

Drei Forscher der Universität Duisburg-Essen haben sich nach mehreren Jahren Forschung mit der Firma Locoslab selbständig gemacht. Locoslab GmbH ist ein 2012 ins Leben gerufene Spin-off der Universität, das cloud-basierte Produkte und Dienstleistungen rund um das Thema Lokalisierung anbietet. Vielfältige Anwendungsmöglichkeiten durch den technischen Einsatz des so genannten RF (Radio Frequencing) Fingerprinting, beispielsweise in den Bereichen Navigation auf Flughäfen, in Einkaufszentren oder auf Messen. So können personalisierte Mobilitätslösungen realisiert werden.

SIMON macht mobil

Die Cloud ermöglicht personalisierte Mobilität

Von Pedro José Marrón

Als Teil der Aktivitäten im Bereich personalisierte Mobilität, haben sich Mitarbeiter*innen der Firma Locoslab gemeinsam mit Forscher*innen der Universität Duisburg-Essen der Herausforderung gestellt, Navigationsanwendungen zu implementieren, die für Personen mit eingeschränkter Mobilität besonders relevant sind. Im Rahmen des SIMON EU-Projekts wurden unter anderem Lösungen für multimodales Routing erforscht, die in der Lage sind, Outdoor- und Indoor-Routing nahtlos zu kombinieren. Die Integration dieser Lösungen ist in einer App namens SIMON Mobile erfolgt, die für Android und iOS verfügbar ist.

Erfahrungen mit Indoor Navigation in SIMON Mobile

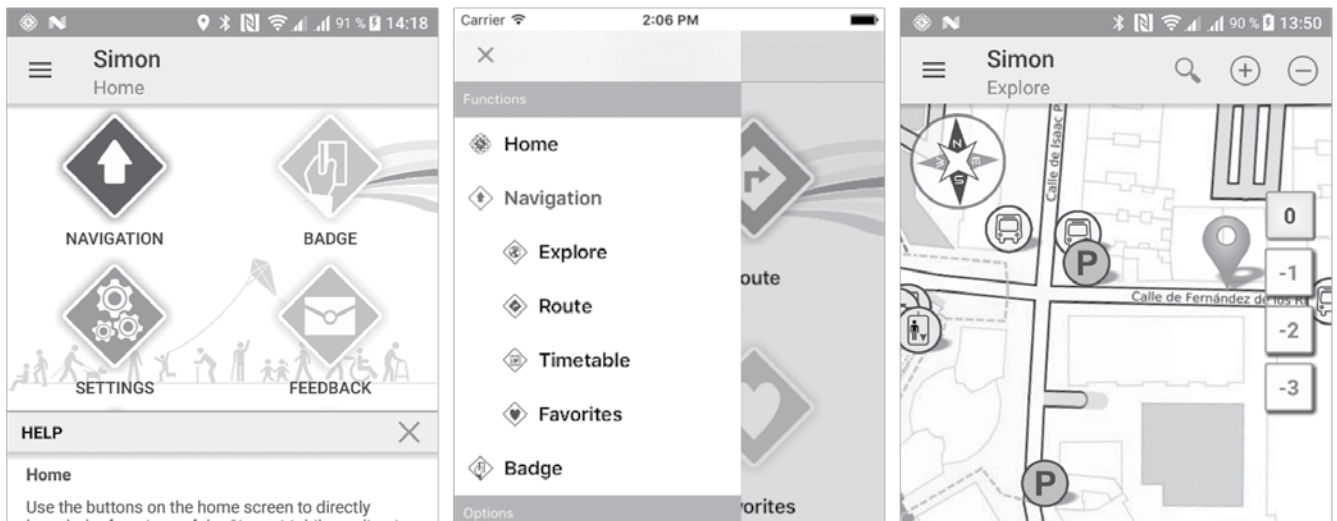
Mit der Ausbreitung GPS-fähiger Geräte haben sich mobile Navigationsanwendungen zu einem weit verbreiteten Werkzeug entwickelt, die Menschen während ihrer täglichen Fahrten unterstützen. Da GPS in Gebäuden nicht verfügbar ist, werden solche Anwendungen meist nur im Außenbereich genutzt. Infolgedessen kann eine nahtlose End-to-End Nutzung nicht gewährleistet werden. Dies ist besonders dann problematisch, wenn man schnell durch ein Gebäude laufen muss, um zum Beispiel eine bestimmte U-Bahn Haltestelle zu erreichen.

Das Ziel des Europäischen Forschungsprojektes SIMON ist die Entwicklung von Tools und technischen Lösungen, die eine Verbesserung der Mobilität von Menschen mit Einschränkungen ermöglichen. Eine dieser Lösungen ist eine mobile Navigationsanwendung, die speziell für Nutzer*innen mit Geh- oder Sehbehinderungen konzipiert wurde. Die Anwendung wird zur Zeit in Lissabon, Parma, Reading und Madrid getestet.

Ein wichtiger Teil der Anwendung ist die Bereitstellung von nahtlosen und multimodalen Navigationsrouten, die sowohl im Außen- wie im Innenbereich funktionieren und verschiedene Verkehrsmittel wie



Pedro José Marrón. Foto: Vladimir Unkovic



(1) SIMON Mobile.
Quelle: eigene Darstellung

Bus, U-Bahn, Auto oder zu Fuß unterstützen. Die Arbeit baut auf Ergebnissen eines früheren EU-Projekts namens GAMBAS auf [2], das an der UDE von unserer Forschungsgruppe koordiniert wurde. Um die neue resultierende Anwendung testen zu können, haben wir ein auf Bluetooth basierendes Indoor-Lokalisierungssystem in Moncloa, eine große Transit-Station in der Stadt Madrid installiert. Darüber hinaus haben wir die Anwendung für Android und iOS über den Google Play Store und den Apple App Store zur Verfügung gestellt. Zum Zeitpunkt des Schreibens hat die Anwendung etwa 1.000 Downloads mit ungefähr 400 aktiven Installationen.

Simon Mobile

Um das Risiko zu minimieren, eine Navigationsanwendung zu bauen, die nicht den Bedürfnissen unserer primären Nutzer*innengruppe entsprechen würde, folgten wir einem partizipativen Designprozess. In einer Reihe von Workshops mit Behörden in den Zielstädten sowie Mitgliedern von Gruppen und Organisationen mit Mobilitätseinschränkungen wurden die Anforderungen und Informationsquellen definiert und zu einem späteren Zeitpunkt validiert.

Im Laufe der Arbeit haben wir schnell erkannt, dass die notwendigen Informationen oft nicht leicht zugänglich waren oder über verschiedene Informationsanbieter und -systeme hinweg verteilt waren. Zum Beispiel musste eine Excel-Tabelle mit den Orten der Aufzüge händisch ausgewertet und mit den Fahrplaninformationen des U-Bahnnetzbetreibers kombiniert werden. Um Routen für die Intercity-Busse berechnen zu können, mussten wir ein Tool entwickeln, das die Buspläne aus etwa 600 PDF-Dateien heraus extrahiert, konsolidiert und in die Cloud-Infrastruktur integriert, die als Basis für die Ausführung der Client-Apps benutzt werden.

Basierend auf den in den Workshops gesammelten Endbenutzer*inneneingaben haben wir die Anwendung entwickelt, die in Abbildung (1) zu sehen ist. Mit einem zweistufigen hierarchischen Menü (1) kann ein*e Benutzer*in auf die Funktionen der Hauptanwendung zugreifen. Jeder Screen ist als Einzelansicht implementiert (Fragmente auf Android und ViewController auf iOS). Für jede Ansicht kann ein*e Benutzer*in eine kontextsensitive Hilfe aufrufen, die entweder auf der Unterseite (im Hochformat) oder auf der rechten Seite (im Querformat) angezeigt wird. Die

Hilfe beschreibt den Zweck der Ansicht sowie deren Funktionalität. Wir haben auch ein Navigationselement integriert (2), das persistent in der oberen linken Ecke angezeigt wird und von jeder Ansicht angeklickt werden kann, indem man eine Swipe-Geste von links nach rechts durchführt. Das Menü bietet einen direkten Zugriff auf wichtigen Funktionen und auf die Hilfe-Ansicht.

Ähnlich wie bei anderen Navigationsanwendungen, ermöglicht eine Explorationsansicht (Abb. 3) dem*der Benutzer*in die Karte unter Verwendung von Panning, Pinching oder Rotieren zu manipulieren. Wenn man den Zoom betätigt, wird die Karte automatisch geladen und relevante Orte werden als Overlay markiert und visualisiert. Dabei werden in der Visualisierung verschiedene Symbole zur Differenzierung der Kategorien verwendet. Außerdem werden Farben benutzt, um dynamische Informationen anzuzeigen (z.B. besetzte Parkplätze mit einem roten „P“ Symbol und verfügbare Parkplätze mit einem grünen „P“ Symbol). Mit einem kurzen Tap auf den Screen kann ein*e Benutzer*in einen Marker setzen, mit einer langen Berührung des Bildschirms wird ein anderer Ort ausgewählt, um Details darstellen zu können. Diese Detailansicht enthält



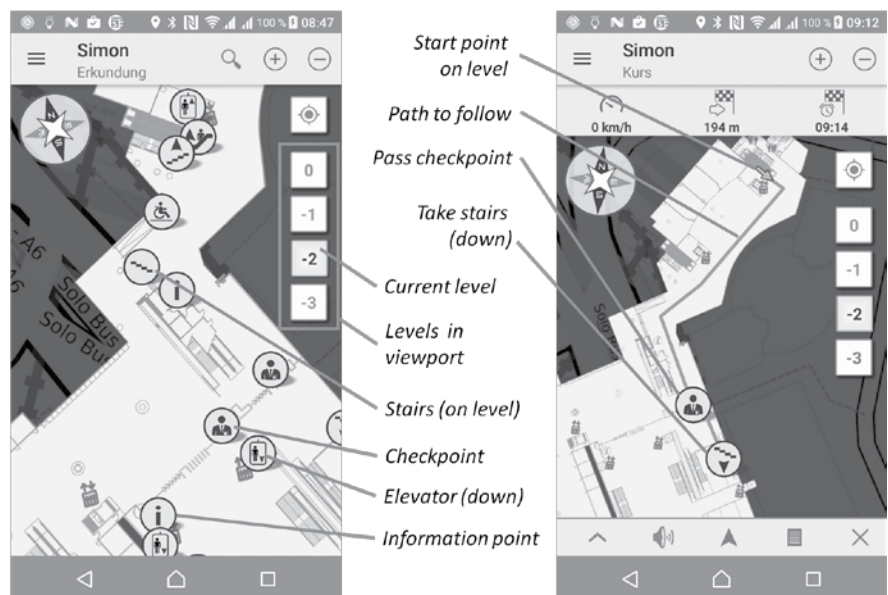
eine Gruppe von Typ-spezifischen Schaltflächen, die es dem*der Benutzer*in ermöglichen, einen Ort als Eingabe für weitere Aktionen zu verwenden (z.B. um einen Zeitplan für eine Bushaltestelle abzurufen oder um den Ort als Start- bzw. Zielort für eine Route auszuwählen).

Bei der Suchansicht (Abb. 4) kann der Benutzer Orte nach Namen oder Adressen suchen. Ob eine Suche lokal oder global durchgeführt wird, hängt davon ab, ob die Informationen mit dem aktuellen Kontext (z.B. Standort) verbunden sind oder nicht. Die globale Suche erfordert von dem*der Benutzer*in Teile eines Stadtnamens einzugeben und diese wird dann mit den gespeicherten Daten in der Cloud verglichen. Die lokale Suche verwendet den aktuellen Benutzer*innenkontext, um die Ergebnisse entsprechend der Distanz zum Standpunkt zu ordnen und einzuschränken. Der*Die Benutzer*in kann auch eine Reihe von Schaltflächen nutzen, um die Ergebnisse auf einen bestimmten Ort zu begrenzen. Wenn ein Ergebnis angezeigt wird, werden die Suchbegriffe und der Suchverlauf gespeichert. Somit ist es dann möglich, zu einem späteren Zeitpunkt vorherige Anfragen schnell wieder durchführen zu können.

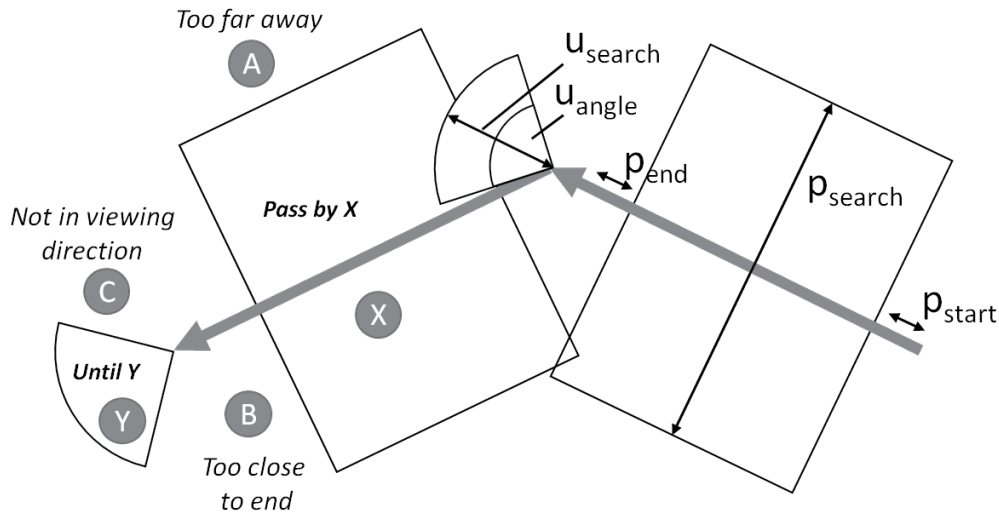
Unter Verwendung einer Zeittabellenansicht (Abb. 5) kann ein*e Benutzer*in an den Haltestellen auf die Abfahrt- und Ankunftszeiten von öffentlichen Verkehrsmitteln zugreifen. Dafür kann der*die Benutzer*in entweder die Explore-Ansicht oder die Suchansicht betätigen. Wenn die Suchansicht verwendet wird, werden die Filtertasten ausgeblendet und die Suchansicht automatisch so eingestellt, dass sie nur Stopps und Stationen anzeigt. Wenn die Abfahrtsinformationen für eine bestimmte Haltestelle angefragt werden, werden die Ergebnisse Zeile für Zeile gruppiert. Für jede Zeile werden nicht nur die Abfahrtsinformationen pro Stunde angezeigt sondern auch Echt-Zeit Informationen und Verhinderungen, die einen Einfluss auf den Verlauf der Route haben können. Bei Bedarf werden diese Informationen in verschiedenen Sprachen automatisch in der Cloud übersetzt. Dafür haben wir unser Backend mit den APIs von Microsoft integriert, die eine automatische Übersetzung zur Verfügung stellen (<https://azure.microsoft.com/en-us/services/cognitive-services/translator-text-api/>).

In der Routenansicht (Abb. 5) kann ein*e Benutzer*in Routen von

einem Start zu einem Ziel unter Berücksichtigung verschiedener Bewegungsmodalitäten (zu Fuß, mit dem Auto oder mit öffentlichen Verkehrsmitteln) berechnen. Bei der Eingabe des Start- beziehungsweise Endpunktes der Reise können verschiedene Ansichten benutzt werden. Falls einer der Punkte nicht spezifiziert wird, benutzt das System die jetzige Position des*der Nutzer*in. Um Reisen planen zu können, kann auch eine Uhrzeit angegeben werden, an der die Reise stattfinden soll. Darüber hinaus erstellt SIMON Mobile weitere Optionen für die Berechnung der Routen, die mit dem Profil des*der Nutzer*in in Verbindung gebracht werden können. Zum Beispiel kann der*die Nutzer*in die Routenberechnung auf „nur Bus“ oder „nur Bahn“ einschränken oder angeben, dass er Mobilitätseinschränkungen hat und nur Aufzüge und rollstuhlgerechte Zugänge auszugeben sind. Die Berechnung der Route findet in der Cloud statt. Die Ergebnisse wiederum werden in der App dargestellt und je nach Einschränkung entweder lediglich visuell angezeigt oder zusätzlich automatisch vorgelesen. Dies ermöglicht, dass Reisende mit Sehheinschränkungen unsere App



(2) Indoor Visualisierung.
Quelle: eigene Darstellung



(3) Berechnung der Landmarks.
Quelle: eigene Darstellung

auch benutzen können. Die Liste der möglichen Routen wird nach Relevanz sortiert und dem*der Benutzer*in präsentiert. Um alternative Ergebnisse schnell vergleichen zu können, verwenden wir verschiedene Icons, die die einzelnen Teile der Route darstellen. So wird zum Beispiel ein Bus oder eine U-Bahn für das Segment benutzt, wo der*die Nutzer*in diese Verkehrsmittel nutzen muss oder ein*e Fußgänger*in, wenn er*sie zu Fuß die angezeigte Anzahl der Meter laufen muss. Darüber hinaus werden auch zusammenfassende Informationen wie Abfahrtszeit, Ankunftszeit, Umstiege oder die Gesamtdauer der Reise berechnet und gemeinsam mit jeder Route dargestellt. Diese Informationen werden entweder in Textform oder auf einer Karte angezeigt, damit der*die Nutzer*in auch die Möglichkeit hat, sich auf der Karte zu orientieren und visuell zu entscheiden, welche Route sich besser eignet.

Wenn die Schritt-für-Schritt-Navigation für eine ausgewählte Route gestartet wird, schaltet die Anwendung zu einer Tracking-Ansicht (8), die die verbleibende Route des Benutzers als Polylinie auf einer Karte visualisiert. Außerdem zeigt die Ansicht die jetzige Position des*der Nutzer*in sowie wichtige

Points of Interest (POIs). Die Ansicht unterstützt zusätzlich einen „Follow-me“-Modus, der den*die Benutzer*in immer in der Mitte der Karte platziert und die Karte schiebt und rotiert, wenn der Nutzer in die eine oder andere Richtung läuft. Um den Drehwinkel zu berechnen, verwenden wir die Ausrichtung des Mobiltelefons, die entweder von GPS oder vom eingebauten Beschleunigungssensor/Magnetometer lokal erfasst wird. Die Tracking-Ansicht zeigt auch die Geschwindigkeit des*der Benutzer*in, die verbleibende Entfernung, sowie die erwartete Ankunftszeit. Der „Follow-me“-Modus wird automatisch deaktiviert, wenn der*die Benutzer*in beginnt, selber die Karte zu manipulieren (z.B. durch Pinching, Panning, usw.), und es kann wieder aktiviert werden durch Drücken des „Locate me“-Buttons in der oberen rechten Ecke der Anwendung.

Während der Schritt-für-Schritt-Navigation kann ein* Benutzer*in weiterhin alle Funktionen der Anwendung benutzen. Wenn die Navigation aktiv ist, zeigt die Anwendung kontinuierlich eine Leiste am unteren Rand, die die nächsten eine bis zwei Anweisungen mit Icons und Text visualisiert. Die Leiste kann aber auch minimiert

werden, um mehr Platz auf dem Screen zu schaffen, ohne die Möglichkeit zur Navigation zu verlieren.

Indoor-Navigation

Um Indoor-Navigation in das System zu integrieren, mussten wir die zugrundeliegenden Visualisierungen, Dienste und Algorithmen anpassen. Insbesondere mussten wir Erweiterungen in der Modellierung und Visualisierung der einzelnen Elemente durchführen.

Modellierung

Für die Außennavigation setzen wir auf Daten des OpenStreetMap-Projekts. Das Datenmodell besteht aus Knoten, Wegen und Beziehungen. Ein Knoten repräsentiert einen bestimmten Punkt auf der Karte, der als WGS84 Koordinate definiert ist. Ein Weg stellt eine Reihe von Punkten dar, die eine Polylinie sein können (z.B. um eine Straße zu repräsentieren) oder ein Polygon (z.B. ein See oder ein Wald, etc.). Unter Verwendung von Relationen ist es möglich, Gruppen von Objekten (z. B. eine Menge von Straßen, die eine Autobahn repräsentieren) zu definieren, die bestimmte Eigenschaften haben (z.B. Beschränkung an einer

bestimmten Straßenkreuzung). Um die Semantik dieser Elemente zu definieren, werden Attribute-Value Pairs benutzt, deren Bedeutung und Verwendung in den Richtlinien des OpenStreetMaps Projekts detailliert beschrieben werden.

Allerdings ist es nicht möglich, diese Daten direkt aus dem OpenStreetMap Projekt zu verwenden. Dafür ist es notwendig, einen Renderer zu schreiben, der diese Daten für die Außennavigation vorbereitet, synthetisiert und basierend auf der semantischen Information, die richtigen geometrischen Elemente extrahiert.

Darüber hinaus benötigen wir eine Routing-Engine, die auf den Graphen arbeitet, die durch die topologische Struktur der Wege definiert werden. Dadurch können Straßen erkannt und zusammen gefasst werden. Für die Indoor-Navigation benutzen wir das gleiche Modell, um die Wege in einem Gebäude zu definieren und als Input für die Routenberechnung bereit zu halten. Um mehrere Etagen zu unterstützen, fügen wir dem Knoten des Modells einen Integer hinzu, der die Etage darstellt, wo sie zu finden sind. So wird eine 0 für das Erdgeschoss benutzt, positive Zahlen für höhere Etagen und negative Werte, um unterirdische Etagen wie bei einer U-Bahn-Station zu repräsentieren.

Wir generieren dann Links zwischen zwei Knoten auf zwei verschiedenen Ebenen, um zwei Etagen miteinander zu verbinden, womit wir die Möglichkeit haben, Treppen, Rolltreppen, Rampen oder Aufzüge zu modellieren. Die Unterschiede zwischen einer Treppe und einem Aufzug wird durch die Nutzung von semantischen Informationen an den jeweiligen Knoten realisiert. Orte, die für den*die Nutzer*in von Interesse sind, werden als Knoten (z.B. ein Geschäft), Wege (z.B. einen Gang) oder Polygone (z.B. eine Einkaufszone) in dem Modell dargestellt und mit besonderen semantischen Informationen versehen.

Um den Berechnungsaufwand zu minimieren, wird dieses Modell als Input der Routing-Engine in der Cloud verwendet. Die Darstellung der einzelnen Grundrisse, Etagen und so weiter werden vorberechnet und als Bild auf die Client-Anwendung übertragen. Durch die Definition von drei Ankerpunkten auf der Basis der OpenStreetMap-Daten, können wir die vorberechneten Bilder auf den WGS84-Koordinaten abbilden, die von den Ankerpunkten dargestellt werden. Durch eine affine Transformation werden dann lokal von der Client-Anwendung die Bilder auf der Karte dargestellt und als Overlay an der korrekten Stelle angezeigt.

Visualisierung

Für die Visualisierung der Innenräume stellen wir zwei Dienste zur Verfügung. Der erste Dienst wird benutzt, um die Overlay-Bilder zu empfangen. Da die Auflösung der vorberechneten Bilder hoch sein kann, werden die Bilder in Teile gesplittet, die individuell mit dem slippy Addressierungsmechanismus abgerufen werden können [4], die von OpenStreetMap und anderen Kartenanbieter verwendet wird. Der zweite Dienst ermöglicht die Berechnung der Etagen, die in einer View der Karte zu finden sind. Dazu werden Bounding-Boxes mit Etage-Informationen aus den Modell-Elementen herausgefiltert und zurückgegeben.

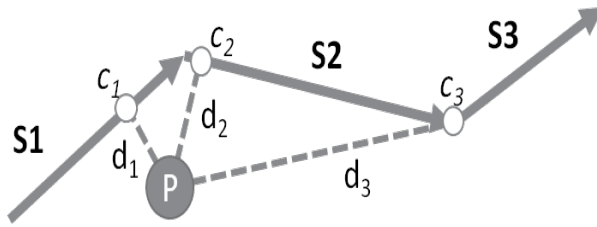
Beim Testen der Visualisierung fanden wir schnell heraus, dass eine zweidimensionale Darstellung der verbindenden Elemente (siehe Abb. 2) auf verschiedenen Etagen (z.B. Treppen, Rolltreppen, usw.) nicht immer von allen Nutzer*innen korrekt interpretiert wurde. Insbesondere kam die Frage „Welche Rolltreppe fährt nach oben und welche nach unten? Welche soll ich nehmen? Manchmal war es so, dass es nicht ersichtlich war, welche zwei Etagen durch eine Rolltreppe verbunden sind und das hat in solchen Fällen für Verwirrung gesorgt. Ein ähnliches Problem hing mit der Darstellung der 3D-Routen zusammen. SIMON Mobile unterstützt zwar die Berechnung von Routen, die über mehrere Etagen laufen, aber die Darstellung von solchen Ergebnissen war für viele Nutzer nicht intuitiv. In manchen Fällen war es nicht klar, welche Pfade mit welcher Etage gekoppelt waren, womit unsere Testgruppe Schwierigkeiten hatte, den Anfang und das Ende der Route auf den jeweiligen Etagen zu finden.

Um diese Probleme zu lösen, fügten wir Informationen über die dritte Dimension hinzu anhand von Annotationen auf der Karte, die nicht nur die Art der Verbindung



(4) Beispielroute mit Textanweisungen.

Quelle: eigene Darstellung



(5) Positionsbestimmung auf einer Linie mit drei Segmenten.

Quelle: eigene Darstellung

zwischen zwei Etagen darstellen, sondern auch die Richtung die für die Route relevant ist. In der Karte sind diese zusätzlichen Informationen für alle sichtbaren Elemente zu finden aber in der Routen-Darstellung werden nur die Markierungen gezeigt, die für die Route relevant sind. Dadurch kann sich der*die Benutzer*in auf die wesentlichen Routeninformationen konzentrieren und wird weniger durch Extra-Informationen abgelenkt. Für den Anfang der Route wurde einen Pfeil hinzugefügt, der die Richtung anzeigt, der der*die Nutzer*in zu folgen hat. Die daraus resultierende Visualisierung ist in Abbildung (2) dargestellt.

Zusammenfassung der dargestellten Informationen

Da mehrere unserer Tester*innen Sehbehinderungen haben, war eine Zusammenfassung der Indoor-Route in Textform notwendig, um die Anweisungen nicht nur visuell sondern auch als Sprachausgabe darstellen zu können. Darüber hinaus sind Anweisungen in Bezug auf Meter oder Winkel nicht so nützlich wie andere, die auf Landmarks basieren und diese als Ankerpunkte gebrauchen [3]. Die richtige Mischung aus Hinweisen und Textbeschreibungen haben wir in der Zusammenarbeit mit den Nutzergruppen des Konsortiums über mehrere Tests herausgearbeitet und in der Application integriert. Folgende Elemente des Modells haben

wir mit den folgenden Eigenschaften erweitert:

- *Links*: Die Beschreibungen der Route beinhalten explizite Informationen über die Verknüpfung der folgenden Etagen wie zum Beispiel „Nehmen Sie den Aufzug runter bis Ebene -2.“. Somit ist es klar, welche zwei Etagen ein Link verbindet und diese kann auch bei den Routing-Anweisungen verwendet werden.

- *Landmarks*: Wie in [3] beschrieben, wurden Landmarks als primäres Element benutzt, um Wendepunkte zu identifizieren. Personen, die das Gebäude gut kennen, orientieren sich eher über bekannte Orte, zum Beispiel „Laufen Sie an Geschäft X vorbei und anschließend biegen Sie rechts ab“. Aber auch andere Nutzer*innen, die sich nicht im Gebäude auskennen, orientieren sich lieber im Innenbereich über Landmarks als über relative Distanzangaben, wie es in Autonavigationsanwendungen üblich ist.

- *Kompression*: Für Anweisungen, die schnell nacheinander durchgeführt werden sollten, weil sie in einer relativ kurzen Distanz auftreten, ist es wichtig, beide Anweisungen in einem Satz auszugeben. Das gibt dem*der Nutzer*in die Möglichkeit, sich vorzubereiten und nicht über die nächste Anweisung überrascht zu werden. Aus diesem Grund generiert unsere Anwendung komprimierte Hinweise, die zum Beispiel wie folgt aussehen: „Gehen Sie die Treppe hinunter und dann biegen Sie rechts ab“.

- *Zonen*: Die Beschreibungen des

Verkehrsbetreibers beinhalten häufig Anweisungen über Zonen, die von den Nutzer*innen eindeutig identifiziert werden können. Zum Beispiel „Gehen Sie in Richtung Metro“ oder „Gehen Sie in Richtung Einkaufszone“. Diese semantischen Informationen müssen in das Modell eingepflegt werden und als Landmark benutzt werden, um die Anweisungen verständlicher zu machen.

- *Anmerkungen*: Interessanterweise wurden auch solche Aussagen und Hinweise von den Nutzer*innen erwartet wie „Bitte beachten Sie die Haltestellenregeln.“ oder „Bitte Vorsicht bei der Einfahrt des Zuges“, sowie Hilfestellungen wie „Ein rollstuhlgerechter Eingang befindet sich auf der linken Seite der Barriere.“. Diese Anmerkungen und Hinweise wurden nicht nur modelliert, sondern auch an der entsprechenden Stelle wie zum Beispiel beim Betreten des Gebäudes oder der U-Bahn Zone ausgegeben.

Basierend auf diesen Beobachtungen und Anforderungen der Nutzer*innen, haben wir Notizen und Zonen zu unserem Modell hinzugefügt, so dass es möglich ist, Sprachausgaben an den jeweiligen Stellen durchzuführen, die für die Route, Zone oder Wendepunkte notwendig sind. Um diese Informationen in unser Modell zu integrieren, haben wir einen Nachbearbeitungsschritt eingefügt, der auf der räumlichen Analyse der Routen basiert und in fünf Schritten durchgeführt wird.

Zuerst gruppieren wir die Pfade, die auf einer gerade Linie auf einer bestimmten Ebene sind. Dafür berechnen wir die Winkel zwischen den Segmenten der Polylinie der Route und teilen sie an den Stellen, an denen der Winkel einen voreingestellten Schwellenwert überschreitet oder Übergänge repräsentiert werden (z.B. Treppen, Aufzüge, Kontrollpunkte, usw.). Das Ergebnis ist eine Reihe von Gruppen, die entweder gerade Linien oder Transitionen repräsentieren. Zwischen den Gruppen wird der Winkel berechnet,

um Abbiege-Anweisungen zu generieren.

In dem zweiten Schritt werden die semantischen Informationen identifiziert, die zu den Zonen gehören, die sich mit der Route schneiden. Dafür wird eine topologische Analyse durchgeführt, die die entsprechenden Elemente aus dem Modell extrahiert, die für die Route relevant sind.

Als dritter Schritt wird für jede Gruppe, die keinen Übergang darstellt, eine räumliche Analyse durchgeführt, um zu bestimmen, welcher Landmark benutzt werden soll, um diesen Teil der Route zu identifizieren. Dafür wird ein Verfahren benutzt, was grafisch in Abbildung (3) zu sehen ist. Dadurch wird auch erkannt, welche Landmarks erreicht werden und an welchen vorbei gelaufen wird.

Bei der Identifizierung der Landmarks, an denen vorbei gelaufen wird, muss gewährleistet werden, dass diese Landmarks nicht zu nah an den Start- und Endpunkten der einzelnen Gruppen sind. Bei der Identifizierung der erreichten Landmarks müssen wir feststellen, ob die Distanz zwischen der Route und des Landmarks bei einem optimalen Winkel kleiner ist, als ein vordefinierter Schwellenwert.

Im vierten Schritt versuchen wir, Gruppen zu identifizieren, die in einer komprimierten Einzelanweisung auftauchen sollen. Dafür identifizieren wir zuerst Gruppen, die eine vordefinierte Länge unterschreiten und keinen assoziierten Landmark oder Zone haben. Um sicher zu stellen, dass wir nicht die komplette Route in einer langen Anweisung komprimieren, limitieren wir die Kompression auf zwei Segment-Längen.

Schließlich verwenden wir die resultierenden komprimierten Gruppen mit Landmarken, Zonen, Notizen und Abbiege-Informationen, um die Anweisungen zu erzeugen. Eine einfache Grammatik wird für die automatische Generierung der Anweisungen benutzt, so dass

mehrere Sprachen leichter unterstützt werden können. Eine einfache Version der Templates, die wir dafür verwenden ist unten angegeben. Die eckigen Klammern zeigen optionale Teile an. Eine Beispielroute mit den resultierenden Textzusammenfassungen ist in Abbildung (4) zu sehen.

A) Nehmen Sie den `<LinkType>` [auf Ebene `<targetLevel>`] [in Richtung `<NextZone>`] [dann wiederum `<TurnType>`] [und dann sofort `<nextTurnType>`]

B) Gehen Sie [durch `<currentZone>`] [vorbei an `<passingPlace>`] [bis `<ErreichtPlace>`] [dann wiederum `<TurnType>`] [und dann sofort `<nextTurnType>`] [in Richtung `<nextZone>`].

Tracking

Schritt-für-Schritt-Navigation erfordert die Lokalisierung des*der Nutzer*in auf der Route. Basierend auf der verbleibenden Route kann die Anwendung die geeignete Beschreibung für die nächste Aktion generieren, auf dem Bildschirm darstellen und gegebenenfalls sprachlich ausgeben.

Die Bestimmung der verbleibenden Strecke wird anhand der Strategie gemacht, die in Abbildung 5 zu sehen ist. Diese Strategie funktioniert gleichermaßen für den Außen- und den Innenbereich und basiert auf der Berechnung des nächsten Punktes (c1–3) zu jedem Liniensegment (S1–3), die die Route definiert. Dadurch werden die Linien berechnet, die senkrecht zu jedem Segment liegen und die durch die erfasste Position des*der Nutzer*in laufen. Das Segment, das die kleinste Distanz aufweist, wird dann als die Position des*der Nutzer*in auf der Route betrachtet.

Ein wesentlicher Unterschied zwischen Indoor- und Outdoor-Routen besteht darin, die Segmente auf verschiedenen Etagen voneinander zu unterscheiden und nur die Segmente zu nehmen, die auf der gleichen Ebene wie die erfasste Position des*der Nutzer*in sind.

Falls die minimale Distanz nicht berechnet werden kann, zum Beispiel weil der*die Benutzer*in sich auf einer Etage befindet, die nicht direkt zu der Route gehört oder die berechnete Distanz zu der zur Zeit aktiven Route zu groß ist, geht das System davon aus, dass der*die Benutzer*in sich nicht auf der Route befindet, vermutlich in eine andere Richtung gelaufen ist und eine neue Berechnung der Route erfolgen muss, damit die richtigen Anweisungen dargestellt werden können.

Im Laufe des SIMON-Projekts haben wir mit Anweisungen auf verschiedenen Granularitätsniveaus experimentiert. Diese gingen von den üblichen Anweisungen wie bei der Outdoor-Navigation (z.B. „in X Meter bitte nach links abbiegen“) bis hin zu allgemeinen Anweisungen wie „Gehen Sie in Richtung Norden“. Bei den Tests mit den Endnutzer*innen haben wir aber festgestellt, dass die Anweisungen, die am besten verstanden und als korrekt eingestuft werden, diejenigen sind, die mit Landmarks arbeiten und nicht unbedingt davon ausgehen, dass die Positionierung eine maximale Ungenauigkeit aufweist. Nicht nur die Positionierung, sondern auch die Bestimmung der Bewegungsrichtung anhand des Magnetometer-Sensors ist nicht in jedem Fall fehlerfrei zu schaffen und darf auch nicht uneingeschränkt benutzt werden. Stromleitungen im Gebäude oder Stahlkonstruktionen spielen hierbei eine besonders wichtige Rolle und führen oft zu Fehlern, die sich dann in falsche Anweisungen übersetzen lassen.

Erfahrungen

Im Laufe des Projektes haben wir 18 Monate gebraucht, bis wir die Backend-Services in der Cloud entwickelt haben, die Datenquellen integriert haben und die erste Version der Client-Anwendungen für Android mit Hilfe unserer Nutzer*innengruppen frei verfügbar gemacht haben. Danach haben wir

kleine Nutzertests gemacht, die uns erlaubt haben, weitere Entwicklungen und Iterationen mit den Nutzern zu machen. Dadurch haben wir 33 fortlaufende Versionen der Applikation innerhalb von fünf Monaten entwickelt, die dann als Beta-Version zu einer kleinen Gruppe von Tester*innen verschickt worden ist. Nachdem diese Tests erfolgreich gewesen sind, haben wir die öffentliche Version der Android Applikation gelauncht und nebenher die Portierung zu iOS initiiert. Sechs Monate später haben wir die iOS Version veröffentlicht. Seitdem haben wir weitere Updates und Bugfixes zur Verfügung gestellt, die die Anwendungen immer besser und einfacher nutzbarer gemacht haben.

Wir haben unsere Anwendung zusätzlich instrumentiert, um verstehen zu können, wie die Endnutzer*innen mit unserer App interagieren. Dafür haben wir ein Logging-Framework integriert, das UI-Interaktionen sowie Interaktionen zwischen der App und des Cloud-Backends protokolliert. Wir haben selbstverständlich Vorkehrungen getroffen, um sicherzustellen, dass die erfassten Daten nicht die Identität des*der Nutzer*in offenlegen und nur von uns sichtbar sind. Wir verwenden auch Pseudonyme, die bei der Installation der App neu generiert werden und sich auch manuell ändern lassen können. Darüber hinaus haben wir die Nutzer*innen beim ersten Start der Anwendung darüber informiert, dass Nutzungsdaten erfasst und für Optimierung- und Verbesserungszwecke anonymisiert in der Cloud evaluiert werden.

Bis heute wurde die Anwendung 1.065 Mal installiert (Android: 803, iOS: 262), von denen schätzungsweise 415 Installationen derzeit noch aktiv sind. Gemeinsam haben die Benutzer*innen nach 13.831 Standorten gesucht, 1.748 Anfragen für Routen gestellt und 292 Fahrpläne ausgesucht. Interessanterweise wurde die Karten-basierte Explorationsansicht 2.806 mal benutzt, was

ähnlich ist wie die Ansicht, die für die Berechnung der Routen benutzt wird (2.212 mal). Es sieht deswegen so aus, dass die Nutzer*innen die Routen-Berechnungs-Ansicht sechsmal häufiger nutzen als die Fahrplanningformationsansicht. Dieses Verhalten unterstreicht die Wichtigkeit, solche Fahrplaninformationen direkt in der Routen-Ansicht darzustellen.

Positiv zu bewerten ist auch die Tatsache, dass alle komplexere Berechnungen in der Cloud durchgeführt werden und die Client-Anwendungen „lediglich“ als Darstellungsoberfläche für die Cloud-Services dienen. Durch die Nutzung einer Cloud-basierten Infrastruktur können verschiedene mobile Applikationstechnologien schneller unterstützt werden. Auch die Verbesserung der Routing-Algorithmen oder Bugfixes in der Software können einfacher durchgeführt werden und in den meisten Fällen benötigen sie nicht mal die Installation einer neuen App. Ausgeschlossen sind natürlich Änderungen, die mit der Benutzer*innenoberfläche der einzelnen Apps zu tun haben.

Zusammenfassung

SIMON Mobile ist eine der wenigen Anwendungen, die in der Lage ist, multimodale Indoor- und Outdoor-Navigation nahtlos zu ermöglichen. Darüber hinaus unterstützt das System nativ auch Nutzer*innen mit Mobilitäts- sowie Seheinschränkungen und passt die Routenberechnung und die Darstellung der Ergebnisse der entsprechenden Zielgruppe an. Die Cloud-basierte Infrastruktur und die Tatsache, dass die Berechnungen in der Cloud durchgeführt werden, hat uns ermöglicht, die Portierung unserer Lösung für verschiedene Plattformen sehr schnell durchzuführen. Das Feedback der Nutzer*innen in Madrid, Parma, Lissabon und Reading ist hervorragend und mit dem ausdrücklichen Wunsch gekoppelt, das System auch nach Ende des Projekts weiter laufen

zu lassen und mit zusätzlicher Funktionalität zu erweitern.

Summary

Three researchers at the University of Duisburg-Essen opened a company called Locoslab after several years of research in the area of localization. This article discusses the experiences of the research team in the design, development and real-life testing of SIMON Mobile, a mobile application that uses a cloud infrastructure to provide support for mobility impaired users for different cities like Madrid, Parma, Lisbon and Reading. The application is the main product of three years of collaborative research as part of the SIMON European project, that involves researchers from Spain, Italy, Portugal and Germany. The main asset of the application is the seamless integration of indoor and outdoor navigation routes and the fact that the computed routes take into account the characteristics of the user in terms of capabilities and preferences. At the time of this writing, SIMON Mobile has been available for several months at the Google and Apple App Stores and has been downloaded over 1000 times with very good user feedback. Additionally, all of the operations from the computation and retrieval of the map tiles to the algorithm that computes the optimal route for the user, are performed in the cloud. The client application simply acts as the local device where the information from the cloud is presented in the appropriate form to the user. This has the advantages that routing algorithms, for example, can be improved on the cloud backend and do not require the update of the application on the client side in order to use the new and improved services.

Danksagung

Wir danken allen Mitgliedern unserer Benutzer*innengruppen, die wertvolles Feedback für die Entwicklung der SIMON Mobile Apps geliefert haben. Darüber hinaus danken wir den Vertretern der Behörden von Lissabon, Parma, Reading und Madrid für den Zugang zu den Datenquellen, die für die Applikation benutzt wurden sowie ihr Feedback und ihre Hilfe während der Entwicklung und Installation der Cloud Dienste, die für die Applikationen notwendig waren.

Anmerkungen/Literatur

- 1) 2017. Google Maps. (2017). <https://www.google.com/maps> Besucht am: 2017-02-02.
- 2) M. Handte, S. Foell, S. Wagner, G. Kortuem, and P. J. Marron. 2016. An Internet-of-Things Enabled Connected Navigation System for Urban Bus Riders. *IEEE Internet of Things Journal* 3, 5 (Oct 2016), 735–744. DOI: <http://dx.doi.org/10.1109/JIOT.2016.2554146>
- 3) Andrew J. May, Tracy Ross, Steven H. Bayer, and Mikko J. Tarkiainen. 2003. Pedestrian navigation aids: information requirements and design implications. *Personal and Ubiquitous Computing* 7, 6 (2003), 331–338. DOI: <http://dx.doi.org/10.1007/s00779-003-0248-5>
- 4) OpenStreetMap Project. 2017. OpenStreetMap Wiki. http://wiki.openstreetmap.org/wiki/Slippy_map_tilenames. (2017).

Der Autor

In Spanien geboren, studierte **Pedro José Marrón** das Fach „Computer Engineering“ an der University of Michigan in Ann Arbor bis 1998. Ab 1999 arbeitete er an der Universität Freiburg an seiner Doktorarbeit, die er 2001 mit Auszeichnung abschloss. Von 2003 bis 2007 leitete er die Arbeitsgruppe „Mobile Data Management and Sensor Network“ an der Universität Stuttgart. Seine erste Professur führte Pedro Marrón 2007 an die Universität nach Bonn. Dort leitete er in der Computerwissenschaft die Arbeitsgruppe „Sensor Networks and Pervasive Computing“. Seit 2009 ist er Professor an der Universität Duisburg-Essen und war bis 2015 gleichzeitig führender Wissenschaftler am Fraunhofer Forschungsinstitut für Kommunikation, Informationsverarbeitung und Ergonomie (FKIE) in Wachtenberg. Pedro Marrón ist einer der Gründer der Locoslab GmbH, einem Unternehmen, das kostengünstige Lösungen zur Lokalisierung in geschlossenen Räumen anbietet. Ferner ist er Vorsitzender der Steuerungskomitees von UBICITEC, kurz für „European Center for Ubiquitous Technologies and Smart Cities“. An diesem virtuellen Zentrum sind etwa 20 europäische Partner aus Wissenschaft und Industrie beteiligt. Das Ziel von UBICITEC ist die Koordination der Forschungsvorhaben zum Thema Smart Cities.

DuEPublico

Duisburg-Essen Publications online

UNIVERSITÄT
DUISBURG
ESSEN

Offen im Denken

ub | universitäts
bibliothek

Dieser Text wird über DuEPublico, dem Dokumenten- und Publikationsserver der Universität Duisburg-Essen, zur Verfügung gestellt. Die hier veröffentlichte Version der E-Publikation kann von einer eventuell ebenfalls veröffentlichten Verlagsversion abweichen.

DOI: 10.17185/duepublico/70378
URN: urn:nbn:de:hbz:464-20190813-121246-1

Erschienen in: UNIKATE 50 (2017), S. 66-75

Alle Rechte vorbehalten.