Konfiguration funkbasierter Orientierungshilfen für behinderte Personen im Fußgängerverkehr

M. Hennig[1], A. Fay[1] und S. Koch[2], [1]Institut für Automatisierungstechnik, Helmut-Schmidt-Universität, Hamburg, hennigm@hsu-hh.de, [2]RTB GmbH & Co. KG, Schulze-Delitzsch-Weg 10, D-33175 Bad Lippspringe

Kurzfassung

Im Projekt SANBI wird ein großräumig angelegtes Informations- und Kommunikationssystem entwickelt, das den technologischen Kern eines Assistenzsystems in öffentlichen Verkehrsanlagen bildet. Ortsbasierte Assistenz-Dienste bieten in der Infrastruktur individuelle Orientierungshilfen für benachteiligte Personen im öffentlichen Fußgängerverkehr an. Das im Projekt entwickelte System soll eine lückenlose Kette von Orientierungspunkten bieten, an der sich der Benutzer sicher auf Ziele in komplexen Räumlichkeiten zubewegen kann. Eine Problemstellung ist der notwendige Konfigurations- sowie Parametrierungsaufwand der stationären Systemeinheiten während der Inbetriebnahme sowie im alltäglichen Betrieb. In jedem Fall ist eine Anpassung an die jeweiligen örtlichen und betrieblichen Gegebenheiten notwendig. Je vielfältiger die Hilfsangebote des Systems und je mehr technische Komponenten eingebunden sind, desto mehr Systemänderungen sind darüber hinaus zu erwarten. Daher bedarf es einer Lösung, die eine einfache und zeiteffiziente Konfiguration sowie Parametrierung durch den Betreiber oder Instandhalter ermöglicht. Dieser Beitrag beschreibt die Verwendung eines graphischen, regelbasierten Editors und die weitere Verarbeitung der damit erzeugten Systemänderungen.

1 Einleitung

Im Projekt SANBI wird ein großräumiges funkbasiertes Informations- und Kommunikationssystem entwickelt, welches den technologischen Kern eines Assistenzsystems sowohl im öffentlichen Raum als auch in Verkehrsanlagen bildet. Ortsbasierte Assistenz-Dienste bieten in der baulichen Infrastruktur individuelle Orientierungshilfen für benachteiligte Personen im öffentlichen Raum an. Das Assistenzsystem ist so konzipiert, dass die komplexen Einheiten (Anker) in der Infrastruktur (Ampeln, Gebäude, Bushaltestellen) integriert sind, während der Benutzer entweder ein spezielles, kostengünstiges und einfach zu bedienendes Benutzergerät bei sich trägt oder die Funktionen über sein Smartphone ansteuert. [1]. Das System soll dabei eine lückenlose Kette von Orientierungspunkten bieten, an der sich der Benutzer in komplexen Räumlichkeiten sicher auf Ziele zubewegen kann. Zusätzlich bietet das System neuartige Assistenzdienste an, wie z.B. Grünanforderung an Lichtsignalanlagen, Fahrstuhlruf oder akustische Ausgaben von dynamischen Fahrtzielanzeigern.

Die hier behandelte Problemstellung betrifft den notwendigen Konfigurations- und Parametrierungsaufwand während der Inbetriebnahme sowie im alltäglichen Betrieb. Für Betreiber und Instandhalter (z. B. in Verkehrsunternehmen oder Kommunen) erhöht sich die Komplexität des Assistenzsystems, je vielfältiger die Hilfsangebote des Systems sind. Es kann aber nicht vorausgesetzt werden, dass das Wartungspersonal des Betreibers, das die Konfiguration des Systems und die zu erwartenden Änderungen vornehmen soll, über das einerseits notwendige technische Verständnis und andererseits über die Kenntnisse in der Mikrocontroller-Programmierung verfügt. Darüber hinaus muss der zeitliche Konfigurationsaufwand möglichst gering gehalten werden, um bei den Betreibern auf Akzeptanz des Systems zu stoßen. Daher bedarf es einer Lösung, die eine einfache und zeiteffiziente Konfiguration ermöglicht.

2 Funktionsweise des Assistenzsystems

Mobilität ist eine Grundvoraussetzung für die Teilhabe am sozialen Leben. Der barrierefreie Zugang ist inzwischen rechtlich verbrieft (§8 BGG 2006, §8 Abs.3 PBefG, 2013). Während sich die bauliche Infrastruktur und das Design von Fahrzeugen deutlich verbessert haben – dies wird insbesondere bei Niederflurfahrzeugen, Aufzügen sowie Blindenleitstreifen deutlich –, bestehen nach wie vor erhebliche Defizite bei den Orientierungshilfen sowie den Informationsangeboten.

Im SANBI-System sollen hilfebedürftige Benutzer durch in die Infrastruktur von Straßen und Verkehrsanlagen integrierte „Anker“ (Funkeinheiten) lokalisiert, ihre charakteristischen Eigenschaften identifiziert und entweder die jeweils notwendigen Assistenzfunktionen automatisch ausgelöst oder ein Dialog angeboten werden.

Es sind verschiedene Assistenzfunktionen denkbar. Darunter beispielsweise Navigations- und Warnhinweise, unterstützte Kommunikation an Informations- und Notrufanlagen, Informationen an das Fahrpersonal über die Anwesenheit von behinderten Personen, der automatisierte Ruf des Personenaufzugs, das Öffnen von Türen, die Erhöhung der Lautstärke von akustischen Zusatzanlagen

an Ampeln oder die automatische Anforderung von „Grün“ an der Fußgängerampel.

2.1 Einige Fallbeispiele

(1) Frau W. ist nach dem Tod ihres Mannes in die Nähe ihrer Tochter gezogen. In der neuen Umgebung kennt sie sich nicht gut aus. Auf den Wegen zum Arzt muss sie häufiger umsteigen und hat Angst, sich zu verfahren oder unnötig lange zu warten, wenn sie den Bus verpasst. Sie hat Mühe, in kurzer Zeit die Fahrplanaushänge zu lesen und zu verstehen. Das geplante System wird sie unterstützen, indem es bei Annäherung an eine Haltestelle automatisch ansagt, welche Linien dort verkehren und welche Fahrzeuge dort wann eintreffen.
(2) Herr K. hat Sorge, auf dem ausgedehnten Gelände des Universitätskrankenhauses die Orientierung zu verlieren. Er muss dort regelmäßig die Ambulanz eines Spezialisten besuchen und wird dort gelegentlich auch in andere Institute geschickt. Das System hilft ihm bei der Auswahl von geeigneten Routen und bestätigt ihm das Erreichen wichtiger Wegepunkte.
(3) Die Studentin Sabine ist mit ihrem Rollstuhl auch gerne ohne fremde Hilfe unterwegs. Das System unterstützt sie auf dem weitläufigen Universitätsgelände die geeigneten Routen (Rampen, Aufzuge, abgesenkte Bordsteine) zu finden, öffnet automatisch Türen und ruft den Fahrstuhl.
(4) Die blinde Frührentnerin Frau M. möchte für alltägliche Wege, die ihr eigentlich vertraut sind und auf denen sie sich sicher fühlt, nicht immer ihre Tochter um Hilfe bitten. Das Problem für sie bestand in der Überquerung der großen Straße auf dem Weg zum Einkaufen. Das System erhöht automatisch die Lautstärke der akustischen Zusatzanlage an der Ampel. Diese ist nun leicht zu finden und die lästige Suche nach dem Taster entfällt auch.
Das System liefert hilfebedürftigen Menschen – unabhängig davon, ob sie blind, sehbehindert oder durch ihr Alter eingeschränkt sind – weit mehr Autonomie. Ihnen wird Flexibilität, Unabhängigkeit sowie ein höheres Maß an Selbstvertrauen zurückgegeben [2].

2.2 Komponenten des Systems

Im Normalfall soll die Interaktion am Endgerät aufgrund der benutzeradaptiven und kontextsensitiven Systemsteuerungen auf wenige Eingaben begrenzt bleiben, sodass sich der Benutzer auf die verkehrliche Situation konzentrieren kann. Daher wird besonderer Wert auf die Auswahl automatisierter Funktionen sowie die Kombination mit optional abrufbaren Diensten gelegt.
Für die technische Umsetzung folgt daraus, dass in dem Assistenzsystem unterschiedliche Komponenten zusammenwirken, die entweder von dem „Anker“ gesteuert werden oder diesen mit Informationen versorgen. Beispiele für typische Komponenten, Features und deren Beziehungen zu einander werden in **Bild 1** dargestellt.

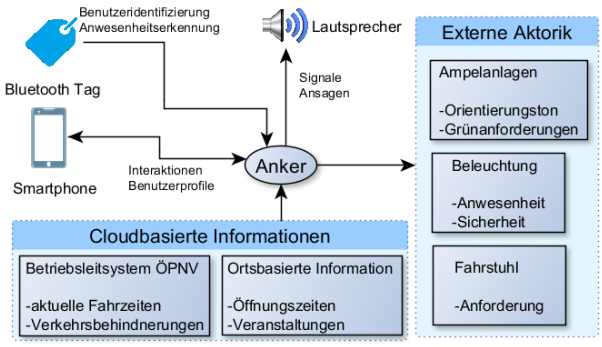

Bild 1 Komponenten und Features des Assistenzsystems

3 Ansatz von der graphischen Regel hin zur Systemkonfiguration

Die oben skizzierten Input- und Output-Beziehungen zwischen den Systemkomponenten können für die Steuerung des Systemverhaltens in Form von Regeln systematisiert werden. Dadurch bietet sich auch die Möglichkeit, das Verhalten des Assistenzsystems bei Vorliegen von zuvor definierten Triggern zu beschreiben – und zwar unter einschränkenden räumlich-zeitlichen Randbedingungen und in Bezug auf variierende Benutzereigenschaften. So kann in Regeln formuliert werden, in welcher räumlichen Entfernung vom Anker (z.B. 4 – 10 m), welche Systemfunktionen (z. B. Orientierungston oder Sprachansage) für einen blinden oder einen gehbehinderten Benutzer bei Verzögerungen von Abfahrtzeiten (zeitliche Randbedingung) ausgelöst werden sollen. Die Parametrierung der in den Regeln angelegten Attribute wird in SANBI über einen graphischen Editor möglich gemacht. Die Übertragung der Regeln erfolgt über ein Bluetooth Low Energy (BLE) - Profil. Ein Regelparser im Anker prüft zur Laufzeit kontinuierlich die in den Regeln enthaltenen Trigger und aktiviert die zugehörigen Aktionen.

3.1 Systemkonfiguration mit Hilfe von Regeln

Zu den wichtigsten Funktionen des SANBI-Systems gehören das Aussenden von Sprachansagen und Orientierungshinweisen, die Anforderung der Ampel-Grünphase sowie diverse Komfortfunktionen, wie beispielsweise ein automatisierter Fahrstuhlruf.
Das Verhalten der Anker wird durch die Konfiguration determiniert. Im vorausgegangen Projekt BUS-ID [1] wurden diese Konfigurationen als fester Programmcode im Mikrocontroller implementiert. Die Nachteile dieser statischen Konfiguration, sind, dass das Verhalten nicht flexibel und die bedingten Anweisungen schlecht lesbar sowie demzufolge nur mit Mehraufwand veränderbar waren.
Ein flexibleres Verhalten und eine einfachere Konfiguration sollen durch die Verwendung von Regeln erfolgen. Die deterministischen Regeln sind dabei der Aussagen-

oder Prädikatenlogik regelbasierter Systeme angelehnt. Einfache formalisierte Konditionalsätze der Form Wenn (**if**) *A* dann (**then**) *B* bieten einen guten Kompromiss zwischen einfacher syntaktischer Form und formaler Beschreibung [5]. Solche Regeln haben sich in Produktionssystemen zur Planung und Steuerung etabliert.
Zur Konfiguration eines jeden Ankers wird eine Zone definiert, die der Entfernung des Nutzers zum Anker entspricht. Bewegt sich der Nutzer in die Zone hinein oder hinaus, ist dies ein Trigger für ein spezifisches Ereignis. Vom Kern her entspricht die Konfiguration der Anker einer 1:1 Verknüpfung von Triggern mit Aktionen. Aktionen repräsentieren Funktionen des Ankers, welche dieser bereitstellt. Als Option können beliebig viele Zonen mit jeweils unterschiedlichen Verknüpfungen erstellt werden.

Eine Regel besteht aus der Beziehung einer Aktion zu einem Trigger, sowie zusätzlich optional aus Bedingungen, einem Benutzerprofil sowie einem Zeitkontext. Bedingungen beschreiben dabei situationsabhängige Filter der Assistenzfunktion. Einige Aktionen müssen verzögert zum Auftreten des Triggers ausgeführt und ggf. mehrfach wiederholt werden. Der Zeitkontext definiert die Ausführung der Aktion und verändert deren Zeitpunkt sowie die Wiederholung der Aktion. Intern im Mikrocontroller führt dies dann zu den entsprechenden Timern der Aktion **Bild 2** zeigt den strukturellen Aufbau einer Regel.

Die Regeln sind gut geeignet, um flexibles Verhalten von Ankern zu ermöglichen, solange der Funktionsumfang (Anzahl und Art der möglichen Aktionen) und der Ereignisse (Trigger) gleich bleibt. Das Hinzufügen neuartiger Aktionen und Trigger ist zwar möglich, jedoch mit zusätzlichem Implementierungsaufwand. Für gleichartige Typen von Ankern können zur erstmaligen Konfiguration Templates (vorkonfigurierte Regelwerke) verwendet werden.

3.2 Unterstützung der technischen Mitarbeiter durch graphische Regeln

Der vorgestellte Ansatz hat zum Ziel, Mitarbeiter mit einem regelbasierten graphischen Konfigurationstool zu unterstützen, welches an der Qualifikation des Betriebs- und Wartungspersonals angepasst ist. Bei technischen Mitarbeitern eines Verkehrsunternehmens oder einer Kommune werden keine Programmierkenntnisse, z. B. in C, vorausgesetzt.

J. Jasperneite et al. beschreiben allgemein, dass komplexe technische Systeme durch die Reduktion der Systemkomplexität zu beherrschen sind, indem die Elemente des Systems und deren Beziehungen untereinander reduziert werden. Oder die gegebene Komplexität wird zugelassen und dem Menschen zugleich eine Unterstützung zur Bedienung des Systems angeboten[6].
Das Paradigma der graphischen Programmierung (Visual Programming Language, VPL) kann den Benutzer bei der Programmierung oder Konfiguration von technischen Systemen deutlich unterstützen. Ein klassisches Beispiel ist die Datenflussorientierte Programmierung mit der G-Code (LabView) Engine der Firma National Instruments. Die graphische Darstellung erfolgt als Blockdiagramm, bei dem die Verbindungen zwischen den Ein- und Ausgängen der Objekte den Datenstrom, den Signalfluss oder die Beziehung modellieren Sie wurde entwickelt um vereinfacht messtechnische Systeme zu entwerfen.
Ein weiteres Beispiel ist das Automatisieren von Prozessen mit dem Lightning Proccess Builder der Online Plattform von SalesForce. Hier können vereinfacht graphische Regeln erstellt werden, die zeitabhängige Aktionen auslösen, sowie Prozesse automatisieren werden, die aus mehreren Schritten bestehen [7]. Der Funktionsumfang des Assistenzsystems ist verglichen mit dem Prozessgenerator geringer.

3.3 Graphisches Tool zur Konfiguration des Regelwerks

Als graphisches Werkzeug wurde ein dynamisches browserbasiertes Visualisierungs-Tool ausgewählt – das Open-Source Tool Tiddly und das Plugin TiddlyMap [8], welches ursprünglich für die Erstellung von MindMaps entwickelt wurde. Das plattformunabhängige Tool läuft innerhalb einer HTML Seite und basiert auf Java sowie der graphischen Java Bibliothek vis.js. Das HTML Tool kann dabei ohne Beschränkungen angepasst werden.
Bild 3 zeigt die graphische Darstellung einer Regel in Form eines Graphen. Über den Editor des Tools können neue Objekten (Aktion, Trigger, Bedingung, Profil, Zeitkontext) hinzugefügt und bearbeitet werden. Praktisch werden die Objekte über Kanten verbunden. Werden eine Aktion und ein Trigger in Beziehung miteinander gebracht, bilden sie eine einzelne gültige Regel ab. Das Blatt kann beliebig viele dieser Beziehungen aufnehmen,

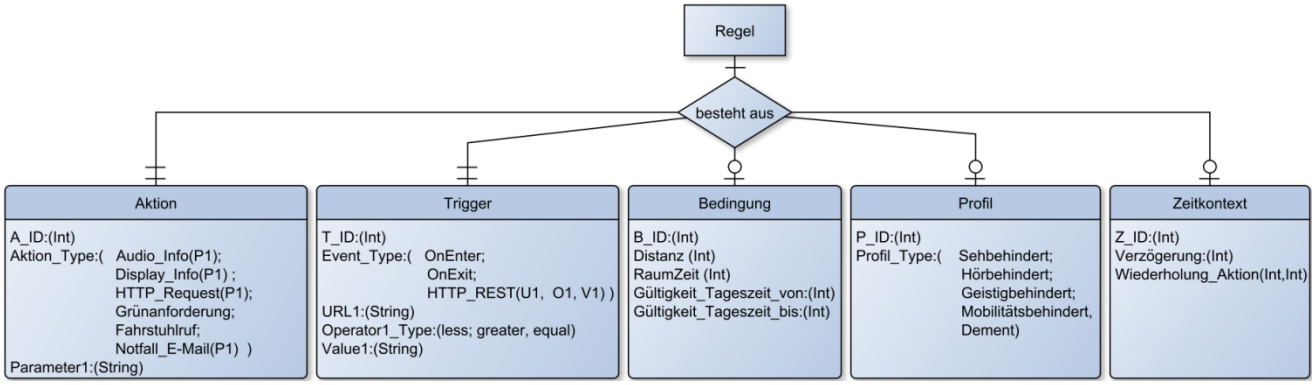

Bild 2 Auszug des Aufbaus einer Regel.

sodass das gesamte Blatt das Regelwerk eines Ankers repräsentiert. Das Konzept erlaubt sich überschneidende Regeln. Ein Beispiel wäre ein Duplikat einer Regel mit unterschiedlichen Bedingungen. Formal entspricht dies einer UND-Verknüpfung der Regel. Offen ist daher eine syntaktische Prüfung auf die Schnittmenge zwischen den gleichartigen Aktion-Trigger-Beziehungen und deren optionalen Bedingungen. Inhaltlich sollten solche Überschneidungen vermieden werden, da sie zu Unübersichtlichkeit führen.

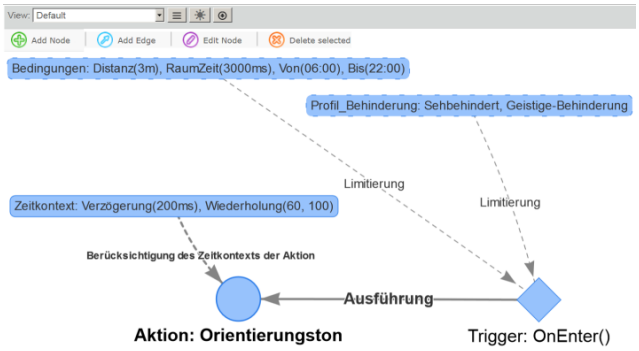

Bild 3 TiddlyMap Beispiel für eine Regel zur Steuerung eines Orientierungstons. Die abgebildete Regel beschreibt die Erzeugung eines Orientierungstons sobald sich eine Person mit mobilem Benutzergerät >3 sec lang innerhalb eines 3-m-Radius um den Anker befindet.

Das Blatt verfügt über eine Export-Funktion, welche am Ende des Bearbeitungsschrittes ausgeführt wird. Für den Export der Regeln wurde die Export-Funktion erweitert. Im ursprünglichen Zustand ließen sich die Regel-Graphen exportieren, jedoch ohne die zusätzlichen Attribute des Knotens, welche für das nachfolgende Vorgehen wichtig sind.

```
  "edges": [{
     "id": "d4541aa2-6939-4c15-badc-1394b8cc721d",
     "from": "d4ac5850-f8f6-433f-97fc-113b0da86f56",
     "to": "e1dec8f9-6b7c-4233-ab93-9aa120453e9c",
    },  ...    ]
"nodes": [{
    "id": "d4ac5850-f8f6-433f-97fc-113b0da86f56",
      {
        "type": "condition",
         "von": "0600",
        "bis": "2200",
        "distanz": "3",
        "raumzeit": "3000",
       },  ...   } ]
```

Bild 4: Auszug aus einer Regel im exportierten JSON Format

Die exportierten Regeln werden als proprietäres JSON-Dateiformat gespeichert. Die Beziehungen zwischen den Knoten und Kanten werden durch id's abgebildet. Das Attribut *type* definiert dabei die Art des Objekts.
Bild 4 zeigt einen einzelnen exportierten Knoten vom Type *Condition* mit seiner dazugehörigen Kante. Die Kanten repräsentieren die Verbindungen zwischen den Objekten.

Weiterhin offen ist die Erweiterung des Tools um eine passende Import-Funktion. Hiermit ließen sich dann Templates für verschieden Typen von Ankern bereitstellen.

3.4 Technische Übertragung der Regeln über die Bluetooth Schnittstelle

Anschließend folgt ein Prozess, in dem die erstellten und exportierten Regeln in den Anker übertragen werden müssen. Eine kabelgebundene Übertragung aus dem vorrausgegangen Projekt hat sich als unkomfortabel erwiesen. Zum einen sind die Anker in der Infrastruktur (z.B. Lichtsignalanlage) nicht direkt zugänglich, weshalb eine Leiter benötigt wird, und zum anderen muss das Gerät geöffnet werden, um an die Schnittstelle zu gelangen. Bluetooth Funkverbindungen unter technischen Geräten nehmen im Alltag deutlich zu.
In allen Anwendungen, in denen Zuverlässigkeit im Sinne der Echtzeitverarbeitung, Datenmengen und Sicherheit nicht die höchste Priorität haben, sind Bluetooth Funkverbindungen vorteilhaft. Bei Hardwarekosten für Bluetooth-Module unter einem Euro und geringen Installationskosten können vernetzte Anwendungen zu einem überschaubaren Investitionsaufwand realisiert werden. BLE besitzt das Potential einer Schlüsseltechnologie, um das *Internet der Dinge* im Alltag zu realisieren. Insbesondere die Tatsache, dass diese Technologie in jedem aktuellen Smartphone vorhanden ist, kennzeichnet sie als zukünftige Schnittelle für die Übertragung von geringen Datenmenge und Sensorwerten. Die Möglichkeit, eigene GATT (Generic Attribute) Profildienste über BLE anzubieten, macht diese Schnittstelle universell mit einem geringen Entwicklungsaufwand und klarer Kommunikationsstruktur.

Bild 5 links zeigt den Aufbau der einzelnen Layer aus dem BLE Standard. Der Standard benutzt zum physikalischen Verbindungsaufbau das GAP (Gerneric Access Profile) und ist als Zugriffssteuerung zu verstehen. Nach dem Bekanntmachen der Netzwerkteilnehmer durch das GAP Profil kann das GATT Profil zum Austausch von Daten zwischen den beiden Teilnehmern verwendet werden.

Der Standard ist auf die allgemeine und energieeffiziente Übertragung kleinerer Datenmengen, wie z.B. Sensordaten (<100 kbits/s) ausgerichtet. Ein deutlicher Vorteil der BLE-Verbindung gegenüber der klassischen Bluetooth-Verbindung ist der deutlich schnellere Verbindungsaufbau zwischen beiden Teilnehmern. In Abhängigkeit von Connection Intervall (200ms) und Slave Latency (3 Slots) liegt der bei maximal 800ms. Der Verbindungsaufbau einer klassischen Bluetooth Verbindung ist in der Praxis deutlich langsamer (>5 Sekunden).
Bild 5 rechts zeigt die Bestandteile eines GATT Profils. Einzelnen Services werden Characteristics zugeordnet, welche die Values sowie die Identifier (UUID) enthalten. Außerdem kann ein Descriptor hinzugefügt werden. Diese Services werden von einem Teilnehmer im BLE-

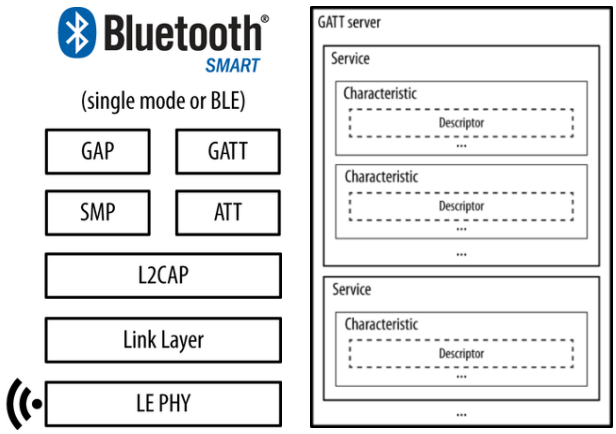

Bild 5: (links) Bluetooth Low Energy **(**BLE) Kommunikationsschichten; **(rechts)** Aufbau des GATT (Generic Access) Profils Quelle: Getting Startet with Bluetooth Low Energy, O'Reilly Media, Inc.

Netzwerk angeboten und können von einem anderen BLE Teilnehmer gescannt werden. So ist es möglich, eigene Services zu kreieren. Danach kann entschieden werden, ob der Scanner auf diese Services reagieren möchte. Auf diese Weise können zwei Bluetooth LE Teilnehmer einfach Daten austauschen. Passend dazu hat die Special Interest Group (SIG), welche für den BLE-Standard verantwortlich ist, allgemeine GATT Profile spezifiziert. Eine Zusammenfassung dieser Profile ist auf der Homepage [4] zu finden. Die Komplexität und der Entwicklungsaufwand, um Signale über das GATT Profil zu übertragen, sind dabei gering. Ein Großteil der Funktionalität wird über den BLE-Stack und das Betriebssystem der Hardware und Bibliotheken bereitgestellt.
Die Regeln aus der exportierten JSON-Datei werden in einem Folge-Prozess von einem Tool geparst und über die BLE-Schnittstelle bereitgestellt. Dabei werden die Regeln sequentiell über die BLE Schnittstelle an den Anker übertragen. Das Tool wurde noch nicht realisiert. Für Tests wurden die Regeln jedoch manuell über die BLE Schnittstelle entsprechend der enthaltenen Konfiguration aus dem JSON Dateiformat übertragen.

Zur drahtlosen Übertragung des Regelwerkes an die Anker werden die einzelnen Regeln unter zur Hilfenahme von 10 Attributen übertragen. Die Anzahl der Attribute ergibt sich aus der Anzahl aller möglichen Parameter-Knoten aus dem JSON Dateiformat. **Bild 4** zeigt diesen Aspekt. Der Objektknoten *Condition* besitzt die vier Parameter „von, bis, distanz, raumzeit". Ergeben sich in Zukunft weitere Parameter, so müssen die Attribute angepasst werden. Durch ein weiteres, zusätzliches definiertes Steuerattribut *TransferComplete* wird dem Mikrocontroller signalisiert, dass zurzeit eine einzelne Regel vollständig an den 10 Attributen der GATT Schnittstelle anliegt und somit übertragen wurde.
Sofern der Mikrocontroller die einzelne Regel verarbeitet, setzt er das Steuerattribut *TransferComplete* zurück und der Prozess der Übertragung weiterer Regeln kann wiederholt werden.

Grundsätzlich könnte die gesamte Kommunikation mit nur zwei Attributen nach dem klassischen Schema der seriellen Übertagung (RX, TX Kanal) realisiert werden. Einerseits wäre dann ein weiterer Parser notwendig der Signale oder Kommandos aus dem Datenstrom auswertet. Andererseits GATT Verbindung sind viel umfangreicher und bieten größere Vorteile. Der GATT Server kann die Zugriffsfunktionen der Attribute genauer spezifizieren. Neben den Operation *READ* und *WRITE* wurde die Operation *NOTIFY* eingeführt. Diese Operation implementiert das *puplish-subscribe Muster*. Hat sich ein Client auf das Attribut registriert, wird automatisch eine Information ausgesendet, sofern sich der Wert ändert. Dies führt zu weniger Funkkommunikation und spart Energie. Konkret wurde das Steuerattribut *TransferComplete* mit dieser Zugriffsoperation versehen. Das aufwendige Pullen der benötigten Information kann dadurch vermeiden werden.

Ein weiterer Vorteil sind Deskriptoren der Attribute. Sie machen die BLE Schnittstelle für den Menschen einfacher lesbar. Analog dem Ansatz von Felfernig[9] könnten sie später dafür genutzt werden, um die Semantik der Schnittstelle zu erhöhen und ggf. flexibler auf verschiedene Firmware Versionen der Anker oder auf Änderungen der Identifier (UUID) der Attribute zu reagieren

3.5 Implementierung und Verarbeitung der Konfiguration auf dem Anker

Die Übertragung der Informationen zur Konfiguration eines Ankers ist ein mehrstufiger hierarchischer Prozess. Der Anker bietet den in 3.4 dargestellten Service an.
Das Steuerattribut *TransferComplete* gibt ein Signal, sofern eine Regel übertragen wurde. Anschließend fügt der Anker die Regel in eine Datei ein. Jede Zeile dieser Datei entspricht den jeweiligen Attributen einer Regel. Anschließend wird das Steuerattribut zurückgesetzt.
Der Anker verarbeitet die Signale der Empfangsfeldstärke (RSSI) des funkbasierten Benutzergerätes, die mit der Entfernung des Senders korreliert, zu Ereignissen. Als Beispiel wertet er die Signale zu den Ereignissen *„Benutzer kommt"* und *„Benutzer geht"* aus. Sofern der Anker ein Ereignis generiert, verarbeitet der Mikrocontroller die Datei mit allen enthaltenen Regeln sequentiell.
Dieser Ansatz ist notwendig, wenn eine unbekannte Anzahl von Regeln auf dem Mikrocontroller verarbeitet werden soll. Der Grund hierfür ist das statische Speichermanagement der Mikrocontroller.
Grundsätzlich kann der Anker eine Reihe von Ereignissen (Trigger) erkennen. Sollen die Anker um neuartige Ereignisse erweitert werden, müssen diese mit entsprechendem Aufwand im Anker implementiert werden. Äquivalent verhält sich dies bei neuartigen Aktionen bzw. Assistent Funktion der Anker. Aktionen haben zusätzlich weitere Randbedingungen: Das Rufen eines Fahrstuhls benötigt die Verbindung zur Fahrstuhlsteuerung, die Grünanforderung erfordert eine Schnittstelle zur Blindenleitakustik der Ampel usw.

Ein weiteres Feature von BLE ist, dass neue Firmware funkbasiert auf die Anker aufgespielt werden kann. So kann der Anker im Rahmen der vorgegeben angeschlossenen Hardware nicht nur in seiner Funktionalität vollständig neu konfiguriert werden, sondern auch während des Betriebes mit neuer Funktionalität versehen werden. Solange der Anker keine Fehlfunktionen aufzeigt und keine neue Hardware integriert wird, gibt es keinen Grund weshalb er zur Wartung geöffnet werden muss, dies minimiert ebenfalls den Aufwand des Betreibers.

3.6 Sicherheit

Grundsätzlich ist der Zugriff auf Services und Characteristics der Ankereinheit immer möglich. Um den möglichen Missbrauch durch Dritte zu verhindern, wurden einige Lösungswege betrachtet. Der BLE-Standard hat die Möglichkeit, über Sicherheitsfeatures nur bestimmten Geräten das Verbinden (Whitelist) oder das Abrufen von Services/Characteristics zu erlauben.
Passwörter sowie Whitelists gewährleisten jedoch nur partielle Sicherheit. Werden die Whitelists oder Passwörter (siehe Telekom DSL Router Standard Passwort) einmal im Internet bekannt, lässt sich dies nur über eine neue Firmware beheben.
Letztendlich obliegt es dem zukünftigen Betreiber, wieviel Sicherheit für sein Assistenzsystem notwendig ist, da hierfür ein entsprechender Aufwand nötig ist. Für die laufende Projektarbeit wurde eine vereinfachte Sicherheitsmethode gewählt. Der Dienst wird für 120 Sekunden von der Ankereinheit angeboten, sofern ein definiertes Service BLE-Signal vom Benutzer (z.B. über eine App) ausgesendet wird. Die spezielle BLE UUID fungiert als Schlüssel. Die UUID ist dabei an eine Verbindung zwischen den Teilnehmern gebunden, wobei die Verbindung gesichert ist. Ein einfaches Mitlauschen (sniffen) des Schlüssels ist daher nur mit spezieller Hardware und höherem Aufwand möglich.

4 Fazit

Der in diesem Beitrag beschriebene Ansatz zeigt die Vorteile eines unterstützten Prozesses zur Erstellung von Verhaltensregeln für technische Assistenzeinheiten. Die Vorteile von graphischen Editoren für regelbasierte Systeme im Vergleich zu bedingten Anweisungen in C werden deutlich aufgezeigt.
Weiterhin wird ein halbautomatisierter Prozesses zur Übertragung der Regeln auf Mikrocontrollern aufgezeigt. Die Vorteile funkbasierte Konfiguration wurden dargestellt. Als Resultat ergibt sich weniger Aufwand und Komplexität bei der Inbetriebnahme und in der Wartung eines solchen technischen Systems.

Bei der Erstellung eines auf diese Art unterstützten Prozesses ist jedoch ein Mehraufwand notwendig. Da sich das Projekt anwendungsnah bewegt, und eine Anforderung lautet den Wartungs- und Installationsaufwand für zukünftige Betreiber solcher Systeme zu minimieren, wurde nicht weiter untersucht ab wann sich der Mehraufwand bei der Entwicklung rechnet.

Die Projektpartner versprechen sich dadurch eine erhöhte Akzeptanz bei zukünftigen Betreibern. Insbesondere dann, wenn keine Gewinne durch Installation und Betrieb zu erwarten sind. Dies ist ein wesentlicher Faktor zur Verbreitung technischer Assistenzsysteme zur Unterstützung hilfebedürftiger Menschen. Die Akzeptanz bei zukünftigen Benutzern, insbesondere blinde und sehbehinderte Nutzer, hat das System mehrfach unter Beweis gestellt[1][2].

Die Grenzen des dargestellten Ansatzes liegen in der bedingten Erweiterbarkeit um weitere Funktionalität. Was nicht im Regel-Editor vorgesehen ist, kann auch nicht im Regel-Parser auf dem Anker verarbeitet werden. Werden neue Funktionen hinzugefügt, so müssen evtl. der Regel-Editor, die BLE Schnittstelle sowie die Firmware im Anker angepasst werden.

5 Literatur

[1] R. Meister, A. Fay, D. Cory, C. Ehring: Bus-ID: Orientierung für Blinde an der Haltestelle. In: atp edition 7-8/2013, S. 36-43.

[2] R. Meister et al.: RFID-basierte Unterstützung der Mobilität blinder und sehbehinderter Menschen im öffentlichen Raum - *Supported Mobillity for Blind and Low Vision Persons based on RFID*. Tagungsband AUTOMATION, Baden Baden, 2013.

[3] T. Leimbach, B. Jost, U. Petersen, J. Barding, S. Hartig, Roberta-Grundlagenband EV3, Fraunhofer Verlag, 2014.

[4] https://www.bluetooth.com/specifications/adopted-specifications, [Stand: 25.02.2016]

[5] C. Beierle et al.: Regelbasierte Systeme. Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen. Vieweg+Teubner 2008, S. 72-97

[6] J. Jasperneite, O. Niggemann: Systemkomplexität in der Automation beherrschen: Intelligente Assistenzsysteme unterstützen den Menschen. In: atp edition 9/2012, S. 36-44.

[7] P. Weinmeister: Practical Salesforce.com Development Without Code: Producing Advanced Automation with Visual Workflow. Apress 2015

[8] www.TiddlyMap.org, [Stand: 25.02.2016]

[9] A. Felfernig et al.: Semantic Configuration Web Services in the CAWICOMS Project. The Semantic Web In: Proceedings ISWC 2002. First International Semantic Web Conference Sardinia, Italy, June 9□12, 2002