

Bedarfsanalysator zur Bestimmung eines Assistenzsystems (PATRONUS)

Bedarfsanalysator for an assistance system assembly (PATRONUS)

Dr.-Ing. J. Denecke¹, T. Felix¹, V. Pfister², B. Steiner²

¹Fraunhofer Institut für Produktionstechnik und Automatisierung IPA, Nobelstraße 12, 70569 Stuttgart, thomas.felix@ipa.fraunhofer.de, julia.denecke@ipa.fraunhofer.de

²BruderhausDiakonie, Ringelbachstraße 211, 72762 Reutlingen, barbara.steiner@bruderhausdiakonie.de, verena.pfister@bruderhausdiakonie.de

Kurzfassung

Durch den sich verstärkenden demographischen Wandel steigt auch die Anzahl der Menschen die sich wünschen bis in ein hohes Alter im Eigenheim zu verbleiben. Moderne Technik kann Senioren dabei unterstützen und ermöglicht es den Angehörigen und auch dem Pflegepersonal die sich ergebenden Herausforderungen zu meistern. Leider befindet sich der aufstrebende Markt der Heimautomatisierung und der Pflorgetechnik in einem sehr fragmentierten Zustand, eine Konsolidierung ist kurzfristig nicht zu erwarten. Für eine Person, die sich nicht täglich mit den Neuheiten und Möglichkeiten des Marktes auseinandersetzt, fällt es sehr schwer, ein aufeinander abgestimmtes Gesamtsystem zusammen zu stellen, das gleichzeitig auch die Voraussetzungen für Sicherheit, Zuverlässigkeit und Komfortfunktionen bietet. Um dieses Problem anzugehen wurde eine Software, der Bedarfsanalysator, entwickelt, das es ermöglicht, um solche Komplettsysteme mit wenig Hintergrundwissen einfach erstellen zu können.

Abstract

Because of the demographic change the number of people who wish to remain in their own house until a high age is ever increasing. Modern technology helps them and is the enabler for relatives and nursing staff to tackle this challenge. The emergent market of home automation and care technology is unfortunately in a very fragmented state and an improvement is not expected at least in the short-term. It is really hard for people who are not constantly keeping themselves up to date with news and latest developments in this area, to put together a complete system which accounts for security, reliability and comfort. To circumvent this problem software was developed, the so called Bedarfsanalysator, to enable everyone without detailed knowledge to put together such a complete system.

1 Problemstellung

Laut einer Vorhersage des Bundesinstitutes für Bevölkerungsforschung liegt der Anteil der jungen Menschen unter 20 Jahren, im Jahre 2020 bei ca. 18%. Menschen über 65 würden dann 25% der Bevölkerung ausmachen [1]. Mit einem sich verstärkendem Trend hin zu einer älteren Bevölkerungsstruktur verschärft sich der demographische Wandel. Es existiert ein immer stärker werdender Bedarf ältere Menschen so lange wie möglich in ihrer Häuslichkeit zu versorgen und zu betreuen. Um dies gewährleisten zu können, nimmt der Bedarf an passgenauer Beratung, sowie Versorgung und Unterstützung zu. Zeitgleich erleben wir den aufstrebenden Markt der Heimautomatisierung, das Smart Home. Intelligente Technik, Sensoren und Aktoren, ziehen in die Häuslichkeit ein und versprechen uns den Alltag einfacher zu gestalten, zum Beispiel indem sie uns bei alltäglichen Aufgaben unterstützen und ggf. überwachen, sofern dies gewünscht ist. Ein Netzwerk aus intelligenten Sensoren und Aktoren wird in der Literatur unter dem Stichwort Internet of Things geführt [2]. Über-

wachung der eigenen Gesundheit spielt auch in der Quantified Self Bewegung eine Rolle, um die Leistungsfähigkeit des Körpers zu erhalten und auf Dauer zu optimieren. Es liegt also nahe, dass auch im Bereich der unterstützenden Pflege, speziell der Altersvorsorge, neue innovative Produkte in der nahen Zukunft auf dem Markt erscheinen werden. Die Herausforderung besteht jedoch im individuellen Anpassen der vorhandenen Technik an die unterschiedlichen Vorgaben und Lebenssituationen der einzelnen Personen. Nur so ist eine optimale Abdeckung der persönlichen Bedürfnisse gewährleistet. Bedingt durch den sich sehr dynamischen entwickelnden Markt der Heimautomatisierungslösungen, welcher fast tägliche neue Produkte hervorbringt, wird vom Betreuer bzw. Berater ein hoher Zeitaufwand abverlangt. Um technische Systeme sinnvoll einsetzen zu können, muss er sich stets weiter bilden um immer auf dem aktuellen Stand der Technik zu bleiben. Ein Zeitaufwand, der gerade im Bereich der Pflege und Beratung, nur sehr schwer oder gar nicht zu leisten ist.

2 Durchgeführte Analysen

Im AAL-Projektvorhaben von „PATRONUS“ wurde ein umfassendes und individuell anpassbares Assistenzsystem entwickelt, welchem die hier aufgeführten Ergebnisse als Grundlage dienen. Dem Projekt zugrundeliegend ist das Modell der selektiven Optimierung und Kompensation nach Baltes [3] sowie ein Assistenzmodell das allgemeine Bedürfnisse älterer Menschen und allgemeine Angebote in unterschiedlichen Bereichen wie z.B. Kommunikation, Hauswirtschaft, Pflege, medizinische Versorgungen und Technik in einer Datenbank erfasst. Das Assistenzmodell kann anhand dieser Grundlage über einen Bedarfsanalysator die individuellen Bedürfnisse bezüglich der Lebens- und Gesundheitsumstände einer Person erfassen, z.B. automatische Notfallerkennung, offene/geschlossene Fenster, passende Angebote wie Erinnerungsfunktionen für Termine, Lieferservice für Lebensmittel, Fahrdienst usw. der Person zuordnen und entsprechende Aktivitäten oder Maßnahmen einleiten.

Zur Erhebung wurden Bedarfsanalysen und Literaturrecherchen durchgeführt, deren Ergebnisse in eine Datenbank integriert wurden. Diese Datenbank liefert Informationen an den Bedarfsanalysator und dient als dessen Grundlage. Insgesamt wurden 21 Dienstleister aus den Bereichen Pflege, Medizin, Anwender und Technik interviewt. Ergänzend wurde eine Gruppendiskussion mit 15 Senioren zum Thema Barrierefreiheit und Technik im Alter durchgeführt. Ziel der Befragung war die Gewinnung von Kenntnissen wie eine Assistenzplattform (technisch) gestaltet sein und was sie umfassen muss, dass sie für ältere Menschen und Dienstleister einen Mehrwert bietet und als unterstützend empfunden wird.

Der Aufbau des Bedarfsanalysators ist in Bild 1 dargestellt.



Bild 1 Aufbau Bedarfsanalysator

3 Bedarfsanalysator

Der Bedarfsanalysator unterstützt die Berater in mehrerlei Hinsicht: zunächst soll der Bedarf der jeweiligen Person ermittelt werden. In einem weiteren Schritt wird aus dem Bedarf eine Sammlung aus, für das Patronussystem compatible, Komponenten erstellt werden. Diese Komponentenliste wird dem Benutzer angezeigt und er kann im letzten Schritt Korrekturen daran vornehmen. Das Ergebnis ist eine übersichtliche Liste aus technischen Komponenten, aber auch Dienstleistungen, welche die Person im Alltag unterstützen würden.

Durch das lokale Abspeichern der Erhebung soll der Benutzer in die Lage versetzt werden nach einer gewissen

Zeit die Analyse erneut zu starten und einen veränderten persönlichen Bedarf erneut zu evaluieren.

3.1 Softwarekonzept

Zur schnellen Entwicklung eines nutzbaren Programmes wurde auf die Microsoft .NET Plattform mit der Programmiersprache C# zurückgegriffen. Dadurch war es möglich viele etablierte Bibliotheken zu nutzen welche die Entwicklungszeit stark verkürzten. So wird zum Beispiel das EntityFramework [4] eingesetzt um eine Anbindung an die MySQL Datenbank herzustellen. Durch den Einsatz dieser serverbasierten Datenbanktechnologie ist es möglich die zentrale Datenbasis online zu verwalten und zu ergänzen. Dies führt dazu, dass man zur Verwendung des Programmes auf eine Internetanbindung angewiesen ist. Man könnte auf eine lokal angelegte Datenbank zurückgreifen, verliert dadurch aber den Vorteil der gemeinsamen Datenbasis. Gerade diese Online-Datenbasis stellt sicher, dass ohne größeren Zusatzaufwand, die neusten Informationen allen Softwareinstallationen jederzeit zur Verfügung gestellt werden können.

Ein weiterer, wichtiger Aspekt der Architektur spiegelt sich im Trennen der Benutzeroberfläche von der Programmlogik wieder.

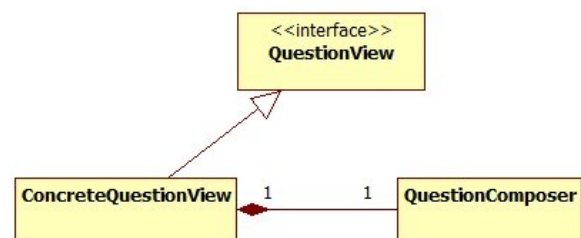


Bild 2 Auslagerung der Geschäftslogik in ein Composer Objekt. Interaktion mit der Anzeige über eine View Implementierung.

Dieses Konzept ermöglicht es relativ leicht die komplette Benutzeroberfläche auszutauschen, die Anwendung mit einer Weboberfläche zu versehen und sie damit im Netz verfügbar zu machen, sollte dies gewünscht sein. Die Anwendungslogik selbst bleibt davon unberührt. Das verwendete Design ist im UML Diagramm in Abbildung 2 visualisiert.

3.2 Datenaufnahme

Die Datenaufnahme zur Bestimmung des Bedarfs gliedert sich in zwei Abschnitte: Zunächst müssen Hintergrundinformationen über das wohnliche Umfeld gesammelt werden. Einige Anforderungen können nur durch den Abgleich mit dieser Datenbasis und folglich durch konkrete Komponenten aufgelöst werden. Ein gutes Beispiel ist der Bedarfsfall einer Einbruchserkennung. Um die Anzahl der Bewegungsmelder ermitteln zu können muss die Wohnfläche in m² bekannt sein.

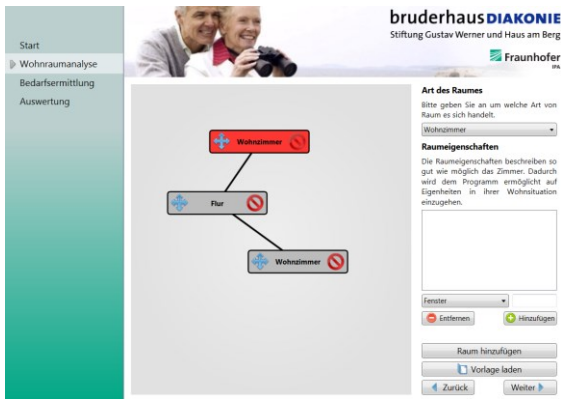


Bild 3 Mit Hilfe des Wohnraumeditors werden grundlegende Informationen über die Wohnung abgefragt.

Im ersten Schritt werden also Informationen über das Wohnungsumfeld der Person gesammelt. Ein intuitiver Wohnraumeditor steht bei diesem Erfassungsprozess den Beratern zur Seite. In diesem Editor werden Wohnräume in Anlehnung an einen Graphen erstellt (siehe Abbildung 3). Die Verbindungslinie stellt einen Durchgang zwischen zwei Räumen dar. Für jeden Raum lassen sich Attribute festlegen die später als Datenbasis zum Vervollständigen der Anforderungsdaten herangezogen werden. Je nach Größe und Ausstattung der Wohnung (wie zum Beispiel die Zahl der Fenster und Art der Räume), unterscheiden sich die, zum Erfüllen der Bedürfnisse, benötigten Produkte.

Zur Feststellung des konkreten Bedarfs einer Person wird im zweiten Schritt ein Fragenkatalog genutzt und abgearbeitet. Die Fragen wurden zunächst im Rahmen des Projektes in Abhängigkeit der vorhandenen Sensoren und Aktoren formuliert. Dabei sollen sie möglichst breit die Lebensumstände einer Person abfragen und je nach gewählter Antwort einen Bedarf anzeigen, der mit der vorhandenen Hardware abgedeckt werden kann. Die Oberfläche einer solchen Befragung ist in Abbildung 4 zu sehen.

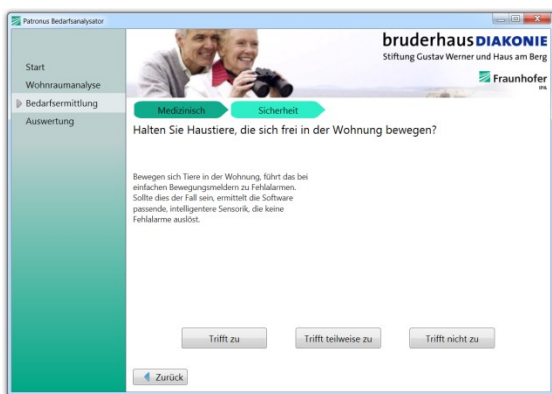


Bild 4 Der Benutzer wird gebeten Fragen zu beantworten.

3.3 Datenmodell

Den Antwortmöglichkeiten der Fragen sind über eine frei wählbare Logik mit einer Anforderung (Requirement) verknüpft. Wird eine Frage entsprechend dieser Logik so beantwortet, dass eine Anforderung aktiv wird, werden die benötigten Einschränkungswerte (in der Datenbank

constraint values genannt) bestimmt. Diese sind entweder statisch in der Datenbank hinterlegt, setzen sich aus der Eingabe des Benutzers auf Fragenoberfläche zusammen oder werden aus der Datenbasis des Wohnungsmodells komplettiert. Es ist daher durchaus möglich, dass zwei Anforderungen von unterschiedlichen Fragen zur identischen Einschränkung aber mit unterschiedlichen Werten führen. Der Begriff „Einschränkung“ ist hier als Zielvorgabe für das System zu verstehen, auf welche im nächsten Schritt passende Komponenten abgebildet werden sollen. Im Nachhinein betrachtet, wäre die Bezeichnung „Eigenheit“ passender gewesen. In Abbildung 5 ist ein vereinfachtes Entity Relationship Diagramm abgebildet, welches die Beziehungen zwischen Fragen, Anforderungen, Einschränkungswerten und Komponenten verdeutlicht.

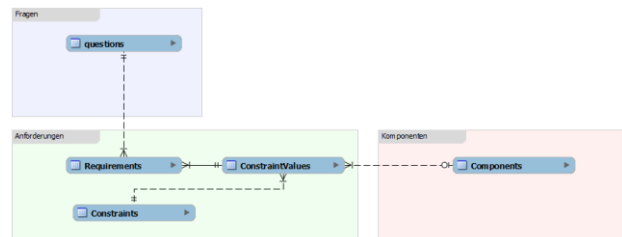


Bild 5 Vereinfachtes ER-Diagramm der Datenbank.

Mehrfach vorhandene Einschränkungswerte müssen nun zusammengefasst werden. Über eine Kumulationseigenschaft, die zu jeder Einschränkung hinterlegt werden muss, wird über die Vorgehensweise der Zusammenfassung entschieden. In Tabelle 1 sind die im System verfügbaren Kumulationseigenschaften aufgelistet.

Kumulationseigenschaft	Vorgehen
IntMaxConstraintCumulator	Der größere Wert der beiden Einschränkungswerte wird übernommen.
IntMinConstraintCumulator	Der kleinere der beiden Einschränkungswerte wird übernommen.
IntConstraintCumulator	Eine einfache Addition beider Werte wird vorgenommen.
BoolConstraintCumulator	Die zwei booleschen Logikwerte werden durch eine ODER-Operation verknüpft und übernommen.

Tabelle 1 Arten der Kumulationsmethoden und deren Funktionsweise.

Die zusammengefassten Einschränkungswerte stellen somit die „Soll-Basis“ dar. Im nächsten Schritt der Auswertung wird eine Sammlung von Komponenten gesucht die eine möglichst gute Erfüllung eben jener Einschränkungswerte ermöglicht.

3.3 Auswertung

Wie bereits erwähnt besitzen die Komponenten des Patronus-Systems ihrerseits einen oder mehrere Einschränkungswerte. In ihrer Rolle sind sie aber als Erfüller dieser

Eigenschaften zu sehen. Es wird nun ein Algorithmus angewendet, um sowohl die Menge der zu erfüllenden Einschränkungswerte (gegeben durch die Anforderungen) sowie die Menge der erfüllten Einschränkungswerte (gegeben durch eine Kombination von Komponenten) zur bestmöglichen Deckung zu bringen. Die Herausforderung besteht in der hohen zu erwartenden Anzahl an Kombinationen von Komponenten. Um eine perfekte Lösung zu finden müsste man alle Kombinationen auf ihre Lösungsgüte überprüfen. Sucht man nur nach einer Menge von 10 Komponenten aus einem Pool von 20 verfügbaren, ergibt dies aber schon über 20.030.010 Kombinationen. Da es in einem gut gepflegten System schnell zu einigen hundert möglichen Komponenten kommen kann, stoßen herkömmliche Algorithmen, welche die optimale Lösung suchen, schnell an ihre Grenzen. Tests mit einem sogenannten Evolutionären-Algorithmus [5] verliefen sehr vielversprechend. Bei diesem Ansatz werden zunächst zufällige Lösungen bestimmt. Eine Lösung besteht aus einer Menge von Komponenten. Im zweiten Schritt wird die Qualität dieser einzelnen Lösungen ermittelt. Durch eine sogenannte Fitnessfunktion wird die Komponentenzusammenstellung unter verschiedenen Gesichtspunkten bewertet:

- Wie viele Einschränkungswerte wurden erfüllt?
- Wie hoch ist die Anzahl der verwendeten Komponenten?
- Mit welcher Qualität wurden die Einschränkungswerte erfüllt?

Für jeden komplett erfüllten Einschränkungswert wird eine hohe Punktzahl vergeben. Je nach Art der Einschränkung und festzulegender Bewertungsfunktion gibt es Abzüge (negative Punkte) wenn man über oder unter dem gesuchten Wert liegt. Die Summe dieser positiven und negativen Punkte ergibt die Gesamtfitness dieser Lösung. Über diese Bewertung werden die besten Lösungsmengen bestimmt und die Komponenten dieser Lösungen erneut miteinander zufällig kombiniert.

Zusätzlich existiert eine geringe Wahrscheinlichkeit für „Mutationen“. Das bedeutet eine Komponente der Lösungsmenge wird zufällig durch eine neue Komponente oder durch gar keine Komponente ersetzt und danach eine erneute Bewertung der Lösungsqualität vorgenommen. Diese Schritte wiederholen sich so lange bis die Fitnessänderung der so gefunden Lösungen zwischen den Einzelschritten gegen Null konvergiert. Die am besten bewertete Komponentenkombination wird als Lösung des Problems angenommen. Der Algorithmus garantiert nicht, dass eine absolut beste Lösung gefunden wird. Dennoch weisen die Lösungen eine hohe Qualität auf und sind in der Lage, auch in sehr großen Lösungsräumen in vertretbarer Zeit eine näherungsweise gute Lösung zu finden.

3.4 Präsentation der Ergebnisse

Je nach Punktevergabe der Fitnessfunktion können leicht unterschiedliche Lösungsmengen mit veränderten Schwerpunktsetzungen gefunden werden. Denkbar wäre zum Bei-

spiel je nach verfügbarem Budget unterschiedliche Lösungskombinationen zu suchen. Die gefundenen Lösungsmengen werden übersichtlich präsentiert. Die Anzeige vermittelt der durchführenden Person sowie dem Berater transparent den Grund für die Auswahl dieser Komponente (der erfüllte Beschränkungswert wird angezeigt), sodass er die Vorauswahl noch einmal überarbeiten und anpassen kann. Die Ergebnisanzeige ist in Abbildung 5 dargestellt.

Die Datenbank des Bedarfsanalytators ist dabei so gestaltet, dass Hersteller auch neue Produkte einpflegen können und somit alle Installationen bei zukünftigen Analysen davon profitieren können. Auch eine nachträgliche Anpassung beziehungsweise Erweiterung des Fragenkatalogs wird dadurch ermöglicht.

Durch diese Maßnahmen wird das Erstellen eines fachgerechten Komponentenkataloges zur Altersassistentz automatisiert und vereinfacht. Diese Maßnahmen erhöhen die Akzeptanz beim Beratungspersonal.

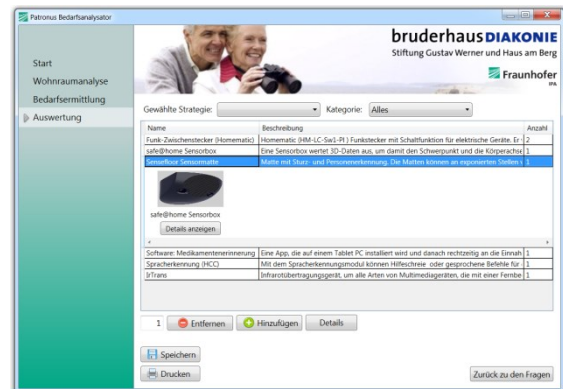


Bild 5 Anzeige der gefundenen Komponenten für das PATRONUS System.

3 Interoperabilität der Sensoren

In aller Regel bieten die Hersteller heutiger Heimautomatisierungslösungen nur Insellösungen an. Das heißt die Komponenten beschränken sich in ihrer Interoperabilität auf herstellereigene Produkte. Eine herstellübergreifende Zusammenarbeit wird in der Regel nur über den Einsatz einer zentralen Management Instanz gewährleistet. In aller Regel ist dies eine eigene Hardware oder ein kleiner, dezidierte Server der in das System eingebracht wird. Mit Hilfe von Gateways werden physisch inkompatible Übertragungsprotokolle auf ein gemeinsames Kommunikationsmedium herunter gebrochen. Hierfür wird aufgrund seiner starken Verbreitung in Privatwohnungen oft LAN oder WLAN verwendet.

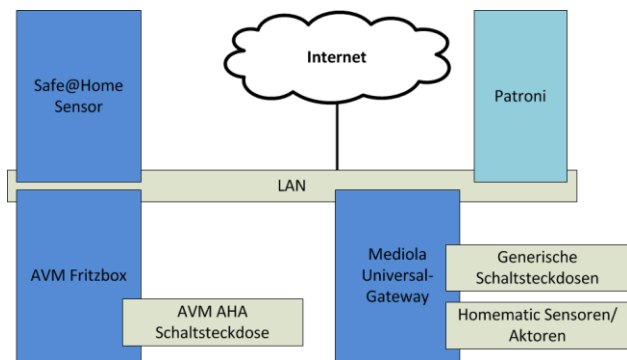


Bild 6 Schematischer Aufbau eines Patronus Systems.

Im Patronus Projekt wurde ein Gateway des Herstellers Mediola verwendet, das die Einbindung der ebenfalls verwendeten Homematic Aktoren und Sensoren ermöglichte (in Grafik 6 ist der schematische und vereinfachte Aufbau im Labor dargestellt). Die Informationen werden über eine REST-Schnittstelle [6] verfügbar gemacht. Kleine, selbstständige Programme, Patroni genannt, greifen auf diese Informationen zu und binden sie in eine interne Auswertelogik ein. Durch diese Logik sind sie in der Lage indirekte Notfälle zu erkennen und diese Statusänderung zu signalisieren. Begibt sich zum Beispiel eine Person ins Badezimmer und verbleibt dort ungewöhnlich lange ließe sich über den Patroni ein entsprechende Warnmeldung absetzen.

4 Ergebnisse

Der Bedarfsanalysator wurde bei einer Informationsveranstaltung dreißig Bewohnerinnen und Bewohnern des betreuten Wohnens vorgestellt und die Funktionen diskutiert. Im Rahmen des Realtests wurde zudem eine Befragung zur Handhabbarkeit des Bedarfsanalysators bei drei Seniorinnen durchgeführt, bei welchen anschließend einzelne Systemkomponenten installiert wurden. Der Fragenkatalog war hierbei auf die verfügbaren Patronus-Komponenten abgestimmt, sodass das Ergebnis der Auswertung technisch umgesetzt werden konnte. Zusammen mit Experten wurden die Fragen von den Seniorinnen beantwortet und eine Analyse der vorgeschlagenen Komponenten durchgeführt. Die vorgeschlagenen Komponenten deckten in allen drei Fällen den Bedarf der Damen ab. Bei zwei der Seniorinnen ergab die Anwendung des Bedarfsanalysators, dass eine automatische Sturzerkennung als einzige Komponente eingebaut werden soll. Bei der dritten Seniorin ergab die Anwendung, dass ein Bedarf an einer Einbruchserkennung besteht. Diese kann technisch ebenfalls über das Sturzerkennungssystem abgedeckt werden.

Eine weitere Erkenntnis war, dass vor allem technische Fragen zu Komfortfunktionen, wie Multimedia-Steuerungen, sehr einfach und präzise formuliert sein müssen um nachvollzogen zu werden.

Zukünftige Arbeiten könnten das Eintragen von neuen Komponenten in das System durch z.B. eine grafische Weboberfläche stark vereinfachen. Weiteres Potential steckt in der Optimierung und Verbesserung der Patroni

Auswertelogik. Hier würden weitere, erprobte Notfallszenarien zu einer verbesserten Alarmierung oder Warnung beitragen. Die Anpassung der Kommunikationsschicht an weitere Technologien anderer Hersteller würden letztlich die Einsatzmöglichkeiten des Systems erweitern und flexibler gestalten.

4 Literatur

- [1] Bevölkerungsentwicklung: Bundesinstitut für Bevölkerungsforschung, Evelyn Grünheid, Christian Fiedler, Bundesinstitut für Bevölkerungsforschung, Wiesbaden, 2013, urn:nbn:de:bib-var-2013-012, S. 12
- [2] The Internet of Things: Connecting Objects, Hakima Chaouchi, Wiley, 2013, ISBN 9781118600177
- [3] Baltès PB, Lindenberger U, Staudinger UM. Life-span theory in developmental psychology. In: Lerner RM, editor. Handbook of child psychology: Theoretical models of human development. Vol. 1. pp. 1029–1143. New York: Wiley, 1998
- [4] Programming Entity Framework: Dbcontext, Julian Lerman, Rowan Miller, O'Reilly, 2012, ISBN 9781449312961
- [5] Evolutionary Algorithms for Solving Multi-Objective Problems, Carlos Coello Coello, Gary B. Lamont, David A. van Veldhuizen, Springer, 2007, ISBN 9780387367972
- [6] REST API Design Rulebook, Mark Masse, O'Reilly, 2011, ISBN 9781449310509