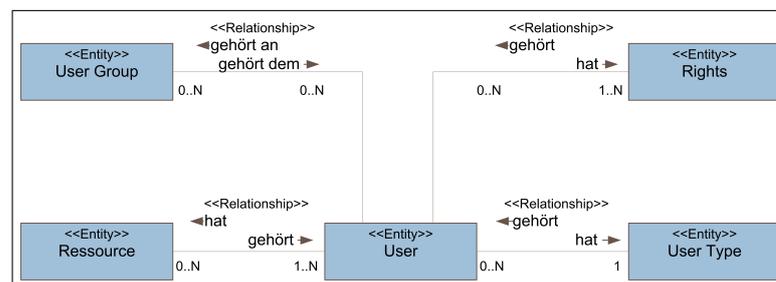


Abbildung 4.19: ER-Diagramm zur Benutzer und seine Abhängigkeiten

4.8.7 Point of Interest

Point of Interest Tabelle speichert alle seiner statischen Eigenschaften. Daneben wird hier auch eine kleine Abbildung dieser geographischen Position gespeichert. Darüber hinaus ist es wichtig, eine Beziehung zwischen einem Point of Interest und seiner geographischer Position abzulegen. Hierzu wird eine Referenzbeziehung zur der Tabelle Vertex hergestellt. Ein Point of Interest ist ebenfalls einem Verwendungstyp zugeordnet. Über diesen wird die erweiterte Navigation benutzergerechte Routen berechnen. Zur Sicherung der Integrität des Verwendungstyps wird diese Eigenschaft in eine zusätzliche Tabelle ausgelagert. Darüber hinaus ist für die erweiterte Navigation wichtig die User Type Tabelle für die Wertung des Point of Interest abhängig von Benutzer durchführen zu können. Natürlich ist für die kollektive Arbeit relevanten Aspekte wichtig auch den Eigentümer des POIs eindeutig bestimmen zu können. Hierzu wird eine Relation zu der Tabelle User aufgebaut. Die Relationen werden in der Abbildung 4.20 dargestellt.

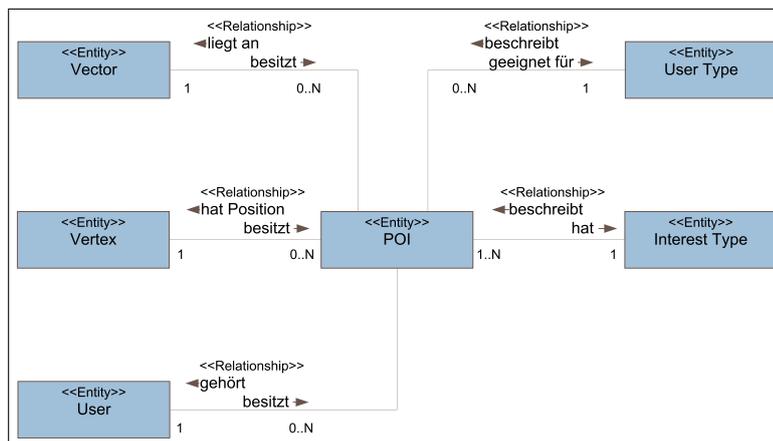


Abbildung 4.20: ER-Diagramm zur Point of Interest und seine Abhängigkeiten

4.8.8 Dynamische Information

Eine dynamische Information kann nur an einen POI angehängt werden. Hierzu besitzt diese eine Referenzbeziehung auf die Tabelle POI. Des Weiteren steht die Tabelle Dynamische Information in Beziehung zu Tabelle Ressource über die die zum Teil Kollektive Arbeit geregelt werden soll (siehe Abbildung 4.21).

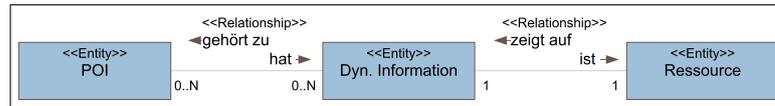


Abbildung 4.21: ER-Diagramm zur Dynamischen Information und ihre Abhängigkeit

4.8.9 Rechte

Mit der Tabelle Rights sollen die Rechte der Benutzer und der Benutzergruppen abgebildet werden. Hierzu steht diese Tabelle in Beziehung zu den eben genannten Tabellen (siehe Abbildung 4.22).

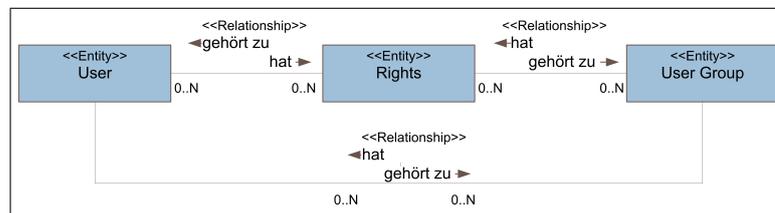


Abbildung 4.22: ER-Diagramm zur Rechten und ihre Abhängigkeiten

4.8.10 Spezialisierte Information

Spec. Information Tabelle speichert alle statischen Ressourcen, die als zusätzliche Beschreibung für diesen gelten. Darüber hinaus steht diese in Beziehung mit der Ressource Tabelle wie das bei der Tabelle Dyn. Information analog verfahren wird (siehe Abbildung 4.23).

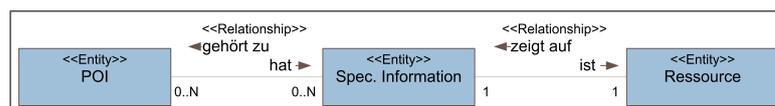


Abbildung 4.23: ER-Diagramm zur Spezialisierte Information und ihre Abhängigkeiten

4.8.11 Ressource

Um zwischen den in dem System gespeicherten Ressourcen, den Benutzern oder Benutzergruppen und dem Benutzerrecht existierende Zusammenhänge im System abbilden zu können, müssen diese vier Tabellen definiert und ihre Relationen zueinander spezifiziert werden. Ein Benutzer besitzt keine oder mehrere Ressourcen, das gleiche gilt für die Benutzergruppen. Auf eine Ressource besitzt dieser bestimmte Rechte, die in der Tabelle Benutzerrechte

persistent abgelegt werden. Darüber hinaus steht die Tabelle Ressource natürlich mit jeder Ressourcenart, also Spezialisierte Karte, Dyn. Information, Spez. Information, Point of Interest und auch der Navigationskarte in Beziehung. Abbildung 4.24 veranschaulicht diesen Sachverhalt.

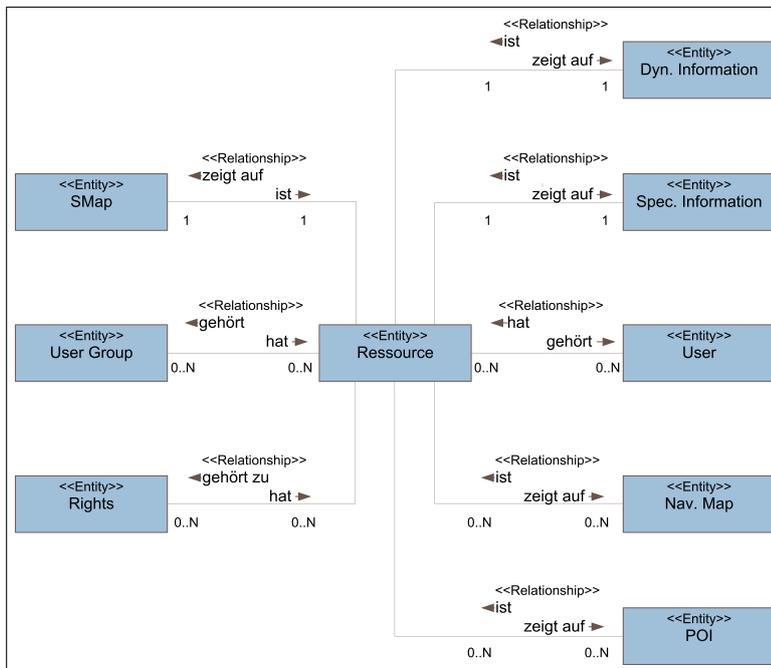


Abbildung 4.24: ER-Diagramm zur Ressource und ihre Abhängigkeiten

4.8.12 Navigationskarte

Eine Navigationskarte aggregiert bereits berechnete Trampelpfade und Routen. Diese werden in Abhängigkeit zu der Benutzerart persistent abgespeichert. Daneben wird aber auch der Besitzer der Navigationskarte festgehalten (siehe Abbildung 4.25).

4.8.13 Diskussion

Bei dem Partizipieren treffen unterschiedliche Sichtweisen der Benutzer aufeinander. Der Austausch solche Sichtweisen sollte auch von dem hier zugrunde liegenden Datenbankmodell unterstützt werden, damit eine divergierte Informationsänderung nicht unnötig entsteht. Bei der Betrachtung vorheriger Modellbetrachtung werden bereits Annotationsobjekte modelliert. Die Diskussionstabelle darf mit einer Annotierung nicht verwechselt werden, da diese hier eine andere Art der Benutzerinformation repräsentiert. Im (Balram und Dragicevic,

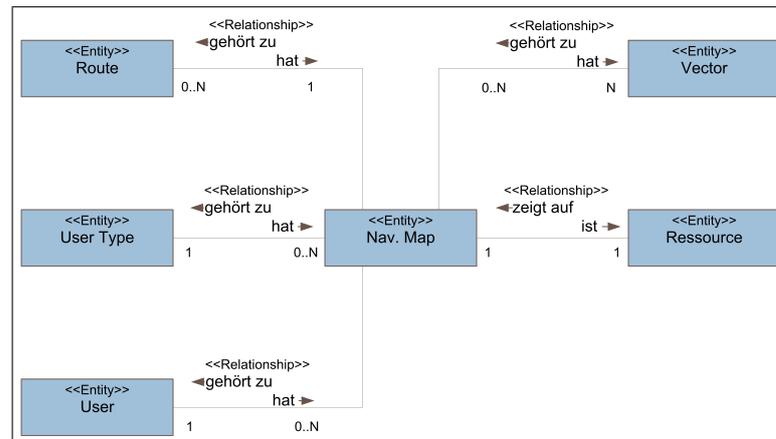


Abbildung 4.25: ER-Diagramm zur Navigationskarte und ihre Abhängigkeiten

2006) diskutierte Modell soll bei der Trailblazers-Plattform speziell für die Abbildung von Diskussionen über Annotationen, welche mit Point of Interest in Beziehung und damit mit einer realen Position stehen. Auf dieser Grundlage soll eine Diskussion möglich sein, die Position und gleichzeitig Annotations-Ressource berücksichtigt. Damit wird die Beziehung zwischen den genannten Objekten selbst zur Diskussion stehen können. Durch die Anwendung dieses Modell wird also jedes Objekt, jede Position und zwischen diesen herrschenden Relationen diskutiert werden könne. Die Abbildung 4.26 verdeutlicht den Aufbau dieses Modell.

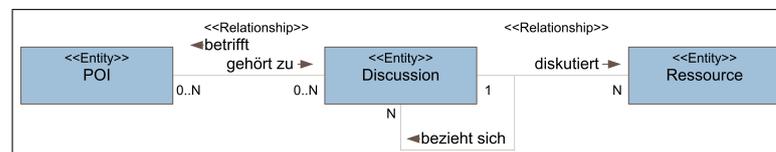


Abbildung 4.26: ER-Diagramm zur Positions- und ressourcenbezogene Diskussion angelehnt an (Rinner, 1999)

4.9 Zusammenfassung

In diesem Kapitel wurde der Entwurf des hier vorgeschlagenen Systems verdeutlicht. Dabei wurde in erster Linie die Architektur und die aus den Vorgaben entstehenden Architekturentscheidungen beschrieben und anschließend diskutiert. Als Ergebnis entstanden Architekturvorgaben, die bei der Realisierung zum tragen kommen werden.

Es wurde wie folgt vorgegangen: Zur Beginn wurde ein grober Entwurf des Systems unter Zuhilfenahme von Methoden, die aus dem objektorientierten Design stammen, durchgeführt.

Dabei wurde das System in Hinsicht auf seine logische sowie physikalische Verteilung gegliedert.

Die in der Analyse durchgeführte Betrachtung der möglichen fachlichen Anwendungsfälle wurde noch einmal unter technisch versierten Gesichtspunkten durchgeführt. Die dabei untersuchten Sachverhalte konkretisierten das Systemverhalten in technischer Hinsicht. Hierbei durchgeführte Arbeiten halfen die Redundanzen des Systemverhaltens aufzuzeigen und diese anschließen zu reduzieren. Damit wurde eine saubere funktionale Trennung geschaffen. Hierbei wurde eine Reihe von Subsystemen definiert, was das hier betrachtete System modular macht.

Das hier entworfene System baut sehr stark auf eine Datenbasis auf, die dabei helfen soll, die gesammelten Daten kohärent und redundanzfrei zu halten sowie diese nach Bedarf wieder auffindbar zu machen. Hierzu wurde eine Untersuchung der in den bestehenden geographischen Informationssystemen enthaltenen Ansätze durchgeführt. Dabei gewonnene Erkenntnisse gingen anschließend in den Entwurf des Datenbankmodells ein. Das Datenbankmodell erfüllt die wichtigsten Anforderungen, um die Struktur der erzeugten geographischen Daten zu bewahren.

5 Realisierung

Dieser Abschnitt hat als Zielsetzung, Umsetzung beziehungsweise den Fortschritt der Realisierung der zuvor in dieser Arbeit erarbeiteten Konzepte zu verdeutlichen. Hierzu wird zuvor mit der Beschreibung der technischen Rahmenbedingungen begonnen, die während der Entwicklungs- und Testphase gegeben waren. Um den Stand der Entwicklung zu verdeutlichen, werden die gestellten Zielsetzungen evaluiert. Der im Rahmen dieser Arbeit betrachtete Lösungsansatz beinhaltet viele Innovationen, die erforderlich sind für seinen Erfolg. Nicht alle diese Herausforderungen können und sollen im Rahmen dieser Arbeit umgesetzt werden. Andererseits sollen die offenen Punkte nicht vorenthalten werden, die bei der Umsetzung ausgelassen wurden. Diese werden in einzelnen diskutiert und mögliche Vorschläge zu deren Fertigstellung unterbreitet. Essentiell für die Wiedergabe des Entwicklungsstands ist auch die Offenlegung der bei der Umsetzung neugewonnenen Erkenntnisse. Die dort erfolgte Diskussion birgt neue Problemstellungen in sich, die in der Weiterentwicklung dieses Lösungsansatzes nicht unberücksichtigt bleiben sollen.

5.1 Technischen Rahmenbedingungen

Zur Beginn sollen die festgelegten technischen Rahmenbedingungen beschrieben werden, die in der Realisierungs- und in der Testphase gelten. Die Abbildung 5.1 verdeutlicht diese einleitend. Danach werden die Einzelheiten konkretisiert.

5.1.1 Entwicklungsumgebung

Für die Umsetzung erforderliche, durch hinreichende Abwägung der Vor- und Nachteile der geprüften Entwicklungsumgebungen fiel die Wahl auf *Microsoft Visual Studio 2005* (Microsoft, 2006k). Dort wird eine breite Palette integrierter Werkzeuge angeboten, die sich in verschiedenen Problembereichen als hilfreich erwiesen haben. Zudem stellt diese Umgebung eine sehr einfache Möglichkeit zur automatisierten Erzeugung Web-basierter Dienste anhand vorher durchgeführter Schnittstellenspezifikation, die anschließend einfach angebunden werden können.

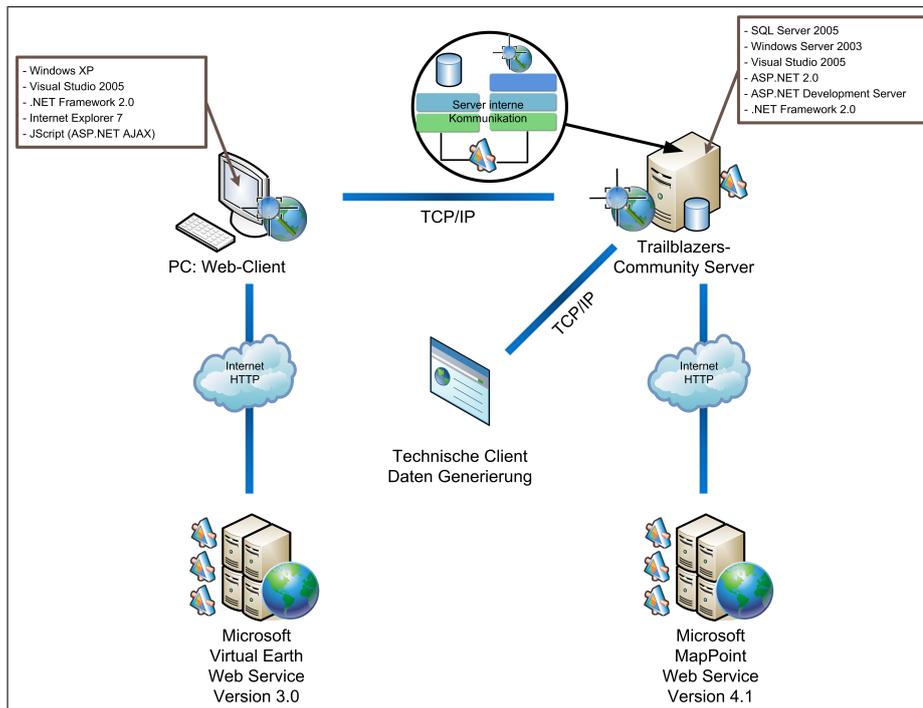


Abbildung 5.1: Skizze der technischen Rahmenbedingungen

Neben der Entwicklung der webbasierte Clientanwendung wurde in Rahmen dieser Arbeit verstärkt die Entwicklung der erforderlichen Datenbanken durchgeführt. Hierzu wurde parallel zur Visual Studio 2005 das *Microsoft SQL Server Management Studio* (Microsoft, 2006d) genutzt. Dieses bietet neben guter Performance auch hervorragende Unterstützung bei dem Entwurf, Analyse und Optimierung einer Datenbank.

Um auf die Betriebssystem-Funktionen zurück greifen zu können, würde bei der Entwicklung der Client- und Server-Programme das *Microsoft .NET Framework 2.0* (Microsoft, 2006j) verwendet. Dieses stellt eine Reihe von Klassenbibliotheken, Dienste sowie die für die Ausführung notwendige Laufzeitumgebung bereit.

5.1.2 Server

Für die Realisierung verschiedener Funktionen, die der Webclient nutzt, waren verschiedene Server seitige Technologien notwendig. Basierend auf dem .NET Framework 2.0 wurde das auch von Microsoft zur entwickelte *ASP.NET*¹ (Microsoft, 2006a) in der Version 2.0 verwendet. Mit dieser Technologie wurde die, für das hier betrachtete Systeme essentielle, Web-

¹Active Server Pages.NET

basierte Benutzer-Schnittstelle realisiert. Diese beiden sich ergänzenden Technologien bieten eine optimale Umgebung für die Prüfung der Tragfähigkeit des erarbeiteten Konzepts.

Die Zuständigkeiten des Trailblazers-Community Servers wurden in den vorangegangenen Abschnitten ausführlich diskutiert. Dieser besteht aus mehreren logischen Schichten, wobei jede separat einer dedizierten Aufgabe nachgeht. Bei der Wahl der Programmiersprache für die Abbildung der erforderlichen fachlichen Zusammenhänge hatte sich *C#* als beste Wahl erwiesen. Diese sehr junge, gleichzeitig sehr ausgereifte objektorientierte Programmiersprache ergänzt die hier eingesetzten Server-Komponenten.

In der hier betrachteten Problemstellung werden große Datenmengen vom Benutzer selber generiert. Diese müssen zentral aggregiert werden, damit sie einer späteren Verarbeitung unterzogen und anderen zur Verfügung gestellt werden können. Hierzu wurde das in Rahmen dieser Arbeit entwickelte Datenbankmodell später auf dem *Microsoft SQL Server 2005* (Microsoft, 2006g), (Dröge und Raath, 2003) umgesetzt. Durch den Einsatz dieser speziellen Datenbank gestaltete sich die Integration der bestehenden Server-Komponenten deutlich einfacher.

5.1.3 Client

Vermeehrt wurde auf das asynchrone Verhalten der heutigen webbasierten Anwendungen eingegangen. Bei der hier realisierten Web-basierten Anwendung ist es ebenfalls sehr wichtig, in der Durchführung der gewünschten Aufgaben dem Anwender eine Kontinuität, vergleichbar mit der von Desktopanwendungen, zur Verfügung zu stellen. Hierzu war der Einsatz eines zusätzlichen Frameworks erforderlich, welches die AJAX Laufzeitumgebung auf dem Client verfügbar macht. In (Woolstone, 2006) wird eine Betrachtung einiger verfügbarer Ajax Laufzeit-Umgebungen durchgeführt. Eine dort betrachtete Realisierung ist *Atlas*, aktuell *ASP.NET AJAX* (Microsoft, 2006c), die durch Abwägung der Vor- und Nachteile als Grundlage für die Realisierung genutzt wurde. Dieses Framework bietet eine Kompatibilität zu den gängigen Browsern, eine objektorientierte Schnittstelle, an Desktop angelehnte Schnittstellen-Komponenten und unterstützt ein deklaratives Modell bei der Verwendung dieser Komponenten.

Bei der Nutzung der AJAX Umgebung wurde verstärkt das *JavaScript* verwendet. Durch das JavaScript wird eine Client seitige Programmierplattform zur Verfügung gestellt. Neben der Verwendung von AJAX wurde aber JavaScript bei der Nutzung der Virtual Earth Web-Schnittstelle verwendet. Die mit JavaScript umgesetzte Logik wurde dann auf den *Internet Explorer 7* interpretiert.

Wie bereits erwähnt wird bei dem Webclient die Funktionalität des *Microsoft Virtual Earth* (Microsoft, 2006h) Web Services genutzt. Dieser wird benötigt, um die gesammelten GPS-

Daten aus der realen Umgebung durch in diesem Web Service gespeicherten Karten in Relation setzen zu können. Die zur Entwicklungszeit eingesetzte Web Services war in der Version 3.0.

5.1.4 Testumgebung

Die Prüfung der Funktionsfähigkeit der umgesetzten Software wurde anhand einiger Testfälle, basierend auf den bereits ausführlich verdeutlichten Anwendungsfällen, geprüft. Die Testumgebung zur Durchführung dieser Testläufe gestaltete sich vergleichbar mit der Entwicklungsumgebung. Dennoch mussten einige Vorbereitungen getroffen werden, um ein Systemverhalten zu simulieren, welches möglichst Praxisnah ist.

Die zentrale Aufgabe bei dem Webclient ist die Bearbeitung von Positionsdaten und damit in Relation stehenden Annotationen. Um die richtige Funktionsweise des Systems prüfen zu können, muss die Datenbasis mit echten Daten gefüllt werden. Hierzu wird eine gut bekannte Umgebung manuell vermessen mit einer ebenfalls realisierten Web-Anwendung und die dabei gewonnenen Positionsdaten in die Datenbasis eingegeben. Der eben genannte webbasierte Testclient hatte ebenfalls die Aufgabe, die Positionsdaten mit Annotationsdaten anzureichern und diese miteinander in Beziehung zu setzen. Dieses wurde erfolgreich mit dem TestClient erledigt.

Beim Test des Webclients war ein Web-Server erforderlich, der die entsprechenden Web Seiten bereitstellt. Hierbei wurde der in Microsoft Visual Studio 2005 integrierte Web-Server verwendet. Der lokal verfügbare *ASP.NET Development Server* erlaube die sofortige Prüfung der umgesetzten Web Seiten.

5.1.5 Testläufe

Im Folgenden soll dem Leser die Umsetzung anhand einiger Testläufe näher gebracht werden. Als Grundlage hierfür wird der in kooperierender Zusammenarbeit mit Sven Stegelmeier (Teammittglied der Trailblazers) entstandene WebClient genutzt. Mit diesem soll der Ablauf der zwei Kernfunktionalitäten verdeutlicht werden, die Erstellung der Routenempfehlung und die Verwaltung der bereits gesammelten Positionsdaten.

Verwaltungsmodus

Die Betriebsart „Verwaltungsmodus“ ist speziell dafür entworfen, die auf dem Trailblazers-Community Server gesammelten Trampelpfade nachzubearbeiten sowie die Definition von

geschützten Arealen durchführen zu können. Der erste Testfall verdeutlicht die Veränderung der Navigationskarte, indem die dort veränderten Trampelpfade gelöscht werden. Dieser Sachverhalt wird anhand des in Abbildung 5.2 skizzierten Aktivitätsdiagramms verdeutlicht.

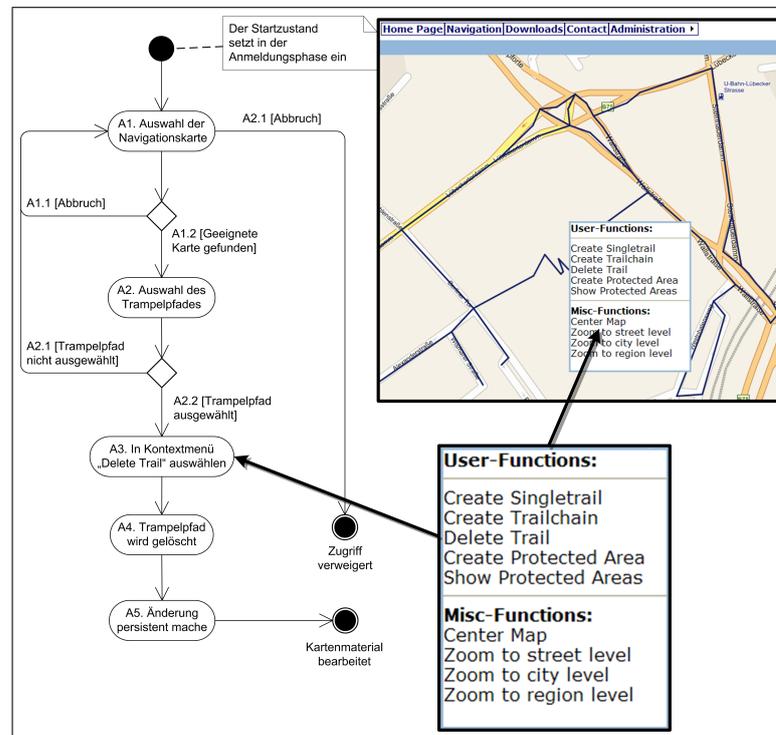


Abbildung 5.2: Aktivitätsdiagramm - Entfernen eines Trampelpfades

Bei diesem Vorgang wird nach der erfolgreichen Anmeldung wie folgt vorgegangen: In der *Aktivität A1* wird eine Navigationskarte ausgewählt, die nachbearbeitet werden soll. Ist der Benutzer berechtigt, diese Karte zu verwenden, geht dieser über zur der *Aktivität A2*. Ist dieser nicht berechtigt diese Karte zu bearbeiten, ist der Vorgang der Bearbeitung für ihn für diese Karte abgeschlossen. Als nächstes wählt der Benutzer den einzelnen Trampelpfad um auf diesem in der folgenden *Aktivität A3* die dafür vorgesehene Bearbeitungsfunktion auszuwählen. In diesem Fall wird die Methode Delete Trail ausgewählt, welche die Entfernung des ausgewählten Trampelpfades bewirkt (*Aktivität A4*). Anschließend wird diese für den Benutzer auf dem Trailblazers-Community Server persistent gemacht (*Aktivität A5*). Der Bearbeitungsprozess Trampelpfad ist damit auch abgeschlossen.

In dem Verwaltungsmodus werden neben der vorgestellten Funktion weitere bereitgestellt. Eine dieser Funktionen ist für die Spezifizierung von sogenannten geschützten Arealen (*Protected Areas*) erforderlich. Durch diese kann der Benutzer private Bereiche spezifizieren, in

dem die Positionsdaten nicht aufgezeichnet werden sollen. Damit verbundenen Aktivitäten werden in der Abbildung 5.3 verdeutlicht.

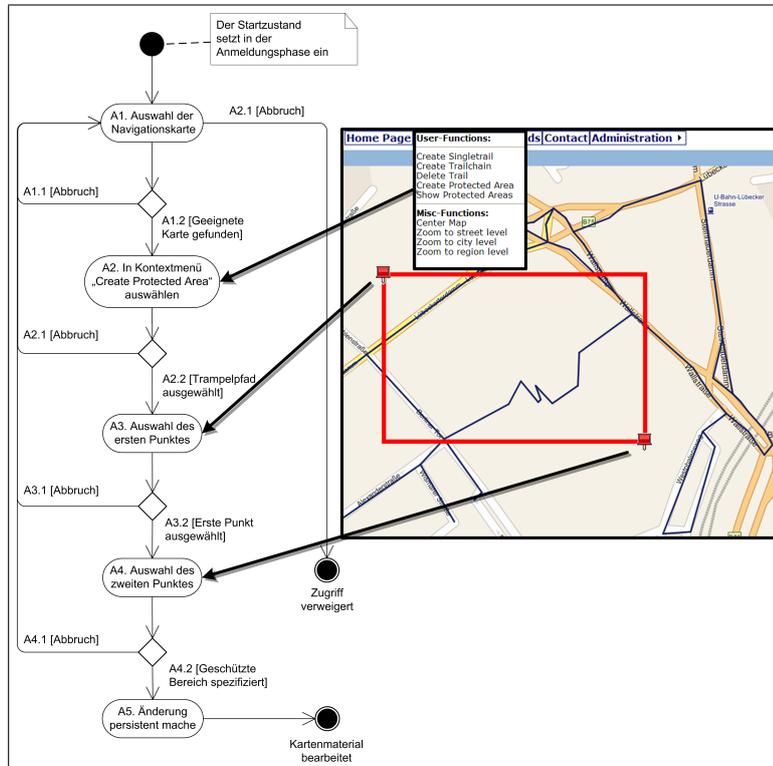


Abbildung 5.3: Aktivitätsdiagramm - Spezifizierung eines geschützten Bereiches

Der Vorgang, bei dem der geschützte Bereich spezifiziert wird, gestaltet sich wie folgt. Nachdem die Anmeldung und Auswahl der Navigationskarte erfolgreich durchlaufen sind, wird über das Kontextmenü die Funktion *Create Protected Area* ausgewählt (*Aktivität A2*). Durch diese Auswahl wird ein weiterer Auswahlmodus gestartet, in dem man den gewünschten Bereich spezifizieren kann. Als Nächstes müssen zwei Punkte ausgewählt werden, die als Grundlage für die Berechnung eines Rechtecks genutzt werden, das den gewünschten Bereich umschließt (*Aktivitäten A3, A4*). Sind diese festgelegt, wird das resultierende Rechteck auf der Navigationskarte dargestellt und in dem Trailblazers-Community Server gespeichert.

Navigationsmodus

In dem Navigationsmodus werden alle Tätigkeiten durchgeführt, die mit der Erstellung einer Wegbeschreibung und Sichtung aller notwendigen Karten-Informationen zu tun haben.

Exemplarisch sollen an dieser Stelle der Ablauf der Erzeugung der Route verdeutlicht werden, mit dem Auslesen der in dem *Point of Interest* enthaltenen Informationen verdeutlicht werden. Die Abbildung 5.4 Skizziert grob bei diesem Vorgang durchlaufene Abläufe.

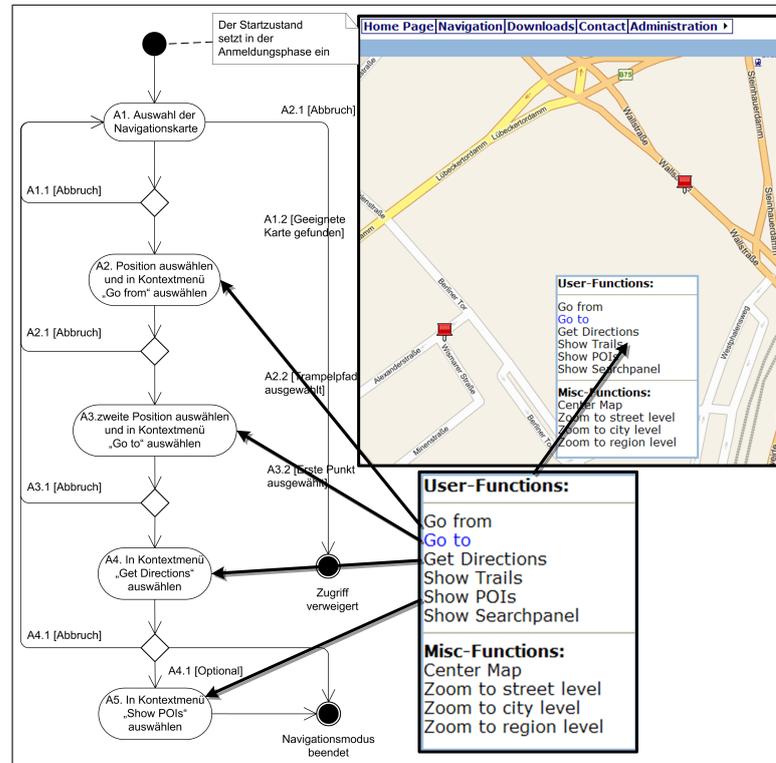


Abbildung 5.4: Aktivitätsdiagramm - Berechnung der Wegbeschreibung

Nachdem die Anmeldung und Auswahl erfolgreich durchlaufen ist, wird mit der Spezifizierung des Startpunktes für die vom Trailblazers-System zu berechneten Strecke begonnen (*Aktivität A2*). Hierzu muss vorher die entsprechende Position auf der Navigationskarte ausgewählt und über das Kontextmenü die Funktion „Go from“ ausgewählt werden. Nach dem Durchführen dieser Schritte wird der ausgewählte Punkt auf der Karte dargestellt (in der Abbildung als rote Pinnadel dargestellt). Als nächstes ist die Zielposition auszuwählen. Hierzu wählt der Benutzer die gewünschte Position und wählt über das Kontextmenü die Funktion „Go to“ aus (*Aktivität A3*). Sind diese beiden Punkte von dem Benutzer festgelegt worden, erscheint erst jetzt im Kontextmenü die Funktion „Get Direction“. Diese muss ausgewählt werden, damit der Trailblazers-Community Server die Wegbeschreibung für die spezifizierten Vorgaben berechnet (*Aktivität A4*). Die berechnete Route wird in der Abbildung 5.5 (a) verdeutlicht. Optional kann der Benutzer sich in der näheren Umgebung lokalisierte Point of Interests anzeigen lassen (*Aktivität A5*). Hierzu wählt dieser ebenfalls über das Kontextmenü die Funktion „Show POIs“ aus. Hierdurch erscheinen auf der Navigationskarte rote Pinnadel, die die Position eines Point of Interest aufzeigen. Führt der Benutzer den Mauszeiger über

eine Pinnadel, erscheint die genauere Information zur den ausgewählten Point of Interest. Die Abbildung 5.5 (b) verdeutlicht diesen Vorgang exemplarisch. In der Abbildung 5.5 (c) wird eine solche Beschreibung eines Point of Interests genauer dargestellt.

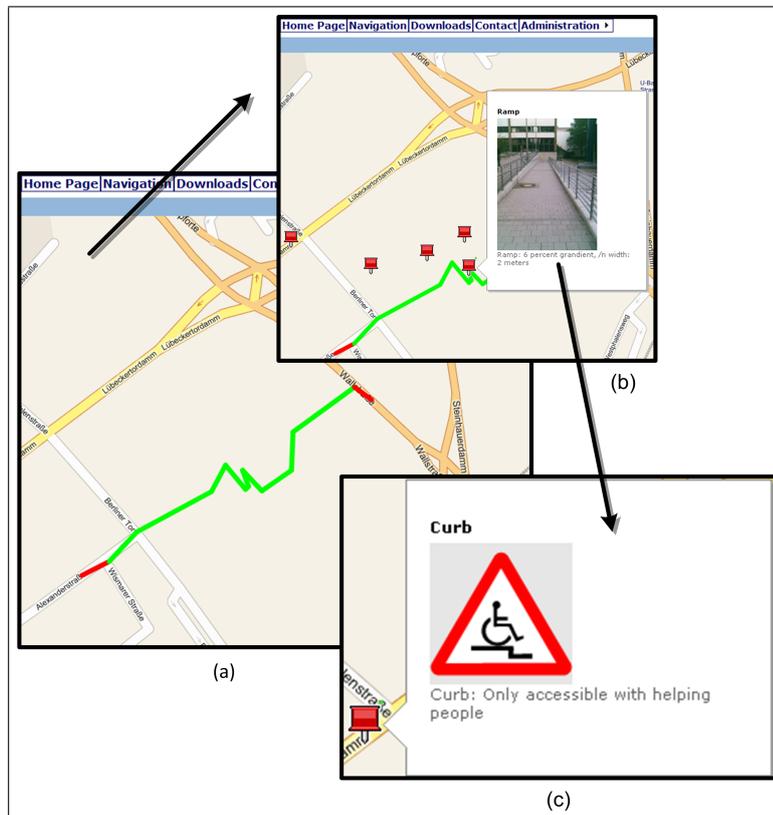


Abbildung 5.5: (a) Der resultierende Pfad. Das besondere hierbei ist, dass dieser durch eine offen, straßenfreie Fläche führt, die von einem Rollstuhlfahrer befahren werden kann. (b) Darstellung der Point of Interests auf der Navigationskarte. Als Beispiel wird hierbei eine Rollstuhl-gerechte Rampe mit ihren Eigenschaften ausgewählt. (c) Verdeutlichung die genauere Information eines Point of Interest.

5.1.6 Durchführung der Testläufe

Die eben beschriebenen Testläufe beschreiben nur exemplarisch die Funktionalität des Webclients. Ungeachtet dessen zeigen diese die prinzipielle Funktionsweise des hier umgesetzten Systems. Durch die prototypische Umsetzung konnte die Machbarkeit praktisch bewiesen werden.

Bei der Durchführung wurden einige Schwächen deutlich, die die bisherige Umsetzung in sich birgt. In dem ersten Testfall wurde sichtbar, dass eine Auswahl von mehreren Trampelpfaden erforderlich ist, bevor diese dann gelöscht werden. Sehr oft entsteht nicht nur ein einzelner Trampelpfad, sondern eine Kette von Trampelpfaden, die mit Hilfe des Webclients berichtigt werden müssen. Eine Berichtigung von vielen einzelnen Trampelpfaden ist einem Benutzer nicht zumutbar.

Für viele Benutzer ist es wichtig, den geschützten Bereich zu spezifizieren. Hierzu wurde mit dem zweiten Testfall die Funktionalität geprüft. Diese funktioniert einwandfrei, muss aber dennoch noch weiter verfeinert werden. In den Prototypen wurde eine einfache rechteckige Fläche zur Spezifizierung des entsprechenden Bereiches bereitgestellt. Dieses wird in vielen Fällen höchst wahrscheinlich ausreichend sein, sollte jedoch durch einen Polygon ersetzt werden. Damit könnte eine genauere Fläche spezifiziert werden und die in vielen Fällen durch Rechteck nicht erfassbaren Bereiche der Pfaderkennung verfügbar gemacht werden.

In dritten Testfall wird eine Wegbeschreibung erarbeitet. Bei den recht überschaubaren Positionsdaten, bzw. daraus resultierenden Trampelpfad-Graphen zeigte die Implementierung des Dijkstra-Routing Algorithmus gute Ergebnisse in Hinsicht auf die Geschwindigkeit der Berechnung. Wenn der Trailblazers-Community Server jedoch mit feineren Daten gespeist wird und diese eine deutlich größere Flächen umspannen werden, wird der Dijkstra höchst wahrscheinlich nicht ausreichend sein um eine schnelle Berechnung durchführen zu können. Aus diesem Grund sind zum einem auf solche Datenmengen optimierte Algorithmen² sowie Strategien zur Optimierung der Graphen³ erforderlich.

Die ebenfalls in dritten Testfall verdeutlichte Darstellung der Umgebungsinformationen anhand der Annotierung durch einen Point of Interest birgt auch mindestens eine Schwäche. Das Virtual Earth Web Service erlaubt einen bestimmten Vergrößerungsgrad bei der Darstellung der dort enthaltenen Karten. Dieser erlaubt eine ausreichende Untersuchung der Karteninformation in den Arealen, welche mit einer kleinen Dichte an Annotierung versehen ist (wie in Testfall dargestellt). Wird ein Areal ausgewählt, welcher sich durch eine hoch Informationsdichte auszeichnet und mit vielen Point of Interests annotiert ist, wird die Darstellung dieser Informationen eher verwirrend wirken. Dieser der Lesbarkeit und damit der Entnahme von Information hinderliche Zustand muss mit geeigneten Methoden beseitigt werden. Zur Beseitigung der eben genannten störenden Darstellungskomplexität könnten Strategien wie Filtermethoden und Mehrschichten-Darstellung von Annotationen hierbei Abhilfe schaffen.

Bis zu einen voll einsatzfähigen, Benutzer-unterstützenden System zur Erstellung von annotierten Navigationskarten ist die Durchführung einiger wichtiger Arbeiten erforderlich. Den-

²Auf Routing-Aufgaben optimierte Algorithmen sind aktuell Gegenstand der Forschung.

³Mehrere Schichten zur Darstellung eines Graphen, durch die abhängig von der Dichte der Graphen und der Zielsetzung die geeignete Darstellung für die Berechnung ausgewählt wird.

noch wurde durch den vorgestellten Prototyp die Realisierbarkeit des hier entworfenen Systems und darin enthaltenen Ansätze bewiesen.

5.2 Gewonnen Erkenntnisse

5.2.1 Server

Bei der Erstellung von Wegempfehlungen stellten sich erhebliche Performance-Probleme heraus, welche hauptsächlich bei der Darstellung der berechneten Trampelpfade zum Vorschein kamen. Durch die nähere Untersuchung zeigte sich, dass die Performance-Probleme ihren Ursprung in dem Virtual Earth Web Service haben. Die Darstellung der berechneten Trampelpfade auf dem Karten wurde mit Hilfe der von diesem Web Service zur Verfügung gestellten Funktionen durchgeführt. Der Verwendungszweck der verwendeten Funktion richtet sich hauptsächlich auf die Darstellung einer Route und nicht wie hier eines ganzen Graphen.

Bei der Berechnung der Wegempfehlung werden viele Zugriffe auf die in der Datenbank enthaltenen Positionsdaten und damit verbundenen Annotierung getätigt. In den Testläufen zeigte sich die Performance der Routen-Berechnung als akzeptabel. Nichtsdestotrotz sind erhebliche Performance-Probleme bei einer zunehmenden Benutzeranzahl zu erwarten. Um dieses Problem abzuschwächen, ist die Verlagerung entsprechende Logik in die Datenbank notwendig⁴.

Die prototypische Prüfung der Lauffähigkeit mit den hier genutzten Werkzeugen stellte sich als äußerst einfach dar. Durch die Bereitstellung sehr ausgereifter, auf einander abgestimmter Komponenten können schnell neue Dienste in den Server integriert werden. Diese Eigenschaft ist sehr wichtig, damit die hier vorgeschlagene Plattform neue Funktionen bereitstellen kann. Durch die Nutzung der Web Service Technik ist es sehr einfach, vergleichbare Dienste, wie z. B. *Google Maps* durch Trailblazers-Community Server anzubieten. Diese Offenheit des entworfenen Servers sichert deren Fortbestehen.

5.2.2 Client

Bei der Nutzung des entworfenen Webclients zeigen sich Schwierigkeiten beziehungsweise Inkompatibilitäten zwischen verschiedenen Webbrowsern. Wie bereits verdeutlicht, nutzt der Web-Client zur Darstellung der Trampelpfade und der berechneten Wegempfehlung den

⁴Durch Abbildung der entsprechenden Funktionalität mit *Store Procedures*.

Virtual Earth Web Service. Bei der Nutzung von *Firefox* zeigen sich Probleme bei der Darstellung der von Virtual Earth zur Verfügung gestellten Informationen. Diese variieren von Fall zu Fall und sollten näher untersucht werden. Möglicherweise gab es in der Version 3 des Virtual Earth Web Services Probleme, die durch die neue Version behoben wurde. Dieses sollte geprüft werden.

Der umgesetzte Webclient bestätigt bereits die Machbarkeit der hier betrachteten Problemstellung. Die erforderlichen Technologien sind vorhanden. Dennoch muss der Web-Client weiter in Zusammenarbeit mit der Zielgruppe verbessert werden. Dieser muss Menschen ebenfalls verfügbar gemacht werden, die unter spastischen Lähmungen leiden. Das hierfür bei der Realisierung notwendige Know-how stand leider nicht zur Verfügung.

5.3 Offenen Punkte

In Verlauf der Arbeit traten weitere wichtige Merkmale hervor, die von der hier entwickelten Anwendung berücksichtigt werden sollten, aus Zeitgründen aber nicht bearbeitet wurden. Diese sollen im Einzelnen besprochen werden.

5.3.1 Benutzer-Schnittstelle

In Rahmen der ASK-IT Konferenz ([ASK-IT, 2006](#)) wurden verschieden Ansätze vorgestellt, bei denen das Augenmerk auf die Gestaltung für Rollstuhlfahrer geeigneter Benutzer-Schnittstellen gelegt wurde. Die dabei vorgestellte Arbeit von Birgit Ringbauer und Verena Schlegel ([Ringbauer und Schlegel, 2006](#)) umfasst Ansätze zur Gestaltung entsprechender Seiten. Diese wurden hier in dem Abschnitt Analyse betrachtet, konnten leider im zeitlichen Rahmen nicht umfassend umgesetzt werden. Auch die daneben betrachteten Konzepte zur Gestaltung von barrierefreien Benutzerschnittstellen müssen umgesetzt werden. Hierbei sollen die von dem ASP.NET 2.0 bereitgestellten Richtlinien und unter Beachtung dieser realisierten Komponente zur Umsetzung barrierefreier Web-Seiten zum Einsatz kommen ([Microsoft, 2006b](#)). Die Einzelheiten hierüber sind aus ([Microsoft, 2007a](#)) zu entnehmen.

Die für die Prüfung der Lauffähigkeit genutzte ASP.NET 2.0 Technologie sollte in der Weiterentwicklung des Webclients möglicherweise durch eine neuere Technologie ersetzt werden. Vor kurzem wurde von Microsoft das *Silverlight* ([Microsoft, 2007b](#)) vorgestellt. Diese ist eine neue Technologie mit der man interaktive Webanwendungen realisieren kann. Damit kann einer Web-Anwendung Eigenschaften verliehen werden, welche vergleichbar mit denen der *Windows Presentation Foundation* sind. Hierzu stellt Microsoft ein Browser Plug-In bereit,

welcher von allen gängigen Browsern genutzt werden kann. Durch Silverlight soll eine Laufzeitumgebung bereitgestellt werden, auf der AJAX, Python, Ruby, Visual Basic und C# für die Realisierung der Web-Anwendungen genutzt werden können.

Bei den technischen Problemen wurde auf die Schwierigkeit bei der Darstellung der in den Trailblazers-Community Server enthaltenen Trampelpfade hingewiesen. Diese höchstwahrscheinlich durch die Komplexität des dargestellten Graphen verursachten Probleme könnten durch eine Verringerung der Komplexität beseitigt werden. Hierzu sollte eine geeignete Strategie erarbeitet werden, durch die nur sehr einfache Graphen berechnet werden und an den Virtual Earth Server gesendet werden. Dem sollte auch die AJAX Technologie Abhilfe schaffen, indem Anfragen an den Server parallelisiert werden und somit eine Kontinuität bei der Bearbeitung der Trampelpfade geschaffen wird.

5.3.2 Regelung der kollektiven Arbeit

Die hier vorgeschlagene Architektur zur Regelung der auf der Trailblazers-Plattform stattfindenden Partizipation bei der Erstellung der Navigationskarten stellt eine prototypische Lösung dar. Diese sollte weiterentwickelt werden, sodass Ansätze aus bereits durchgeführten Arbeiten in diese einfließen können. Eine genauere Betrachtung der Collaborative Work unterstützende Systeme wird in der Diplomarbeit von Roman Bartnik gegeben ([Bartnik, 2006](#)). Auch die neu erarbeiteten Vorgehensweisen in Rahmen des Wikipedia Projektes sollen nicht unberücksichtigt bleiben. Hierbei wurden Verfahren zur Sicherung der kollektiv erarbeiteten Inhalte erarbeitet. Zur Bewertung der Brauchbarkeit dieser Strategien für die Trailblazers-Plattform bedarf es einer tieferen Analyse.

5.3.3 Sicherheits-Modell

Die Trailblazers-Plattform baut sehr stark auf die Offenheit der involvierten Benutzer. Jedoch soll diese nicht die Benutzer von der Teilnahme ausschließen, die auf ihre Privatsphäre nicht verzichten möchten. Hierzu muss ein ausgeklügeltes Rechtemodell realisiert werden. Der hier vorgeschlagene Ansatz deckt diese Anforderungen nur ansatzweise. Bei der Weiterentwicklung des Sicherheitsmodells kann die Arbeit von Horst Mund berücksichtigt werden. In dieser Arbeit wurde ein solches Sicherheitsmodell entworfen und prototypisch realisiert ([Mund, 2006](#)).

Nicht unberücksichtigt sollte das von Microsoft zur Verfügung gestellte *Active Directory* beziehungsweise *Active Directory Application Mode* (ADAM) ([Microsoft, 2006I](#)) bleiben. Neben den bereits genannten Möglichkeiten zur Beschreibung von Benutzerrechten ist ADAM eine mögliche Alternative. ADAM wurde speziell für Webanwendungen konzipiert und erlaubt die

Abbildung einer Benutzerdatenbank mit einer dazu gehörenden Rechte- und Rollenverwaltung.

Das von dem Fraunhofer-Institut FIT und OrbiTeam Software GmbH entworfene BSCW Share-Workspace-System ([GmbH, 2007](#)) bietet bereits umfassende Funktionalität, die von der Trailblazers-Plattform bereitgestellt werden muss. Damit ist in erster Linie das spezifizieren von selbstorganisierenden Arbeitsgruppen und deren Verantwortlichkeiten gemeint. Gleichzeitig ist das gemeinsame Erarbeiten von Dokumenten sowie deren Versionierung gegeben. Möglicherweise stellt diese Software Möglichkeiten bereit, durch die diese in bestehende Applikationen wie z. B. Trailblazers-Plattform integriert werden kann. Durch den Einsatz einer bereits geprüften Komponente könnte die Entwicklung der Trailblazers-Plattform an anderen Stellen vorangetrieben werden.

5.3.4 Suche

Google ist das beste Beispiel dafür, dass Suche einer der wichtigsten Funktionen ist, wenn man mit der immer größer werdenden Mengen an Informationen zurecht zu kommen möchte. Auch die Trailblazers-Plattform sollte eine solche Funktion anbieten, um die dort gespeicherten Ressourcen wieder auffindbar zu machen. Darüber hinaus ist bei einer steigenden Zahl der Mitglieder die Auffindung dieser ebenfalls nicht trivial. Die Bereitstellung einer Suchfunktion kann als mächtiges Werkzeug die Handhabbarkeit dieser Plattform erheblich steigern.

5.3.5 Mobilität

Diese Arbeit konzentrierte sich vollständig auf die Gesichtspunkte der Annotierung der Navigationskarten. Ihr Einsatz auf mobilen Geräten wurde nicht untersucht und ist Gegenstand der Arbeit von (Thomé, 2007) ([Thome](#)).

5.3.6 Datensammlung

Wie bereits verdeutlicht spielt bei dem Web 2.0-Ansatz und damit auch bei dieser Ausprägung der Sozialen Software, die datenzentrische Sicht eine fundamentale Rolle. Damit muss das System eine Möglichkeit zur Motivierung (Motivationsinstrument) der Benutzer, dieses verstärkt zu nutzen (also ein *Incentivierungssystem*) beinhalten. Wie bei *LycosIQ* ([Lycos, 2007](#)) sollten auch bei dem Trailblazers-System aus Sicht des Systems gute Aktionen eines Benutzers durch sogenannte Statuspunkte belohnt werden. Diese Statuspunkte könnten bei den Benutzern ein Wettstreiten entfachen, gutes Navigationsmaterial zu erstellen und

der Community bereitzustellen, da diese zum Ansehen des Benutzers beiträgt. Das Ansehen könnte man z. B. durch Verleihung von entsprechenden Titeln bewirken. Damit kann ein Benutzer von einem gelegentlichen Spaziergänger bis zu Magellan oder Columbus aufsteigen.

Neben der Vergütung von guten Aktionen mit immer steigendem Ansehen kann eine weitere Art von der Belohnung eingeführt werden. So könnten z. B. Bonuspunkte vergeben werden, wenn ein Benutzer dem System eine vordefinierte Menge an Informationen über eine geographische Umgebung in das System einspeist. Auf der anderen Seite können auch Routenvorschläge, beziehungsweise die Bereitstellung von neuen spezialisierten Karten, mit einem Bonus belohnt werden. Mit diesen Punkten kann der Benutzer dann weitere Aktionen in dem System ausführen, die für ihn bis dahin vorenthalten waren. Diese wichtigen Aspekte konnten in Rahmen dieser Arbeit nicht realisiert werden.

5.3.7 Notifikation

Um auf die Änderung der Umgebung sofort reagieren zu können, muss das Wahrnehmen der neu eingefügten Informationen durch geeignete Techniken verbessert werden. In der Analyse erwies sich die Feed Technologie als geeignet um eine Verteilung der neu erstellten Informationen durchzuführen. Die Anbindung einer konkreten Ausprägung in die Web-basierte Anwendung konnte in Rahmen dieser Arbeit nicht durchgeführt werden.

5.3.8 Performance

Wie zuvor beschrieben wird die Verbindung zwischen der Datenbank und dem Server intern über einen Web-Service realisiert. Bei einer heterogenen Serverlandschaft ist dieses Vorgehen immer zu empfehlen. Gestaltet sich jedoch die Serverlandschaft eher homogen, ist das *.NET Remoting* zu empfehlen. Durch den Einsatz von *.NET Remoting* wird die intern aufgebaute Kommunikation deutlich an Performance dazugewonnen. Daneben wird durch die Nutzung von *.NET Remoting* ein traditionelles Objekt-Verteilungsmodell eingeführt. Damit können einzelne Objekte dediziert verwaltet werden, also einzeln aktiviert, deren Lebenszyklus gesteuert sowie den Nachrichtenaustausch basierend auf der Kanal-Kommunikation durchgeführt und verwaltet werden ([Rossberg und Redler, 2006](#)). Durch das Objekt-Verwaltungsmodell wird gleichzeitig eine bessere Basis für die Realisierung einer skalierbaren Anwendung gegeben ([Hashimi und Steffan, 2005](#)). Demnach empfiehlt es sich, in der Fortführung der Arbeit die interne Kommunikation basierend auf *.NET Remoting* zu realisieren.

5.3.9 Virtual Earth

Aktuell ist das Virtual Earth in der Version 5.0 verfügbar und enthält damit neue Funktionalität, die das Zusammenarbeiten mit diesem Web-Service verbessert. In dieser neuen Version wird verstärkt auf die Anpassungsfähigkeit der Umgebung und auf die Ansprüche des Benutzers, Verbesserung der Darstellung der Controls, sowie eine verbesserte Kontrolle bei der Erstellung und Verarbeitung der Annotationen eingegangen. Darüberhinaus wurde in der vorherigen Versionen die 3D-Darstellung der Karten und das hinzufügen von benutzerdefinierten Karten ermöglicht. Die sich eröffnenden Möglichkeiten sollten von dem Trailblazers-System zur qualitativen Steigerung der angebotenen Funktionen genutzt werden.

5.4 Fazit

Die Realisierungsphase hat gezeigt, dass das System in seinen Grundfunktionen realisierbar ist. Sicherlich existieren an einigen Stellen noch unbeantworteten Fragen, die aber nicht auf technische Sachverhalte abzielen. Die technische Basis zur Realisierung des hier betrachteten Systems ist vorhanden, wie die prototypische Umsetzung dieses bestätigt. Es ist nur eine Frage der Zeit bis alle einzelnen System-Komponenten einen hohen Qualitätsgrad erreichen.

Der Schwerpunkt dieser Arbeit lag klar in der Analyse der Problemstellung sowie hierbei zum Einsatz kommenden Ansätzen. Damit wurden aus dieser Betrachtung resultieren Vorgaben an eine verteilte Anwendung, welche das Partizipieren bei der Erstellung von Navigationskarten und deren Annotierung ermöglichen sollten, hervorgebracht. Daneben lag das Hauptaugenmerk auf dem Entwurf einer sauberen, nach Prinzipien des objektorientiertem Designs ausgerichteten Systemarchitektur. Dabei sollten die bereits bestehenden Ansätze durch weitere ergänzt werden. Die hierbei verbundenen Aufgaben wurden weitgehend abgeschlossen.

Die hierbei entworfene System-Architektur wurde auf eine gründliche Untersuchung der fachlichen Eigenschaften der Problemstellung, welche vorher anhand des definierten Szenarios identifiziert wurden, durchgeführt. Die daraus resultierenden, auf die Umsetzung abzielenden Vorgaben wurden in mehreren Schritten erarbeitet. Zur Beginn wurde die Betrachtung der Sachverhalte aus Benutzersicht durchgeführt. Hierzu wurden alle aus der fachlichen Sicht möglichen Anwendungsfälle identifiziert und auf ihre Merkmale näher untersucht. In späteren Verlauf wurde eine Konkretisierung des Systemverhaltens vollzogen und damit den technischen Verhalten des Systems genährt. Diese Untersuchung erlaubte eine eindeutige Klassifizierung der Verarbeitungsschritte, welche in dem Prozess der Erstellung und Annotierung von Navigationskarten enthalten sind. Die hierbei gewonnen Klarheit half die Komplexität des Systems zu entschärfen und lieferte damit notwendige Systemanforderungen.

Diese wurden anschließend bei dem erfolgreich durchgeführten Systementwurf berücksichtigt. Dabei wurde eine Mehrschicht-Architektur erarbeitet und damit ein modular aufgebautes, erweiterbares System entworfen, welches später leichter gewartet werden kann.

Parallel zu der technisch versierten Analyse wurde eine ebenso wichtige Betrachtung der Chancen sowie der Risiken, die das hier realisierte System mit sich bringt, durchgeführt. Dabei identifizierten Aspekte wurden auf deren positive sowie negative Effekte untersucht. Hierbei wurde ein Katalog von Möglichkeiten aber auch Hindernissen identifiziert.

Die hier entworfene Anwendung stützt sich sehr stark auf den Ansätzen des CSCW und unterstützt die Collaboration Awareness. Auf der anderen Seite bekommen durch den asynchronen Charakter dieser Anwendung einige Problemstellungen des CSCW hierbei eine untergeordnete Rolle. Die positiven Aspekte, also das Group Awareness, die wiederum zur Förderung der Partizipation essentiell sind, bleiben dennoch erhalten. Die bei der Untersuchung betrachteten Community-gestützten Ansätze lieferten viele wichtige Richtlinien, um ein solches System erfolgreich umsetzen zu können. Diese Vorgaben wurden in Rahmen hier durchgeführten Analyse erarbeitet.

Neben anderen Benutzergruppen wurden die Rollstuhlfahrer vorrangig als Zielgruppe des hier entworfenen Systems betrachtet. Durch die häufig damit verbunden motorischen Einschränkungen können diese nur bedingt kompliziert gestalteten Anwendungen bedienen. Hierzu sollten existierende Ansätze zur Gestaltung von Webanwendungen gesucht werden, die eine einfache Bedienung des Systems den Betroffenen ermöglichen und gleichzeitig eine benutzergerechte Darstellung erlauben. Die an solche Schnittstellen gestellten Anforderungen konnten erfolgreich identifiziert werden. Die Umsetzung dieser im Prototyp ist noch ausgeblieben.

Das Kartographieren einer Umgebung durch einen Rollstuhlfahrer involviert Informationen, welche einem Nichtrollstuhlfahrer höchstwahrscheinlich entgangen wären. Um die Funktionalität des Systems für solche Informationen zu sensibilisieren, muss die Wahrnehmung einer Umgebung durch einen Rollstuhlfahrer untersucht werden. Dabei sollen für diese Anwendergruppe relevanten Informationen identifiziert werden. Die damit verbundene Betrachtung wurde durchgeführt und bracht die Merkmale einer solchen Wahrnehmung hervor. Die bestehende Umsetzung ermöglicht zum größten Teil eine Integrierung der hierbei identifizierten Merkmale. Dennoch sollte eine Fortführung der Arbeiten in Hinsicht auf das Involvieren dieser Merkmale in die Routenberechnung erfolgen, da dieses in den Rahmen dieser Arbeit nur ansatzweise durchgeführt wurde.

Die Hauptaufgabe des Trailblazers-Systems ist die Bereitstellung der Navigation für Menschen, die in ihrer Mobilität eingeschränkt sind. Um eine Navigation durchführen zu können, sind Informationen über geeignete Wege erforderlich. Hierzu soll das System dabei helfen, die notwendigen Daten zu sammeln. Dafür musste ein geeignetes Datenbankmodell entworfen werden. Dieses wurde erfolgreich durchgeführt. Zuvor wurden verschiedene bestehende

Ansätze untersucht und bei dem Entwurf des Datenbankschemas berücksichtigt. Ungeachtet dessen muss die entworfene Datenbank auf die in der Praxis lauenden Problemstellungen erweitert werden.

6 Fazit

Mobile Navigation bekommt immer mehr Zuspruch, was durch eine aktuelle Statistik untermauert wird. Eine durch BITCOM veranlasste Marktuntersuchung ergab, dass in April über 3 Millionen Benutzer in Deutschland bereits über ein Navigationsgerät verfügen (BITCOM, 2007). Dies ergibt gleichzeitig eine Steigerung von 200 Prozent gegenüber den vergangenen Monaten. Nach Gardner Research (Stuart, 2006) tragen in Deutschland mehr als 82 Millionen Menschen ein Mobiles Gerät mit sich. Diese Geräte haben inzwischen ausreichende Rechenleistung um darauf neuartige Anwendungen ausführen zu lassen. Zudem bieten die Mobilfunk-Anbieter zunehmend neben der Telefon-Flatrate auch eine Internet-Flatrate an. Hierdurch entsteht eine Plattform für neuartige, Web-basierte Anwendungen wie z. B. Navigations-Software.

Immer mehr Anbieter versuchen ihre Navigationssoftware an den Endbenutzer zu verkaufen. Die hierfür notwendige Hardware wird immer kleiner und kostengünstiger. Alle diese Navigations-Systeme haben nach Auffassung des Autors einen wesentlichen Nachteil: sie stellen Navigations-Informationen bereit die von einer zentralen Stelle eingekauft wurde. Damit sind diese zum Teil veraltet und vor allem sehr teuer. Ein Ausweg aus dieser Lage kann die in dieser Arbeit vorgestellte Trailblazers-Plattform darstellen. Diese bietet die nötige Werkzeuge mit denen man zum einen Kartenmaterial sammeln, diesen dann bearbeiten und somit die Anforderungen des Benutzers anzupassen kann. Mit diesem Ansatz erstelltes Kartenmaterial dient dann als Grundlage zur Ermittlung von bestmöglichen Routen, welche dem Benutzer entsprechend seiner Vorstellung an den Ort seiner Wahl führen.

Systeme, die sich auf das Konzept der kollektiven Partizipation Arbeit stützen, also auch die Trailblazers-Plattform, müssen eine gewisse Menge an Benutzern als Mittglieder dazugewinnen. Um diese kritische Menge zu erreichen müssen geeignete Werkzeuge, die Anforderungen wie Intuitive Benutzung, in deren Realisierung berücksichtigt werden. Desweiteren sollten diese ein dynamisches Verhalten aufweisen, damit jeder Benutzer diese Schnittstelle seinen Anforderungen anpassen kann. Gerade diese Merkmale sind für Benutzer wie Rollstuhlfahrer, die nicht selten in der Motorik eingeschränkt sind, eine der wesentlichsten Anforderungen.

Die kollektive Erarbeitung von annotierten Navigationskarten hilft dabei nicht nur, Kosten zu sparen, sondern auch die eigene Wahrnehmung in die Darstellung der umliegenden Umgebung einzufügen. Noch wichtiger ist die spätere Bereitstellung der so erarbeiteten Navigati-

onskarten. Zudem ist die Vereinheitlichung des Kartenmaterials in ein einziges und dadurch einheitliches Format eine deutliche Verbesserung in deren späteren Nutzung. Durch diese Eigenschaft steigt die Lernkurve bei der Benutzung des Systems erheblich. Um das alles zu bewerkstelligen, versucht diese Arbeit einen Teil der Trailblazers-Plattform-Funktionalität in Form von webbasierten Werkzeugen zur Verfügung zu stellen.

Begonnen wird mit der Untersuchung der heute gängigen Konzepte auf die sich zunehmend immer mehr Anwendungen stützen. Dabei werden neben den technischen Aspekten auch die sozialen diskutiert, die einen wesentlichen Einfluss auf die Realisierung haben. Nachdem ein Konsens aus den bestehenden Ansätzen gebildet wurde, wurden Anhand eines Beispielszenarios, welches auf die Rollstuhlfahrer abzielt, vorgestellt, das veranschaulichen soll wie die hier entwickelte Plattform zu einer Verbesserung der Selbstständigkeit in der Fortbewegung für diese Benutzergruppen führen würde. Damit wurde das Potential dieser Plattform verdeutlicht. Um dieses gestellte Ziel erreichen zu können, wurden dann die an solche Plattform gestellten Anforderungen identifiziert.

Das hier offene System orientiert sich an den identifizierten technischen sowie fachlichen Anforderungen. Deren Ursprünge liegen in Konzepten wie *Soziale Software* und *Web 2.0*. Das führt dazu, dass erarbeitete Navigationskarten für jeden, überall und zum jeden Zeitpunkt verfügbar sein müssen. Des Weiteren muss das System im Stande sein, die hierbei gemeinsam arbeitenden Benutzer bei der Gemeinschaftsarbeit zu unterstützen. Damit ist ein Sicherheitsmodell gemeint, welches den Zugriff auf die hier zum Einsatz kommenden Ressourcen regelt.

Nachdem die grundlegenden technischen und fachlichen Aspekte geklärt wurden, wurde der Entwurf der Architektur durchgeführt. Dabei wurde ein Paradigma eines offenes, lose gekoppeltes System gewählt, welches die neuesten Webanwendungen kennzeichnet. Durch diese Maßnahme wurde eine spätere Anbindung von weiteren externen Diensten ermöglicht, welche die hier zugrunde liegende Plattform in ihre Funktionalität und damit der Erfüllung weiterer Anforderungen erweitert. Der Entwurf sieht eine Vier-Schichten Architektur vor, welche physikalisch in der Anwendungsebene zwischen Client und dem Server getrennt wird.

Das hier vorgestellte Konzept konnte prototypisch auf seine Tauglichkeit geprüft werden. Viele der hierbei vorgeschlagenen Ideen konnten aber im zeitlichen Rahmen nicht realisiert werden.

6.1 Erreichte Ziele

Das hier vorgestellte Projekt wurde bereits auf verschiedenen Veranstaltungen vorgestellt und damit der Öffentlichkeit zugänglich gemacht. Eine dieser ersten Veranstaltung war der von Microsoft organisierte Wettbewerb *Imagine Cup* (Microsoft, 2006e). Hierbei wurde das

Projekt mit dem ersten Platz in Deutschland ausgezeichnet und dürfte damit bei dem weltweiten Finale teilnehmen und vorgestellt werden. Dabei wurde über diese Trailblazers-Plattform mehrfach in der Presse geschrieben, sodass der erste Grundstein für die Öffentlichkeitsarbeit gelegt wurde.

Bei der Entwicklung der Trailblazers-Plattform sollten die Prinzipien der Benutzer-orientierte Gestaltung im Vordergrund stehen. Hierzu war eine Kontaktaufnahme mit der Zielgruppe sehr wichtig. Dieses wurde erfolgreich durchgeführt, sodass Benutzer aus der Zielgruppe die entwickelte Software testen und damit ihre konstruktive Kritik in die Verbesserung dieser Plattform einfließen lassen konnten.

Zur Sichtung der Möglichkeiten des Marktes wurden potentiellen Kunden ausfindig gemacht. Dabei konnte als Beispiel einer der führenden Rollstuhlhersteller von den Prinzipien der Trailblazers-Plattform überzeugt werden. Damit haben sich weitere Türen für spätere Kooperation geöffnet.

Um das Trailblazers-Projekt aus Sicht des Marktes noch besser zu positionieren wurde das von Microsoft finanzierte *Innovation Accelerator*-Projekt hierzu in Anspruch genommen. Dabei wurden verschiedene Markt technische Untersuchungen durchgeführt, sowie die zukünftige Marktstrategie erarbeitet. Zudem wurden neue, für die Markteinführung wichtige Kontakte aufgebaut.

Eine weitere Bestätigung bekam das Trailblazers-Projekt in Rahmen des Hamburger INNOTECH-Wettbewerbes ([Hep, 2007](#)). Bei diesem Wettbewerb, der gleichzeitig ein Projekt zur Förderung innovativer Geschäftsideen ist, wurde das Trailblazers-Projekt mit dem ersten Platz geehrt. Durch diese Auszeichnung wurden weitere Weichen zur erfolgreichen Realisierung der Trailblazers-Systems gestellt.

Aber auch in der Welt der Wissenschaft hatte das Trailblazers-Projekt seinen hohen Innovationsgrad beweisen können. Bei der Konferenz ASK-IT, welche sich mit dem Einsatz der IT zur Erhöhung der Lebensqualität der in der Mobilität eingeschränkten Menschen auseinandersetzt, konnte das Trailblazers-Projekt vor einem breiteren wissenschaftlichen Publikum vorgestellt werden. Neben der positiven Resonanz wurde die, in diesem Rahmen verfasste wissenschaftliche Publikation ([Stegelmeier u. a., 2006](#)) zudem als beste wissenschaftliche studentische Publikation gewürdigt.

6.2 Ausblick

Für die vollständige Realisierung der Trailblazers-Plattform sind weiter Schritte zum Ausbau von möglichen Kooperationen erforderlich. Damit sollen nicht nur potentielle Test-Benutzer

dazugewonnen werden, sondern auch Partner mit notwendigem Know-how und nach Möglichkeit Ressourcen. Diesbezüglich wird eine Zusammenarbeit mit der Stadt Kiel beziehungsweise mit der dortigen Stadtplanung geplant. Damit soll das Trailblazers-System in der gesamten Stadt für die Rollstuhlfahrer eingesetzt und damit ein erster größerer Feldversuch gestartet werden.

Natürlich ist die Startphase bei der Produkteinführung einer der kritischsten Phasen. Um das hierbei entstehende Risiko zu reduzieren können verschiedene Maßnahmen ergriffen werden. Eine eher technisch versierte Möglichkeit wäre der Kauf von bereits bestehenden Karteninformationen. Diese könnten dazu benutzt werden um für bestimmte Regionen die zur Beginn nicht vorhandenen Navigationskarten durch kommerzielle Daten zu füllen. Damit würden die ersten Benutzer nicht auf einer leeren Datenmenge arbeiten und somit sofort in diesem Bereich alle Funktionen der Navigation in vollem Ausmaß nutzen können.

Vermehr wurde in dieser Arbeit darauf hingewiesen, dass der Erfolg Trailblazers-Plattform und die mit ihr bereitgestellte Funktionalität sehr stark von der Größe der Community abhängt. Damit die kritische Masse der aktiven Benutzer dieser Plattform erreicht werden kann, muss eine sehr kostspielige Werbekampagne betrieben werden. Um die damit verbundenen Kosten zu minimieren wurden weitere Partner dazugewonnen. Dabei sind in Verbindung mit verschiedenen Veranstaltungen Präsentationen des Trailblazers-Systems geplant. Zudem wurde von einer weiteren Seite eine Bereitschaft bekundet, eine bereits bestehende Community von über 3000 Benutzer speziell auf die Evaluierung der Trailblazers-Plattform anzusetzen.

Die Weiterentwicklung muss natürlich auch bei Trailblazers finanziell weiter unterstützt werden. Der Europäische Kommission sind die Schwierigkeiten der, in der Mobilität eingeschränkten Menschen bekannt. Um dieser Menschengruppe eine Verbesserung der Mobilität in der EU zu ermöglichen fördert die Europäische Kommission auf diese Problemstellung abzielende Projekte. Auch hierbei soll das Trailblazers-Projekt Unterstützung erfahren.

Sichtung weiterer EU Projekte in Richtung auf Unterstützung der bestehenden und in Kürze entstehenden Navigationssysteme. Die Europäische Kommission plant die Unterstützung und Ausbau des Galileo Satelliten-Navigationssystems mit mehr als 3 Milliarden Euro zu fördern. In diesem Rahmen sollen auch diese Gelder zum Teil als Fördermittel für Entwicklungsprojekte bereitgestellt werden. Ein Teil davon soll auch zur Unterstützung des Trailblazers-System weitere Unterstützung zugute kommen.

6.3 Zusammenfassung

Die Arbeit verdeutlicht die Möglichkeiten, die eine Plattform zur Annotierung von Navigationskarten und deren spätere Nutzung in dem mobilen Zustand, bieten können. Die hier

entworfene Web-basierende Werkzeuge der Trailblazers-Plattform sind eine neue Art, die uns umgebende Umgebung zu kartographieren und die eigene Wahrnehmung der Gemeinschaft zur Verfügung zu stellen. Die Werkzeuge entsprechen der Forderung des *Web 2.0*, demnach unterliegen diese der fortwährenden Weiterentwicklung. Eine benutzerzentrierte Navigation stellt eine neue Möglichkeit dar, preisgünstig unbekannte Umgebung zu kartographieren um das erstellte Kartenmaterial für eine Barrieren-freie Navigation zu verwenden. Dass ein Bedarf für eine solche Plattform besteht wurde bereits durch die involvierten Test-Benutzer bestätigt.

Die in dieser Arbeit durchgeführte Untersuchung zeigte, wie eine Community-gestützte Erstellung von Navigationskarten möglich ist. Damit wurde ein Schritt in eine Welt gemacht, in der jeder dazu beitragen kann die Welt von neuem zu erforschen. Ungeachtet des enormen innovativen Fortschrittes, die diese Plattform mit sich bringt, wurde ebenfalls durch diese Arbeit ersichtlich, dass ein solches System in seinem Aufbau nicht trivial ist. Es verdeutlicht das bei solchen Systemen viele Wissenschaftsdisziplinen miteinander kooperieren beziehungsweise ihre Erkenntnisse austauschen, damit eine solche Plattform realisiert werden kann. Die prototypische Realisierung zeigt in den Anfängen den Grad der Machbarkeit dieser Anwendung. Ungeachtet den bisher erreichten Fortschritten in der Realisierung, steht die Entwicklung in einem Anfangsstadium was auch im Umfeld des Web 2.0 gerechtfertigt ist.

Durch die hohe Verbreitung des PC und der immer weiter steigenden Bandbreite wird die erfolgreiche Durchsetzung dieser neuen Art von Anwendungen, bei den große Gemeinschaften Partizipierend an gemeinsamen Zielsetzungen zusammenarbeiten, begünstigt. Die heute verfügbare Technologie kann diese Zusammenarbeit ortsunabhängig zu den unterschiedlichen Zeitpunkten ablaufen lassen. Trailblazers ist eine auf diesen Technologien basierte Plattform und steht damit technologisch betrachtet vor überwindbaren Herausforderungen. Ob sich diese auf dem Markt durchsetzen kann, beziehungsweise welche Marktschancen dieses System besitzt, ist schwierig abzuschätzen. Untersucht man den die aktuelle Marktlage, wird ersichtlich, dass dort bereits eine breite Palette an Navigationssystemen angeboten wird. Besteht also die Notwendigkeit eines weiteren Navigationssystems? Wenn ja, was ist der Grund hierfür?

„*Map on Demand*“ ist ein neues Schlagwort, welches seit kurzem in Japan die Runde macht (Köhler, 2007). Hinter diesem Begriff verbirgt sich ein System, welches dem Autofahrer Navigationskarten anbietet, die Änderungen im Straßenbild sofort nach Erscheinen auf der Karte anzeigt. Damit bekommt der Autofahrer eine Möglichkeit, Hindernisse rechtzeitig umfahren zu können. Das hier von Toyota entwickelte GPS-basierte Navigations-System heißt *G-Book mX*. Die Informationen¹, die als Grundlage zur der Annotierung der Navigations-Karten genutzt werden, sollen von den Fahrzeugen selbst automatisiert gesammelt werden. Eine Parallele zu dem Trailblazers-System ist also eindeutig zu erkennen. Neben der beschriebenen

¹Die anonymisiert erhobenen Daten handelt es sich um Position und Geschwindigkeit.

Funktionalität des G-Book mX bietet das Trailblazers-System aber noch weitere, auf die Verbesserung der Navigation abzielenden Innovationen, die in Rahmen dieser Arbeit vorgestellt wurden. Darüber hinaus steht das Trailblazers-System allem Autofahrern zur Verfügung, bei G-Book mX nur dem Toyota Besitzern.

Betrachtet man den Freizeitsektor, findet man auch dort verschiedene Ausprägungen vergleichbaren Lösungsansätze. Für das von Apple vor kurzem auf dem Markt gebrachte *iPod Nano* wird zusätzlich ein *iPod Sport Kit* ([Apple, 2007](#)) angeboten. Dieses wird für den Aufbau einer Verbindung zwischen dem iPod und dem in Nike Sportschuh integrierten Sensor eingesetzt. Ein in dem Schuh eingebauter Beschleunigungssensor erfasst die Bewegungen des Läufers. Dabei gesammelte Daten werden dann an das iPod gesendet und auf diesem aufgezeichnet. Diese Bewegungsdaten können dann über das Internet mit dem Nike Server ([Nike, 2007](#)) synchronisiert und dann in verschiedenen Formen ausgewertet werden. Dies natürlich Community-gestützt mit einem Wettkampf Charakter. Wichtiger ist jedoch die Möglichkeit der Anzeige der Rennstrecke basierend auf Google Maps. Leider werden diese Rennstrecken derzeit noch manuell aufgezeichnet. Das Trailblazers-System unterstützt ebenfalls die Auswertung von Bewegungssensoren. Diese werden derzeit zur Kontexterkenkung der Bewegung verwendet, können aber auch auf weitere Funktionen ausgeweitet werden. Wichtiger ist jedoch die Aufzeichnung der Rennstrecken, welche neben dem von Nike vorgestellten manuellen Ansatz, auch eine automatisierte Erzeugung solche Wege vom Trailblazers-System unterstützt wird. Nike und Apple zeigen eine erfolgreiche Einsatzmöglichkeit welches auch von Trailblazers bereitgestellt wird.

Die Unterstützung der Autofahrer und Jogger umfasst bereits einen sehr großen Markt, so dass eine Sicherung des Bestehens der Trailblazers-Plattform gegeben ist. Das Trailblazers-System spricht aber durch die Bereitstellung eines breiteren Funktionsumfanges weitere Zielgruppen an. Welche das sein können zeigt ein vergleichbares Projekt *GPSies* von Klaus Bechtold ([Bechtold, 2007](#)). Dieses Portal ist eine Anlaufstelle für verschiedene Freizeitsportler und mehr, auf dem man seine aufgenommen GPS-basierten Karten hochladen und anderen zur Verfügung stellen kann. Die hochgeladene GPS-Daten können auf dem Google Maps Kartenmaterial projiziert und besichtigt werden. Anschließend kann die Karte in verschiedenen Formaten herunter auf einen GPS Gerät geladen werden um bei dem Begehen der ausgewählten Strecke nicht von dem Pfad abzukommen. Die dort gespeichert Karten umfassen verschiedene Benutzergruppen, die diese benutzen können. Alle diese Gruppen werden auch von dem Trailblazers-System unterstützt. Vergleichbar betrachtet, stellt das Trailblazers-System zusätzlich noch den hier betrachteten Karteneditor bereit, durch den eine neue Dimension der Kartenannotierung der aufgenommen Strecken eingeführt wird. Dieses Merkmal bietet dem Benutzer genügend Gestaltungsfreiraum um seine Sichtweise auf die Welt in diesem System abzubilden. Damit hebt sich das Trailblazers-System auch in diesem Fall von der hier genannten Plattform ab.

Durch die genannten Beispiele sollte verdeutlicht werden, dass der Zeitpunkt der Bereitstellung solcher Plattformen bereits überschritten ist. Diese Plattformen erfreuen sich eines immer mehr steigenden Zuspruchs von Seiten der Benutzer. Die Trailblazers-Plattform ist mit ihrem Funktionsumfang in diesem Marktsegment genau richtig aufgestellt um sich neben der immer größer werden Konkurrenz sich zu etablieren.

Literaturverzeichnis

- [Aarts 2004] AARTS, Emile: Ambient Intelligence: A Multimedia Perspective. In: *IEEE MultiMedia* 11 (2004), Nr. 1, S. 12–19
- [Aarts u. a. 2002] AARTS, Emile ; HARWIG, Rick ; SCHUURMANS, Martin: Ambient intelligence. (2002), S. 235–250. ISBN 0-07-138224-0
- [Adler 2005] ADLER, Steven: WebOS: say goodbye to desktop applications. In: *netWorker* 9 (2005), Nr. 4, S. 18–26. – ISSN 1091-3556
- [Alby 2006] ALBY, Tom: *Web 2.0. Konzepte, Anwendungen, Technologien*. München, Wien : Hanser Fachbuchverlag, 2006
- [Apple 2007] APPLE: *Ipod+Nike*. 2007. – URL <http://www.apple.com/de/ipod/nike/>. – Stand: 04.05.2007
- [ASK-IT 2006] ASK-IT: *ASK-IT - Ambient Intelligence System of Agents from Knowledge based and Integrated Services for Mobility Impaired Users*. 2006. – URL <http://www.ask-it.org/>. – Stand: 07.05.2007
- [Atlas 2007] ATLAS, Tele: *Tele Atlas*. 2007. – URL <http://www.teleatlas.com/index.htm>. – Stand: 11.05.2007
- [Balram und Dragicevic 2006] BALRAM, Shivanand ; DRAGICEVIC, Suzana: *Collaborative Geographic Information Systems*. United States of America : IDEA GROUP PUBLISHING, 2006
- [Balzert 2005] BALZERT, Heide: *Lehrbuch der Objektmodellierung. Analyse und Entwurf mit der UML 2*. München : Spektrum Akademischer Verlag: 2.A., 2005
- [Bartnik 2006] BARTNIK, Roman: *Weiterentwicklung einer Technologiebasis für interaktive Gruppenräume*. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/diplom/bartnik.pdf>. – Stand: 07.05.2007
- [Bechtold 2007] BECHTOLD, Klaus: *GPSies*. 2007. – URL <http://gpsies.de/>. – Stand: 04.05.2007
- [Berners-Lee 1989] BERNERS-LEE, Tim: *Information Management: A Proposal*. 1989. – URL <http://www.w3.org/History/1989/proposal.html>. – Stand: 19.10.2006

- [Berners-Lee 1999] BERNERS-LEE, Tim: *Transcript of Tim Berners-Lee's talk to the LCS 35th Anniversary celebrations*. 1999. – URL <http://www.w3.org/1999/04/13-tbl.html>. – Stand: 03.12.2006
- [BITCOM 2007] BITCOM: *Telematik - 3 Millionen Deutsche besitzen bereits ein mobiles Navigationsgerät*. 2007. – URL http://www.bitkom.org/45523_45415.aspx. – Stand: 30.04.2007
- [Bosworth 2005] BOSWORTH, Alex: *Ajax Mistakes*. 2005. – URL http://sourcelabs.com/ajb/archives/2005/05/ajax_mistakes.html. – Stand: 19.03.2007
- [Brickley und Miller 2005] BRICKLEY, Dan ; MILLER, Libby: *FOAF Vocabulary Specification*. 2005. – URL <http://xmlns.com/foaf/0.1/>. – Stand: 03.10.2006
- [Brown und Haas 2003] BROWN, Allen ; HAAS, Hugo: *Web Services Glossary*. 2003. – URL <http://www.w3.org/TR/2003/WD-ws-gloss-20030514/>. – Stand: 09.10.2006
- [Bush 1945] BUSH, Vannevar: *As We May Think*. In: *The Atlantic Monthly* 176 (1945), Nr. 1, S. 101–108
- [Cho 2006] CHO, I S.: *Weblogagenten mit local trust metric für den automatischen Nachrichtenaustausch*. 2006. – URL http://www.tzi.de/fileadmin/user_upload/wearlab/downloads/wearlab-Publication/weblogmitLTM_I_Seong_Cho.pdf. – Stand: 18.12.2006
- [Christopher 2004] CHRISTOPHER, Allan: *Tracing the Evolution of Social Software*. 2004. – URL http://www.lifewithalacrity.com/2004/10/tracing_the_evo.html. – Stand: 08.11.2006
- [Coates 2002] COATES, Tom: *On the augmentation of human social networking abilities...* 2002. – URL http://www.plasticbag.org/archives/2002/12/on_the_augmentation_of_human_social_networking_abilities/. – Stand: 29.11.2006
- [Coates 2005] COATES, Tom: *An addendum to a definition of Social Software*. 2005. – URL http://www.plasticbag.org/archives/2005/01/an_addendum_to_a_definition_of_social_software/. – Stand: 12.01.2007
- [Commision 2001] COMMISSION, European: *IST Advisory Group (ISTAG)*. 2001. – URL <http://cordis.europa.eu/ist/istag.htm>. – Stand: 02.02.2007
- [Drexler 1997] DREXLER, Erik: *Hypertext Publishing and the Evolution of Knowledge*. 1997. – URL <http://www.islandone.org/Foresight/WebEnhance/HPEK1.html>. – Stand: 25.10.2006
- [Dröge und Raath 2003] DRÖGE, Ruprecht ; RAATH, Markus: *Microsoft SWL Server 2005 -Konfigurierung, Administration, Programmierung*. Microsoft Press Deutschland, 2003

- [Engelbart 1962] ENGELBART, Douglas: *AUGMENTING HUMAN INTELLECT: A Conceptual Framework*. 1962. – URL <http://www.bootstrap.org/augdocs/friedewald030402/augmentinghumanintellect/ahi62index.html>. – Stand: 06.11.2006
- [ESRI 2007] ESRI: *Geodatabase*. 2007. – URL <http://www.esri.com/software/arcgis/geodatabase/>. – Stand: 28.04.2007
- [Gamma u. a. 2004] GAMMA, E. ; HELM, R. ; JOHNSON, R.: *Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software*. München : Addison-Wesley, 2004
- [Garrett 2005] GARRETT, Jesse J.: *Ajax: A New Approach to Web Applications*. 2005. – URL <http://www.adaptivepath.com/publications/essays/archives/000385.php>. – Stand: 18.02.2007
- [GmbH 2007] GMBH, OrbiTeam S.: *Basic Support for Cooperative Work*. 2007. – URL <http://www.bscw.de/>. – Stand: 07.05.2007
- [Google 2006] GOOGLE: *Google Maps*. 2006. – URL <http://maps.google.com/>. – Stand: 03.02.2007
- [Hashimi und Steffan 2005] HASHIMI, Sayed ; STEFFAN, Scott: *Pro Service-Oriented Smart Clients with .NET 2.0*. Apress, 2005
- [Hep 2007] HEP: *Hep-Hamburger Existenzgründungs Programm*. 2007. – URL http://www.hep-online.de/cms/?page=de.1120741648_193&lang=de. – Stand: 04.05.2007
- [Himpsl 2006] HIMPSL, Klaus: *An addendum to a definition of Social Software*. 2006. – URL http://himpsl.htldornbirn.vol.at/these/2006_09_23_Masterthese_Himpsl.pdf. – Stand: 12.12.2006
- [Huch 2006] HUCH, Michael: *Getting networked Getting networked for the new Ambient Assisted Living (AAL169) programme*. 2006. – URL <http://www.aal169.org>. – Stand: 2.02.2007
- [Huffaker und Calvert 2005] HUFFAKER, David ; CALVERT, Sandra: *Gender, Identity, and Language Use in Teenage Blogs*. 2005. – URL <http://jcmc.indiana.edu/vol10/issue2/huffaker.html>. – Stand: 02.12.2006
- [Interakt 2005a] INTERAKT: *AJAX: Asynchronously Moving Forward*. 2005. – URL http://www.interaktonline.com/Support/Articles/Details/AJAX%3A+Asynchronously+Moving+Forward.html?id_art=36. – Stand: 26.02.2007
- [Interakt 2005b] INTERAKT: *AJAX: Vorteile und Probleme*. 2005. – URL <http://www.getthe-code.de/code/javascript/ajax/ajax-benefitsproblems.htm>. – Stand: 19.09.2006

- [Jini.org 2006] JINI.ORG: *Jini*. 2006. – URL www.jini.org/. – Stand: 25.09.2006
- [Kahlbrandt 2001] KAHLBRANDT, Bernd: *Software Engineering mit Unified Modeling Language*. Berlin : Springer, 2001
- [Kalz 2006] KALZ, Marco: *Social Software für das technologiegestützte Lernen Blogs und Wikis*. 2006. – URL http://dSPACE.ou.nl/bitstream/1820/789/1/kalz_tukl_171006.pdf. – Stand: 17.01.2007
- [Köhler 2007] KÖHLER, Angela: Exklusiver Verkehrsdienst. In: *Wirtschaftswoche* (2007), April, Nr. 18
- [Kolbitsch und Maurer 2006] KOLBITSCH, Josef ; MAURER, Hermann A.: The Transformation of the Web: How Emerging Communities Shape the Information we Consume. In: *J. UCS* 12 (2006), Nr. 2, S. 187–213
- [Kropla 2005] KROPLA, Bill: *Beginning MapServer: Open Source GIS Development*. United States of America : Apress, 2005
- [Lagh-e.V. 2004] LAGH-E.V.: *Hamburger Stadtführer für Rollstuhlfahrer - Sonderausgabe: Tourismus, Freizeit & Kultur*. Hamburg : Elbe-Werkstätten GmbH, 2004. – Hamburger Landesarbeitsgemeinschaft für behinderte Menschen e.V.
- [Last.fm 2007] LAST.FM: *Last.fm*. 2007. – URL <http://www.last.fm>. – Stand: 10.01.2007
- [Lycos 2007] LYCOS: *Lycos IQ*. 2007. – URL <http://iq.lycos.de/>. – Stand: 07.05.2007
- [Maier 2006] MAIER, Gerhild: *AJAX von A bis X*. 2006. – URL http://krottmaier.cgv.tugraz.at/docs/seminar/sem2005_ajax.pdf. – Stand: 27.10.2006
- [Mauerer 2006] MAUERER, Jürgen: *AJAX und ASP.NET - Technologien für moderne Web-Anwendungen*. 2006. – URL <http://www.microsoft.com/germany/msdn/library/web/AJAXUndASPNET.aspx?mfr=true>. – Stand: 03.02.2007
- [McLaughlin 2005] MCLAUGHLIN, Brett: *Ajax meistern*. 2005. – URL <http://www.oreilly.de/artikel/ajax1/index.html>. – Stand: 19.01.2007
- [McLellan 2005] MCLELLAN, Drew: *Very Dynamic Web Interfaces*. 2005. – URL <http://www.xml.com/pub/a/2005/02/09/xml-http-request.html>. – Stand: 27.12.2006
- [Microsoft 2002] MICROSOFT: *Application Architecture for .NET: Designing Applications and Services*. 2002. – URL <http://msdn2.microsoft.com/en-us/library/ms978348.aspx>. – Stand: 24.01.2007
- [Microsoft 2006a] MICROSOFT: *ASP.NET*. 2006. – URL <http://www.asp.net/>. – Stand: 03.05.2007

- [Microsoft 2006b] MICROSOFT: *ASP.NET - ASP.NET Controls and Accessibility*. 2006. – URL [http://msdn2.microsoft.com/en-us/library/ms227996\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms227996(VS.80).aspx). – Stand: 03.05.2007
- [Microsoft 2006c] MICROSOFT: *ASP.NET AJAX*. 2006. – URL <http://ajax.asp.net/>. – Stand: 03.05.2007
- [Microsoft 2006d] MICROSOFT: *Einführung in SQL Server Management Studio*. 2006. – URL <http://msdn2.microsoft.com/de-de/library/ms174173.aspx>. – Stand: 03.05.2007
- [Microsoft 2006e] MICROSOFT: *Imaginecup*. 2006. – URL <http://imaginecup.com/default.aspx>. – Stand: 04.05.2007
- [Microsoft 2006f] MICROSOFT: *Microsoft MapPoint*. 2006. – URL <http://www.microsoft.com/mappoint/default.msp>. – Stand: 03.02.2007
- [Microsoft 2006g] MICROSOFT: *Microsoft SQL Server*. 2006. – URL <http://www.microsoft.com/sql/default.msp>. – Stand: 03.05.2007
- [Microsoft 2006h] MICROSOFT: *Microsoft Virtual Earth*. 2006. – URL <http://www.microsoft.com/virtualearth/default.msp>. – Stand: 05.05.2007
- [Microsoft 2006i] MICROSOFT: *MSR MapCruncher for Virtual Earth*. 2006. – URL <http://research.microsoft.com/mapcruncher/>. – Stand: 17.11.2006
- [Microsoft 2006j] MICROSOFT: *.NET*. 2006. – URL <http://www.microsoft.com/net/>. – Stand: 03.05.2007
- [Microsoft 2006k] MICROSOFT: *Visual Studio 2005*. 2006. – URL <http://msdn2.microsoft.com/de-de/vstudio/default.aspx>. – Stand: 03.05.2007
- [Microsoft 2006l] MICROSOFT: *Windows Server 2003 Active Directory Application Mode*. 2006. – URL <http://www.microsoft.com/windowsserver2003/adam/default.msp>. – Stand: 07.05.2007
- [Microsoft 2007a] MICROSOFT: *ASP.NET - Accessibility Support in ASP.NET*. 2007. – URL [http://msdn2.microsoft.com/en-us/library/ms228004\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/ms228004(VS.80).aspx). – Stand: 07.05.2007
- [Microsoft 2007b] MICROSOFT: *Microsoft Silverlight*. 2007. – URL <http://msdn2.microsoft.com/en-us/asp.net/bb187358.aspx>. – Stand: 07.05.2007
- [Microsystems 2003] MICROSYSTEMS, Sun: *JavaSpaces Service Specification*. 2003. – URL http://www.sun.com/software/jini/specs/js2_0.pdf. – Stand: 16.11.2006
- [Millard und Ross 2006] MILLARD, David E. ; ROSS, Martin: Web 2.0: hypertext by any other name? In: *Hypertext*, 2006, S. 27–30

- [Minar u. a. 2001] MINAR, Nelson ; HEDLUND, Marc ; SHIRKY, Clay ; O'REILLY, Tim ; BRICKLIN, Dan ; ANDERSON, David ; MILLER, Jeremie ; LANGLEY, Adam ; KAN, Gene ; BROWN, Alan ; WALDMAN, Marc ; CRANOR, Lorrie F. ; RUBIN, Aviel ; DINGLEDINE, Roger ; FREEDMAN, Michael ; MOLNAR, David ; DORNFEST, Rael ; BRICKLEY, Dan ; HONG, Theodore ; LETHIN, Richard ; UDELL, Jon ; ASTHAGIRI, Nimisha ; TUVELL, Walter ; WILEY, Brandon: *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. (2001)
- [Mitchell und McKenna 2005] MITCHELL, Tyler ; MCKENNA, Jeff: *MapServer - Vector Data Access*. 2005. – URL http://mapserver.gis.umn.edu/docs/reference/vector_data. – Stand: 02.05.2007
- [Möller 2006] MÖLLER, Erik: *Die heimliche Medienrevolution - wie Weblogs, Wikis und freie Software die Welt verändern (2. Auflage)*. Hannover, Deutschland : Heise Zeitschriften Verlag, 2006
- [MomentumSI 2006] MOMENTUMSI: *Service Oriented Architecture*. 2006. – URL http://www.serviceoriented.org/service_oriented_architecture.html. – Stand: 19.12.2006
- [Morville 2005] MORVILLE, Peter: *Ambient Findability: What We Find Changes Who We Become*. United States of America : O'Reilly Media, Inc., 2005
- [München 1997] MÜNCHEN, Technische U.: *CSCW - Computer-Supported Cooperative Work*. 1997. – URL <http://www.telekooperation.de/cscw/cscw.html>. – Stand: 04.11.2006
- [Mund 2006] MUND, Horst: *Berechtigungsstrukturen in kollaborativen Umgebungen*. 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/diplom/mund.pdf>. – Stand: 07.05.2007
- [Navteq 2007] NAVTEQ: *Navteq*. 2007. – URL <http://www.navteq.com/>. – Stand: 10.05.2007
- [Nehmer u. a. 2006] NEHMER, Jörgen ; BECKER, Martin ; KARSHMER, Arthur I. ; LAMM, Rosemarie: Living assistance systems: an ambient intelligence approach. In: *ICSE, 2006*, S. 43–50
- [Nelson 2006] NELSON, Ted: *Xanalogical Structure, Needed Now More than Ever: Parallel Documents, Deep Links to Content, Deep Versioning, and Deep Re-Use*. 2006. – URL http://www.cs.brown.edu/memex/ACM_HypertextTestbed/papers/60.html. – Stand: 22.11.2006
- [Nike 2007] NIKE: *Nikeplus*. 2007. – URL <http://www.nike.com/nikeplus/>. – Stand: 04.05.2007
- [Oestereich 2006] OESTEREICH, Bernd: *Analyse und Design mit UML 2.1 : objektorientierte Softwareentwicklung*. München : Oldenbourg, 2006

- [Ontoworld 2007] ONTOWORLD: *Semantic MediaWiki*. 2007. – URL http://ontoworld.org/wiki/Semantic_MediaWiki. – Stand: 27.01.2007
- [O'Reilly 2006] O'REILLY, Conferences: *Web 2.0 conference*. 2006. – URL <http://www.web2con.com/>. – Stand: 17.10.2007
- [O'Reilly 2005] O'REILLY, Tim: *What Is Web 2.0, Design Patterns and Business Models for the Next Generation of Software*. 2005. – URL <http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web.20.html>. – Stand: 09.10.2006
- [Pandora 2006] PANDORA: *Pandora*. 2006. – URL <http://www.pandora.com>. – Stand: 17.04.2007
- [Ringbauer und Schlegel 2006] RINGBAUER, Brigitte ; SCHLEGEL, Verena: *Inter-modal travel planning for wheelchair users*. (2006)
- [Rinner 1999] RINNER, Claus: *Argumentation Maps: GIS-based discussion support for online planning*. Sankt Augustin : GMD Research, 1999
- [Rossberg und Redler 2006] ROSSBERG, Joachim ; REDLER, Rickard: *Pro Scalable .NET Application Designs*. Apress, 2006
- [Roush 2005] ROUSH, Wade: *Social Machines - Computing means connecting*. 2005. – URL <http://www.technologyreview.com/Infotech/14664/>. – Stand: 27.11.2006
- [Schmidt 2006] SCHMIDT, Jan: *Weblogs: eine kommunikationssoziologische Studie*. Konstanz, Deutschland : UVK, Universitätsverlag, 2006
- [Shadbolt 2003] SHADBOLT, Nigel: *Ambient Intelligence*. In: *IEEE Intelligent Systems* 18 (2003), Nr. 4, S. 2–3
- [Sheth u. a. 2006] SHETH, Amit ; VERMA, Kunal ; GOMADAM, Karthik: *SEMANTICS TO ENERGIZE THE FULL SERVICES SPECTRUM*. In: *COMMUNICATIONS OF THE ACM* 49 (2006), Nr. 7, S. 54–61. – ISSN 1091-3556
- [Shirky 2003] SHIRKY, Clay: *Social Software and the Politics of Groups*. 2003. – URL http://shirky.com/writings/group_politics.html. – Stand: 23.1.2006
- [Simoes u. a. 2006] SIMOES, Anabela ; GOMES, Ana ; EVANGELOS, Bekiaris: *Use Cases for ASK-IT*. 2006. – URL <http://www.ask-it.org>. – Stand: 06.12.2006
- [Steg u. a. 2005] STEG, Horst ; STRESE, Hartmut ; HULL, Jerome ; SCHMIDT, Sophie: *Europe Is Facing a Demographic Challenge Ambient Assisted Living Offers Solutions*. 2005. – URL http://www.iserd.org.il/ist/documents/AAL169_000.pdf. – Stand: 18.03.2007

- [Steg u.a. 2006] STEG, Horst ; STRESE, Hartmut ; HULL, Jerome ; SCHMIDT, Sophie: *1. nationaler Workshop zu Ambient Assisted Living (AAL)*. 2006. – URL http://www.temas.ch/wwwtemas/temas_homepage.nsf/vwRes/AAL_Introduction_d/. – Stand: 19.03.2007
- [Stegelmeier u. a. 2006] STEGELMEIER, Sven ; STEIN, Martin ; THOME, Mark ; WENDT, Piotr: *Trailblazers - A Community Driven Navigation System For Mobility Impaired People*. (2006)
- [Stein 2006] STEIN, Vera: *Trotzdem. Behindert ist man nicht, behindert wird man. Aufrüttelnder Erfahrungsbericht einer beharrlichen Kämpferin*. Schardt, 2006
- [Stuart 2006] STUART, Donald: *Mobile and Wireless Services and Services Providers in Germany, 2006 Update / Gartner Research*. 2006. – Forschungsbericht
- [Surrowiecki und Beckmann 2005] SURROWIECKI, James ; BECKMANN, Gerhard: *Die Weisheit der Vielen*. München : Bertelsmann, 2005
- [Symeonidis und Mitkas 2005] SYMEONIDIS, Andreas L. ; MITKAS, Pericles A.: *Agent Intelligence Through Data Mining*. United States of America : Springer, 2005
- [Tanenbaum und van Steen 2002] TANENBAUM, Andrew S. ; STEEN, Maarten van: *Distributed Systems*. Prentice Hall/Pearson Education, 2002. – 803 S
- [Thatcher u. a. 2006] THATCHER, Jim ; BOHMAN, Paul ; HENRY, Michael ; REGAN, Shawn ; SWIERENGA, Bob ; SARAH ; URBAN, Mark D. ; WADDELL, Cynthia: *Constructing Accessible Web Sites - Web Standards and Regulatory Compliance*. New York - United States of America : friends of ED, 2006
- [Thome] THOME, Mark: *Ortsbezogene Dienste im Paradigma des Web 2.0*. – Voraussichtliche Erscheinungstermin: Mai 2007
- [W3C 1999] W3C: *Web Content Accessibility Guidelines 1.0*. 1999. – URL <http://www.w3.org/TR/WCAG/>. – Stand: 10.01.2007
- [W3C 2006] W3C: *The XMLHttpRequest Object*. 2006. – URL <http://www.w3.org/TR/XMLHttpRequest/>. – Stand: 19.12.2006
- [W3Schools 2006] W3SCHOOLS: *Browser Statistics*. 2006. – URL www.w3schools.com/browsers/browsers_stats.asp. – Stand: 25.09.2006
- [Weber u. a. 2005] WEBER, William ; RABAEY, J.M. ; AARTS, E.: *Ambient intelligence*. Berlin : Springer Verlag, 2005
- [Weiser 1991] WEISER, Mark: *The computer for the 21st century*. In: *Scientific American* (1991), S. 19–25. – URL http://www.sis.pitt.edu/~dist/coursePages/IS2470/weiser_overview.pdf
- [Weiss 2005] WEISS, Aaron: *The power of collective intelligence*. In: *netWorker* 9 (2005), Nr. 3, S. 16–23. – ISSN 1091-3556

- [Wikinews 2006] WIKINEWS: *Wikinews*. 2006. – URL <http://de.wikinews.org/wiki/Hauptseite>. – Stand: 21.04.2007
- [Wikipedia 2006a] WIKIPEDIA: *Alert-Dienst*. 2006. – URL <http://de.wikipedia.org/wiki/Alert-Service>. – Stand: 27.09.2006
- [Wikipedia 2006b] WIKIPEDIA: *Internet Relay Chat*. 2006. – URL http://de.wikipedia.org/wiki/Internet_Relay_Chat. – Stand: 02.02.2007
- [Wikipedia 2006c] WIKIPEDIA: *ISO/IEC 9126*. 2006. – URL http://de.wikipedia.org/wiki/ISO_9126. – Stand: 15.12.2006
- [Wikipedia 2006d] WIKIPEDIA: *JavaScript*. 2006. – URL <http://de.wikipedia.org/wiki/JavaScript>. – Stand: 14.12.2006
- [Wikipedia 2006e] WIKIPEDIA: *Representational State Transfer*. 2006. – URL http://de.wikipedia.org/wiki/Representational_State_Transfer. – Stand: 21.04.2007
- [Wikipedia 2006f] WIKIPEDIA: *Selbstorganisation*. 2006. – URL <http://de.wikipedia.org/wiki/Selbstorganisation>. – Stand: 17.12.2006
- [Wikipedia 2006g] WIKIPEDIA: *Semantisches Wiki*. 2006. – URL http://de.wikipedia.org/wiki/Semantisches_Wiki. – Stand: 10.04.2007
- [Wikipedia 2006h] WIKIPEDIA: *Ward Cunningham*. 2006. – URL http://de.wikipedia.org/wiki/Ward_Cunningham. – Stand: 17.11.2006
- [Wikipedia 2006i] WIKIPEDIA: *Web 2.0*. 2006. – URL http://de.wikipedia.org/wiki/Web_2.0. – Stand: 27.03.2007
- [Woolstone 2006] WOOLSTONE, Daniel: *Pro Ajax and the .NET 2.0 Platform*. Apress, 2006

Versicherung über Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne des §22(4) der Prüfungsordnung des Masterstudiengangs Informatik an der Hochschule für Angewandte Wissenschaften Hamburg vom 22. November 2001, geändert am 7. Dezember 2004, ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 16. Mai 2007

Ort, Datum

Unterschrift