

# BMBF Verbundprojekt iserveU

## Intelligente modulare Serviceroboter- Funktionalitäten im menschlichen Umfeld am Beispiel von Krankenhäusern

### Gemeinsamer Abschlussbericht

Förderkennzeichen 01IM12008



Robert Bosch GmbH	BOS	GU	Dr.-Ing. Tobias Gindele
RWTH Aachen	RWTH	HS	Prof. Dr. Bernhard Rumpe
TU Clausthal	TUC	HS	Prof. Dr. Andreas Rausch
FAU Erlangen-Nürnberg	FAU	HS	Prof. Dr.-Ing. Martin Vossiek
Symeo GmbH	SYM	GU	Dr. Peter Gulden
HS Ulm	HSU	HS	Prof. Dr. Christian Schlegel
REC GmbH	REC	KMU	Dr. Christian Verbeek

19. Oktober 2016

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen 01IM12008A gefördert. Die Verantwortung für den Inhalt dieser Veröffentlichung liegt beim Autor.



# Inhaltsverzeichnis

<b>1</b>	<b>Kurzdarstellung</b>	<b>5</b>
1.1	Aufgabenstellung . . . . .	5
1.1.1	Geplante repetitive Transportaufgaben . . . . .	6
1.1.2	Ungeplante ad-hoc Transportaufgaben . . . . .	6
1.1.3	Begleitdienste und kooperative Transportaufgaben . . . . .	6
1.1.4	Arbeitsplan . . . . .	6
1.2	Voraussetzungen, unter denen das Vorhaben durchgeführt wurde . . . . .	8
1.3	Planung und Ablauf des Vorhabens . . . . .	10
1.4	Wissenschaftlicher und technischer Stand, an den angeknüpft wurde . . . . .	13
1.4.1	Roboterbasisplattform Robotino . . . . .	13
1.4.2	Sensorik zur Hinderniserkennung . . . . .	13
1.4.3	Mensch – Maschine – Interaktion . . . . .	14
1.4.4	Funkortung . . . . .	14
1.4.5	Komponenten und modellgetriebene Entwicklung . . . . .	15
1.4.6	Verifikation und Remote Management . . . . .	17
1.5	Zusammenarbeit mit anderen Stellen . . . . .	17
<b>2</b>	<b>Eingehende Darstellung</b>	<b>19</b>
2.1	Verwendung der Zuwendung und des erzielten Ergebnisses im Einzelnen, mit Gegenüberstellung der vorgegebenen Ziele . . . . .	19
2.1.1	AP1: Szenarien und Gesamtarchitektur . . . . .	19
2.1.2	AP2: Kostengünstige Ortungs- und Navigationssensorik . . . . .	24
2.1.3	AP3: Situationsverständnis, -modellierung und Planung . . . . .	38
2.1.4	AP4: Mensch-Roboter Interaktion über innovative Nutzerschnitt- stellen . . . . .	42
2.1.5	AP5: Service-orientierte Architektur für Fallback-Sicherheit mittels Remote Management . . . . .	47
2.1.6	AP6: Durchgängige komponenten- und modellbasierte Entwicklung . . . . .	53
2.1.7	AP7: Prototypischer Roboteraufbau . . . . .	65
2.1.8	AP8: Pilotierung und Evaluierung im Krankenhaus . . . . .	71
2.2	Wichtigste Positionen des zahlenmäßigen Nachweises . . . . .	74
2.3	Notwendigkeit und Angemessenheit der geleisteten Arbeit . . . . .	74
2.4	Voraussichtlicher Nutzen, insbesondere Verwertbarkeit des Ergebnisses im Sinne des fortgeschriebenen Verwertungsplanes . . . . .	74
2.4.1	Robert Bosch GmbH . . . . .	74
2.4.2	Technische Universität Clausthal . . . . .	75

2.4.3	Hochschule Ulm . . . . .	75
2.4.4	Symeo GmbH . . . . .	76
2.4.5	FAU . . . . .	76
2.4.6	RWTH Aachen . . . . .	77
2.4.7	REC . . . . .	77
2.5	Während der Durchführung des Vorhabens bekannt gewordener Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen . . . . .	78
2.6	Erfolgte oder geplante Veröffentlichungen des Ergebnisses . . . . .	78
2.6.1	Robert Bosch GmbH . . . . .	78
2.6.2	FAU . . . . .	78
2.6.3	Symeo GmbH . . . . .	78
2.6.4	RWTH Aachen . . . . .	79
2.6.5	TU Clausthal . . . . .	79
2.6.6	REC . . . . .	80
2.6.7	Hochschule Ulm . . . . .	80

# 1 Kurzdarstellung

## 1.1 Aufgabenstellung

Ziel des Projekts iserveU war die Entwicklung eines prototypischen intelligenten Transportsystems (ITA) und der dafür notwendigen Technologien für die Ausführung von Logistikaufgaben in Krankenhäusern. Im Rahmen des Projekts sollten die folgenden drei Aufgaben bzw. Szenarien bearbeitet und näher untersucht werden:

- geplante repetitive Transportaufgaben
- ungeplante (ad-hoc) Transportaufgaben
- Begleitsdienste und kooperative Transportaufgaben

Entscheidend für die Lösung dieser komplexen Aufgabenstellung ist die interdisziplinäre Zusammenarbeit (siehe Abbildung 1.1)

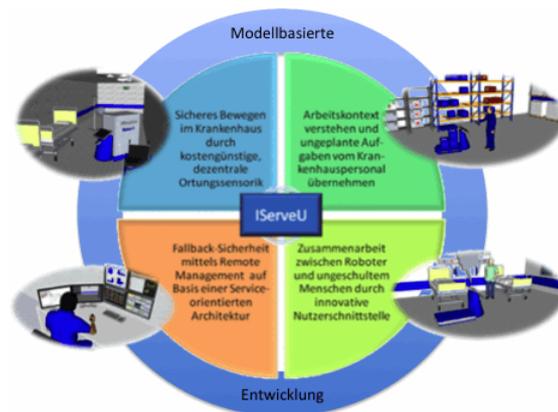


Abbildung 1.1: Forschungsgebiete iserveU

Das Vorhaben adressierte das Gebiet Servicerobotik im Rahmen des Förderprogramms „IKT 2020 – Forschung für Innovationen“ und stand im Kontext der „Hightech-Strategie 2020 für Deutschland“. Das Vorhaben stellte einen Beitrag zum Bedarfswelt „Kommunikation: Intelligente Objekte“ dar. Die in der Aktionslinie angesprochenen Service-Robotik Anwendungen im Gesundheitsbereich wurden konkret durch die Entwicklung eines intelligenten und autonomen Assistenten für Transportaufgaben im Krankenhaus aufgegriffen. Durch die Entwicklung eines Gesamtsystems, das durch seinen Logistikanteil Potenzial besitzt, auch in der Industrie eingesetzt zu werden, wurden die im Förderprogramm angestrebten übergreifenden Systemlösungen im Bereich Service-Robotik vorangetrieben.

### 1.1.1 Geplante repetitive Transportaufgaben

Viele Transportaufgaben im Krankenhaus fallen regelmäßig an. Charakteristisch dafür ist, dass Aufnahme- und Abladeorte sowie der Transportumfang fixiert sind (z.B. der tägliche Transport von Warenlieferungen von der LKW-Laderampe ins Zentrallager). Geplante repetitive Aufgaben stellen die einfachsten Transportaufgaben für ITAs dar: Kapazitäten können zentral geplant und Transportwege vorprogrammiert werden. Ein ITA muss feste Routen abfahren und dabei statischen und vor allem dynamischen Hindernissen ausweichen können. Die besondere Herausforderung in diesem Szenario ist das sichere Bewegen im menschlichen Umfeld mit nicht geschulten Personen.

### 1.1.2 Ungeplante ad-hoc Transportaufgaben

Lassen sich regelmäßige Transportaufgaben noch gut planen, stellen insbesondere die nicht weniger häufigen, dynamisch auftretenden und daher nicht planbaren Transportaufgaben eine Krankenhaus-Logistik vor große Probleme. Muss beispielsweise ein Kasten Wasser oder medizinisches Material aus dem Zentrallager geholt werden, so ist diese Transportaufgabe zeitaufwändig (im Katharinenhospital in Stuttgart rechnet man ca. 30-40 Minuten) und bringt die Arbeitsabläufe auf den Stationen durcheinander. Zusammen mit den geplanten Transporten nehmen derartige Logistikaufgaben derzeit ca. 60% der Arbeitszeit des Pflegepersonals ein. Für einen ITA bestehen hier die besonderen Herausforderungen in leistungsfähigerer Lokalisation und Navigation, effektiver Verfügbarkeit sowie insbesondere in der Kooperation mit Menschen. ITAs für ungeplante ad-hoc Transportaufgaben müssen deutlich stärker mit Menschen zusammenarbeiten, was sich insbesondere in der Auftragsannahme, der Interpretation von Befehlen oder im Folgen von Personen äußert.

### 1.1.3 Begleitdienste und kooperative Transportaufgaben

In dem vorherigen Szenario waren ITAs selbständig unterwegs oder haben mit geschultem Personal interagiert. Darauf aufbauende weitere Transportaufgaben lassen sich erschließen, wenn ITAs in der Lage sind, mit ungeschultem „Publikum“, im Krankenhaus also primär mit Patienten, zu kooperieren. Für Krankenhäuser sind Begleitdienste von besonderem Interesse: Diagnose- und Therapiestationen bleiben täglich über anderthalb Stunden ungenutzt, da Patienten nicht oder verspätet erscheinen. Die besonderen Herausforderungen liegen hier in den Anforderungen an das Situationsverständnis und die intuitive Interaktion. Ein ITA muss erkennen, ob Patienten noch folgen, klassifizieren warum Patienten nicht mehr folgen, und geeignet reagieren. Die Interaktion muss intuitiv sein.

### 1.1.4 Arbeitsplan

Im einzelnen waren folgende Arbeitspakete (AP) geplant:

#### 1. AP1: Szenarien und Gesamtarchitektur

Dieses Arbeitspaket, welches zu Beginn des Projektes steht, erarbeitet die konkre-

te Konzeption und Vorgehensweise. Es werden die konkreten Handlungsszenarien in der Domäne Krankenhaus, die man mit dem ITA bedienen möchte, definiert. Aus den sich daraus ergebenden Anforderungen wird die Gesamtarchitektur des Prototypsystems entworfen.

2. **AP2: Kostengünstige Ortungs- und Navigationssensorik**

Je nach Aufgabenstellung und Kontext sind unterschiedliche Sensoren notwendig. Dieses Arbeitspaket hat als Zielsetzung die Erforschung und Realisierung einer kostengünstigen, anwendungsorientierten Ortungs- und Navigationssensorik sowie die Realisierung einer Navigationslösung für menschengeprägte Umgebungen.

3. **AP3: Situationsverständnis, -modellierung und Planung**

Zur Übernahme heterogener Aufgaben benötigt der ITA ein Verständnis der Situation, in der er sich aktuell befindet. Dazu benötigt er Informationen der Sensorik, der Aktorik und weitere Kontextinformationen. Die relevanten Informationen sollen in einer domänenspezifischen Sprache zur Situationsbeschreibung modelliert werden. Auf Basis des konkreten Modells wird die Planung der nächsten Aktionen erfolgen. Dadurch erhält der ITA ein Verständnis seiner Situation, kann diese bewerten und angemessen reagieren.

4. **AP4: Mensch-Roboter Interaktion über innovative Nutzerschnittstellen**

Das Arbeitspaket hat zum Ziel, ein Konzept für die Interaktion zwischen Personen und dem ITA zu entwickeln. Die in dem Konzept erarbeiteten qualitativen Anforderungen müssen über geeignete Studien auf quantitative Größen heruntergebrochen werden, die dann innerhalb der Algorithmen für die Navigation berücksichtigt werden.

5. **AP5: Service-orientierte Architektur für Fallback-Sicherheit mittels Remote Management**

Ziel des Arbeitspaketes ist die Realisierung eines Fallback-Sicherheitsmechanismus für den Fall, dass der Serviceroboter in Situationen gerät die ihm unbekannt sind und die mit den ihm bekannten Strategien, nicht beherrschbar sind.

6. **AP6: Durchgängige komponenten- und modellbasierte Entwicklung**

Entwicklung einer modularen Softwarearchitektur und der zugehörigen Werkzeugkette für die modellbasierte Entwicklung von Softwarekomponenten für Serviceroboter, die die Integration der verteilten Funktionen von Servicerobotern und Anforderungen in verschiedenen Anwendungsdomänen und Plattformen sowie deren situative Orchestrierung zur Laufzeit ermöglichen und unterstützen.

7. **AP7: Prototypischer Roboteraufbau**

Aufbau einer für die Evaluierung notwendigen Roboterplattform mit der dazugehörigen Basissoftware. Zusätzlich wird in diesem Arbeitspaket die notwendige Entwicklungs- und Simulationsumgebung aufgebaut, so dass im Projekt eine einheitliche Werkzeugkette vorhanden ist.

## 8. AP8: Pilotierung und Evaluierung im Krankenhaus

Evaluierung der Arbeitsergebnisse in einer realen Krankenhausumgebung

### 1.2 Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Das Konsortium bestand aus Hochschul- und Universitäts- Partnern (HSU, RWTH, TUC, FAU), sowie mittelständischen Unternehmen und Großunternehmen (SYM, REC, BOS). Durch diese Zusammensetzung war eine optimale Abdeckung und Aufteilung der Projektinhalte und -ziele möglich. Alle Partner hatten bereits die nötige Projekterfahrung und Robotikaffinität, um die gestellte Aufgabe zu lösen.

Das Konsortium bestand aus folgenden Partnern, die Expertenwissen aus den folgenden Gebieten einbrachten

**Robert Bosch GmbH (BOS)** Ultraschallsensorik, prototypischer Roboteraufbau, Navigation und Mensch-Maschine Interaktion. Die Robert Bosch GmbH brachte ihre langjährige Erfahrung aus dem Sensorbaubereich ein, um die Entwicklung und die Anwendung des neuen 3D Ultraschallsensors voranzubringen. Die gesammelte Erfahrung aus vorangegangenen Robotikprojekten, die sich ebenfalls schon mit Transportrobotik und mobilen Plattformen befassten, stellte die Grundlage für die weiteren Entwicklungen in diesem Projekt dar. Speziell auf dem Gebiet der Mensch-Maschine-Interaktion konnte auf einen reichen Erfahrungsschatz zurückgegriffen werden, um die Entwicklung der Interaktion und des Navigationsverhaltens zu leiten.

**RWTH Aachen (RWTH)** Intelligente Steuerung, Situationserkennung, Simulation. Die RWTH Aachen brachte ein hohes Maß an Erfahrung und Expertise in den Gebieten Sprachdesign und agiler Softwareentwicklung ein. Das am Lehrstuhl für Software Engineering der RWTH Aachen entwickelte Framework MontiCore [KRV08] brachte ideale Voraussetzungen mit sich, in einem agilen Entwicklungsprozess Modellierungssprachen zur Spezifikation der Fähigkeiten und Eigenschaften autonom agierender Roboter, zur Modellierung ihrer Einsatzkontexte, und zur Situationsbeschreibung, zu entwickeln. Die von MontiCore bereitgestellte Codegenerierungsinfrastruktur und die, am Lehrstuhl für Software Engineering der RWTH Aachen entwickelte, Architekturbeschreibungssprache MontiArcAutomaton [RRRW15] eigneten sich bestens zur Modellierung und Generierung komponentenbasierter Softwarearchitekturen für Servicerobotikanwendungen [HMSNR<sup>+</sup>15].

**Technische Universität Clausthal (TUC)** Service-orientierte Softwarearchitektur und Absicherung durch Remote-Management. Der Projektpartner TU Clausthal verfügte über eine langjährige Erfahrung im Bereich der Entwicklung und Weiterentwicklung von Frameworks für dynamisch-adaptive, komponenten-basierte Systeme, welche u.a. im Robotik-Umfeld eingesetzt wurden. So wurde bereits im Jahr 2006 das Framework DAiSi entwickelt, welches in der Lage ist, Sensoren und Aktuatoren sowie Softwarekomponenten zu vernetzen und zur Laufzeit zu rekonfigurieren. Zudem

besitzt der beteiligte Lehrstuhl “Software Systems Engineering” (Rausch) intensive Projekterfahrung sowohl bei der Beteiligung an nationalen als auch internationalen Projekten.

**Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)** Mehrdimensionale Lokalisierungsverfahren auf Basis der Angle-of-Arrival und Roundtrip time-of-Flight Methoden. Der Lehrstuhl für Hochfrequenztechnik (LHFT) der FAU verfügt über langjährige Erfahrung in der Entwicklung von Sensorik und Signalverarbeitung für die Funkortung. Sehr leistungsfähige Roundtrip time of flight (RTOF) Ortungsverfahren auf Basis von 24 GHz Technik wurden bereits von der Forschergruppe von Prof. Vossiek (FAU) [Vea96],[RGV08] in Zusammenarbeit mit der Firma Symeo entwickelt. Hier wird eine Ortungsgenauigkeit im cm-Bereich bei einer Reichweite von einigen 100m erreicht. Die vorangegangenen Arbeiten zeigen die Fähigkeit, komplexe Aufgabenstellungen sowohl auf Hardware- als auch auf Signalverarbeitungsebene realisieren zu können.

**Symeo GmbH (SYM)** Radar-basierte Ortungssensorik und relative Ortung. Die Symeo GmbH brachte in das Vorhaben ihre umfangreiche Kompetenz und Vorarbeiten im Bereich Ortung ein. So konnten in den letzten Jahren eine Vielzahl von 2D Positionierungslösungen in der Schwerindustrie realisiert werden, die die weltweit führende Stellung der Symeo GmbH zeigen. Im Rahmen des Projekts wurde dann sowohl die Hochfrequenzkompetenz (Design, Messtechnik) als auch die Kompetenz im Embedded Bereich (FPGA, C, Linux) benötigt. Dadurch konnte den Projektpartnern bereits zu einem frühen Stadium eine erste Ortungssensorik zur Verfügung gestellt werden und die entsprechenden Schritte zur Systemintegration durchgeführt werden.

**Hochschule Ulm (HSU)** Service-basierte Komponentenarchitektur und modellgetriebene Softwareentwicklung. Die HSU brachte mit dem von ihr entwickelten und gepflegten Robotik Software Framework *SmartSoft* [Sch04, Ser] bereits zu Beginn ein leistungsfähiges Framework mit einer Reihe von konkreten Basiskomponenten für Roboter ein (z.B. Navigation, Lokalisierung, Koordination, etc.). Eine erste Version der SmartMDS Toolchain, ebenfalls entwickelt und gepflegt von der HSU, konnte als Ausgangspunkt für die modellgetriebenen Arbeiten genutzt und weiter entwickelt werden.

**REC GmbH (REC)** Roboterbasisplattform, Simulation. Durch den Einsatz der von der REC GmbH entwickelten mobilen Roboterplattform Robotino bestand die Möglichkeit, das System durch Modifikation der Hardware und Software optimal an die Anforderungen innerhalb des Projektes anzupassen. Darüber hinaus konnte der für Robotino von der REC GmbH entwickelte Simulator Robotino SIM ebenfalls optimal an die durch das Projekt gestellten Erfordernisse angepasst werden.

## 1.3 Planung und Ablauf des Vorhabens

In der Vorbereitung zu diesem Projekt kam es bereits zu einem intensiven Austausch zwischen den Projektpartnern, um die komplexe Aufgabenstellung sinnvoll zu adressieren. Im Rahmen dessen wurde auch schon Kontakt zu Krankenhäusern aufgenommen und diese besichtigt, was zu einer klaren Vorstellung der Herausforderungen und Anforderungen an ein intelligentes Transportsystem führte. Die Konzeption und Aufteilung der Arbeitspakete richtete sich nach diesen Anforderungen und spiegelte gleichzeitig die Expertisen der jeweiligen Projektpartner wider. Der vorgeschlagene zeitliche Plan ist Abbildung 1.2 zu entnehmen.

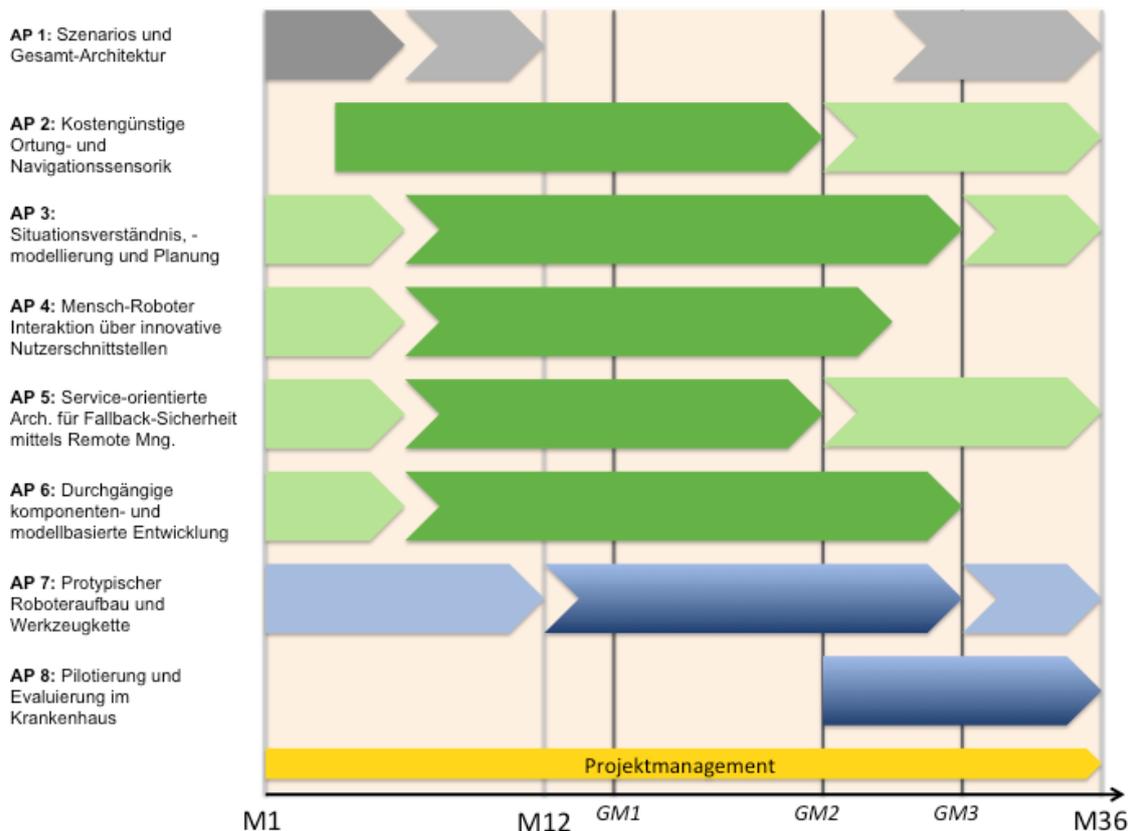


Abbildung 1.2: Geplanter zeitlicher Arbeitverlauf

Der Projektplan definierte mehrere Meilensteine:

**GM1:** Prototyp mit Standardsoftware und Standardsensorik (M15)

**GM2:** Prototyp mit neuer Sensorik (M24)

**GM3:** Voll integrierter Prototyp (M30)

Alle Meilensteine wurden zeitnah erreicht und dem Projektträger präsentiert. Der genaue Ablauf der wichtigsten gemeinsamen Veranstaltungen und Präsentationen der Meilensteine ist hier aufgelistet:

<b>Veranstaltung</b>	<b>Datum</b>
Kickoff	11.04.2013
Workshop zur Software-Systemarchitektur in Schwieberdingen	07.06.2013
Workshop mit allen Projektpartnern und Projektträger DLR: Vorstellung des Standes und Abnahme vom <b>Gesamtmeilenstein 1 (GM1)</b>	29.10.2014
Workshop mit allen Projektpartnern und Projektträger DLR: Vorstellung des Standes und Abnahme vom <b>Gesamtmeilenstein 2 (GM2)</b>	22.04.2015
Vorbereitung Krankenhaustest und Integrationsworkshop bei Bosch in Renningen	26.10.2015 – 28.10.2015
Systemtest im Katharinenhospital Stuttgart	29.10.2015 – 30.10.2015
Evaluation des ITA-Roboters im Katharinenhospital Stuttgart	23.11.2015 – 27.11.2015
Vorbereitungstreffen für die Abschlussveranstaltung	15.02.2016 – 18.02.2016
Abschlussveranstaltung mit allen Projektpartnern und Projektträger DLR: Abnahme vom <b>Gesamtmeilenstein 3 (GM3)</b> sowie Gesamtabnahme des Projekts inklusive umfangreicher Live-Demonstration mit dem Roboterprototyp	25.02.2015

Darüber hinaus fanden mehrere Integrations- und Abstimmungsworkshops, meist mit der Beteiligung von zwei oder drei Projektpartnern, statt. In regelmäßigen Telefonkonferenzen wurde der Projektfortschritt abgeglichen, neue Ergebnisse präsentiert und das weitere Vorgehen abgestimmt. Insgesamt wurde der Projektplan weitestgehend eingehalten. Zeitweise kam es zu Verzögerungen, die im Laufe des Projekts aber wieder aufgeholt werden konnten. Die Verzögerungen waren bedingt durch

- Probleme bei der Fertigung der neuen Ultraschallsensoren
- Anpassung der Ortungshardware auf Grund der Ergebnisse der ersten praktischen Erprobung: Es wurden die 8 Empfangsantennen, eine Sendeantenne sowie die Zig-Bee Kommunikation angepasst auf 16 Empfangs-, 2 Sendeantennen und 24 GHz Kommunikation.
- Ausgründung eines Bosch Startups verbunden mit mehreren Personalwechselln
- Mitarbeiter in Elternzeit

Darüber hinaus wechselte 2014 zweimal die Projektleitung.

Ein wichtiger Aspekt, der sich durch die gesamte Bearbeitung des Projekts zog, war die modellgetriebene Entwicklung. Methodisch wurde dies durch die Hochschule Ulm eingebracht und von allen Partnern umgesetzt. Bereits in den ersten Monaten des Projekts wurde eine gemeinsame service-orientierte komponenten-basierte Softwarearchitektur definiert, welche über die Projektlaufzeit iterativ verfeinert wurde. Entsprechend des durchgängigen modellgetriebenen Entwicklungsansatzes und auf Basis von Komponentenmodellen und Services konnten schon zu diesem Zeitpunkt Zuständigkeiten und Abhängigkeiten identifiziert und geklärt werden. Um einen reibungslosen Start aller Arbeitspakete zu ermöglichen, wurden existierende Standard-Komponenten der *HSU* zu Beginn des Projekts und für GM1 eingesetzt. Den Partnern standen somit wichtige Grundfunktionalitäten sehr früh zur Verfügung. Einige der Standard *HSU* Komponenten, z.B. Lokalisierung und Navigation, konnten dann, wie im Projektplan vorgesehen, über die Projektlaufzeit sukzessive und transparent durch die von den Partnern entwickelten und auf den Anwendungsfall abgestimmten Komponenten ersetzt werden. Die von der *HSU* bereits in den ersten Monaten des Projekts zur Verfügung gestellte SmartMDSD Toolchain und *SmartSoft* als Robotik Framework, ermöglichten den Projektpartnern von Beginn an eine kontinuierliche Integration auf Basis eines durchgängigen komponenten- und modellgetriebenen Entwicklungsansatzes. Die initiale Konzeption der Softwarearchitektur mit der funktionalen Zerlegung in einzelne Komponenten mit wohldefinierten Schnittstellen zahlte sich aus, da es eine fortlaufende Integration der Ergebnisse in das Gesamtsystem ermöglichte und auch vereinfachte.

Innerhalb des Projekts sollte die neueste Version von Robotino eingesetzt werden. Die Entwicklung stand zum Projektstart kurz vor dem Abschluss. Das System war in Hardware aber noch nicht verfügbar. Aus diesem Grund bestand der erste Arbeitsschritt in der Entwicklung eines Robotermodells für den Simulator, so dass bereits mit der Softwareentwicklung ohne reales System begonnen werden konnte. Die in den ersten Wochen abgeleiteten Anforderungen an die Hardware von Robotino konnten noch in die laufende Entwicklung einfließen, so dass einige Wochen nach Projektstart die ersten Robotersysteme zur Verfügung standen. Die für iserveU gemachten Anpassungen der Roboterhardware sind jetzt Bestandteil der Robotino Standardversion.

Im Verlauf des Projekts kristallisierte sich eine Änderung zum ursprünglichen Plan heraus. Aus praktischen Gründen war es sinnvoller die ursprünglich bei REC in München geplante Teststrecke bei Bosch in Renningen aufzubauen. Die dadurch freiwerdenden Kapazitäten bei REC wurden genutzt, um über den ursprünglichen Umfang hinausgehende Versuche in Richtung Multi-Agenten Systeme durchzuführen.

Gegen Ende der Projektlaufzeit wurde eine Evaluation im Katharinenhospital in Stuttgart durchgeführt. Dabei wurden die definierten Use-Cases wie Transport von Gegenständen, Personen führen und Personen folgen mehrfach durchgeführt und die Leistung des Robotersystems evaluiert.

Die Zusammenarbeit zwischen den Partnern verlief vorbildlich und war von einer hohen Hilfsbereitschaft und Lösungsorientierung geprägt. Die Kollaboration und Sammlung der Software, Deliverables und Dokumente wurde durch ein Versionsverwaltungswerkzeug unterstützt, das die RWTH einrichtete. Die wissenschaftlichen Ergebnisse wurden fort-

laufend auf Fachkonferenzen und in Zeitschriften veröffentlicht (s.a. Literaturverzeichnis).

## **1.4 Wissenschaftlicher und technischer Stand, an den angeknüpft wurde**

### **1.4.1 Roboterbasisplattform Robotino**

Die Entwicklung und Programmierung mobiler Robotersysteme ist aufgrund der hohen Komplexität sowohl die Hardware als auch die Software betreffend fehleranfällig und zeitintensiv. Die Erfahrung hat zudem gezeigt, dass eine Entwicklung der Software im Zusammenspiel mit einem Robotersimulator zwar den Entwicklungsprozess beschleunigt, diese Beschleunigung jedoch aufgrund einer Abstraktion von der realen Welt erkauft wird. Innerhalb eines Simulators entwickelte Software ist auf einem realen Roboter daher meistens nicht vergleichbar lauffähig. Aus diesem Grund stand der Einsatz eines einheitlichen mobilen Robotersystems im Fokus, so dass die Entwicklung immer auch am realen System getestet werden kann.

Das Roboterbasissystem Robotino in der zum Projektstart erhältlichen Version 2 war als Versuchsträger aufgrund von Traglastbeschränkungen und einer zu geringen Rechenleistung des Onboard-PC nicht geeignet. Mit der durch dieses Projekt beeinflussten Entwicklung des Nachfolgemodells Robotino 3 stand eine Plattform mit einer optimalen Größe und ausreichend Rechenleistung zur Verfügung.

### **1.4.2 Sensorik zur Hinderniserkennung**

Im Bereich der Hinderniserkennung und Sicherheitssensorik ist der Laserscanner Stand der Technik. Es sind Serienprodukte verfügbar, die Tracking-Algorithmen für Objekte bereits in der Standardkonfiguration mitbringen. Die Firma SICK fertigt mehrere Systeme in diesem Bereich. Kamera-Systeme sind aufgrund der bereits ausreichenden Lösung durch „konventionelle“ Systeme weniger interessant. Die Forschung erstreckt sich hier auf Objekt- und Wegfindungen sowie auf das Erkennen von Personen und deren Stimmungen. Dies wird relevant, sobald Roboter Patienten betreuen. Eine kostengünstigere Alternative zum Laserscanner als Sicherheitssensor existierte derzeit nicht.

Ultraschallsensoren können zur Umfelderkennung verwendet werden. Ultraschall-Arrays, welche neben dem Abstand zu einem Hindernis auch dessen Winkel relativ zum Sensor liefern, können ähnlich wie mit einem Laserscanner Hindernisse detektieren und klassifizieren und bieten sich als kostengünstige Alternative an. Ultraschall-Arrays sind nicht anfällig gegenüber wechselnden Beleuchtungssituationen – was gerade für die Transportwege in Krankenhäusern wichtig ist – und sind robust gegenüber durchsichtigen Hindernissen wie Fenstern oder Glastüren. Darüber hinaus liefert ein Ultraschall-Array Sensor ‚echte‘ 3D Höheninformationen (z.B. Tischkante, auf dem Boden liegende Person), die über einen normalen 2D Laserscanner nicht zur Verfügung stehen. Die Herausforderung bei der Arbeit mit Ultraschall-Arrays liegt hier in der Bestimmung eines geeigneten Layouts, in der parallelen Verarbeitung aller Einzelsensorinformationen und einer effizienten

Auswertelgorithmen. Die Verwendung von Kamera-basierter Sensorik (wie z.B. Microsoft Kinect) zur Gewinnung von Tiefeninformationen (RGBD-Daten) ist derzeit in der wissenschaftlichen Community weit verbreitet. Gegen den Einsatz als Sicherheitssensor sprechen aber nicht nur das begrenzte Sichtfeld und der zu große minimale Sichtabstand, sondern auch die zu große Störempfindlichkeit gegenüber extremem Licht (z.B. Sonneneinstrahlung).

### 1.4.3 Mensch – Maschine – Interaktion

Die Akzeptanz von Robotik-Lösungen im Krankenhaus wird maßgeblich durch den Umgang des Roboters mit dem Menschen und seinem Verhalten im menschlichen Umfeld bestimmt [Bö01]. Dies betrifft die Interaktion des Roboters mit geschultem Personal, aber insbesondere auch mit ungeschulten Menschen, also den Patienten bzw. Besuchern im Krankenhaus. In Untersuchungen wurde bereits gezeigt, dass möglichst vorhersagbares, intuitives, sinnfälliges und natürliches Verhalten solcher Roboter notwendig ist, um den Erwartungen aller Nutzer gerecht zu werden, die Interaktion zu erleichtern und somit Unzufriedenheit, Störungen oder sogar Ängste zu vermeiden [Bö01] [Wal08] [FND02] [FLC<sup>+</sup>03] [KSF09]. Neben den technischen Fähigkeiten bedarf es demnach einer Vielzahl von sozialen Fähigkeiten für die Umsetzung von angemessenen sozial akzeptierten Verhaltensweisen [SDW<sup>+</sup>06] [WDK<sup>+</sup>07] [DWW<sup>+</sup>06]. Sozial akzeptables Verhalten umfasst hierbei sowohl Dimensionen der direkten (Sprache, Gestik, Haptik) als auch der indirekten Interaktion („Körpersprache“ des Roboters - Bewegungsverhalten) zwischen Menschen und Robotern (Christensen, 2005). Obwohl die Wichtigkeit dieser Erkenntnisse von einer Reihe von Forschern proklamiert wird [SDW<sup>+</sup>06] [WDK<sup>+</sup>07] [DWW<sup>+</sup>06] [FND02], bestand vor Projektbeginn insbesondere ein Mangel an systematisch gewonnenen empirischen Daten zum Themenkomplex „Mensch-Roboter Bewegungsverhalten“ (z.B. Mensch-Roboter Proxemik, Geschwindigkeit, Annäherungsrichtung, Ausweichverhalten) [Bö01] [Wal08]. Dabei war bereits gezeigt worden, dass angemessene sozial akzeptable Bewegungsattribute eines ITA wesentlich zu einem „smart maneuvering“ beitragen [CP05]. Bereits bestehende autonome Transportsysteme (z.B. Aethon's TUG) zeigten noch sehr stockende, unnatürliche und vor allem nicht vorhersagbare Bewegungsmuster [MF08]. Roboter, die Wege blockieren, zu dicht heran- oder vorbeifahren, oder den Weg abschneiden, führen zu verärgerten Nutzern und verdeutlichen die menschliche Sensibilität für ein angemessenes Fahrverhalten eines Roboters [MF08]. Sehr lange Wartezeiten und generell zu langsam manövrierende Systeme werden ebenfalls als störend empfunden. Außerdem waren derzeit viele Handlungen von Servicerobotern nicht transparent genug und die Nutzer bekommen nur unzureichendes oder häufig auch unverständliches Feedback über aktuelle Systemzustände [MF08]. Die Ergebnisse sind Unzufriedenheit und eine erhebliche Störung des Workflow auf Seiten der Nutzer [MF08].

### 1.4.4 Funkortung

Neueste Entwicklungen im Bereich integrierter Schaltungen für Radaranwendungen im ISM Band bei 24 GHz ermöglichen die Realisierung miniaturisierter, energie-effizienter

Sensorknoten mit Mikrowellen-basiertem Ortungsvermögen. Die im Frequenzband zugelassene Bandbreite von 250 MHz weist eine Verbesserung bzgl. der erreichbaren Genauigkeit auf, insbesondere im Vergleich zu bisher weit verbreiteten Mikrowellensensoren bei 2,45 GHz. Die im Vergleich zu konventionellen Funkortungssystemen reduzierte Wellenlänge der 24 GHz Sensorknoten führt zur Reduzierung der geometrischen Abmessungen der Sende- und Empfangsantennen. Mehrere derartiger Empfänger können in einem einzigen Sensorknoten integriert und deren Antennen in einem Array akzeptabler Größe zusammengefasst werden, wodurch ein zusätzliches Ortungsverfahren, die Winkelschätzung, implementiert werden kann. Ein solcher mehrkanaligen „single input multiple output“(SIMO) Sensorknoten erlaubt zusätzlich zu der RTOF basierten Entfernungsmessung Aussagen bzgl. der Position des Zielobjekts in Azimut und Elevation. Aufgrund der 3D-Positionsbestimmung mit nur einem einzigen Sensorknoten stellt die RTOF-AoA-Funkortung eine vielversprechende Alternative dar zu den bisher verbreiteten Ansätzen auf Basis von TOA- oder TDOA-Verfahren (TOA = time of arrival; TDOA = time difference of arrival), wobei eine Vielzahl von Messknoten für eine Positionsbestimmung notwendig sind.

Seitens Symeo wurde die grundlegende LPR Technologie zur Entfernungsmessung mittels round-trip time-of-flight verwendet. Diese Technologie ist durch die Patentfamilien um DE10157931 und DE112005000779.0 geschützt und wurde beispielsweise in [RGV08] veröffentlicht. Erste grundlegende Kombinationen mit einer Winkelmessung zwischen 2 Einheiten wurden im RFTACH Projekt durchgeführt und in [RSRV15] veröffentlicht. Mit dieser Ausgangsbasis konnten dann die Arbeiten im Bereich AP2 (Entwurf tragbarer beacon, Empfänger mit bis zu 16 Kanälen, Funkschnittstelle, Positionierung und Sensorfusion mit Sparse Infrastructure) durchgeführt werden.

#### 1.4.5 Komponenten und modellgetriebene Entwicklung

Bereits vor Projektbeginn waren eine Vielzahl von unterschiedlichen Frameworks zur Realisierung von Servicerobotik-Systemen verbreitet, die allesamt spezifische Schwerpunkte setzen, aber zumeist wenig interoperabel sind. Beispiele sind: Player [CMG05], Robot Operating System (ROS) [QCG<sup>+</sup>09], YARP (Yet Another Robot Platform) [MFN06], OPRoS (Open Platform for Robotic Services) [JLJ<sup>+</sup>16], OROCOS (Open Robot Control Software) [Opea], OpenRTM (RT-middleware, Robotics Technology Middleware) [Opeb], Microsoft Robotics Studio [Mic] und SmartSoft [SSL12a, SSL12b].

Aus diesen sticht ROS als größte Plattform mit einer bisher beispiellosen Verbreitung in der Community hervor. Selbst für ROS sind jedoch Stabilität (in Bezug auf Änderungen von Schnittstellen und Mechanismen), Semantik (explizite Spezifikation von Verhalten und Strukturen unabhängig von einer konkreten Implementierung) sowie Robustheit (in Hinblick auf Qualitätskontrolle) noch offene Herausforderungen.

Gemeinsam ist vielen Ansätzen, dass bisher nicht der Schritt gemacht wurde in Richtung einer Modellierung der hinter den strukturgebenden Elementen (z.B. Services, Komponenten) steckenden Prinzipien und Softwarekonstrukten. Typischerweise ist deshalb auch keine exakte und verlässliche, von der konkreten Implementierung unabhängige semantische Spezifikation verfügbar, die Innovationszyklen in der Softwaretechnologie

überstehen würde (z.B. Austausch von Middlewaresystemen: CORBA, DDS). Typischerweise werden auch in der Robotik mit jeder neuen Softwaretechnologie erneut erhebliche Anpassungen auf Code-Ebene notwendig. Desweiteren ist durch die fehlende Explikation relevanter Softwareeigenschaften deren korrekte Einstellung und Überprüfung beispielsweise bei der Systemintegration und Wiederverwendung von SW-Komponenten nur aufwändig möglich.

In jüngerer Zeit vor Projektbeginn gab es in der Robotik signifikante Aktivitäten in Richtung eines konsequent modellgetriebenen Vorgehens in der Softwareentwicklung, z.B. [ASK<sup>+</sup>05, GBWK09, SHLS09, SSL12a, SSL12b]. Vorarbeiten in Richtung eines modellgetriebenen Ansatzes zur Softwareentwicklung in der Robotik bilden Arbeiten zum Component-Based Software Engineering (CBSE) in der Robotik [Sch06]. Ein Überblick hierzu wird in [BS10, BS09] gegeben.

Das bereits im Projektantrag erwähnte FP7 EU-Projekt BRICS [BRI], welches sich zu Projektbeginn im Abschluss befand, wollte gezielt bewährte Praktiken in der Robotik aufgreifen und per Modellebene verfügbar machen. Projektziel bis Anfang 2013 war es, per MDE die einfachere Wiederverwendbarkeit bewährter Lösungen zum Zwecke der Reduktion des Entwicklungsaufwandes zu ermöglichen. Das dort vorgestellte rudimentäre Komponentenmodell von BRICS (BCM - BRICS Component Model) berücksichtigt im Gegensatz zu dem im Projekt verwendeten und weiterentwickelten Komponentenmodell SmartMARS noch nicht die vielfältigen Abhängigkeiten beispielsweise zwischen Ports, Komponenten-Lifecycle, Dynamic Wiring, Monitoring etc. Das BCM zieht auf Roundtripping zwischen unterschiedlichen Modellverfeinerungen (RTT, OpenRTM, ROS) via BCM, was definitiv eine andere Fokussierung darstellt. In SmartMARS der HSU wird dagegen konsequent der Top-Down-Ansatz verfolgt: das SmartMARS-Meta-Modell spezifiziert, wie Komponentenhüllen auszusehen haben. Durch schrittweise Anreicherung von Zusatzinformation, beispielsweise über die Zielplattform, werden mittels M2M / M2T-Transformationen schließlich die Komponentenhüllen generiert. Der Ansatz des *stepwise refinement* ermöglicht die konsequente Trennung der Rollen als wichtige Voraussetzung für einen strukturierten Entwicklungsprozess.

Weit fortgeschritten sind die vor allem asiatisch getriebenen Standardisierungsbemühungen der OMG Robotics Task Force [RTC08]. Derzeit wird in den meisten asiatischen Entwicklungsprojekten zur Servicerobotik der Einsatz von RT (Robotics Technology) [ASK<sup>+</sup>05] gefordert. Eine weit verbreitete Implementierung ist openRTM-aist. Durch die Bereitstellung einer Modellebene können verlässliche Komponentenschnittstellen zur Sicherstellung der Interoperabilität gewährleistet werden. Auch können zur gleichen Spezifikation unterschiedliche Implementierungstechnologien von unterschiedlichen Anbietern mit gleichem Verhalten in Konkurrenz zueinander treten. Die Arbeiten im Umfeld openRTM-aist sind am ehesten mit den Arbeiten um SmartMDS/SmartSoft vergleichbar. Allerdings umfasst es nur ein recht stark vereinfachtes datenflussähnliches Kommunikationsmodell und adressiert nicht den besonders wichtigen Aspekt der Rollentrennung.

### 1.4.6 Verifikation und Remote Management

In den Situationen, die einem Serviceroboter unbekannt oder von ihm nicht mehr beherrschbar sind, kann es die Möglichkeit eines manuellen Eingriffs durch einen externen Operator über ein Netzwerk geben. In diesem Fall spricht man auch von Remote Management. Die kritische Situation, in der sich der Roboter befindet, kann unterschiedliche Auslöser haben, wie die Roboterkonfiguration und die Umgebung, in welcher der Roboter agiert. Ein Beispiel, in dem die Roboterkonfiguration der Auslöser für eine kritische Situation sein kann, sind defekte Sensoren. Der externe Operator wird in diesem Fall eine Rekonfiguration des Roboters vornehmen.

Eine andere Ursache für eine kritische Situation kann die Dynamik der Betriebsumgebung sein. Zur Entwurfszeit des Roboters werden gewisse Annahmen über die Umgebung gemacht, z.B. die Obergrenze für die Geschwindigkeiten von möglichen dynamischen Hindernissen. Es kommt aber oft vor, dass die Betriebsumgebung viel komplexer und heterogener ist als angenommen, z.B. die Geschwindigkeit der dynamischen Hindernissen viel größer ist als die angenommene Obergrenze. Das Roboterverhalten, modelliert in einem abstrakten Systemmodell, kann zusammen mit den Annahmen über die Umgebung gegen eine vordefinierte Systemspezifikation des Roboters verifiziert werden. Im Fall einer Verletzung der Systemspezifikation wird das Ergebnis an den externen Operator gesendet und um Hilfe gebeten.

Die Fachliteratur enthält zahlreiche Arbeiten zum Thema Absicherung von Systemeigenschaften sowohl zur Entwurfszeit durch formale Methoden als auch zur Laufzeit durch Laufzeitüberwachung und Laufzeitverifikation. Mit formalen Methoden wie Model-checking [War96, Hol97, DKFW10, LV01, VHB<sup>+</sup>03] kann man ein abstraktes Systemmodell zusammen mit den Annahmen über die Umgebung gegen die Systemeigenschaften verifizieren. Da die Annahmen über die Umgebung oft nicht mit der Umgebung zur Betriebszeit übereinstimmen, muss die Verifikation der Systemeigenschaften auch zur Laufzeit vorgenommen werden. Allerdings erkennen damals aktuelle Methoden, z.B. Design-by-Contract [Mey92, LBR06, BLS05], monitor-orientierte Programmierung [CR07], regelbasierte Systemen [BGHS04, BRH07]) die Verletzung von Systemeigenschaften nur dann, wenn solche Verletzungen auftreten. In diesem Fall, kommt eine Reaktion des externen Operators zu spät, da die kritische Situation nicht mehr verhindert werden kann. Der Projektpartner TU Clausthal hat Mechanismen und Methoden zur Laufzeitabsicherung einer Systemspezifikation untersucht und entworfen. Das Ziel war, kritische Situationen, verursacht durch die Umgebungsdynamik, zur Betriebszeit des Systems möglichst gut vermeiden zu können.

## 1.5 Zusammenarbeit mit anderen Stellen

In der Anfangsphase des Projekts gab es eine Zusammenarbeit mit Aethon Inc. Pittsburgh, die einer der führenden Anbieter von Robotik-Lösungen für Transport-Anwendungen im Krankenhäusern ist ([www.aethon.com](http://www.aethon.com)). Die in Pittsburgh, Pennsylvania, USA, ansässige Firma wurde 2001 als Spin-off der Carnegie Mellon University gegründet. Das von Aethon entwickelte und produzierte „Automated Robotic Delivery System“ umfasst

Roboter (sogenannte „TUGs“), die autonom im Krankenhaus navigieren und Transportaufgaben insbesondere zwischen Stationen und im menschlichen Umfeld wahrnehmen. Aethon verkauft jährlich knapp 100 Systeme und ist mittlerweile in über 100 Krankenhäusern in den USA mit Robotern aktiv. Die Robert Bosch Venture Capital GmbH hielt zum Zeitpunkt der Antragsstellung ca. 10% Anteile an Aethon, Inc und stand in einem engen Forschungsaustausch. In diesem Rahmen stand Aethon, Inc. als assoziierter Partner dem Projektvorhaben in der Anfangsphase zur Seite, um insbesondere kundennahe Anforderungen und Erfahrungen aus dem Feldeinsatz einzuspeisen.

## 2 Eingehende Darstellung

Im Projekt iserveU wurden Aspekte der intelligenten Transportrobotik erforscht und prototypisch entwickelt als Grundlage weitergehender Produktentwicklungen. Das Konsortium hatte dabei das gesamte System im Blick, so dass die verschiedenen Bereiche der Software, Hardware, Systemintegration und Applikation adressiert wurden. Im Bereich der Hardware lag der Fokus der Entwicklung auf einer neuen Ultraschall Sensortechnologie, einer neuen Beacon-basierten Radarortungstechnologie sowie dem Roboterprototypenaufbau. Im Bereich der Software lag der Fokus auf der durchgehenden modellgetriebenen Entwicklung, einer auf die neue Sensorik abgestimmten Ortungstechnologie und einer Missions- und Bewegungsplanung, um komplexe Aufgaben im Bereich der intelligenten Transportlogistik zu realisieren. Im Bereich der Applikation stand die Mensch-Maschine Interaktion, die Absicherung des Robotiksystems durch Remote-Management und die Entwicklung einer domänenspezifischen Beschreibungssprache für die Aufgabenprogrammierung im Vordergrund. Das entwickelte Gesamtsystem wurde im Rahmen mehrerer Tests in einem realen Krankenhaus evaluiert und abschließend erfolgreich demonstriert.

### 2.1 Verwendung der Zuwendung und des erzielten Ergebnisses im Einzelnen, mit Gegenüberstellung der vorgegebenen Ziele

#### 2.1.1 AP1: Szenarien und Gesamtarchitektur

**Ziel:** Dieses Arbeitspaket, welches zu Beginn des Projektes steht, erarbeitet die konkrete Konzeption und Vorgehensweise. Es werden die konkreten Handlungsszenarien in der Domäne Krankenhaus, die man mit dem ITA bedienen möchte, definiert. Aus den sich daraus ergebenden Anforderungen wird die Gesamtarchitektur des Prototypsystems entworfen.

**Inhalt:** Dieses Arbeitspaket beschäftigt sich mit der Gesamtarchitektur des ITA. Hierzu ist es notwendig, die Domäne Krankenhaus detailliert zu durchleuchten und so die Systemgrenzen des ITA detaillierter definieren zu können. Daraus lassen sich dann wichtige Anforderungen an das System ableiten. Gleichzeitig ergibt sich auch ein klareres Bild, mit welchen Personen der ITA in welchen Situationen interagiert bzw. interagieren muss. Für eine Definition der Gesamtarchitektur eines späteren Produktes ist die Kenntnis der relevanten Normen und Gesetze notwendig. Da für den Bereich Servicerobotik noch keine gesonderte Norm existiert, müssen hier die relevanten Normen zusammengetragen und auf ihre Anwendbarkeit geprüft werden. Wichtig ist hier auch die Berücksichtigung

der am Stuttgarter IPA Institut entstehenden Norm für die Servicerobotik. Aus dem Domänen- und Normenverständnis muss eine Produktgesamtarchitektur abgeleitet werden und eine Definition der Systemkomponenten erfolgen. Zum Ende der Projektlaufzeit kann aufbauend auf den Erkenntnissen hinsichtlich der Normen (Task 1.3), der Sensorik (AP2) und des Interaktionskonzepts (AP4) eine Elektrik- und Elektronikarchitektur für ein mögliches späteres Produkt entstehen.

**Ergebnis:** Im Rahmen des ersten Arbeitspakets wurden in der Anfangsphase des Projekts die Grundlagen für die Realisierung des ITA gelegt. Der erste Schritt war die Definition der Systemgrenzen und die Ausarbeitung der Use-Cases, die behandelt werden sollten (D1.1). Die betrachteten und ausgearbeiteten Use-Cases waren:

1. (Ad hoc) Transportaufgaben
2. Folgen von Personen
3. Führen von Personen
4. Remote Management zur Unterstützung bei unvorhergesehenen, nicht automatisch lösbaren Situation

Es wurden einige besondere Fälle als sekundäre Aspekte in den Use-Cases ausgenommen so z.B. automatisches Laden, Wartung, Sperr- und Gefahrenzonen. Diese sind für eine mögliche Produktentwicklung zwar wichtig, hätten aber den Rahmen des Projektes gesprengt und wurden deshalb nicht betrachtet. Die primär zu behandelnden Ausnahmen und Sonderfälle wurden explizit in den den Use-Case Beschreibungen aufgenommen.

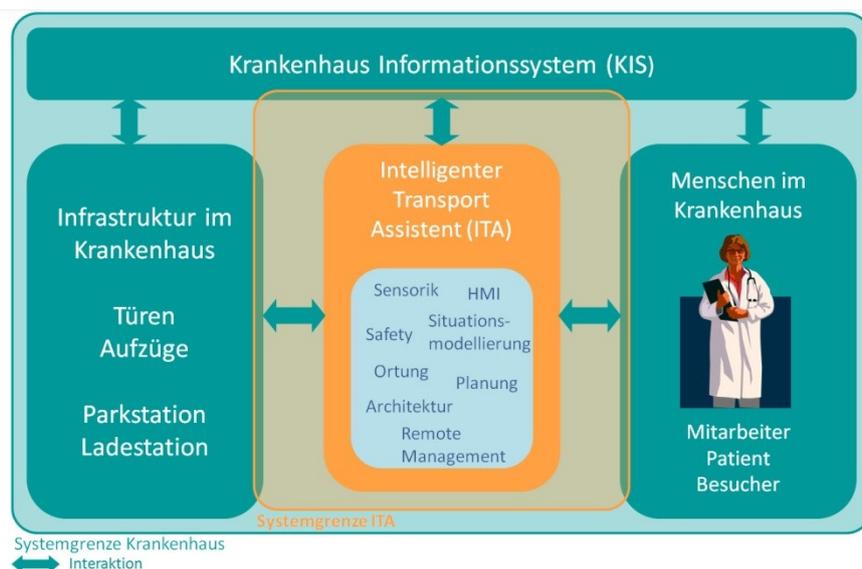


Abbildung 2.1: Systemgrenzen des intelligenten Transportassistenten (ITA) im Krankenhaus

Auf Basis der Anforderungen, die sich aus den Use-Cases ableiteten, wurde die Systemarchitektur entwickelt (D1.2). Dies umfasste unter anderem den schematischen Aufbau des Prototypen, die Verteilung der einzelnen Funktionen auf Komponenten und die Definition der Schnittstellen (Software und Hardware). Diese Separierung ermöglichte eine parallele Entwicklung der Hard- und Softwarekomponenten mit iterativer Integration der jeweiligen Fortschritte. Die initial konzipierte Systemarchitektur wurde im Lauf des Projekts nur noch leicht an wechselnde Anforderungen angepasst. Abbildung 2.2 zeigt einen Überblick über die Software-Systemarchitektur des ITA in Schichten. Die Realisierung der Software-Systemarchitektur modelliert in der SmartMDS Toolchain zeigt Abbildung 2.3.

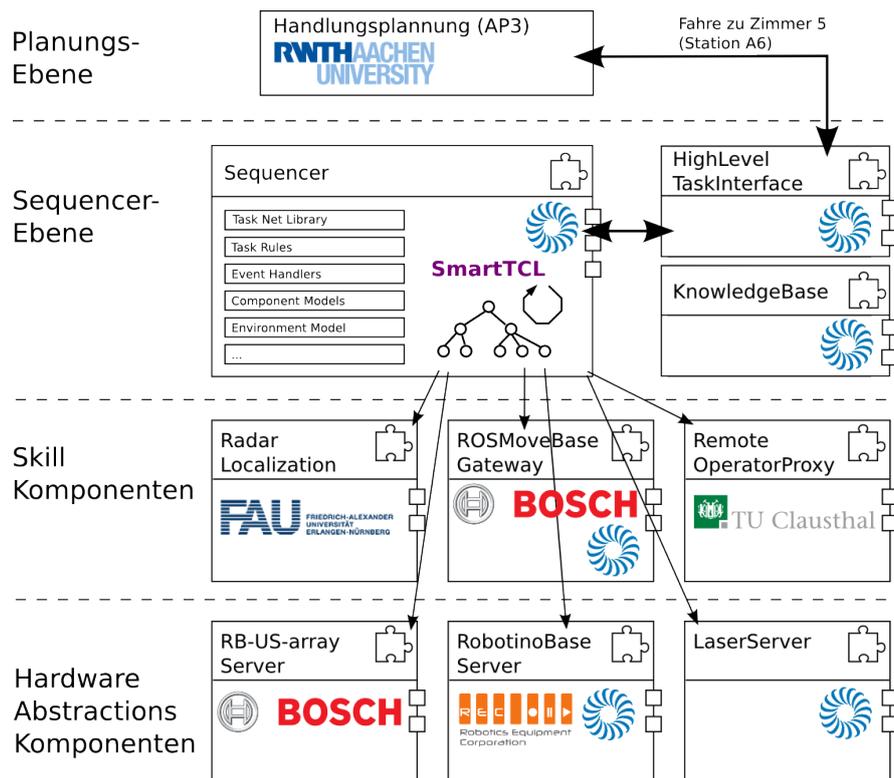


Abbildung 2.2: Überblick über die Software-Systemarchitektur des ITA in Schichten

Um die Entwicklung des Systems durch die Anforderung für eine nachfolgende Produktisierung zu leiten, wurden alle relevanten Normen gesichtet und die Konsequenzen erfasst (D1.3). Ein wichtige Voraussetzung für die Einführung eines Transportroboters in den europäischen Wirtschaftsraum ist die Erfüllung der Maschinenrichtlinie 2006/42/EG. Bei Erfüllung der entsprechenden (harmonisierten) Normen für den entsprechenden Gerätetyp zusammen mit einer Risikobeurteilung wird von Einhaltung der Maschinenrichtlinie ausgegangen (Vermutungswirkung), so dass es ausreicht, sich auf diese zu konzentrieren. Die Umsetzung der geforderten Risikobeurteilung muss nach DIN EN ISO 12100 erfolgen. Aus der Analyse möglicher Risiken beim Betrieb wurde ein Sicherheitsintegri-

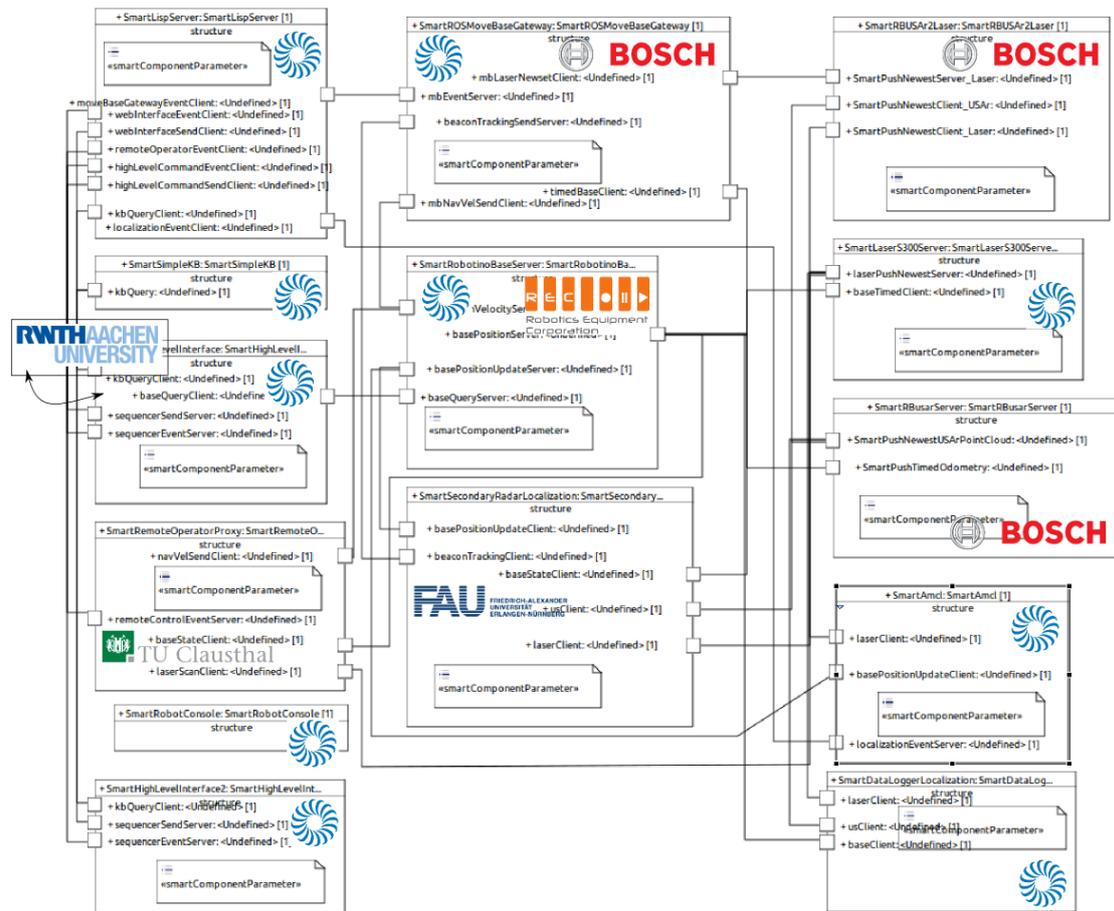


Abbildung 2.3: Realisierung der Software-Systemarchitektur, Systemintegrationsmodell modelliert in der SmartMDS Toolchain.

tätslevel (SIL) der Stufe 3 identifiziert. Das verantwortliche Szenario ist, dass der Roboter die Treppe herunter fällt und dabei auf einen Menschen stürzt. Durch das Gewicht des Roboter ist ein Todesfall nicht unwahrscheinlich.

Als Basis für eine weiterführende Produktisierung wurde gegen Ende des Projekts eine Elektrik- und Elektronikarchitektur (E/E Architektur) des entwickelten Roboterprototypen erstellt.

### 2.1.2 AP2: Kostengünstige Ortungs- und Navigationssensorik

**Ziel:** Erforschung und Realisierung einer kostengünstigen, anwendungsorientierten Ortungs- und Navigationssensorik sowie die Realisierung einer Navigationslösung für menschengeprägte Umgebungen.

**Inhalt:** Grundvoraussetzung für die effektive und effiziente Navigation von Servicerobotern ist deren eindeutige Positionsbestimmung. Je nach Aufgabenstellung und Kontext sind unterschiedliche Sensoren notwendig. Erst die Kombination verschiedener Sensorsysteme erlaubt eine hinreichend zuverlässige Ortung und Erfassung der Umgebung. Die heute übliche Positionsbestimmung auf Basis von Lasersystemen stößt an Grenzen, wenn notwendige Referenzpunkte nicht mehr sichtbar sind. Dies ist in öffentlichen Räumen mit Publikumsverkehrs häufig der Fall. Funkortungssysteme stellen eine geeignete Alternative dar.

Eine weitere Grundvoraussetzung für eine sichere Navigation ist die zuverlässige Erkennung von Hindernissen. Dafür soll ein Hinderniserkennungssystem basierend auf einer neuen 3D Ultraschallsensorik entwickelt werden. Diese Sensorik hat den grundsätzlichen Vorteil gegenüber 2D Laser Sensoren, dass sie nicht nur Tiefeninformationen in einer Sicht-ebene liefert sondern auch Höheninformationen in einem ganzen Bereich und damit eine 3D-Sicht auf die Umgebung. Dadurch wird eine sichere Hinderniserkennung ermöglicht.

Auf Basis einer zuverlässigen Lokalisierung und Hinderniserkennung soll dann ein Navigationsalgorithmus entwickelt werden, der den Roboter in die Lage versetzt, in menschlich geprägten Umgebungen wie Krankenhäusern Transportaufgaben auszuführen. Dabei fließen die Erkenntnisse zu Mensch-Roboter Interaktion aus AP4 und die Interaktion mit der Verhaltensplanung aus AP3 mit ein.

**Navigation und Hinderniserkennung (2.1):** Entwicklung und Evaluation eines Hinderniserkennungssystems auf Basis der neuen 3D Ultraschallsensorik. Realisierung einer Navigationslösung zur Ausführung von Transportaufgaben unter Berücksichtigung von Mensch-Maschine Interaktionsaspekten.

**Relativortung (2.2):** Erforschung einer bilateralen Funkortungstechnik mit dem Ziel, Hardware und Software zu entwickeln, die Roboter-Folgen-Führen Funktionalitäten ermöglichen. Dafür soll ein 24 GHz Funkortungssystem mit einer SIMO-Antennenstruktur und zugehörigem personengetragenen, kompakten Transponder („Follow-me-Beacon“) entwickelt werden. Die Unsicherheiten der Entfernungsmessung sollte dabei weniger als 10 cm und die Unsicherheit der Winkelmessung weniger als 5° betragen.

**Lokale Funkortung innerhalb eines Gebäudes (2.3):** Entwurf neuartiger Sensorfusionsalgorithmen zur Lokalisierung innerhalb einer Gebäudekarte mit Ortungsgenauigkeiten besser als 10 cm und Orientierungsunsicherheit kleiner als 5°. Die Ortungstechnik soll Sensor-Rohdaten des 24 GHz Sekundärradars (vom Projektpartner Symeo GmbH), des Ultraschall-Arraysensors (vom Projektpartner Robert Bosch

GmbH) und Odometriedaten des Roboters verwenden. Hohe Priorität hat die Entwicklung von Konzepten mit minimalem Infrastruktur-Aufwand.

**Funknetzwerk (2.4):** Realisierung eines Master-losen, verteilten Ad-Hoc Kommunikations- und Synchronisationsnetzwerks. Die Sekundärradar Knoten sollen anhand dieses Netzwerkes in der Lage sein, sich zu synchronisieren und Daten auszutauschen

**Ergebnis:** *Hinderniserkennung und Navigation:* Von Bosch wurde die Hinderniserkennung für die Navigation entwickelt (D2.1.1). Dafür sollte die bei Bosch entwickelte, neue 3D-Ultraschallsensorik verwendet werden. Diese bringt potenzielle Vorteile gegenüber am Markt verfügbarer Sensorik mit sich, insbesondere was den Wahrnehmungsbereich, die Kosten und die Sicherheitszertifizierbarkeit angeht. Das Messprinzip beruht darauf, mithilfe kleiner Sender Schallimpulse im nicht-hörbaren Ultraschall-Frequenzbereich zu emittieren und mit einem Array von kleinen Mikrofonen (US-Array) die an der Umgebung reflektierten Schallwellen wieder zu empfangen. Da sich die Mikrofone im Array an leicht unterschiedlichen Stellen befinden, können aus den Laufzeiten und Laufzeitunterschieden die Distanzen und Richtungen zu Reflektionspunkten in der Umgebung mithilfe einer Technik namens Beamforming berechnet werden. Zur Verbesserung der Messergebnisse werden die Reflektionspunktschätzungen zeitlich und räumlich gefiltert. Diese Sensordatenverarbeitung (siehe Abbildung 2.5) wurde für iserveU implementiert und die Hardware aufgebaut. Beispiele für Messergebnisse sind in Abbildung 2.6 zu sehen. Das Messsystem besteht dabei aus mehreren Komponenten und ist in Abbildung 2.4 abgebildet. Die Komponenten sind wie folgt:

- Mehrere Mikrofone zum Aussenden von Schallwellen
- Eine Triggerbox, welche die Aussendung der Schallwellen steuert und zeitlich synchronisiert
- Zwei Ultraschallarrays zum Empfangen der reflektierten Schallwellen
- Zwei FPGAs für die Berechnung der Reflektionspunkte

Der Ultraschallsensor wurde im Laufe des Projekts überarbeitet, um eine günstigere Fertigung mittels Mems-Sensoren zu ermöglichen. Dies gelang nach anfänglichen Fertigungsproblemen bei der Bestückung der Sensoren. Die Ultraschallsensoren auf Mems-Basis verbesserten auch die Qualität der Messergebnisse: Weniger Messrauschen, dichtere Messungen, Reichweite größer 20m, Blickwinkel größer 180°.

Die Funktionalität wurde erst an einem statischen Aufbau entwickelt und getestet und dann später auf den Roboter integriert. Es konnte gezeigt werden, dass Hindernisse zuverlässig erkannt werden und daraus ein Stoppsignal generiert wird. Ein Problem, das sich in manchen Situationen zeigte war, dass, aufgrund des prototypischen Charakters des Sensorsystems, die Latenzen teilweise sehr hoch sein können (ca. 200ms). Dies kann zur Folge haben, dass bei sehr schnellen dynamischen Objekten in der Umgebung nicht

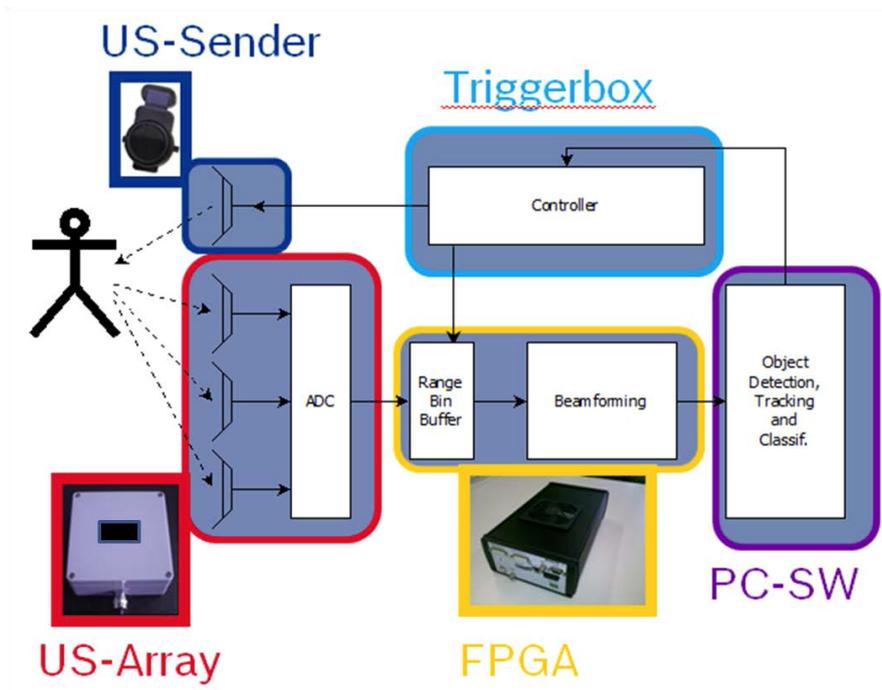


Abbildung 2.4: Komponenten des Ultraschallsensorsystems

mehr rechtzeitig vollständig abgebremst werden kann. Dies wird man bei einer Produktisierung des Sensorsystems verbessern, auch um eine Sicherheitszertifizierung erreichen zu können. Für die Hinderniserkennung werden die Abstände zu Hindernissen in der Umgebung überwacht. Wird kein Weg um die Hindernisse gefunden, bleibt der Roboter stehen. Darüber hinaus findet eine zusätzliche Verifikation statt, so dass es bei einer Unterschreitung eines Sicherheitsabstandes zu einem Sofortstopp kommt (siehe AP5). Beim Vergleich von 3D-Ultraschallsensorik und Laser zeigte sich wie erwartet, dass die Messergebnisse aufgrund der unterschiedlichen Messprinzipien teilweise orthogonal sind. So liefert ein 2D-Laserscanner eine dichte Punktschätzung in einer Sichte ebene, wohingegen ein 3D-Ultraschallsensor eine spärliche Punktwolke mit zusätzlicher Höheninformation liefert. Zusätzlich kann ein Ultraschallsensor auch Glasscheiben zuverlässig erkennen. Die besten Ergebnisse könnten deshalb mit der Kombination beider Sensoren erzielt werden.

Für die Realisierung der Roboternavigation entwickelte Bosch eine Bewegungsplanungskomponente (D2.1.2). Diese war angekoppelt an die Hinderniserkennung, die Verhaltensplanung, die Lokalisierung und die Umgebungskarte. Es wurde ein hierarchischer Ansatz verfolgt. Die obere Planungsschicht, globaler Planer genannt, plant Pfade zu den Zwischenzielen, welche von der Verhaltensplanung vorgegeben werden. Für diese Pfade werden dann von der unteren Planungsschicht, genannt lokaler Planer, Trajektorien mit einem Zeithorizont von 1-2 Sekunden geplant. Die Entwicklung der Navigationskomponente basierte auf der Vorlage des ROS<sup>1</sup> Navigation Stacks. Der globale Planer führt die

<sup>1</sup>Robot Operationg System

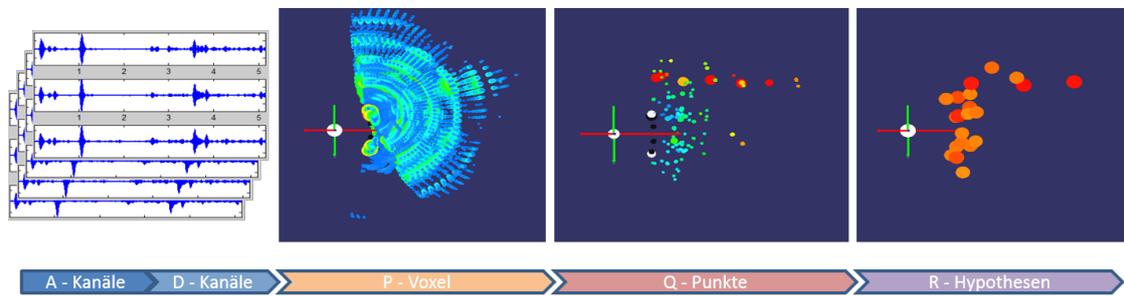


Abbildung 2.5: Visualisierung der Verarbeitungsschritte bei der Ultraschallmessung. Ausgehend von den empfangenen Signalen werden anhand der Laufzeiten und -unterschiede die Reflexionspunkte berechnet und zeitlich und räumlich gefiltert, um zu den Messungen zu gelangen

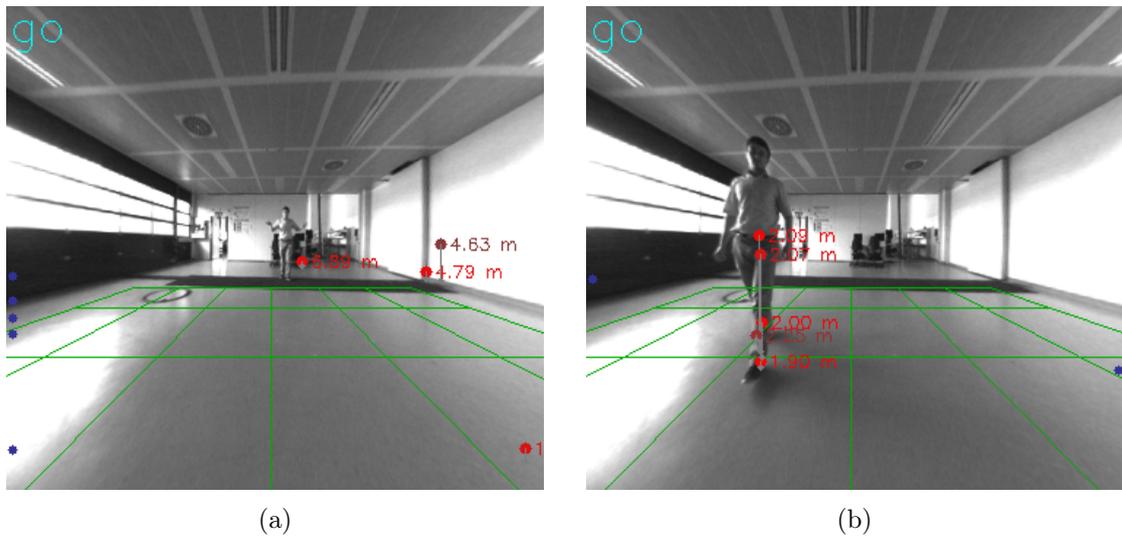


Abbildung 2.6: Beispielmessungen des 3D-Ultraschallsensors. Reflexionspunkte und deren Entfernungen sind in rot dargestellt

Planung mithilfe des Dijkstra-Algorithmus auf einer Gridmap aus, welche Informationen aus der Umgebungskarte und den wahrgenommenen Hindernissen enthält. Der lokale Planer verwendet einen erweiterten Dynamic Window Approach [TFB97], um Trajektorien zu planen, welche möglichst optimal die Pfade des globalen Planers abfahren und dabei Kollisionsfreiheit mit Hindernissen sicherstellen. Die Bewegungsplanung wurde so erweitert, dass nicht nur die Länge des geplanten Weges optimiert wird, sondern Teile der Ergebnisse der Mensch-Maschine Interaktion aus AP4 bei der Planung berücksichtigt werden. So werden z.B. sozial erwartete Abstände zu Menschen und Hindernissen eingehalten soweit es die Situation zulässt.

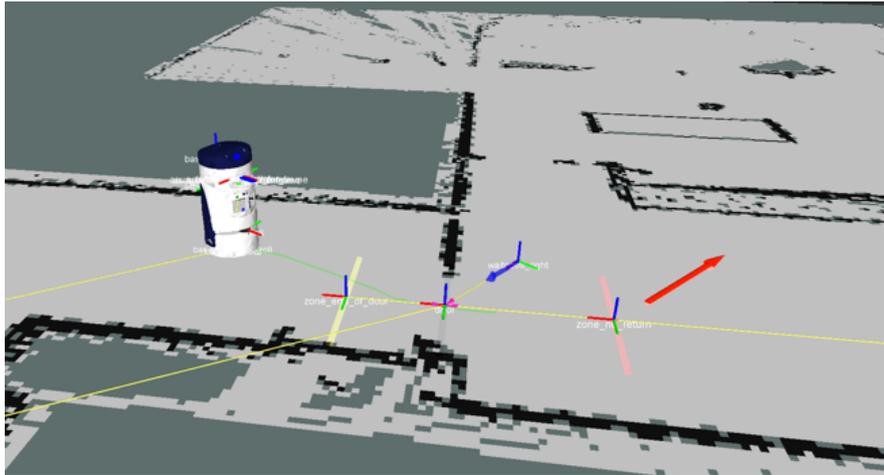


Abbildung 2.7: Visualisierung der Bewegungsplanung

*Relativortung:* Für Task 2.2 wurde von Symeo die HW für die einzelnen Knoten und das tragbare Tag entworfen. Für die ortsfesten Knoten wurde ein SIMO System mit 16 Empfangskanälen für die diskrete separate Antenne entworfen und aufgebaut. Dabei wurde besonders Augenmerk auf die Simulation der Übergänge für die 24 GHz Leitungen gelegt. Durch die externe Antenne können verschiedene Konfigurationen (Array, Ring, usw.) getestet werden. Abbildung 2.8 zeigt das Blockschaltbild des entworfenen Systems. Das realisierte RF Board ist in der Abbildung 2.9 zu sehen. Als tragbare Einheit wurde ein System mit nur einem Sendekanal und zwei Empfangskanälen realisiert (siehe 2.10.). Dafür wurde das komplette Design simuliert und an die aktuellen HF Chipsätze angepasst. Es wurden Formfaktor, Stromverbrauch und Power-up sequencing optimiert. Auch die Herausforderung im Bereich Taktverteilung zum HF Board wurde zufriedenstellend gelöst.

In Task 2.2 Relativortung wurde von FAU Erlangen-Nürnberg die von Symeo realisierte Hardware in Betrieb genommen. Bei der Hardware handelt es sich um ein 24 GHz FMCW 8-Empfangskanal SIMO Sekundärradarsystem, vgl. Abbildung 2.11 a). Für die Radarknoten wurden optimierte Antennenarrays realisiert, um neben der Abstandsmessung auch Winkelmessung zu ermöglichen. Dazu wurde ein Algorithmus basierend auf einem dreidimensionalen räumlichen Optimalfilter entwickelt, vgl. Abbildung 2.11 b).

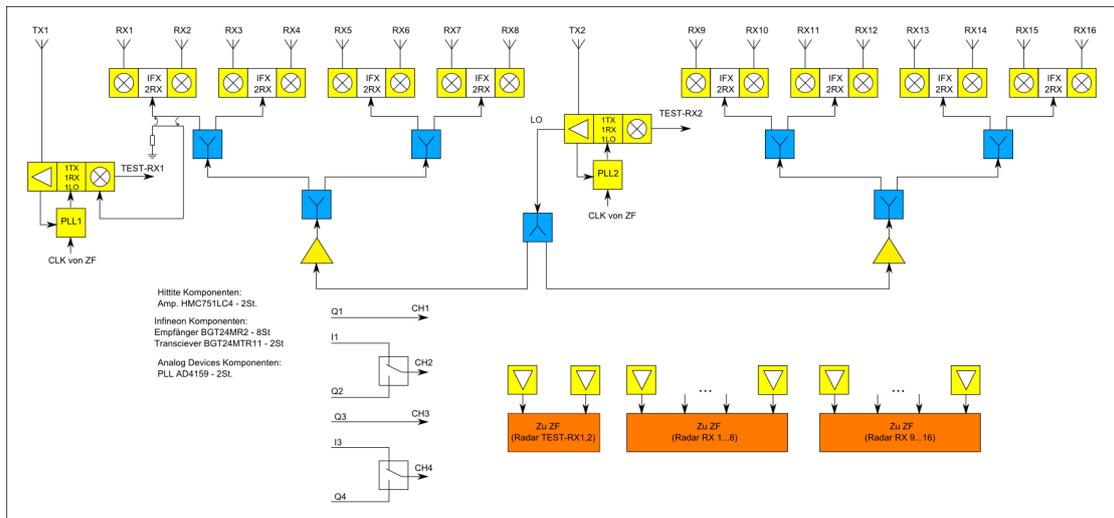


Abbildung 2.8: Blockschaltbild RF Frontend 16 Empfangskanäle.

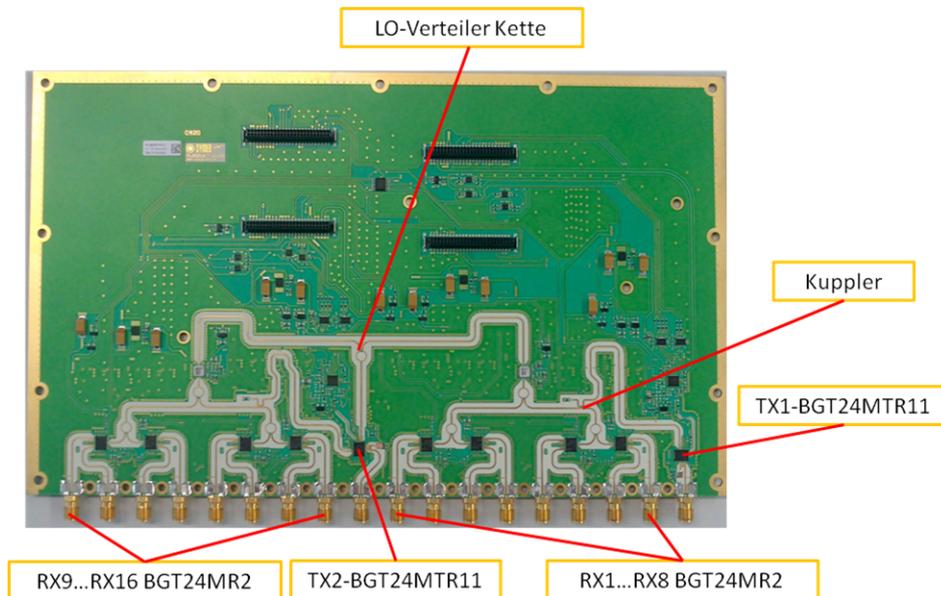


Abbildung 2.9: Realisiertes RF-board.



Abbildung 2.10: Realisiertes Kompaktmodul.

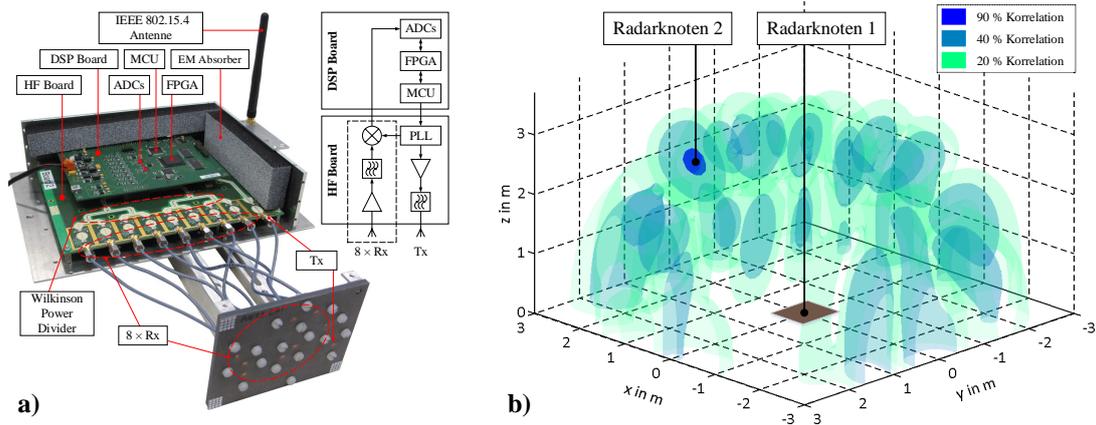


Abbildung 2.11: a) Bild und vereinfachtes Blockdiagramm eines Sekundärradarknotens. b) Ortungsergebnis mithilfe eines dreidimensionalen räumlichen Optimalfilters.

Der Algorithmus wurde auf einer eingebetteten Plattform implementiert, wodurch eine Messrate von ca. 30 Hz erreicht werden konnte.

Während in der ersten Projektphase (Labormuster) ein großer Radarknoten zur Ortung einer Person relativ zum fahrenden Roboter benutzt wurde, wurde für die zweite Projektphase (Funktionsmuster und Evaluation) ein tragbarer kompakter Radarknoten („Follow-me-Beacon“) entwickelt, vgl. Abbildung 2.15 c). Zur Ortung des von der Person getragenen Follow-me-Beacons wurden die vom mobilen Radarknoten gemessenen Abstände und Winkel in einem erweiterten Kalman-Filter zu einer Personenposition relativ zum Roboter verrechnet. Zusätzlich werden Geschwindigkeit und Drehgeschwindigkeit

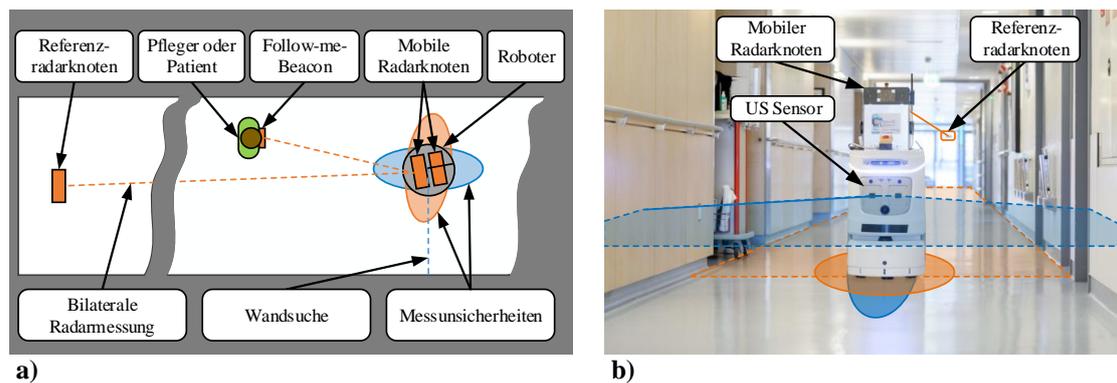


Abbildung 2.12: a) Vereinfachte schematische Darstellung des Ortungskonzepts (Draufsicht). b) Ortungskonzept im Flur des Katharinenhospitals.

der Person aus den Radardaten ermittelt, um eine höhere Robustheit gegen Störungen zu erreichen. Ein exemplarisches Ergebnis der Relativortung einer Person während der Fahrt eines mobilen Roboters ist in Abbildung 2.16 c) dargestellt. Bei dieser Messreihe wiesen 68,3 % der Messungen einen Abstandsmessungsfehler vom Roboter zur Person unter 9,8 cm und einen Winkelfehler unter  $2,5^\circ$  auf. 95,4 % der Messungen hatten eine Abweichung unterhalb 29,3 cm bzw. unterhalb  $4,2^\circ$ . Angesichts der sehr schwierigen Messsituation – schmaler Gang mit vielen Mehrwegeausbreitungen – stellt dies ein sehr gutes Ortungsergebnis dar und war ausreichend für die erfolgreiche Navigation des Roboters.

*Lokale Funkortung innerhalb eines Gebäudes* Das Grundkonzept des in diesem Projekt realisierten Ortungssystems ist in Abbildung 2.12 dargestellt. Am Fluranfang und -ende in der HNO-Abteilung im Stuttgarter Katharinenhospital wurde jeweils ein statischer Referenzradarknoten angebracht. Die Position der Referenzradarknoten wurde mithilfe eines präzisen optischen Tachymeters eingemessen. Auf dem mobilen Roboter befanden sich 2 weitere Radarknoten. Ein Radarknoten (der Follow-me-Beacon) wurde von einer Person getragen, die relativ zum Roboter geortet wurde.

Die Position des Roboters in Fahrtrichtung wird hauptsächlich anhand Sekundärradar-messungen ermittelt. Die Position quer zur Fahrtrichtung und die Orientierung wird durch Abstands- und Winkelmessungen zu den Wänden unterstützt. Dafür wird die vom Ultraschall-Array gelieferte „Punktwolke“ verwendet. Die in der Teststrecke vorhandenen Wände lassen sich als Umgebungsmerkmale identifizieren und deren Position relativ zum Roboter ermitteln. Diese Information wird mithilfe einer Raumparametrisierung anhand der Hough-Transformation ermittelt. Der Ablauf ist in Abbildung 2.13 graphisch dargestellt. Auf dem Roboter ist ein Laserscanner standardmäßig eingebaut. Der vom Laserscanner gelieferte „Laserscan“ besitzt eine höhere Abtastungsdichte als die „Punktwolke“ des Ultraschallsensors, allerdings nur in einer Sichtebene. Die im Rahmen des Projektes realisierten Algorithmen zur Wandsuche können sowohl Laserscandaten als auch Ultraschalldaten verarbeiten. Anhand der Algorithmen lassen sich die Wände mit sehr guter Genauigkeit identifizieren, was ein wichtiger Aspekt für der Qualität der Roboterortung, besonders für die Orientierungsschätzung, ist.

Die Funktionen zur Entfernung- und Winkelmessung vom Sekundärradar wurden embedded auf den Knoten realisiert. Dabei werden die im FPGA realisierten Funktionen (z.B. AD/Wandlerdaten) über eine Schicht im eingebetteten Linux der Anwendungsebene zur Verfügung gestellt. Die Berechnung der ungefilterten Entfernung- und Winkelwerte wurde in einem separaten Thread (Multi-Channel Spectral estimation) umgesetzt. Dieser Thread übernimmt auch die Ablaufsteuerung und Koordination der Messungen. Die ermittelten Entfernung- und Winkelwerte werden dann an die Fusion Engine zur weiteren Verarbeitung übergeben. Abbildung 2.14 zeigt die Architektur des eingebetteten Systems.

Zur multimodalen Sensorfusion wurde ein erweitertes Kalman-Filter realisiert. Mithilfe des Filters kann die komplette 2D Pose des Roboters (Position und Orientierung) aus den Sekundärradar-, Wandsuche- und Odometriedaten bestimmt werden. Die Pose des mobilen Roboters wird im Koordinatensystem der als Occupancy-Grid-Map vorhandenen Gebäudekarte bestimmt. Eine Voraussetzung für das Filter ist die Kenntnis der absoluten Pose der Referenzradarknoten und der relativen Pose der mobilen Radarknoten im

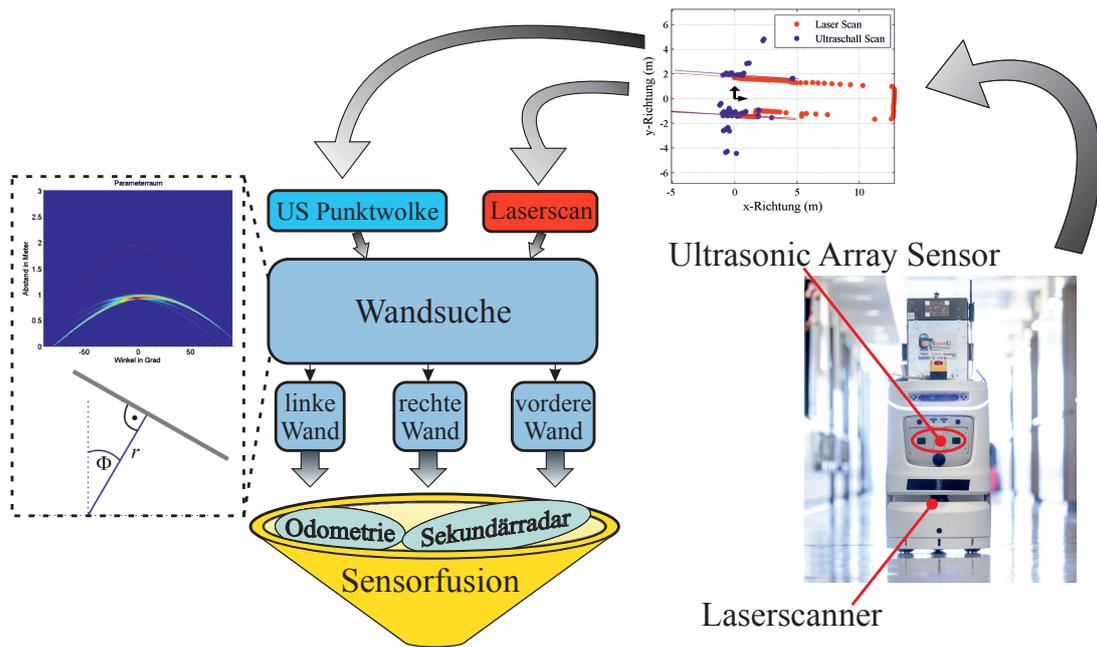


Abbildung 2.13: Fusion der Wandsuche im Lokalisierungsalgorithmus

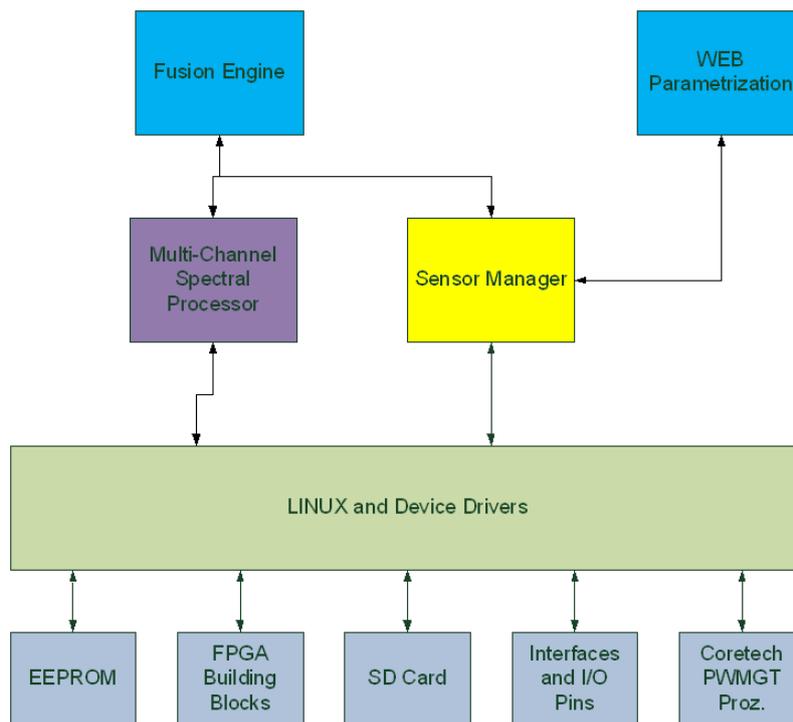


Abbildung 2.14: Embedded Software Architektur.

Roboterkoordinatensystem. Es war von zentraler Bedeutung, die Messunsicherheit der einzelnen Sensoren im Filter unter Berücksichtigung der jeweiligen Besonderheiten korrekt abzubilden. Da die Messungen stets in für die Sensorik sehr anspruchsvollen Räumen stattfanden (viele Mehrwegeausbreitungen an Wänden, Objekten und Anwesenheit von Personen), war es auch besonders wichtig, dass das Filter fehlerhafte und gestörte Messungen erkennen und filtern konnte. Das realisierte Filter ist darüber hinaus in der Lage, mit einer völlig unbekanntem Roboterpose zu starten und sich bei einer Störung selber neu zu initialisieren, was die Zuverlässigkeit des Systems deutlich erhöht. Das Filter wurde auf einer eingebetteten Plattform umgesetzt, so dass eine Ortung in Echtzeit möglich war. Das von FAU und Symeo realisierte Ortungssystem wurde in Zusammenarbeit mit Bosch und HSU in den finalen Demonstrator integriert.

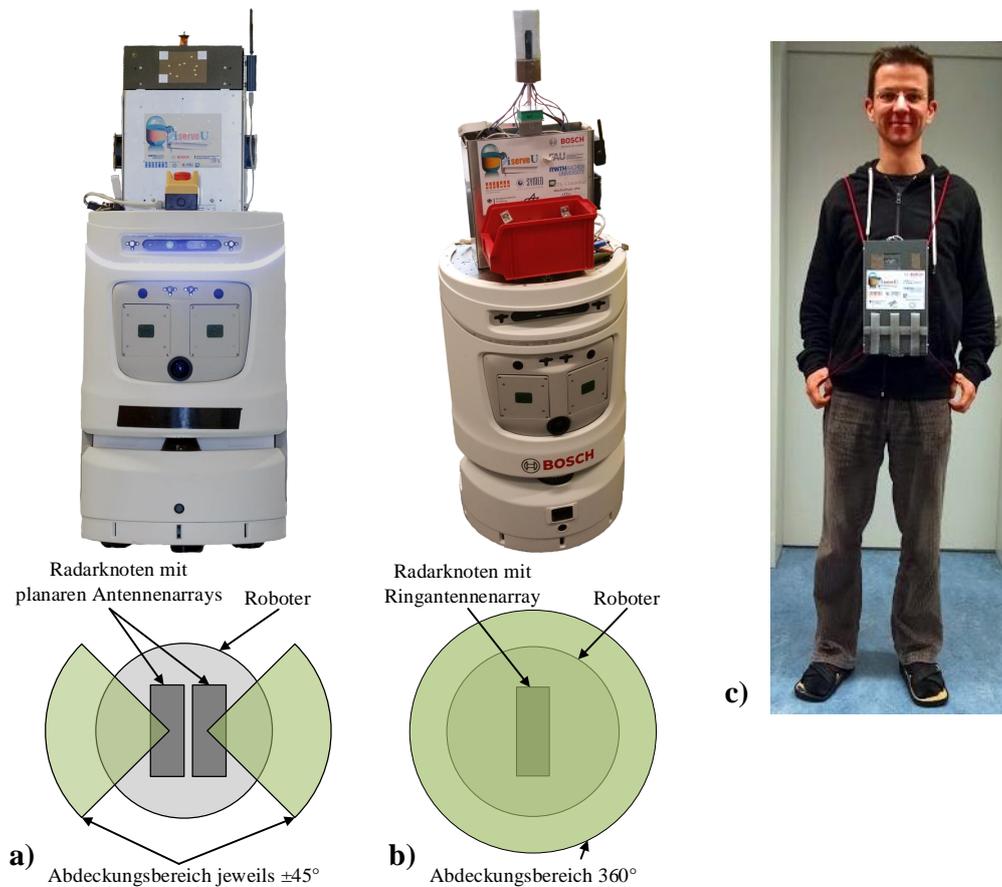


Abbildung 2.15: Untersuchte Ortungskonzepte: a) Auf dem Roboter befinden sich 2 Radarstationen mit einer Abdeckung von jeweils  $\pm 45^\circ$ . b) Auf dem Roboter befindet sich nur eine Radarstation mit einer Abdeckung von  $360^\circ$ . c) Person mit dem kompakten tragbaren Radarknoten (Follow-me-Beacon).

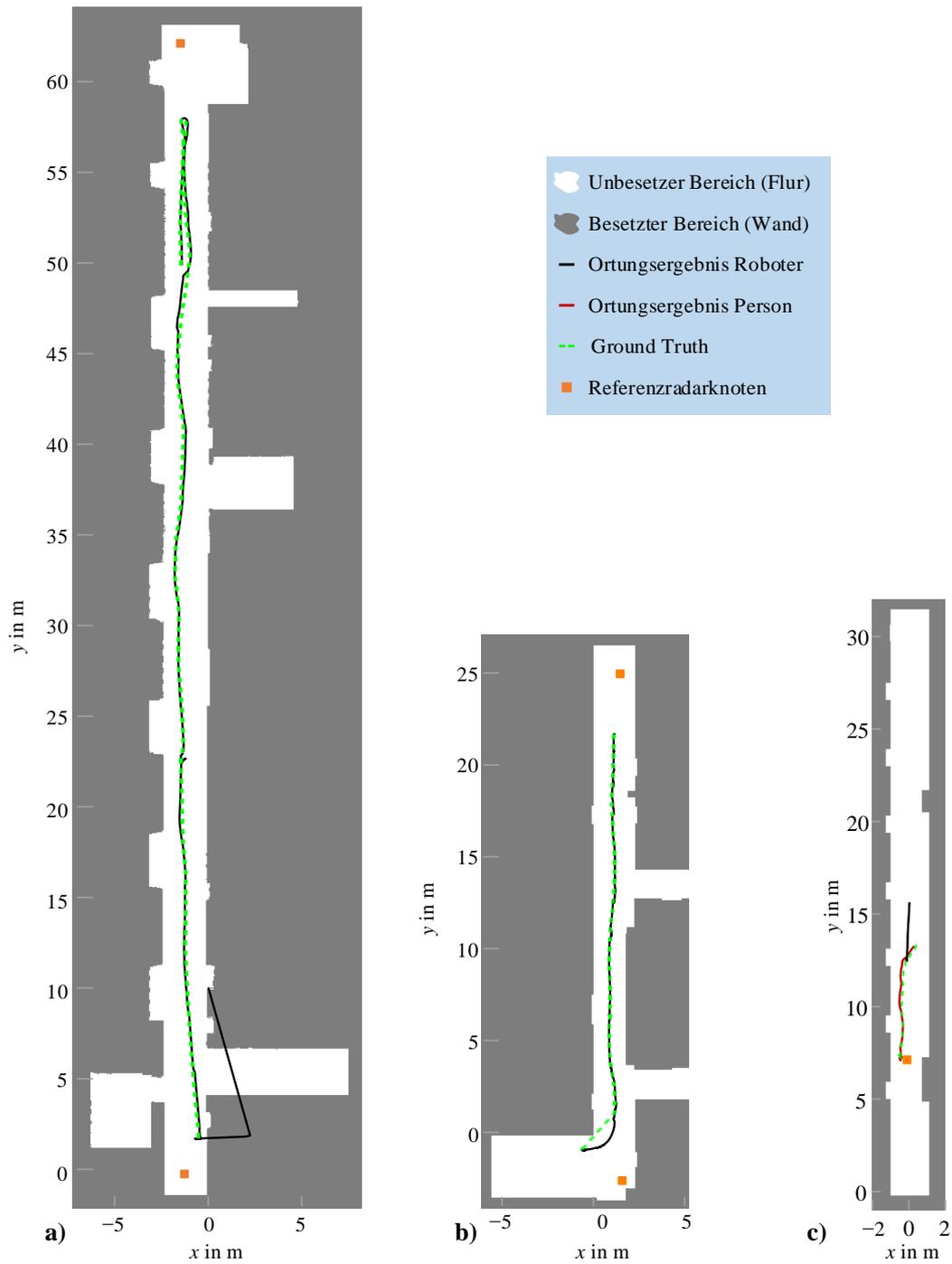


Abbildung 2.16: Ortungsergebnis in 3 verschiedenen Szenarien: a) Roboterortung im Katharinenhospital in Stuttgart. b) Roboterortung am Bosch-Standort in Renningen. c) Ortung einer Person relativ zum Roboter mithilfe des Follow-me-Beacons an der Universität Erlangen-Nürnberg.

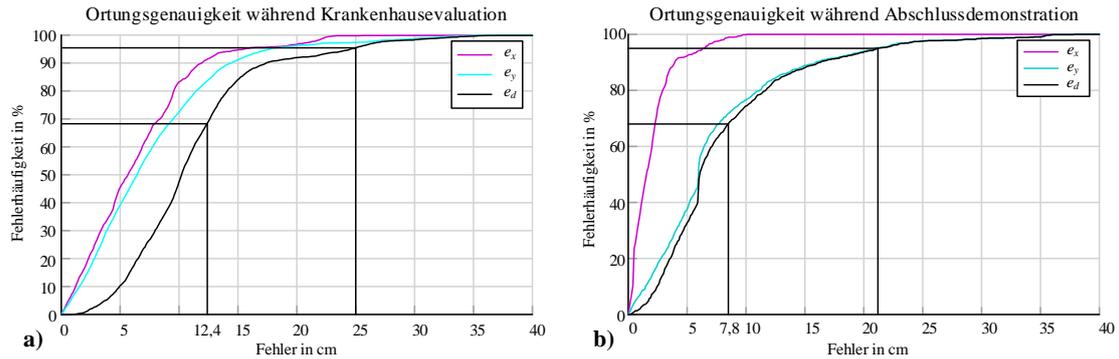


Abbildung 2.17: Kumulierte Fehlerhäufigkeit jeweils einer Messung im Rahmen der a) Krankenhausevaluation und b) Abschlussdemonstration.

Während der Evaluation im Katharinenhospital wurde das in Abbildung 2.15 a) dargestellte Systemkonzept verwendet. Bei diesem Konzept befanden sich 2 Radarknoten mit planaren Antennenarrays mit einem Abdeckungsbereich von jeweils  $\pm 45^\circ$  in Azimutrichtung auf dem Roboter. Das Ortungsergebnis mit dem entwickelten System in der Krankenhausumgebung ist in Abbildung 2.16 a) dargestellt. Die zugehörige kumulierte Fehlerhäufigkeit ist in Abbildung 2.17 a) dargestellt. Wie der Abbildung zu entnehmen ist, war der 2D Ortungsfehler kleiner als 12,4 cm bei 68,3 % der Messungen bzw. kleiner als 25,0 cm bei 95,4 % der Messungen. Da keine Referenz für die Roboterorientierung vorhanden war, konnte die Genauigkeit nicht quantitativ ausgewertet werden. Qualitativ war die Genauigkeit der Roboterorientierung für eine erfolgreiche Navigation in jedem Fall ausreichend. Auch wenn das im Krankenhaus eingesetzte System genaue Ergebnisse lieferte, gab es dennoch vereinzelt Aussetzer. Dies lag meistens daran, dass das Radarsystem quer zur Fahrtrichtung blind war.

Um die Robustheit des Systems zu erhöhen, wurde ein zweites Systemkonzept entwickelt, vgl. Abbildung 2.15 b). Hier wurde nur ein Radarknoten auf dem Roboter verwendet, der jedoch mit einem Ringantennenarray mit einem Abdeckungsbereich von  $360^\circ$  ausgestattet war. Mithilfe dieses Systemkonzepts konnten eine höhere Stabilität der Ortung und ein geringerer Hardwareaufwand bei näherungsweise gleichbleibender Genauigkeit erreicht werden. Dies wurde erfolgreich auf der Projektabschlussdemonstration bei Bosch in Renningen vorgestellt. Die erzielten Messergebnisse sind in Abbildung 2.16 b) und Abbildung 2.17 b) dargestellt. Der 2D Ortungsfehler war kleiner als 7,8 cm in 68,3 % bzw. kleiner als 21,3 cm in 95,4 % der Messungen. Dieses Ergebnis ist gut geeignet für autonome Navigation in der getesteten Umgebung. Im Rahmen der Abschlussdemonstration wurde auch parallel eine Ortung mithilfe eines Laserscanners entsprechend dem Stand der Technik durchgeführt. Diese erwies sich um einige cm ungenauer als das in diesem Projekt entwickelte Ortungssystem.

*Funknetzwerk:* In der ersten Projektphase (Labormuster) wurde von FAU und Symeo ein Funknetzwerk basierend auf dem IEEE 802.15.4 Standard aufgebaut. Mithilfe des Netzwerks wird die für das eingesetzte Messprinzip notwendige Zeitsynchronisation

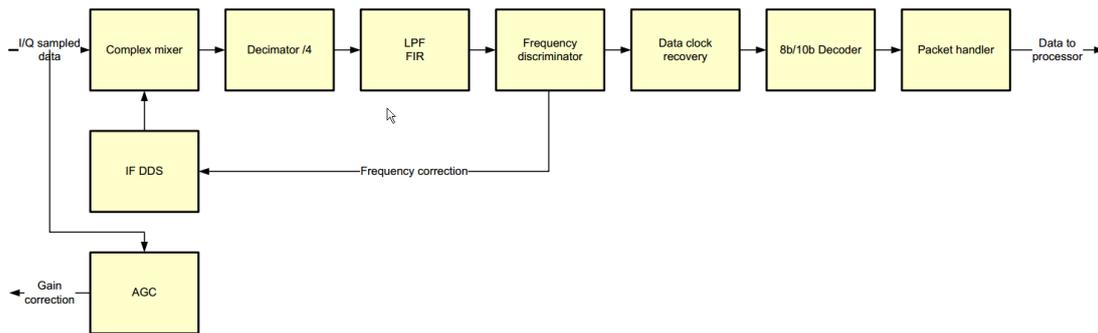


Abbildung 2.18: FSK Kanal Blockdiagramm.

zwischen den Radarstationen erreicht. Desweiteren dient das Funknetzwerk zur Übertragung der zeitgestempelten Messdaten von den Referenzradarknoten zum mobilen Radarknoten auf dem Roboter, wo die Sensorfusion stattfindet. In der zweiten Projektphase (Funktionsmuster und Evaluation) wurde die Vorsynchronisierung und Kommunikation zur Erhöhung der Zuverlässigkeit stattdessen mithilfe eines FSK-modulierten Signals im 24 GHz ISM-Band implementiert (in diesem Frequenzband finden auch die Messungen statt). Dies erhöht die Zuverlässigkeit des Systems und verringert den Hardwareaufwand, da keine zusätzliche Kommunikationshardware notwendig ist.

Die realisierte 2,4 GHz Schicht kann auch in Kombination mit der 24 GHz Schicht betrieben werden, um beispielsweise in komplexeren Anordnungen Messabsprachen durchzuführen. Die 24 GHz Schicht wurde als direkte FSK realisiert, da diese entsprechend stromsparend und mit geringer Latenz umsetzbar ist. Besonders wichtig sind in dem System die data clock recovery und der decoder, der die diversity mit mehreren Kanälen ausnutzt. Abbildung 2.18 zeigt die Umsetzung im FPGA. Die 24 GHz Kommunikationsschicht übernimmt die Kommunikation für den Messablauf (Einleitung Messablauf, Messung selber) sowie die Übermittlung der Ergebnisse an die andere Station.

### 2.1.3 AP3: Situationsverständnis, -modellierung und Planung

**Ziel:** Zur Übernahme heterogener Aufgaben benötigt der ITA ein Verständnis der Situation, in der er sich aktuell befindet. Dazu benötigt er Informationen der Sensorik, der Aktorik und weitere Kontextinformationen. Die relevanten Informationen sollen in einer domänenspezifischen Sprache zur Situationsbeschreibung modelliert werden. Auf Basis des konkreten Modells wird die Planung der nächsten Aktionen erfolgen. Dadurch erhält der ITA ein Verständnis seiner Situation, kann diese bewerten und angemessen reagieren.

**Inhalt:** Der Inhalt dieses Arbeitspakets umfasst die Elemente Situationskatalog, Situationsanalyse, Planung und Adapter. Primäres Ziel ist das Sprachdesign einer domänenspezifischen Sprache zur Modellierung von Situationen. Damit werden dann einzelne Situationen modular entworfen und im Situationskatalog abgelegt. Auf Basis eines konkreten Modells (Situation) erfolgt die Planung, so dass der ITA ein Verständnis seiner Situation, sowie die Möglichkeit zur adäquaten Reaktion erhält.

Die Formulierung und Ausführung komplexer heterogener Servicerobotik-Aufgaben benötigt Expertise verschiedener Domänen: Experten der Anwendungsdomäne tragen Wissen über logische Zusammenhänge derartiger Aufgaben bei. Robotikexperten implementieren robotische Fähigkeiten. Softwareentwicklungsexperten integrieren diese. Von Expertenteams in Allzweckprogrammiersprachen entwickelte Lösungen sind häufig nur sehr schwierig in anderen Kontexten wiederverwendbar, wodurch sowohl die Weiterentwicklung komplexer Servicerobotikanwendungen, als auch der Eintritt neuer Ausrüster in den Markt für Servicerobotik erschwert wird.

Modellgetriebene Softwareentwicklung kann diese Komplexität reduzieren, indem abstrakte Modelle zu primären Entwicklungsartefakten erhoben werden. Diese Modelle können in domänenspezifischen Sprachen formuliert auf die - aus Perspektive der jeweiligen Experten - wesentlichen Aspekte einer Anwendung abstellen und automatisiert in qualitativ hochwertige Software übersetzt werden. Für die Formulierung von Roboterverhalten haben sich zustandsbasierte Formalismen etabliert. Die Modellierung derartiger Zustände – Situationen in denen sich der Roboter befinden kann – ist spezifisch für die Anwendung in einer konkreten Domäne. Dieses generisch oder anwendungsspezifisch zu beschreiben geht regelmäßig über das Domänenwissen beteiligter Robotik- und Softwareexperten hinaus. Stattdessen sollen Servicerobotik-Aufgaben sowie relevantes Kontextwissen von Domänenexperten modelliert und automatisiert in Roboterhandlungen übersetzt werden können.

**Ergebnis:** Anhand der mit den Konsortialpartnern entworfenen Szenarien (AP1) wurde ermittelt, dass sich eine zustandsbasierte, sequentielle Darstellung von Situationen als Konditionen über Roboter und Umwelt gut für deren Beschreibung eignet. Hierzu wurden zwei Modellierungssprachen für die Beschreibung von Aufgaben als Sequenzen von Zielen (Situationen) entwickelt, deren Komplexität im Hinblick auf die Modellierung durch Domänenexperten so gering wie möglich gehalten wurde. Derartige Modellierung unterstützt fernerhin die online Planung (siehe AP 3.3), da jeweils nur Handlungsfolgen zwischen zwei Situationen geplant werden müssen und so flexibel auf sich ändernde

Herausforderungen reagiert werden kann. Weiterhin erlaubt die deklarative, modulare, plattformunabhängige Beschreibung von Zielen deren Wiederverwendung in verschiedenen Kontexten und mit verschiedenen Robotern. Die Abbildung 2.19 zeigt eine Übersicht über die relevantesten Systemkomponenten und Deliverables mit Beteiligung der RWTH Aachen.

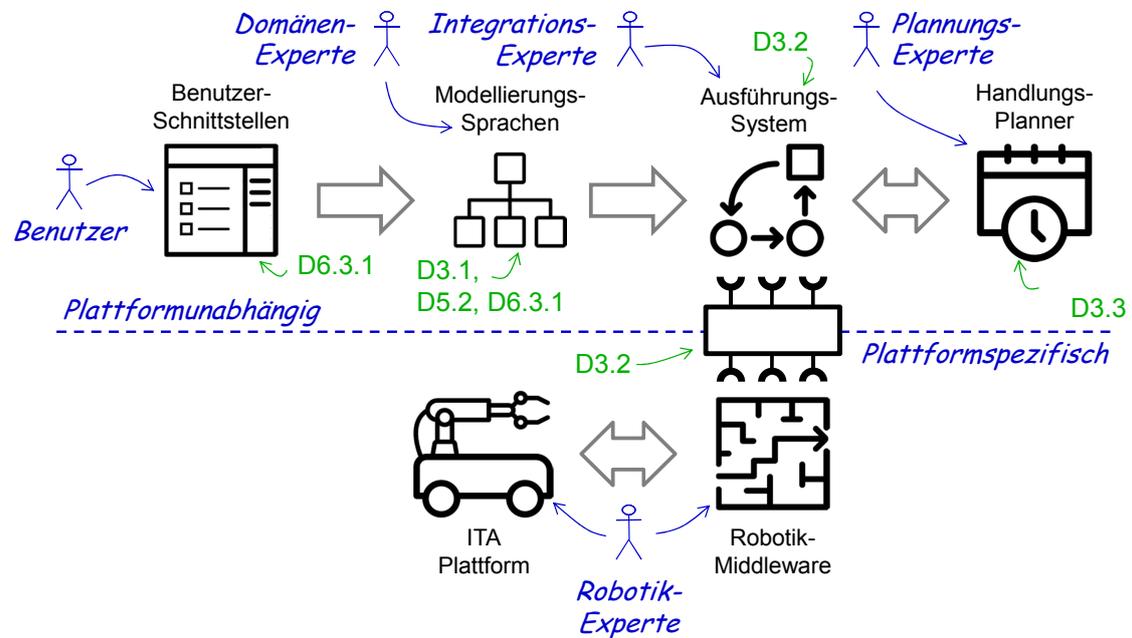


Abbildung 2.19: Übersicht über die relevantesten Systemkomponenten und Deliverables, sowie deren Interaktionen, aus Arbeitspaketen mit Beteiligung der RWTH Aachen.

Die Modellierungssprachen wurden mit der Language Workbench MontiCore [KRV08, KRV10] entwickelt und integriert. Die integrierten Modellierungssprachen für Aufgaben und Ziele wurden mit den Deliverables **D3.1** und **D6.3.1** abgeliefert. Die Möglichkeit das Kontaktieren des Remote-Operators vorzusehen wie in **D5.2** beschrieben wurde in die Aufgabenbeschreibungssprache integriert und in **D3.1** geliefert.

*Kontextmodellierung:* Serviceroboter erfüllen Aufgaben und Ziele im Kontext ihrer Fähigkeiten, Eigenschaften und der Ausführungsumgebung. In diesem Arbeitspaket wurde dazu eine Modellierungssprache entwickelt und in die Modellierungssprachen aus **AP 3.1** integriert. Modelle dieser Sprache beschreiben Entitäten (Roboter, Umwelt) über ihre Eigenschaften (beispielsweise Lage von Objekten im Krankenhaus oder Position des Roboters) und Aktionen (samt Vor- und Nachbedingungen). Sowohl Aufgaben und Ziele als auch Entitäten operieren im Kontext von Domämentypen. Letztere werden in **iserveU** in Form einer vereinfachten Klassendiagrammsprache in Anlehnung an die UML/P [Rum11] realisiert. Die Schnittstelle zwischen Verhaltensmodellierungsschicht und Plattform wurde in **D3.1** abgeliefert.

*Planung auszuführender Aktionen:* Die in **AP 3.1** modellierten Aufgaben und Ziele wer-

den im Kontext der in AP 3.2 modellierten Entitäten und Domänentypen ausgeführt. Hierzu benötigt es ein Ausführungssystem, das nicht nur Aufgaben löst und entsprechende Pläne berechnet, sondern weiterhin deren Ausführung kontrolliert und Interaktion mit Benutzern ermöglicht. Mit der Architekturbeschreibungssprache MontiArcAutomaton wurde hierzu eine Referenzarchitektur entwickelt, die verschiedene S/W Bausteine (vgl. Abbildung 2.19) umfasst, um Aufgaben zu empfangen, deren Zielerfüllung nacheinander zu planen und die aus Aktionen bestehenden Pläne in der Realität auszuführen. Jede Situation wird dazu zur Systemlaufzeit in ein PDDL [MGH<sup>+</sup>98, FL03] übersetzt. Relativ zur aktuellen Situation wird eine Folge von PDDL Aktionen berechnet, die das Ziel erreichen würde. Diese Folge wird in Aktionen der Entitäten aus AP 3.2 übersetzt und mittels der Referenzarchitektur und deren Abbildung auf Funktionen der Plattform in der Welt ausgeführt. Dieses partielle Planen erlaubt flexibel auf Änderungen der Situation zu reagieren und verhindert mehrfache vollständige Neuplanung. Zur weiteren Absicherung werden die Vor- und Nachbedingungen von Aktionen vor deren Ausführung erneut durch die Architektur validiert. Sofern die Aufgabenerfüllung Benutzerinteraktion erfordert, beispielsweise zum Be- und Entladen des ITA, bedient sich die Referenzarchitektur angeschlossener Komponenten für die Mensch-Maschine Interaktion und setzt die Ausführung erst nach entsprechenden Bestätigungen fort. Der Aktions-Planer wurde in D3.3 abgeliefert in Form der Referenzarchitektur samt PDDL-fähigem Planer.

*Folgen und Führen von Personen:* Der Lehrstuhl für Software Engineering stellte die grundlegenden Aufgaben- und Zielmodelle für das erfolgreiche Folgen und Führen von Personen zur Verfügung. Dies beinhaltet auch Benutzerinteraktionsziele. Bosch entwickelte die Navigationsfähigkeiten zum Folgen und Führen von Personen (D3.4 und D3.5). Diese Fähigkeiten wurden mit Zustandsautomaten implementiert. Die Knoten im Automaten kapseln verschiedene Zustände, die der Roboter im Rahmen der Ausführung durchlaufen kann. Für das Führen von Personen musste beispielsweise berücksichtigt werden, dass die Person dem führenden Roboter auch tatsächlich folgt und nicht zu weit zurückfällt. Passiert dies dennoch, weil z.B. die geführte Person eine andere bekannte Person trifft, so wird die Person wieder vom Roboter eingesammelt und die Führung fortgesetzt. Beim Folgen von Personen ist das Wichtigste, den Abstand zu dieser Person nicht zu groß werden zu lassen und gleichzeitig auch nicht zu aufdringlich zu sein. Bei beiden Fähigkeiten ist zentral, dass der Roboter sich nicht nur auf die geführte oder zu folgende Person konzentriert, sondern auch auf die Umgebung und kollisionsfrei navigiert. Abbildung 2.20 zeigt den entwickelten Zustandsautomaten für Folgen und Abbildung 2.21 zeigt den Zustandsautomaten für Führen. Die Realisierung der Fähigkeiten stützt sich dabei auf die in AP2 entwickelte Navigation und die Lokalisierung des Roboters als auch der Person.

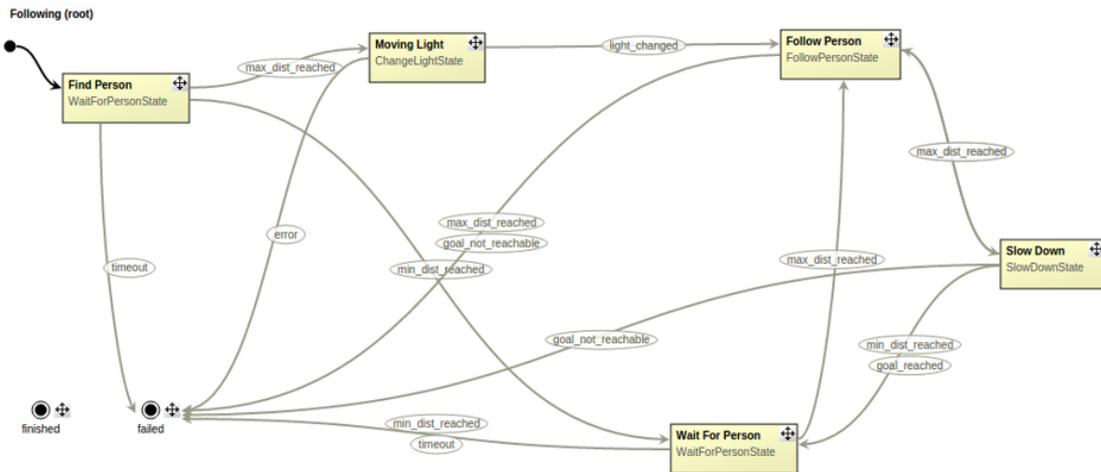


Abbildung 2.20: Zustandsautomat für die Fähigkeit Folgen von Personen

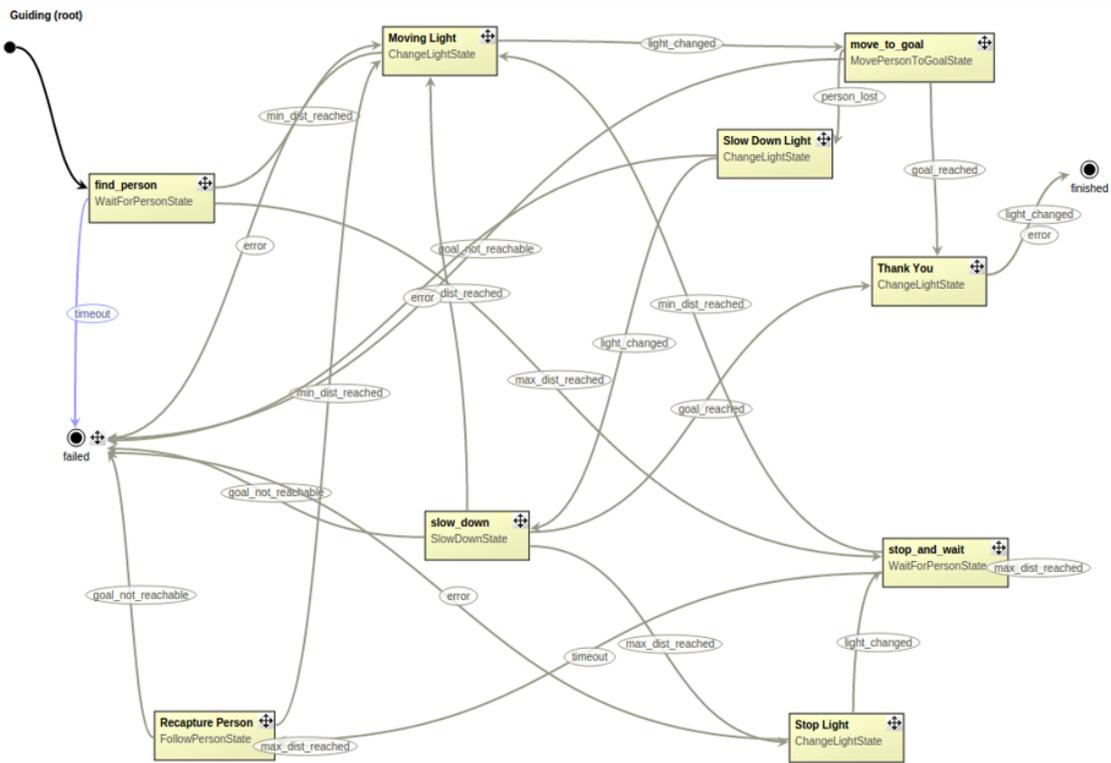


Abbildung 2.21: Zustandsautomat für die Fähigkeit Führen von Personen

#### 2.1.4 AP4: Mensch-Roboter Interaktion über innovative Nutzerschnittstellen

**Ziel:** Das Arbeitspaket hat zum Ziel, ein Konzept für die Interaktion zwischen Personen und dem ITA zu entwickeln. Die in dem Konzept erarbeiteten qualitativen Anforderungen müssen über geeignete Studien auf quantitative Größen heruntergebrochen werden, die dann innerhalb der Algorithmen für die Navigation berücksichtigt werden.

**Inhalt:** Die Akzeptanz von Robotiklösungen im Krankenhaus wird maßgeblich durch den Umgang des Roboters mit dem Menschen und seinem Verhalten im menschlichen Umfeld bestimmt. Dies betrifft die Interaktion des Roboters mit geschultem Personal, aber insbesondere auch mit ungeschulten Personen, also den Patienten bzw. Besuchern im Krankenhaus.

Die Mensch-Roboter Interaktion erfolgt auf einer Vielzahl von Ebenen, beginnend bei reinem (Bewegungs-)Verhalten des Roboters, bis hin zur expliziten direkten Interaktion in Form von Touchscreens oder natürlich sprachlicher Ein- und Ausgabe. Fragestellungen für die indirekte Interaktion sind dabei beispielsweise:

- Mit welcher Geschwindigkeit soll ein ITA in welchen Situationen fahren?
- Wie sind Beschleunigung und Verzögerung zu gestalten?
- Welche Ausweichmuster – insbesondere um Menschen herum – sind akzeptabel?
- Wie folgt der ITA einer Person? (Abstand, neben/hinter, welche Seite)
- Wie verhält sich der ITA bei Türen, in und vor Aufzügen etc.?

Eine weitere Ebene der Interaktion betrifft die Erteilung von Aufträgen an den Roboter bzw. die Rückmeldungen des Roboters über sein aktuelles Verständnis der Situation oder auch seinen Betriebszustand. Für die skizzierten Aufgaben des ITA bieten sich hier, je nach Benutzergruppe, verschiedene Ansätze an: geschultes Klinikpersonal wird z.B. über Taster oder Touchscreens Aufträge erteilen können. Patienten und Besucher hingegen können eine andere Interaktion bevorzugen, z.B. durch Sprache oder Gesten. Hier gilt es, passend abgestimmte Konzepte zu entwickeln und verfügbare Technologien bestmöglich zu integrieren. Allen Aktivitäten liegt die Erkenntnis zugrunde, dass ein akzeptiertes Mensch-Maschine Interaktionsverhalten nur von den Nutzern (Krankenhauspersonal, Patienten) bewertet werden kann und nicht von den Entwicklern.

Die Interaktionsformen sind folgendermaßen gegliedert:

**Direkte Interaktion:** Beinhaltet ein konkretes Ziel, welches der interagierende Mensch mit dem ITA erreichen möchte, direkten physischen Kontakt und/oder intendierten Informationsaustausch (Anzeigen/ Bedienung) zwischen einem konkreten Menschen und dem ITA. Auch das Folgen und Führen von Personen zählt damit zur direkten Interaktion, da der Mensch sein konkretes Ziel mit dem ITA erreicht.

**Indirekte Interaktion:** Wechselseitiger Informationsaustausch oder wechselseitige Beeinflussung zwischen Mensch(en) und ITA, wobei kein gemeinsames Ziel verfolgt wird. Typischerweise finden kein direkter physischer Kontakt und keine Bedienung des ITA durch den Menschen statt. Beispiele für indirekte Interaktion sind die Navigation und das Verhalten, insbesondere Bewegungsverhalten, des ITA in einem sozialen Kontext. Auch das Aussenden von Signalen des ITA zur Anzeige seiner Anwesenheit oder seiner geplanten Route zählen zur indirekten Interaktion.

Der Schwerpunkt im Projekt liegt auf der Gestaltung der indirekten Interaktion, das heißt der Einwirkung eines ITA auf seine Umwelt sowie seine Kommunikation mit seinem Umfeld, in dem sich Menschen aufhalten und bewegen, die den ITA nicht selbst für eigene Zwecke nutzen und möglicherweise noch keine bisherigen Erfahrungen mit dem ITA haben. Ziel ist es, sein Verhalten im menschlichen Umfeld nachvollziehbar und akzeptiert zu gestalten, so dass durch den ITA keine Verunsicherungen oder Störungen anwesender Menschen erzeugt werden. Dabei sind die besonderen Rahmenbedingungen im Krankenhaus zu berücksichtigen, z.B. die Anwesenheit von medizinischem Personal, Patienten und Besuchern sowie typische räumliche Bedingungen wie beengte Platzverhältnisse oder die Anwesenheit vieler Engstellen. Die Relevanz einer angemessenen Gestaltung der indirekten Interaktion im Krankenhaus zeigen [MF08]. In ihrer Analyse des sozialen Verhalten eines Krankenhausroboters zeigten sich mehrere Defizite, die als unhöflich und unakzeptabel wahrgenommen wurden und teilweise dazu führten, dass der Einsatz eines Transportroboters im Krankenhaus als gefährlich eingestuft wurde, wenn es z.B. zur Blockade von Nottransporten im Krankenhaus durch den Roboter kommt. Im Projekt galt es zu erforschen, welches Verhalten eines Transportroboters im sozialen Umfeld als akzeptabel empfunden wird und welche Parameter des Bewegungsverhaltens besonders ausschlaggebend für eine angemessene soziale Navigation sind. Daneben wurden grundlegende Lichtsignale im Projekt umgesetzt, um die Kommunikation mit dem Umfeld weiter zu verbessern und das Gesamtverhalten selbsterklärend zu gestalten. Für die Gestaltung der direkten Interaktion galt es, wesentliche Arbeitsschritte und Rollen für die Aufgaben Beladen und Entladen zu definieren sowie für die Aufgaben Führen von Personen und Folgen einer Person grundlegende Anforderungen zu erstellen.

Die konkreten Aufgabenstellungen waren deshalb wie folgt:

1. Ableitung relevanter Szenarien der Mensch-Maschine Interaktion für die identifizierten Use-Cases (Übernahme autonomer Transporttätigkeiten, Folgen, Führen von Personen) sowie der zu betrachtenden technischen Gestaltungsparameter,
2. Qualitative Untersuchungen der Erwartungen an das Verhalten eines autonomen Transportroboters im menschlichen Umfeld und insbesondere im Umfeld des Krankenhauses,
3. Quantitative Analyse relevanter Gestaltungsparameter in den identifizierten relevanten Szenarien,
4. Ableitung eines Interaktionskonzeptes zur indirekten Interaktion,

5. Prototypische Umsetzung des Interaktionskonzeptes auf dem Prototyp und Vorbereitung der Evaluation im Krankenhaus (AP 8).

**Ergebnis:** Die Erforschung und Untersuchung Mensch-Maschine Interaktionen wurde von Bosch durchgeführt. Ausgangspunkt für die Arbeiten im AP 4 war die basierend auf der Use-Case-Beschreibung D1.1 entwickelte Situationssystematik für die Ableitung relevanter Szenarien und zu betrachtender technischer Gestaltungsgrößen. Durch die Beachtung verschiedener baulicher Umgebungen, anwesender Personen (bis zu 2 betrachtet), wesentlicher Personenmerkmale (z.B. Rollstuhlfahrer), der intendierten Bewegungsrichtungen von Transportroboter und Menschen sowie die Anwesenheit/Abwesenheit temporärer Hindernisse entsteht eine sehr hohe Zahl möglicher Konstellationen, die nicht erschöpfend in Einzelanalyse betrachtet werden kann. Aus den möglichen Kombinationen wurden daher besonders typische Szenarien ausgewählt (D4.1), für die verschiedenartige Anforderungen an eine geeignete Mensch-Roboter Interaktion zu erwarten sind, welche also in der Lage sind, grundlegende Anforderungen an das Verhalten des ITA abzudecken. Bei der Auswahl hoch priorisierter Szenarien wurde so darauf geachtet, dass folgende Aspekte des Bewegungsverhaltens adressiert werden:

- Abstandshaltung zum Mensch
- Ausweichverhalten gegenüber dem Menschen
- Verhalten an (Tür-)Engstellen
- Verhalten an Kreuzungen

Daneben wurde der Bedarf identifiziert, situationsübergreifende Parameter wie Geschwindigkeit, Abbiegeverhalten und typische Pfadverläufe zu adressieren. Von der erstellten Situationssystematik wurden mehrere Studien abgeleitet, um generelle Erwartungen an das Bewegungsverhalten zu erheben und genaue akzeptierte Ausprägungen der Gestaltungsparameter einzugrenzen. Aufgrund der Vielzahl an durchgeführten Studien und vorgelegter Ergebnisse soll hier für einen Gesamtüberblick auf die jeweiligen Berichte verwiesen werden, D4.2 sowie D4.3. Genutzt wurden qualitative Ansätze, die allgemeine Erwartungen an Transportroboter, auch über seine indirekte Interaktion hinaus, ermittelten (z.B. Erwartungen an seine äußere Gestaltung sowie Erfordernisse für die äußere Gestaltung aufgrund der im Krankenhaus gegebenen Rahmenbedingungen), sowie quantitative Studien, um relevante technische Parameter einzugrenzen und für eine Formalisierung vorzubereiten, die durch die Navigation genutzt werden kann. Als Beispiel sollen durchgeführte Analysen zur Akzeptanz des Abstandsverhaltens genannt werden, welche im Vergleich zu dem zu Projektbeginn existierenden wissenschaftlichen Stand folgende u.a. folgende methodische Verbesserungen aufwiesen (s. auch [Lau15] für eine dedizierte Literaturanalyse):

- Untersuchung sowohl frontaler Annäherungs- als auch seitlicher Passierdistanzen, welche gleichermaßen in der indirekten Interaktion in Gängen von Relevanz sind,

- Kombination verschiedener methodischer Herangehensweisen, u.a. dem Erleben eines vollständig autonom bewegten Prototypen ohne Möglichkeiten der Einflussnahme durch die Person,
- Ableitung von Akzeptanzkurven durch Analyse größerer Stichprobenumfänge unter Beachtung strengerer Kriterien zur Stichprobenwahl (z.B. keine Vorerfahrungen mit dem Prototyp),
- Hoch kontrolliertes Versuchssetting und Analyse von genau mittels Laserscannern gemessenen Abständen; durchgängige Verwendung der Methodik zur Sicherstellung der Vergleichbarkeit von Ergebnissen.

In den Arbeiten wurden Akzeptanzkurven ermittelt, welche es ermöglichen, akzeptierte Mindestabstände für die frontale Annäherung und das seitliche Vorbeifahren abzuleiten sowie die Einflüsse von verringerten Abständen auf die Akzeptanz quantitativ zu bewerten, so dass eine mathematische Kombination mit weiteren formalisierten Anforderungen ermöglicht wird, welche das unter gegebenen Rahmenbedingungen optimale Verhalten ermittelt.

Das indirekte Interaktionskonzept fasst diese und weitere Projekterkenntnisse zur Gestaltung der indirekten Interaktion zusammen (D4.4). Darüber hinaus wurde das Verhalten des Prototypen soweit technisch möglich an die Vorgaben angepasst und aus den ermittelten Ergebnissen eine Checkliste zur Bewertung des ITA-Verhaltens im Rahmen des AP 8 im Krankenhaus abgeleitet. Diese ist in Abbildung 2.22 dargestellt.

Die Untersuchung der Lokalisierungsgenauigkeit ergab, dass diese in den meisten Fällen ausreichend genau ist, so dass keine spezielle Anpassung der Mensch-Maschine-Kommunikation nötig ist (D4.5). In den seltenen Fällen, bei denen es zu einem Totalausfall der Lokalisierung kommt, ist dieses Problem nicht über eine Änderung der Interaktionsart lösbar. Die Konsequenz daraus ist deshalb eher eine zukünftige Weiterentwicklung der Lokalisierungsrobustheit.

Die erarbeiteten Grundlagen für eine angemessene soziale Interaktion mit Personen bilden auch über den Projektabschluss hinaus eine wesentliche Grundlage zur Entwicklung von Robotik-Plattformen, die sich selbständig im offenen menschlichen Umfeld bewegen und dabei die Erwartungen ihres Umfeldes an ihr Verhalten erfüllen sollen. Durch die Ermittlung quantitativer Parameter und deren Auswirkungen auf die Akzeptanz wird die Formalisierung von Anforderungen im Rahmen der Navigation möglich.

### Checkliste Bewertung des Verhaltens des ITA durch Beobachter

ITA-Verhalten	ja	nein, Defizite	nicht relevant
Es gab keine Kollisionen zwischen dem ITA und anderen Menschen oder Objekten.			
Der ITA hält angemessene Abstände zu Objekten, Wänden und Hindernissen.			
Der ITA hält angemessene Abstände zu Personen (frontal mind. 1 m Abstand, seitlich mind. 50 cm Abstand).			
Die Fahrtgeschwindigkeit ist angemessen (wenn möglich, zwischen 0,6 und 0,8 m/s).			
Die Fahrtgeschwindigkeit verändert sich angemessen/ das Geschwindigkeitsprofil ist glatt.			
Der Bewegungspfad des ITA ist angemessen, z. B. bevorzugtes Befahren der rechten Gangseite.			
Richtungsänderungen des ITA sind angemessen und gut vorhersehbar.			
Weitere relevante Verhaltensänderungen sind rechtzeitig vorhersehbar.			
Das Verhalten an Engstellen ist angemessen.			
Der ITA hat keine Personen oder Personentransporte blockiert.			
Das Umfahren von Personen/ Ausweichen ist angemessen.			
Das Umfahren von Objekten/ Ausweichen ist angemessen.			
Der ITA verhält sich angemessen gegenüber kreuzenden Personen.			
Der ITA fährt angemessen an Zimmertüren vorbei.			
Der ITA hat vorhandene Bewegungsfreiräume genutzt, d.h. er ist nicht stehen geblieben/ ausgewichen, obwohl genug Platz ist.			
Das Verhalten beim Anfahren und Ankommen ist angemessen.			
Der ITA löst alle Situationen selbstständig.			
Der ITA unterbricht seine Bewegung nicht, um sich neu zu orientieren oder seine Umgebung vollständig zu erfassen.			
Es gab keine unnötigen Wiederholungen des Verhaltens des ITA.			

Abbildung 2.22: Bewertungscheckliste des ITA-Verhaltens

### 2.1.5 AP5: Service-orientierte Architektur für Fallback-Sicherheit mittels Remote Management

**Ziel:** Ziel des Arbeitspakets ist die Realisierung eines Fallback-Sicherheitsmechanismus für den Fall, dass der Serviceroboter in Situationen gerät, die ihm unbekannt sind und mit den ihm bekannten Strategien nicht beherrschbar sind.

**Inhalt:** Um Dienstleistungen, die bislang von Menschen erbracht wurden, übernehmen zu können, wird von der Robotik ein hohes Maß an Zuverlässigkeit und Verfügbarkeit vorausgesetzt. Zur Entwurfszeit des Roboters werden gewisse Annahmen über die Umgebung gemacht. Zudem wird eine präzise Spezifikation definiert, die der Roboter erfüllen muss. Ein abstraktes Systemmodell des Roboters kann zusammen mit den Annahmen über die Umgebung zur Entwurfszeit gegen seine Systemspezifikation verifiziert werden. Dadurch kann formal bewiesen werden, dass der Roboter gemäß der Vorgaben der Systemspezifikation entwickelt wurde.

Durch die Heterogenität und Komplexität der Betriebsumgebung, gibt es jedoch unendlich viele Situationen, die zur Entwurfszeit nicht vorgesehen werden können. Solche Situationen können dazu führen, dass die Umgebungsannahmen in der Umgebung in welcher der Roboter agiert ungültig werden. Um hier Abhilfe zu schaffen, muss eine Überwachungskomponente parallel zu der Robotersoftware laufen und ständig überprüfen, ob sich der Roboter in einer kritischen Situation befindet. Falls das passiert, wird die Überwachungskomponente sowohl den Roboter benachrichtigen als auch einen externen Operator in der Servicezentrale um Hilfe bitten. Dort übernimmt der Operator die Kontrolle über den Roboter und steuert ihn aus der kritischen Situation.

In diesem Arbeitspaket sollte eine Laufzeitverifikationskomponente entwickelt werden, um das Roboterverhalten während seines Betriebes ständig gegen seine Systemspezifikation zu überprüfen. Außerdem sollte zum einen die Schnittstelle erweitert werden, um das Remote Management durch den externen Operator zu erleichtern. Zum anderen sollte ein entsprechendes Frontend zur Fernsteuerung durch einen Operator bereitgestellt werden.

#### Aufgabenstellungen von AP5

- Ermöglichung des Remote Managements von Servicerobotern
- Modellierung von überwachten Aktionsausführungen, Erkennung von Not-Situationen und Einsteuern angemessener Reaktionen
- Entwicklung eines Remote Management Frontends
- Aufbereitung der Sensordaten des 3DHinderniserkennungssystems für die Visualisierung

**Ergebnis:** Um das Remote Management des Roboters zu ermöglichen, wurde, der Architektur entsprechend, eine SmartSoft Komponente entwickelt, namens *SmartRemoteOperatorProxy*, welche die Anbindung an die von der TUC entwickelten Remote Operator Lösung unterstützt. Mit Hilfe von *Webservices* wurde eine Schnittstelle realisiert (**D5.1**),

welche die Datenübertragung über das Netzwerkprotokoll TCP zwischen der SmartSoft Komponente und einem externen Operator ermöglicht. Der Webservice wurde in der Programmiersprache C++ implementiert, um die Kommunikation und Integration mit der oben genannten SmartSoft Software Komponente zu erleichtern. Der Webservice kann von unterschiedlichen Geräten angesprochen werden, z.B. Smartphones, Tablets, oder Arbeitsrechnern, wie in Abbildung 2.23<sup>2</sup>dargestellt ist. Die Schnittstelle des Webservices wurde in dem String-basierten JSON (JavaScript Object Notation) Format [JSO] implementiert.

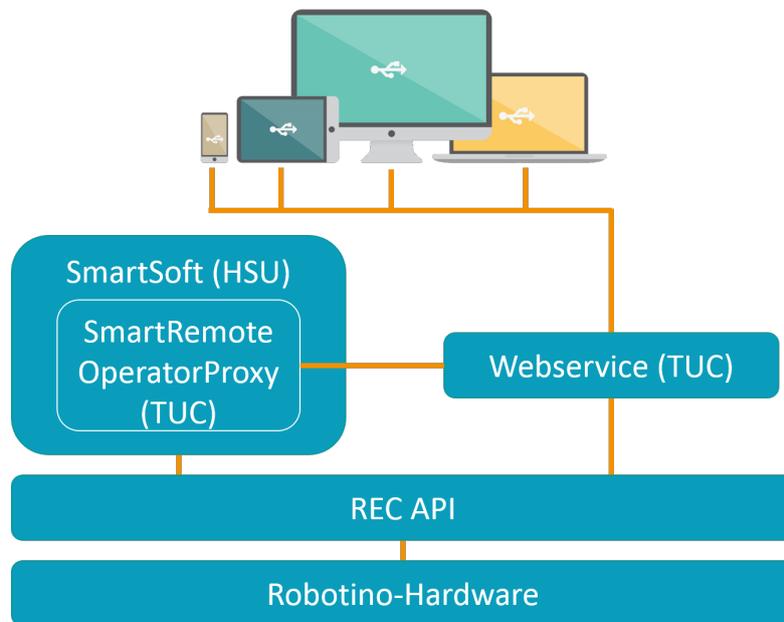


Abbildung 2.23: Webservice Überblick.

Mittels des Webservices kann der Roboterzustand abgefragt und gesetzt werden. Der Roboter hat vier Zustände, gezeigt in Abbildung 2.24:

- OK – der Roboter bewegt sich autonom.
- HELP – der Roboter befindet sich in einer Situation, die er selbst nicht mehr lösen kann und fragt den externen Operator um Hilfe.
- CONTROL – der Roboter hat um Hilfe gefragt und der externe Operator hat die Kontrolle übernommen.
- CANCEL – der externe Operator hat dem Roboter bekannt gegeben, dass die

<sup>2</sup>In Abbildung 2.23 wurde ein Vektorbild verwendet um die Geräte die sich mit dem Webservice verbinden können darzustellen. Alle Rechte zu dem Vektorbild gehören Freepik. Weitere Informationen können auf folgende Webseite gefunden werden: [http://www.freepik.com/free-vector/flat-collection-of-screen\\_714026.htm#term=designers](http://www.freepik.com/free-vector/flat-collection-of-screen_714026.htm#term=designers).

Fernsteuerung erfolglos gewesen ist. Dadurch wird das weitere Handlungsvorgehen dem Roboter übergeben.

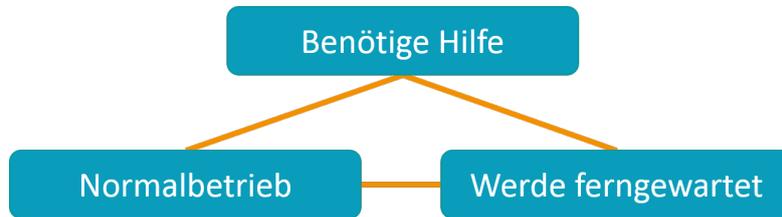


Abbildung 2.24: Roboterzustände.

Der Webservice erlaubt die Abfrage von Session-IDs, die es verhindern, dass mehrere Nutzer gleichzeitig den gleichen Robotern ansprechen. Zudem können die Sensordaten (Ultraschallpunktswolken und Laserscans), das Kamerabild, die Grundrisskarte und die Roboterpose über den Webservice abgefragt werden.

Für die Modellierung von überwachten Aktionsausführungen des Roboters, wurde, mit Hilfe der Logiksprache TCTL (Timed Computational Tree Logic) [GB04], die Systemspezifikation für den Roboter entworfen. Die Spezifikation wurden anhand eines Roboter- und Umgebungsmodells mit dem UPPAAL Model-checker [GB04] verifiziert. Darüber hinaus wurden die Standardreaktionen definiert, die gewählt werden im Falle einer Verletzung der definierten Systemeigenschaften. Wenn beispielsweise der Roboter den Sicherheitsabstand verletzt, muss er halten und die Kontrolle an einen externen Operator übergeben.

Um die Überprüfung der Systemspezifikation des Roboters während seines Betriebs zu ermöglichen, wurde eine *Laufzeitverifikationskomponente* entwickelt (**D5.2**), welche Abstände zu Hindernissen zur Laufzeit überwacht und das Einhalten der Systemspezifikation überprüft. Die Spezifikation gab vor, dass der Roboter einen vordefinierten Sicherheitsabstand in Bezug zu den Objekten/Menschen in seiner Umgebung einhalten muss. Wenn der Sicherheitsabstand verletzt wird, wird der Roboter in einen sicheren Zustand gebracht. Ein externer Operator wird gleichzeitig benachrichtigt, um die Kontrolle über den Roboter zu übernehmen. Die Komponente wurde mit dem Webservice und dem Operator Frontend integriert und im Krankenhaustest erfolgreich eingesetzt. Die Abbildung 2.25 zeigt einen Überblick über die Kommunikation und Integration zwischen dem Webservice, dem Frontend des externen Operator und der Laufzeitverifikationskomponente.

Um das Remote Management durch den externen Operator zu unterstützen, wurde ein entsprechendes Frontend zur Fernsteuerung für einen Operator bereitgestellt (**D5.3**). Dafür wurden die Sensordaten der Hinderniserkennung zusammen mit Bosch für die Visualisierung aufbereitet (D5.4). Das Frontend für die Anzeige und Steuerung besteht aus der *RemoteOperator-App*, einer Applikation für mobile Geräte und aus der *RemoteControlDesktop*, eine entsprechende Software für Arbeitsrechner.

Die RemoteOperator-App wurde nach den Prinzipien von Xbox Controllern aufgebaut. Die graphische Schnittstelle stellt sowohl Knöpfe für Vorwärts- und Rückwärtsbewegung

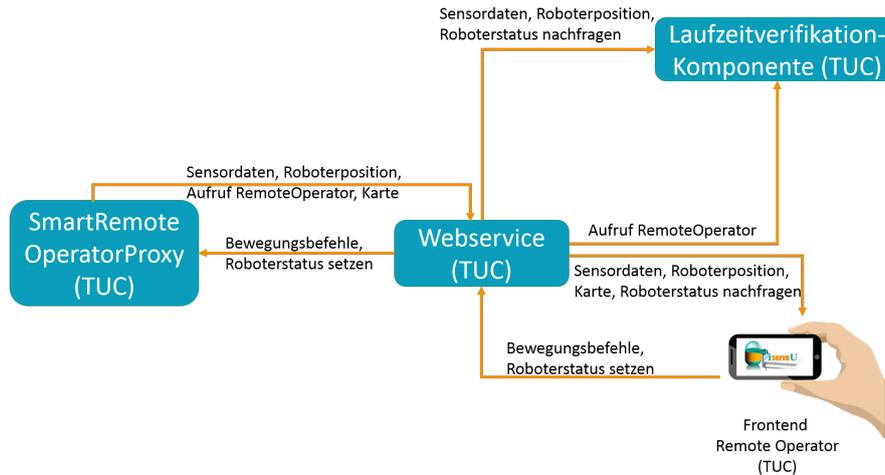


Abbildung 2.25: Laufzeitverifikation-Komponente integriert mit dem Webservice und dem Frontend des externen Operator.

als auch für Drehbewegung zur Verfügung. Darüber hinaus kann man die Bewegungsgeschwindigkeit und Drehgeschwindigkeit des Roboters durch die entsprechende GUI-Elemente einstellen. Die RemoteOperator-App ermöglicht die Visualisierung einer Reihe von Sensoren: Abstandssensoren, Ultraschallsensoren, Laserscanner, und Kameras. Darüber hinaus, wurde eine Alternative zur Darstellung der Kamera in Form eines Kompasses auf einem Raster implementiert. In Abbildung 2.26 wird eine Bildschirmaufnahme des Fernsteuerung-Reiters in der Smartphone-App dargestellt.

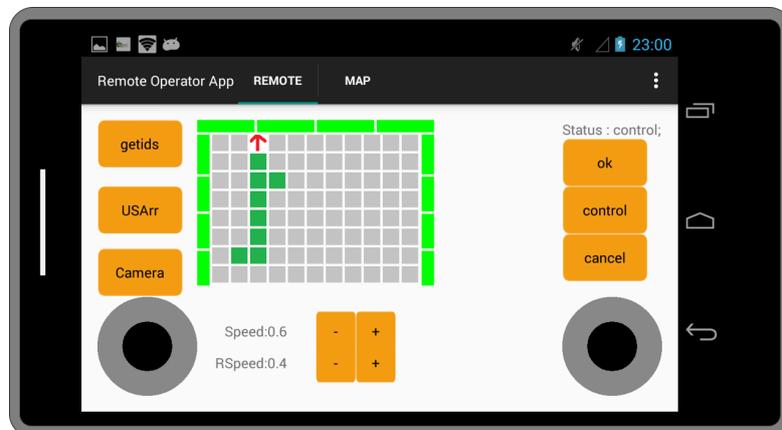


Abbildung 2.26: Smartphone-Applikation mit Kompass und Robotersensoren.

Auf diesem Raster wurde mit Hilfe der Wegmessungsdaten sowohl gezeigt, welchen Weg der Roboter zurückgelegt hat als auch in welche Richtung er gefahren ist. Zusätzlich dazu wurde die Visualisierung einer Grundrisskarte des Krankenhauses ermöglicht (siehe Abbildung 2.27). Darauf konnte mit Hilfe der Wegmessungsdaten die Roboterpose gezeigt

werden.

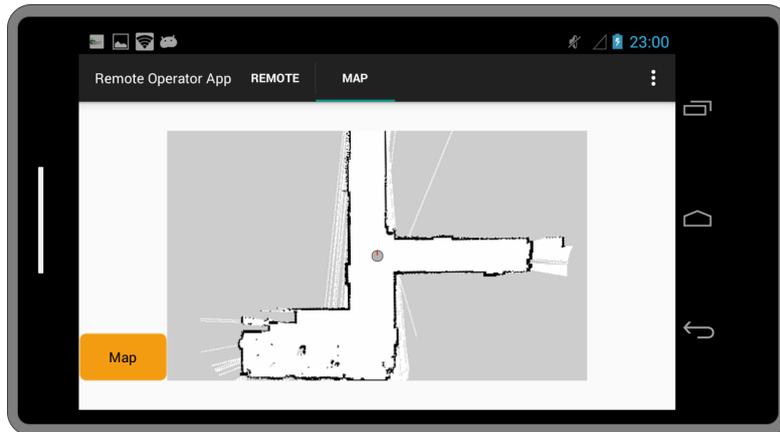


Abbildung 2.27: Smartphone-Applikation mit Grundrisskarte.

Die Darstellung des Roboterzustands, der Sensordaten, etc. wird in einer auf die Tätigkeit des externen Operators zugeschnittenen Form präsentiert. Hierzu werden die Abstandsmessungen zu Hindernissen in mehrere Bereiche aufgeteilt und visualisiert. Die Anzahl der Bereiche wurde nach der Größe des abgedeckten Blickfeldes parametrisiert, z.B. wurden bei einem 180 Grad großen Blickfeld zwölf Bereiche konstruiert. Für jeden Bereich wird der minimale Wert ermittelt. Die in der Remote-Schnittstelle dargestellten Abstandsbalken werden entsprechend dem Wert eingefärbt.

Mit dem Ziel die Datenmenge, welche zwischen Roboter und RemoteOperator ausgetauscht wird, auf ein notwendiges Minimum zu reduzieren, wird auf dem Roboter eine Vorverarbeitung und Verdichtung der Information vorgenommen. So werden zum Beispiel die hochfrequenten Sensordaten (z.B. Laser), in ihrer Frequenz reduziert und in eine, für die Darstellung ausreichende, reduzierte Form konvertiert und übertragen. Diese Vorverarbeitung wird auf dem Roboter in der SmartRemoteOperatorProxy Komponente vorgenommen.

Um eine gewisse Verbindungsstabilität zwischen dem Webservice und der Smartphone-App zu ermöglichen, wurde der Webservice über Timer regelmäßig angesprochen. Um Abstürze der Smartphone-App zu vermeiden, werden die Timer beendet, sobald das Smartphone in den Ruhezustand (Standby) geht oder die App in den Hintergrund geht. Darüber hinaus wurden weitere Optimierungen über Timer-Intervalle vorgenommen, um das Blockieren der Smartphone-App durch eine hohe Anzahl von Anfragen zu vermeiden.

Die Software RemoteControlDesktop wurde als graphisches Frontend für Operatoren entwickelt, die einen Arbeitsrechner für die Fernsteuerung des Roboters benutzen. Das Frontend ermöglicht die Beobachtung einer Flotte von mehreren Robotern, die sich im Krankenhaus im Einsatz befinden. Wenn die Laufzeitverifikation-Komponente oder der Roboter selbst eine kritische Situation melden, kann der externe Operator das zum jeweiligen Roboter gehörende Icon in der GUI anklicken und diesen per Joystick fernsteuern. Um eine Konsistenz mit der Smartphone-App zu sichern, wurden sowohl die Werte der

Abstandsensoren als auch die Kamera in der Desktop-Software entsprechend dargestellt (siehe Abbildung 2.28).



Abbildung 2.28: Fernsteuerung eines einzelnen Roboters in RemoteDesktopControl.

### 2.1.6 AP6: Durchgängige komponenten- und modellbasierte Entwicklung

**Ziel:** Kern der Aufgabenstellung von AP6 ist die Entwicklung einer modularen Softwarearchitektur und der zugehörigen Werkzeuge für die modellgetriebene Entwicklung von Softwarekomponenten für Serviceroboter. Diese sollen die Integration der verteilten Funktionen von Servicerobotern und Anforderungen in verschiedenen Anwendungsdomänen und Plattformen sowie deren situative Orchestrierung zur Laufzeit ermöglichen und unterstützen.

**Im Einzelnen standen Forschungs- und Entwicklungsarbeiten zu folgenden Themen des S/W-Engineerings für die Servicerobotik im Fokus:**

- Dynamisch-adaptives komponenten-basiertes Framework für Serviceroboter (Laufzeitumgebung für die S/W-Komponenten)
- Modulare Softwarearchitektur (modellgetrieben)
- Werkzeuge zur Realisierung eines modellgetriebenen Softwareentwicklungs-Ansatzes
- Methoden zur Modellierung und Nutzung von Variabilität (Designzeit (Entwickler, Integrator) und Laufzeit (Roboter))
- Modellierungssprachen zur Verhaltensmodellierung (Methoden, Sprachen, Werkzeuge)
- Werkzeuge und Methoden für die Modellierung von Handlungsabläufen und deren situative flexible Ausführung

**Inhalt:** Der in diesem Arbeitspaket durch die HSU entwickelte komponenten- und modellgetriebene Entwicklungsansatz bildet die Grundlage der erfolgreichen Integration von Softwarekomponenten aller Projektpartner. Die Strukturierung des Robotik-Entwicklungsprozesses unter Einbeziehung universell einsetzbarer und wiederverwendbarer Komponenten ermöglicht die Entwicklung von übergreifenden Systemlösungen, weg von isolierten Einzellösungen. Das Explizieren und Realisieren des Entwicklungsprozesses unter Verwendung von modellgetriebener Softwareentwicklung ermöglicht die klare Trennung der Verantwortlichkeiten unterschiedlicher Rollen in einem gemeinsamen Entwicklungsprozess. Diese Trennung bildet die Grundlage für ein Business Ecosystem, in welchem verschiedene Partner mit ihrer jeweiligen Expertise beitragen können, ohne alle Details einer möglichen Anwendung (Gesamtsystem) im Blick haben zu müssen. Die Entwicklung lässt sich in vier Bereiche, respektive Rollen, aufteilen (Abbildung 2.29), welche die SmartMDS Toolchain als zentrales Entwicklungswerkzeug (IDE) zusammenbringt: Design (anwendungsgetrieben, Projekt-Architekten), Implementierung (technologiegetrieben, Komponentenentwickler), Integration (anwendungsgetrieben, Systemintegrator), Run-Time (umgebungs- bzw. situationsgetrieben, Roboter). Die Möglichkeit, sich auf Bausteine (Softwarekomponenten) und Schnittstellen (Services) anderer Rollen und Experten verlassen zu können, ist die Grundlage für die effiziente Wiederverwendung von

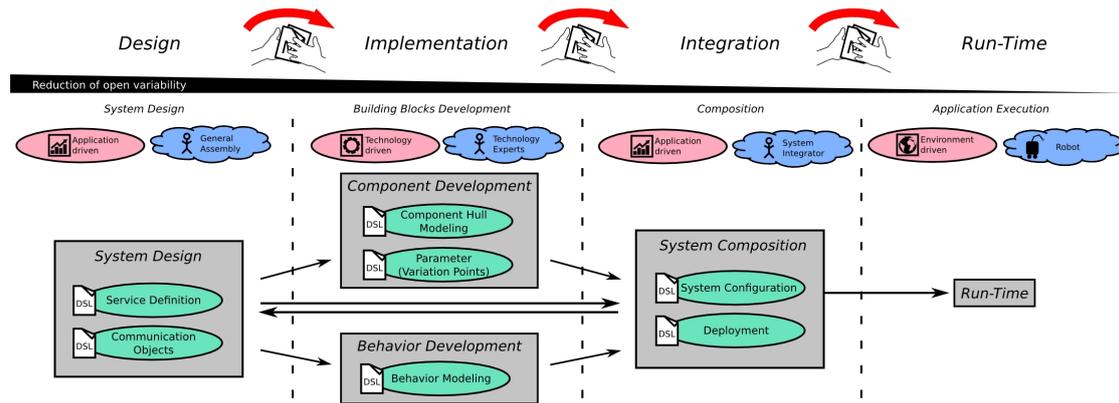


Abbildung 2.29: Im durchgängigen modellgetriebenen Entwicklungsprozess reduziert sich die offene Variabilität schrittweise. Die Verbindung (Modell-Handover) zwischen den Entwicklungsschritten, Rollen und Sichten sowie die entsprechenden DSLs werden von der SmartMDS Toolchain unterstützt.

Komponenten. Die Modellierung der strukturebenen und dahinter stehenden semantischen Elemente ermöglicht eine saubere Trennung der architektur-relevanten Elemente (z.B. Services) von der Implementierung, so dass ein Technologiewechsel (z.B. die darunter liegende Middleware (CORBA, ACE, DDS)) nicht notwendigerweise Änderungen in der Implementierung (sprich User-Code) nach sich zieht.

Die Realisierung des komponenten- und modellgetriebenen Entwicklungsansatzes in entsprechenden integrierten Werkzeugen (SmartMDS Toolchain) unterstützt sowohl die jeweiligen Rollen des Entwicklungsprozesses als auch die Übergabe der Modelle und Ergebnisse zwischen den Rollen. Eine solche einfach zu bedienende Entwicklungsumgebung, welche auf Basis von Softwaremodellen die Entwicklung und konsistente Integration der Komponenten sicherstellt, macht die Realisierung und Anwendung eines durchgängigen komponenten- und modellgetriebenen Entwicklungsansatzes überhaupt erst möglich. Die konsistente Weiterentwicklung der SmartMDS Toolchain [SLLS16] als solches Entwicklungswerkzeug stand deshalb im Hauptfokus der Arbeiten der *HSU* innerhalb dieses Arbeitspaketes.

**Ergebnis:** Die im Arbeitspaket entwickelten Methoden und Strukturen zu einem modellgetriebenen Softwareentwicklungs-Ansatz, umgesetzt in der Weiterentwicklung der SmartMDS Toolchain sowie dem Robotikframework *SmartSoft*, stellte die grundlegende Plattform für die entwickelte Softwarearchitektur sowie für die modellgetriebenen Arbeiten in diesem Arbeitspaket dar. Die in diesem Arbeitspaket entwickelte modulare Softwarearchitektur (Abbildung 2.30) besteht aus lose gekoppelten *SmartSoft* Softwarekomponenten mit expliziten, stabilen Schnittstellen (Services). Sie ermöglichte dem Projekt eine kontinuierliche und verlässliche Integration, die Wiederverwendung von existierenden Komponenten, das einfache Verpacken von Bibliotheken in Komponenten, sowie die Laufzeit-Orchestrierung aller Komponenten.

Im folgenden sind die Ergebnisse der im AP6 erarbeiteten Methoden, Strukturen und Werkzeuge erst stichpunktartig und dann im Detail erläutert.

### **Ergebnisse (Step-Changes) der Arbeiten aus AP6 in Stichpunkten:**

- Erschließen von modellgetriebener Softwareentwicklung (MDSO) für die Robotik
- Entwicklung und Umsetzung eines modellgetriebenen Entwicklungsansatzes, welcher die Expertisen der Partner einbindet (Rollentrennung)
- Die Entwicklung entsprechender Werkzeuge (SmartMDSO Toolchain) auf hohem technologischen Niveau (TRL6), um den modellgetriebenen Entwicklungsansatz für die Nutzer zugänglich zu machen.
- Leitung der Software Integration im Projekt durch die Moderation des Vorgehens (iterativ modellgetrieben), State of the Art Robotik Funktionalitäten (bspw. Lokalisierungs-, Navigations-, Koordinations-Komponenten), die Ausführungsumgebung (Dynamisch-adaptives komponenten-basiertes Framework - SmartSoft) und das entsprechende Werkzeug, um all dies einfach zugänglich zu machen (SmartMDSO Toolchain).
- Integration der Roboter-Verhaltensentwicklung in einen durchgängigen modellgetriebenen Entwicklungsansatz.
- Einbindung von Robotik in übergreifende Systeme in konkreten Anwendungsszenarien mit KMU Partner (z.B. Robotino mit Festo MPS, REC)

### **Durchgängiger modellgetriebener Softwareentwicklungsansatz und Werkzeuge: SmartMDSO Toolchain**

Die HSU konnte im Projekt einen systematischen und durchgängigen modellgetriebenen Entwicklungsansatz [LSLS14] erforschen, welcher die Trennung der Rollen und Verantwortlichkeiten vorsieht. Dieser berücksichtigt die Trennung der Rollen mit ihrer jeweiligen Expertise und Aufgabe und stellt auf Modellebene die Konformität der Komponenten zueinander sicher, so dass eine Integration trotz loser Kopplung möglich bleibt. Dieser neue Ansatz, welcher MDSO für die Robotik erschließt, stellt eine qualitative Verbesserung und einen Step-Change im Robotik Software Engineering dar.

Die SmartMDSO Toolchain realisiert diesen Ansatz, unterstützt die unterschiedlichen Rollen bei der Softwareentwicklung und macht den modellgetriebenen Entwicklungsansatz einfach zugänglich. Die sukzessive Konsolidierung und Weiterentwicklung der SmartMDSO Toolchain und das Erreichen eines signifikanten TRL (Technology Readiness Level, "technology demonstrated in relevant environment") für die modellgetriebene S/W-Entwicklung in der Servicerobotik waren Kern der Arbeiten der HSU in diesem Arbeitspaket. Über die Projektlaufzeit konnte der Reifegrad entsprechend zu TRL 6 ("technology demonstrated in relevant environment") [euR15] gesteigert werden (Step Change). Die

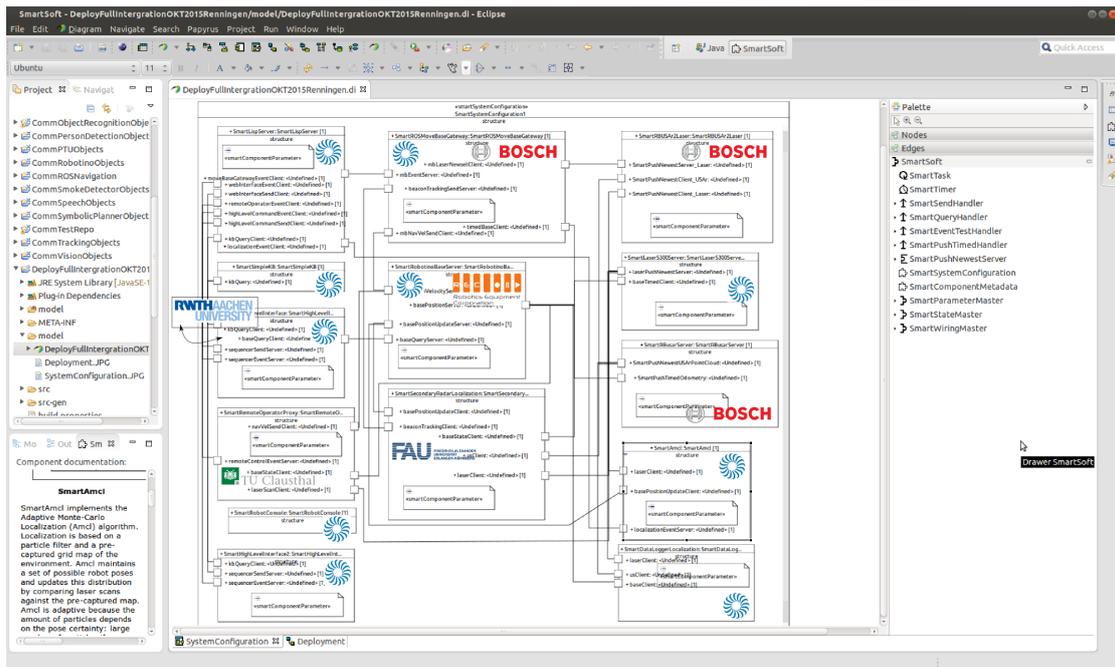


Abbildung 2.30: Systemkonfigurationsmodell (Krankenhausevaluation) der *SmartSoft* Komponenten aller Partner des Projekts, aus Sicht des Systemintegrators, in der SmartMDS Toolchain.

Projektpartner waren mit der SmartMDS Toolchain in der Lage, eigenständig Komponenten zu modellieren und Deployments zu modellieren und auszuführen (Integration und Tests). Die konsistente Integration dieser Komponenten zu einem Gesamt-Demonstrator wurde durch die Verwendung der Toolchain sichergestellt.

### Dynamisch-adaptives komponenten-basiertes Framework für Serviceroboter

Mit dem komponenten-basierten und service-orientierte Robotik-Framework *SmartSoft* stand dem Projekt eine leistungsfähige Ausgangsbasis zur Verfügung. Durch die service-orientierte Architektur erlaubt *SmartSoft* die Spezifikation von semantisch klar definierten Kommunikations-Schnittstellen, welche sowohl die Daten als auch das Kommunikationsverhalten, unabhängig von der darunterliegenden Kommunikations-Middleware, beschreiben. Die Trennung der Zuständigkeiten zwischen Framework (Komponentenhülle) und den Implementierungen des Komponententwicklers, sowie die Abstraktion der darunterliegenden Kommunikations-Middleware und deren Strukturen, sind weitere wichtige Grundlagen, welche das *SmartSoft* Framework schafft, um aufbauend darauf Komponenten- und Systemmodelle erstellen zu können.

Im Rahmen dieses Arbeitspaketes konnte das *SmartSoft* Framework um wichtige Aspekte, welche die Konfiguration der Softwarekomponenten betrifft, erweitert werden. Ausgehend von den von der HSU im Projekt erarbeiteten Methoden zur Modellierung von

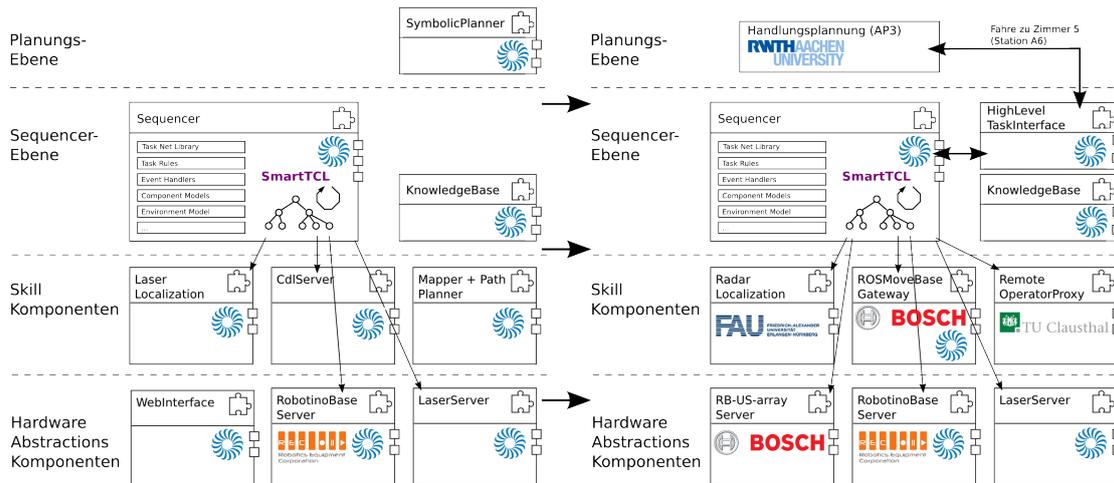


Abbildung 2.31: Wichtige Funktionalitäten konnten in Einzel-Demos der HSU bereits früh demonstriert werden. Die iterative Integration der Projektergebnisse der Partner war aufgrund des modellgetriebenen Entwicklungsansatzes, welcher auf stabile Softwarekomponenten aufsetzt, transparent möglich.

Softwarevariabilität wurden die Fertigkeiten der Komponenten-Konfiguration des *Smart-Soft* Frameworks entsprechend ausgebaut. Hiermit ist die Grundlage gelegt, um die in Modellen ausgedrückte Software-Variabilität systematisch zu managen. Die Variabilität kann schrittweise bis hin zur Laufzeit selbst verfeinert werden, was dann in den entsprechenden (Laufzeit) Konfigurationen von Komponenten resultiert. Entscheidend ist dies besonders für komplexe Systeme, die in offenen Umgebungen agieren müssen, z.B. Serviceroboter. Die immense Anzahl an möglichen Situationen und Eventualitäten in solchen Umgebungen macht einen strukturierten Ansatz, um Variabilität zur Design-Zeit auszudrücken und zur Laufzeit zu binden, gegeben den dann verfügbaren Informationen, unabdingbar. Eine Lösung durch klassisches „Ausimplementieren“ ist auf Grund der immensen Anzahl an Varianten bei solch komplexen Systemen in der Regel nicht effizient handhabbar.

### Kontinuierliche Integration und Realwelt Evaluation

Integration, als eine der großen Herausforderungen der Robotik, stand als wichtiger Punkt der Arbeiten der *HSU* in AP6 im Fokus. Komplexe Robotiksysteme bestehen aus einer Vielzahl von Technologien aus unterschiedlichen Domänen. Dies macht die Systemintegration zu einer Aufgabe, die systematisch gelöst werden muss. Der von der HSU entwickelte modellgetriebene Entwicklungsansatz expliziert und unterstützt die Rolle des Systemintegrators, welcher auf Modellebene die Softwarekomponenten als Ergebnis der Entwicklung anderer Rollen integriert. Gerade auch in Verbundprojekten, in welchen alle Projektpartner ihre Beiträge sukzessive zu einem Demonstrator integrieren, ist die systematische

Integration von essentieller Bedeutung. Die Umsetzung und das „gelebt werden“ des von der HSU entwickelten Softwareentwicklungsansatzes im Projekt, welches verschiedenste Rollen und Partner, von Radar Hardware Entwicklung bis zu Symbolischer Planung, von Akademia über KMUs und Industrie beinhaltet, ermöglichte eine erfolgreiche Integration bis hin zu einem gemeinsamen Real-Welt Projekt-Demonstrator. Damit diente die erfolgreiche Projektintegration aufgrund der Vielfalt der Rollen und Partner zugleich als relevanter Testfall für den Reifegrad und den Gewinn einer modellgetriebenen Systemintegration.

Abgestimmt auf den beschriebenen modellgetriebenen Entwicklungsansatz wurde ein iterativer Integrationsansatz im Projekt maßgeblich von der HSU moderiert. Die HSU konnte auf Grund ihrer Expertise in der Robotik und entsprechender Vorarbeiten die Tragfähigkeit der entwickelten Architektur und des Entwicklungsansatzes selbst mit existierenden eigenen „State of the Art“ Robotik-Komponenten demonstrieren, bis hin zu GM1 (Videos Real-Welt und Simulation Szenarien siehe [You]). Durch dieses iterative Vorgehen, der Demonstration für das Projekt wichtiger Technologien durch die HSU und der späteren Integration der Ergebnisse der Partner (siehe Abbildung 2.31), standen dem Projekt wichtige Grundfunktionalitäten, wie zum Beispiel Navigation und Lokalisierung, sehr früh zur Verfügung. Die von den Partnern entwickelten und auf den Anwendungsfall abgestimmten Funktionalitäten konnten schnell in entsprechende Komponenten verpackt werden. Die transparente Austauschbarkeit dieser Komponenten konnte durch die Modellierung der Komponentenhülle und die saubere Trennung von User- und Framework-Code bzw. die stabilen Schnittstellen der Komponentenhülle sichergestellt werden. Ein Betrachten oder gar Ändern der entsprechenden Implementierung der Komponente war nicht notwendig („Black-Box“).

Die SmartMDS Toolchain unterstützte die Systemintegration und das iterative Vorgehen mit speziell auf die Rolle des Systemintegrators zugeschnittenen und angereicherten Informationen aus den Modellen (Komponenten, Services, etc.). Das einfache Zusammenfügen von Komponenten auf Modellebene, ohne die Implementierung der Komponenten kennen zu müssen (BlackBoxes), ermöglicht die schnelle iterative Integration von Teilergebnissen (Komponenten). Die Mächtigkeit dieses Ansatzes wurde in mehreren Integrationsworkshop mit den Projektpartnern bis hin zu den GM1/2/3 Projektdemonstratoren so wie der Krankenhausevaluation gezeigt. Die einzelnen Projektpartner konnten mit diesem Ansatz selbstständig Teiltests und Demonstrationen ihrer Beiträge unter Wiederverwendung von existierenden Softwarekomponenten durchführen. Die Projektpartner waren in der Lage, verschiedenste Deployments (Szenarien) wie in der Projektarchitektur vorgesehen selbstständig und toolunterstützt auf den Robotern zu realisieren.

Der Einsatz der SmartMDS Toolchain (realisiert den modellgetriebenen Entwicklungsansatz) von Projektbeginn an sowie deren kontinuierliche Weiterentwicklung ermöglichten die kontinuierliche und iterative Integration der Projektergebnisse, im Gegensatz zur sonstigen Vorgehensweise der „Big Bang“ Integration am Projektende mit großem Risiko bis zum Ende.

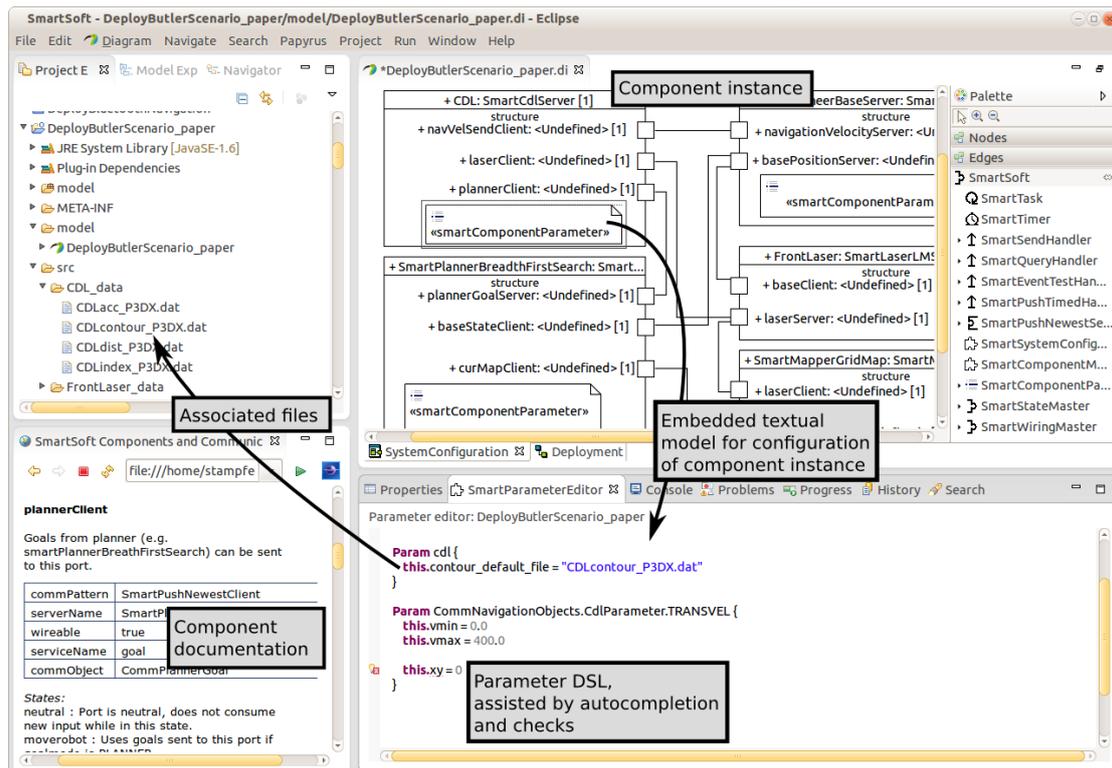


Abbildung 2.32: Sicht des Systemintegrators, schrittweise Verfeinerung der von den importierten Komponenten noch offen gelassenen Variationspunkte (Konfigurationen). Modellbasierte Dokumentation unterstützt den Nutzer kontextabhängig.

## Modellierung von Softwarevariabilitäten

Die SmartMDS Toolchain macht Rollentrennung und Wiederverwendung für die verschiedenen Rollen nutzbar. Wiederverwendung von Softwarekomponenten in verschiedenen Anwendungen benötigt zwingend die Explizierung von Konfigurationsmöglichkeiten (Variabilitätspunkten).

Die Modellierung von Softwarevariabilitäten (ausgedrückt durch Parametrierungsmodelle) in der SmartMDS Toolchain konnte bereits ab der Toolchain Version (V2.0) Ende 2013 vor dem ersten Deliverable D6.1 (M12) erreicht und veröffentlicht werden. Softwarevariabilitäten können hiermit in der modellgetriebenen SmartMDS Toolchain durchgängig modelliert, behandelt und schrittweise durch die verschiedenen Rollen verfeinert und genutzt (Roboter zur Laufzeit) werden (Abbildung 2.29). So kann nun bspw. der Systemintegrator ausgewählte Konfigurationen an Softwarekomponenten vornehmen (Abbildung 2.32), welche er als „black boxes“ vom Komponentenentwickler übernimmt (Darstellen von Variabilität für den Systemintegrator, „Variability Management“ entlang einer Rollentrennung im Entwicklungsprozess) [LIRS<sup>+</sup>14]. Weiter bilden die modellier-

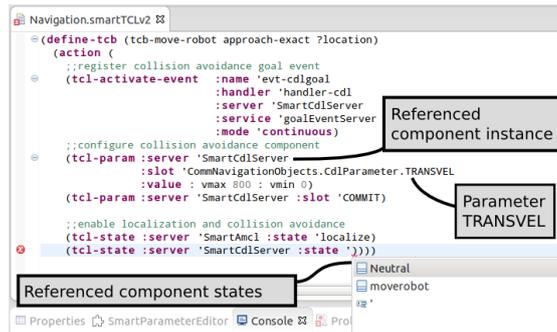


Abbildung 2.33: SmartMDS Toolchain aus der Sicht der Verhaltenentwicklung. Diese wird kontextabhängig mit Modell-Checks, Vervollständigung und Dokumentation unterstützt. Die Verhaltensmodelle sind unter anderem mit den modellierten Variationspunkten in den Softwaremodellen verknüpft.

ten Softwarevariabilitäten eine wichtige Grundlage für die explizite Modellierung der Wechselwirkung von Softwaremodellen und Verhaltensmodellen. Der Verhaltensentwickler wird von der SmartMDS Toolchain kontextabhängig mit Model-Checks, Vervollständigung und Dokumentation unterstützt (Abbildung 2.33). Der direkte Link zwischen Verhaltens- und Softwaremodellen (Komponenten und Systemkonfiguration) ermöglicht die konsistente Modellierung von Verhaltensmodellen [SLLS16]. Hiermit ist eine Eingliederung der Verhaltensentwicklung in einen durchgängig modellgetriebenen Entwicklungsansatz (Rollentrennung, Wiederverwendung, Ecosystem, etc.) möglich.

### Modellbasierte integrierte Dokumentation

Die SmartMDS Toolchain unterstützt die unterschiedlichen Rollen bei der Modellierung und Implementierung durch eine neu hinzugekommene ‘‘Dokumentationsansicht‘‘. Diese ermöglicht eine einfache und in die SmartMDS Toolchain integrierte Nutzung der Modelldokumentation durchgängig durch alle Rollen. Insbesondere der Rolle des Systemintegrators stehen zur Integration notwendige Informationen zu Komponenten und Services unmittelbar und aufbereitet zur Verfügung.

Die integrierte Dokumentation erlaubt es, menschenlesbare Erläuterungen über das Verhalten der Komponente und deren Services zu verfassen und mit dem sonst rein technischen Komponentenmodell zu verknüpfen. Daraus lässt sich eine vollumfängliche menschenlesbare Spezifikation und Dokumentation über Komponenten und Services generieren, welche besonders in der Integration von Bedeutung ist. So werden Inkonsistenzen der Dokumentation, vor allem bei Änderungen der S/W-Komponente, vermieden als auch überhaupt sichergestellt, dass für die Wiederverwendung relevante Informationen aus den Modellen für den User in aufbereiteter Form stets verfügbar sind.

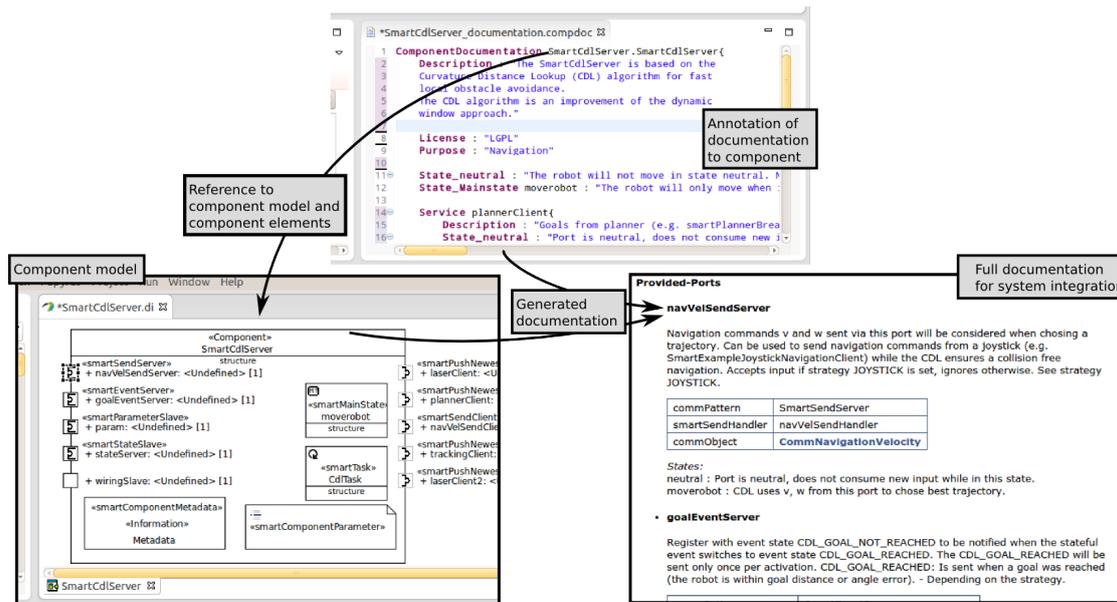


Abbildung 2.34: SmartMDS Toolchain, integrierte modellbasierte Dokumentation, unterstützt die unterschiedliche Rollen in ihrer Tätigkeit.

## Video Tutorials und Handbuch zur Nutzung

Über die Projektlaufzeit wurde die Dokumentation für die Nutzung der SmartMDS Toolchain sowie der von der HSU bereitgestellten Komponenten in verschiedenen Formen aufgebaut, aktualisiert und veröffentlicht. Für die Projektpartner (und für projektexterne Nutzer: Outreach and Dissemination) von besonderer Wichtigkeit sind die SmartMDS Toolchain Tutorial Videos [You] (Abbildung 2.35), die das Benutzen aller Entwicklungsschritte in der aktuellen Toolchain anschaulich demonstrieren. Zusätzlich zu den demonstrierenden Videos wurde ein Handbuch [FSS] zur Nutzung der Toolchain geschrieben. Dieses bietet eine umfassenden Anleitung für die Verwendung der Toolchain.

## Modellierungssprachen zur Verhaltensmodellierung

Mit SmartTCL stand dem Projekt bereits zu Beginn eine leistungsfähige domänenspezifische Sprache (DSL) zur Verhaltensmodellierung als Ausgangspunkt zur Verfügung. SmartTCL ermöglicht die kaskadierende (sich verfeinernde) Modellierung des Roboterhaltens in Tasknetzen, welche zur Laufzeit der Situation und Umgebung entsprechend expandiert und verfeinert werden. Mit SmartTCL stehen dem Entwickler Methoden zur Verfügung, um elegant und ausdrucksstark (ohne alle aufkommenden Fälle modellieren zu müssen) auf „Abweichungen“ im normalen Verhalten des Roboters reagieren zu können (z.B. blockierte Pfade, nicht erkannte Personen, etc.). SmartTCL, auf der Sequencer Ebene, bindet „nach oben“ an die Planungsebene an, um z.B. symbolische Planer (Metric-FF [Hof03], LAMA [RW08], etc.) zur Problemlösung zu nutzen. Generierte Pläne werden dann unter Nutzung der Mächtigkeit (Abweichungen vom Plan bzw. Fehlerbehandlung,

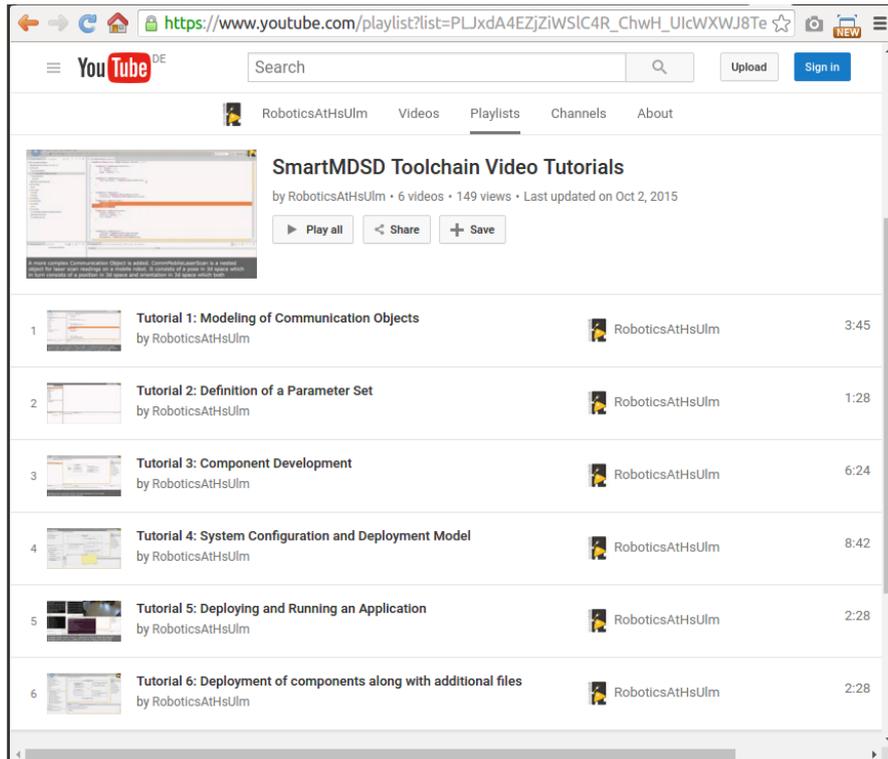


Abbildung 2.35: Die SmartMDS Toolchain Video Tutorials, zeigen alle Schritte der Entwicklung einer Robotik Anwendung mit Hilfe der SmartMDS Toolchain.

etc.) von SmartTCL abgearbeitet. Dies ist auch die Schnittstelle zur im Projekt entwickelten Planungs-Software der RWTH Aachen (Abbildung 2.31). SmartTCL orchestriert (Sequencer Ebene) alle Komponenten im Robotiksystem, die Skill Komponenten („unterhalb“) stellen von der konkreten Implementierung und der konkreten Roboterplattform abstrahierte Funktionalitäten zur Verfügung. Die Integration von SmartTCL in den von der HSU entwickelten modellgetriebenen Entwicklungsansatz und die SmartMDS Toolchain ermöglichen die rollenspezifische und integrierte Unterstützung der Verhaltensentwicklung sowie die effektive Wiederverwendung von Verhaltensmodellen [SLLS16]. Die Arbeiten zur Verhaltensmodellierung der RWTH Aachen im Projekt setzen auf die oben kurz beschriebene Planungsschnittstelle auf hohem Niveau oberhalb der Sequencer-Ebene auf und sind im Bericht zu AP3 beschrieben.

### SmartMDS Toolchain, Reifegrad und Stimmen aus dem Projekt

Die SmartMDS Toolchain wurde konsequent verbessert und erweitert sowie quellenoffen veröffentlicht. Die bereits zu Projektbeginn zur Verfügung stehende SmartMDS Toolchain konnte zu Beginn des Projekts rasch auf Eclipse in der Version Juno und damit auf eine technisch aktuelle Basis migriert werden (Abbildung 2.36). Kontinuierlich sind

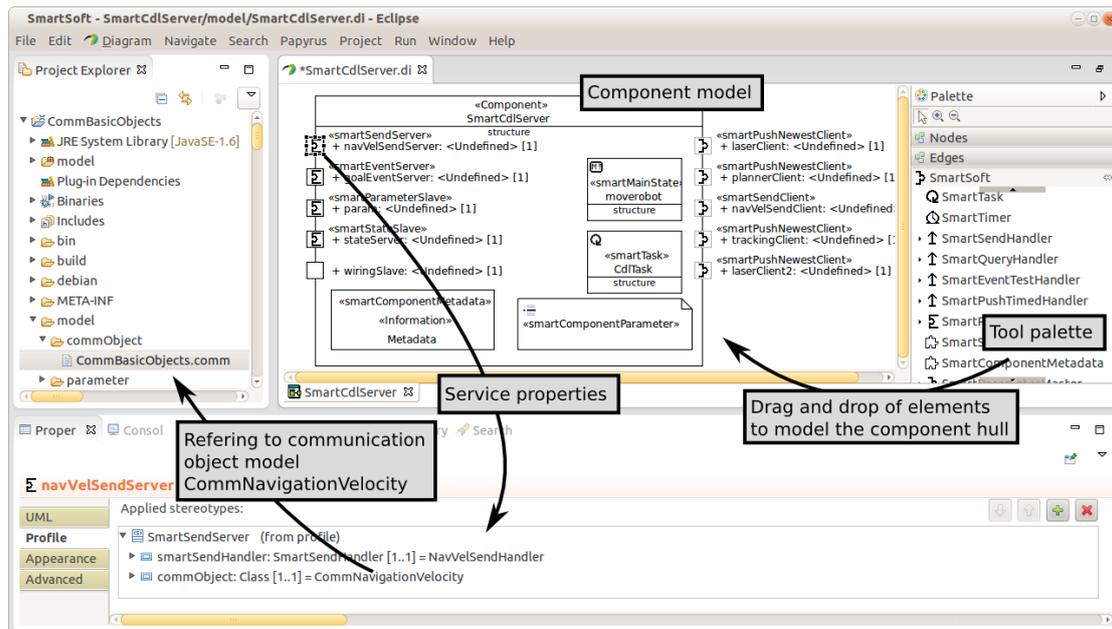


Abbildung 2.36: SmartMDS Toolchain V2.10, basierend auf Eclipse Juno, Sicht des Komponenten-Entwicklers

Bugfixes zur Stabilisierung in die Toolchain eingeflossen. Die Art und Anzahl der Bugfixes und die eigenständige Verwendung der Toolchain von Projektpartnern unterstreichen das Erreichen eines Reifegrades vergleichbar zu TRL 6 (Technology Readiness Level, “technology demonstrated in relevant environment”) [euR15].

Die über die Projektlaufzeit gepflegte SmartMDS Toolchain konnte im letzten Projektabschnitt durch gezielte Performance und Speicheroptimierung erheblich in ihrer Leistungsfähigkeit gesteigert werden. Insbesondere beim gleichzeitigen Nutzen vieler *SmartSoft* Komponenten (und somit Eclipse-Projekte), wie z. B. während der Integration durch den Systemintegrator, wirkt sich die Optimierung positiv aus.

Eine Nutzerstudie [SLLS16], welche überwiegend unter den Projektpartnern dieses Projektes und des Projekts FIONA [FIO] zum Abschluss der Krankensevaluation durchgeführt wurde, bestätigt die erfolgreiche Integration und den Nutzen, den die SmartMDS Toolchain in den Projekten bisher erzeugt hat:

- SmartSoft and the Smart-MDS Toolchain did a “fundamental contribution to the composition of software building blocks (components) in order to effectively build new applications” (94%)
- The presented approach clearly “helped to structure project collaboration towards demonstrators” (93%) and provided guidance for all participants (81%). They found (participant quote:): “that developing prototypes is easier since you can easily avoid

communication problems that normally arise at the beginning of a project.”

- Also 81% of participants felt that (agreed that) “the toolchain provides the necessary technical workflow-support and aids all developers along the concepts of SmartSoft by guiding them through the overall development process.”
- Compared to other approaches, they say that they made less errors in system integration (69%). In case they encountered “errors or problems[,] it was easy to identify the specific architectural element that caused it. The problem or error was fixed within that element alone without further influence to others.” (77%). In order to correct them, “ it was easy to identify who (component or partner) is affected and has to become active” (94%).

### 2.1.7 AP7: Prototypischer Roboter Aufbau

**Ziel:** Aufbau einer für die Evaluierung notwendigen Roboterplattform mit der dazugehörigen Basissoftware. Zusätzlich wird in diesem Arbeitspaket die notwendige Entwicklungs- und Simulationsumgebung aufgebaut, so dass im Projekt eine einheitliche Werkzeugkette vorhanden ist.

**Inhalt:** Für die Tests und Evaluierung der Arbeitspaketergebnisse wird im AP7 eine Prototypenplattform aufgebaut. Um die Partner schnell in die Lage zu versetzen, entsprechende Tests ihrer Entwicklungen vornehmen zu können, wird keine neue Plattform entwickelt, sondern auf das Robotersystem „Robotino“ in der Version 3 von Festo Didactic mit der entsprechend verfügbaren Basissoftware zurückgegriffen. Die Plattform bietet mit ihrer hohen Traglast eine sehr gute Voraussetzung, um auch neue, nur als Prototyp vorhandene, Sensoren zu testen und zu evaluieren. Die Robotino 3 Plattform wird dabei mit den gleichen Sensoren und Schnittstellen ausgerüstet, wie auch der im realen Krankenhausszenario eingesetzte ITA, so dass sowohl die Übertragbarkeit der im Testszenario erzielten Ergebnisse auf das Krankenhaus als auch der im Krankenhaus erkannten Probleme auf das Testszenario möglich sind. Darüber hinaus wird die Plattform mit entsprechender Sicherheitssensorik ausgestattet, die es erlaubt, im öffentlichen Raum die in AP2 entwickelte Sensorik zu testen und zu evaluieren. In Vorbereitung auf die Evaluierung im Arbeitspaket 8 wird der Prototyp in den AP2 - AP6 verwendet. Hierzu wird eine Teststrecke aufgebaut, in welcher die typischen Aufgaben intelligenter Transportassistenten (ITA) entwickelt und getestet werden können. Eine Simulationsumgebung basierend auf der von der REC GmbH entwickelten Robotersimulationsumgebung Robotino SIM Professional wird für Tests und Evaluierungen zur Verfügung gestellt. Die Integration der Ergebnisse aus AP2 – AP6 ist ebenfalls Inhalt dieses Arbeitspaketes.

**Ergebnis:** In der Anfangsphase des Projekt wurde die Entwicklung Version 3 der Robotinobasisplattform vollendet und allen Partnern verfügbar gemacht. Ein Robotino ist in Abb. 2.37 zu sehen. Es handelt sich dabei um eine mobile Plattform, die über eine Recheneinheit und einen omnidirektionalen Antrieb verfügt. Das mobile Robotersystem Robotino 3 wurde im Hinblick auf die Projektanforderungen modifiziert. Insbesondere wurde eine hardwareseitige Abschaltung der Motoren bei einer aufgetretenen Schutzfeldverletzung am Laserscanner eingebaut. Für ein verbessertes Fahrverhalten bei hohen Traglasten wurde ein spezielles Omnirad für Robotino entwickelt. Um einem Robotino eine adäquate Fahrgeschwindigkeit auch bei hoher Traglast zu ermöglichen (Schritttempo 6km/h), wurde die Verwendung stärkerer Elektromotoren untersucht.

Für die Ansteuerung der neu entwickelten Sensorhardware, d.h. Ultraschallsensorik und Radarlokalisierung, wurden Treiber entwickelt und in Smartsoft Komponenten integriert (D7.2). Dies ermöglichte die Ansteuerung der Sensorik und Verarbeitung der Sensordaten in den einzelnen Komponenten.

Im Laufe des Projekts stellte sich heraus, dass es praktischer wäre, eine Teststrecke in Renningen bei Bosch zu haben statt bei REC in München. Aufgrund dieser Entscheidung wurde bei Bosch eine Teststrecke realisiert (D7.3). Diese ermöglichte es, die Navigation



Abbildung 2.37: Robotino 3 Plattform (links) und neu entwickeltes Omnirad (rechts)

des Roboters zu entwickeln und zu testen. Flexible Raumbaulemente ermöglichten das einfache Aufbauen von Testszenarien und das Nachstellen verschiedener Situationen. Die Teststrecke diente auch als Basis für die Integrationsworkshops. So konnte die Bewegung des Robotersystems in Korridoren oder Kreuzungen, die Reaktion des Robotersystems auf statische und dynamische Hindernisse, die Verständigung zwischen Robotersystem und menschlichem Instrukteur sowie der Einfluss der Lokalisierung auf die Navigation als auch auf das Remote Management getestet und verbessert werden.

Die gemeinsame Entwicklungsumgebung stellt die SmartMDS Toolchain basierend auf Eclipse dar (D7.4). Als Programmiersprachen kamen neben verschiedenen DSLs, C++, Python und LISP zum Einsatz.

Für den Robotersimulator Robotino SIM wurde eine Krankenhausumgebung entwickelt (D7.5). Der simulierte Robotino wurde um Laserscanner und die Simulation eines Ultraschallarrays erweitert. Für eine möglichst realistische Umgebung wurde die Simulation von Glasscheiben entwickelt. Die Darstellung in Abbildung 2.38 zeigt eine Krankenhausumgebung in Robotino SIM. Möbel und Menschen werden so simuliert, dass die von dem Roboter erfassten Sensordaten weitestgehend den Sensordaten in einer realen Umgebung entsprechen. Dadurch kann eine sinnvolle und zielführende Softwareentwicklung mit dem Simulator als Testumgebung erreicht werden.

Die Robotinoplattform stellt die Basis für die Entwicklung des ITA-Prototypen dar. Aufbauend auf der Basis wurde die neue Sensorik in eine Trägerkonstruktion integriert und um zusätzliche Recheneinheiten, ein zusätzliches Batteriesystem und Netzwerkrouthinhardware erweitert. Abbildung 2.39a zeigt den Aufbau des ITA ohne Hülle und noch ohne Lokalisierungshardware. Im Lauf des Projekts wurde eine Hülle für den ITA Roboter designed und gefertigt, um die zukünftige Produktvision greifbarer zu machen. Beim Design wurden die Anforderungen aus den Mensch-Maschine Interaktion berücksichtigt.

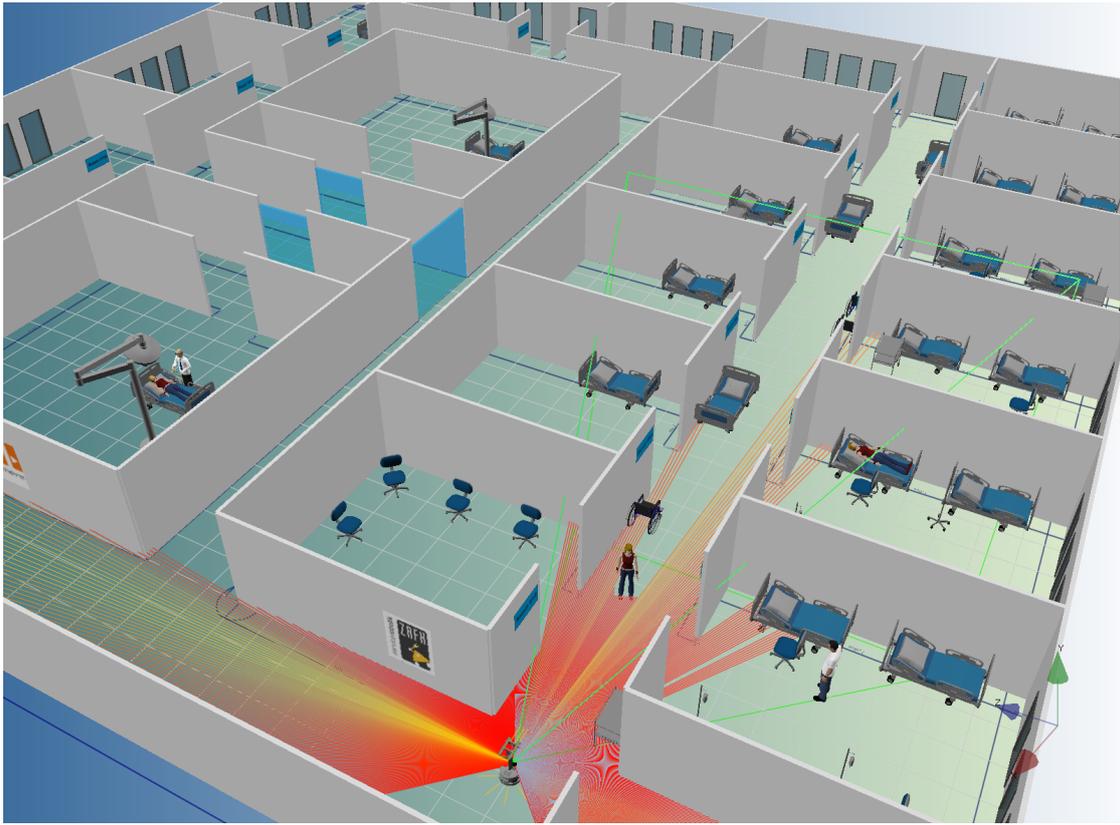


Abbildung 2.38: Simulationsumgebung Robotino SIM

Ein Bild des vollständigen Roboters, wie er auch in der Endevaluation im Krankenhaus verwendet wurde, ist in Abbildung 2.39b zu sehen.

Im Laufe des Projekts wurden alle implementierten Funktionalitäten in den Roboterprototyp integriert und deren Zusammenspiel optimiert. Dies umfasste unter anderem die Situationsanalyse und Verhaltensmodellierung (D7.7), die Fähigkeit des Roboters zu Führen und zu Folgen (D7.8), das direkte Interaktionsdesign (D7.9) und die Bereitstellung des Remote Managements (D7.10).

Die freigewordenen Kapazitäten durch das Wegfallen des Aufbaus einer Teststrecke bei REC wurden genutzt, um weiterführende Forschungsarbeiten im Bereich der Flottenkoordination durchzuführen. Es wurden SmartSoft-Komponenten entwickelt, die eine koordinierte Bewegung einer Flotte von Robotern auf vordefinierten Pfaden ermöglichen. Für die Definition von Pfaden wurde das Web-Interface von Robotino um die Möglichkeit erweitert, auf der Navigationskarte Pfade anzulegen. Diese Pfade werden von einer Koordinationskomponente eingelesen und in kleine Abschnitte unterteilt. Abbildung 2.40 zeigt ein Beispiel dafür. Sobald ein Roboter von einem Start- zum Zielpunkt fahren möchte, fragt er bei der Koordinationskomponente an und erhält den zu befahrenden Pfad. Während des Abfahrens des Pfades wird das aktuelle und nächste Teilstück des Pfades von der Koordinationskomponente für andere Roboter gesperrt. Sobald die Teilstücke wie-



(a) ITA Roboterprototyp ohne Hülle und Lokalisierungshardware



(b) Finaler ITA Roboterprototyp im Krankenhaus

Abbildung 2.39: ITA Roboterprototyp

der freigegeben werden, dürfen andere Roboter diese Teilstücke befahren. Auf diese Weise erfolgt eine ortsbasierte Koordination der Bewegung aller Roboter in der Roboterflotte.

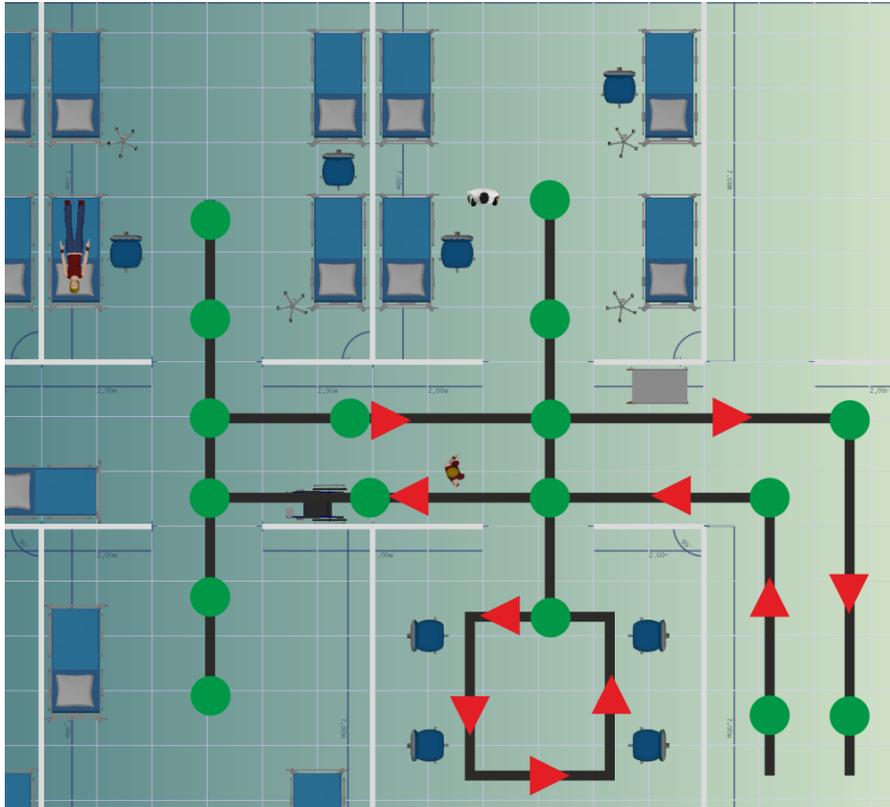


Abbildung 2.40: Flottenbetrieb

Aufbauend auf das Projekt iserveU wurde an der Universität Hohenheim eine Modellanlage von Festo Didactic mit dem in diesem Projekt entwickelten Flottenkoordinierungsalgorithmus ausgestattet (s. Abb. 2.41 und Video [You]). Der Betrieb wurde mit zwei Robotinos getestet. Mittlerweile kann die Funktionsweise des entwickelten Algorithmus auch an der Hochschule Ulm mit vier Robotinos demonstriert werden (siehe Videos [You]).



Abbildung 2.41: Teststrecke an der Universität Hohenheim

### 2.1.8 AP8: Pilotierung und Evaluierung im Krankenhaus

**Ziel:** Evaluierung der Arbeitsergebnisse in einer realen Krankenhausumgebung

**Inhalt:** Dieses Arbeitspaket startet zum Ende des Projekts und hat zum Ziel, die in den vorherigen Arbeitspaketen erarbeiteten Ergebnisse im Krankenhaus zu evaluieren. Damit soll sichergestellt werden, dass die erarbeiteten Lösungen nicht nur im Laborbetrieb, sondern auch in einer realen Umgebung funktionieren. Dies gilt insbesondere für die neuen Sensorkonzepte aus dem Arbeitspaket 2.

**Ergebnis:** Gegen Ende der Projektlaufzeit wurde die Evaluation in einem realen Krankenhaus geplant und durchgeführt. Als Krankenhaus wurde das Klinikum Stuttgart – Katharinenhospital<sup>3</sup> ausgewählt. Zum einen aufgrund der örtlichen Nähe zu Renningen, zum anderen aufgrund des bestehenden Kontakts durch die Sondierungsgespräche im Zuge der Projektantragsvorbereitung. Nach grundsätzlicher Einigung über die Form und Umfang der Kooperation erfuhr das Konsortium breite Unterstützung von Seiten des Klinikums für die Durchführung der Experimente im laufenden Stationsbetrieb. Die Durchführung wurde dem Konsortium auf der HNO Station (Haus D) erlaubt. Abbildung 2.42 zeigt eine Übersichtskarte der Station mit interessanten Punkten. Für diese Station wurde eine Teststrecke definiert (D8.1) und vorbereitet (D8.2). Für die Einsatzfähigkeit waren nur Hardwareaufbauten nötig, keine Umbauten der bestehenden Infrastruktur. Die Aufbauten erstreckten sich auf die Aufstellung der Radarbeacons, WLAN Infrastruktur und Kontrollrechner. Im Vorfeld wurden mit dem Krankenhaus alle Aspekte abgeklärt zu den Themen Datenschutz, Sicherheit, IT, Verhaltensregeln während der Durchführung und Information von Krankenhauspersonal, Patienten und Besuchern. Vor der Durchführung der eigentlichen Evaluation fand ein Vortest erst bei Bosch in Renningen und dann noch für zwei Tage im Krankenhaus statt, um das Gesamtsystem zu erproben und feinabzustimmen.

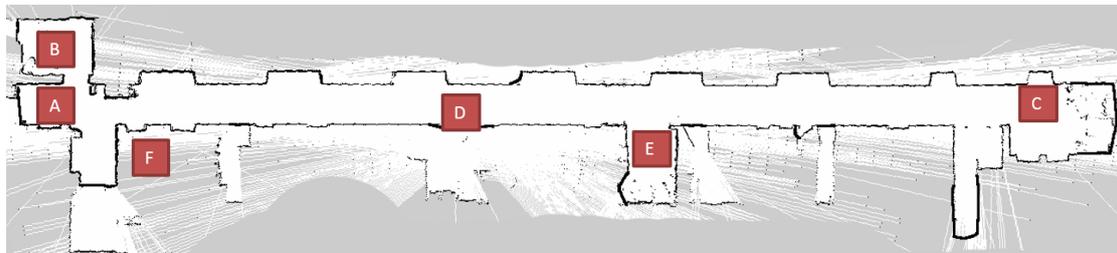


Abbildung 2.42: Karte der Station auf der die Evaluation des ITA Roboters durchgeführt wurde. Die wichtigsten Orte waren: (A) Startposition, (B) Warteraum, (C) Gangende mit Fahrstühlen, (D) Stationsempfang, (E) Kaffeküche, (F) Fahrstühle

Die Evaluation fand dann Ende November 2015 statt. Im Vorfeld wurden Storyboards

<sup>3</sup>Klinikum Stuttgart – Katharinenhospital, Kriegsbergstraße 60, 70174 Stuttgart

erarbeitet, um ein systematisches Testen und Auswerten der Versuche zu ermöglichen. Die Storyboards beinhalteten zum einen die Spezifikation der durch den Roboter durchzuführenden Aufgabe, zum anderen die Beschreibung von Ereignissen, wie z.B. Personengruppe versperrt den Weg am Fahrstuhl oder Person kommt um die Ecke und kreuzt den Weg des ITAs, um eine gewisse Dichte an interessanten Situationen zu garantieren. Diese Ereignisse wurden dann während der Durchführung durch Mitarbeiter des Konsortiums durchgeführt. Die Aufgaben umfassten die in AP1 definierten Use Cases in verschiedenen Varianten:

1. (Ad hoc) Transportaufgaben
2. Folgen von Personen
3. Führen von Personen
4. Remote Management zur Unterstützung bei unvorhergesehenen, nicht automatisch lösbaren Situation

Die verschiedenen Aufgaben wurden sowohl einzeln als auch in einer kombinierten Mission getestet. So lautete eine Mission beispielsweise: Starte am Ganganfang (A), nimm einen Gegenstand im Warteraum (B) entgegen, bringe den Gegenstand zum Gange (C), hole eine Person am Empfang (D) ab, führe Sie in die Kaffeeküche (E) und folge der dort wartenden Person. Die Versuche wurden tagsüber während dem laufenden Stationsbetrieb durchgeführt, so dass es neben den Ereignissen, die durch Mitarbeiter provoziert wurden, zu einer Vielzahl interessanter Begegnungen zwischen Roboter, Personal, Patienten und Besuchern kam. Alle Menschen waren dazu angehalten, sich möglichst natürlich zu verhalten. Während der Evaluation variierte der Verkehr auf den Gängen tageszeitabhängig sehr stark. Abbildung 2.43 zeigt Bilder von der Evaluation. Während der Ausführung der Versuche wurden die Daten für die spätere Auswertung aufgezeichnet und dann unter folgenden Gesichtspunkten evaluiert:

- Sensorik (D8.3)
- Navigation (D8.4)
- Aufgabenplanung (D8.5)
- Remote-Management (D8.6)
- Interaktionskonzept (D8.7)

Insgesamt konnte gezeigt werden, dass der entwickelte ITA Roboter die gestellten Aufgaben im realen Betrieb in den meisten Fällen erfüllen konnte. Zwei Aspekte erschwerten die Durchführung. Zum einen kam es zum Ausfall des Ultraschallarrayprototyps während der Evaluation, so dass die Messungen mithilfe des Laserscanners durchgeführt werden mussten. Zum anderen kam es vereinzelt zu Missionsabbrüchen aufgrund von Fehllokalisierungen. Dies führte dazu, dass der Planer aufgrund falsch geschätzter Position und Orientierung keinen Weg mehr zum Ziel fand und deshalb einen Stopp veranlasste. Dieses



(a) ITA beim Ausführen einer Transportaufgabe



(b) ITA beim Folgen einer Person

Abbildung 2.43: Impressionen von der Roboterevaluation im Krankenhaus

Problem trat vorwiegend an bestimmten Stellen auf der Station auf. Die Analyse ergab, dass die Ursache für die Fehllokalisierung hauptsächlich auf den Sensoraufbau zurückzuführen war, da es bei Drehungen über  $45^\circ$  zu Abschattungen kam. Dies konnte durch den verbesserten Sensoraufbau 2.15 gelöst werden.

Neben der technischen Evaluation wurde auch der Eindruck des Roboters und seines Verhaltens auf Personal, Patienten und Besucher untersucht. Bei der Befragung anwesender Personen wurden überwiegend positive Eindrücke erfasst. Kritiken passen zu den in den Beobachtungen identifizierten Schwachstellen des sozialen Verhaltens des ITA, z.B. beim frühzeitigen Ausweichen gegenüber Personen. Bei der Befragung des Krankenhauspersonal zeigte sich eine hohe Erwartungshaltung an die Fähigkeiten des Systems. So wurden z.B. auch Funktionen nachgefragt, die weit über die Transportfunktionalität hinausgehen, wie z.B. natürliches Sprachverstehen oder Feinmanipulation von Objekten. Zudem wurden Folgen des Einsatzes eines solchen Roboters im Krankenhaus in mehreren Aspekten kritisch hinterfragt. Ebenfalls wurde von den Mitarbeitern gesehen, dass es sich bei der Einführung einer solchen Technologie um einen langfristigen Prozess handelt, der eine schrittweise Gewöhnung aller Beteiligten an diese beinhaltet.

## 2.2 Wichtigste Positionen des zahlenmäßigen Nachweises

Im Projekt iserveU waren die wesentlichen Kosten die Personalkosten. Diese wurden voll ausgeschöpft. Die genauen Werte können dem zahlenmäßigen Nachweis entnommen werden.

## 2.3 Notwendigkeit und Angemessenheit der geleisteten Arbeit

Der Verlauf der Arbeit im Projekt folgte der im Projektantrag formulierten Zielsetzung. Alle im Arbeitsplan formulierten Aufgaben wurden erfolgreich bearbeitet, es waren keine zusätzlichen Ressourcen für das Projekt nötig.

## 2.4 Voraussichtlicher Nutzen, insbesondere Verwertbarkeit des Ergebnisses im Sinne des fortgeschriebenen Verwertungsplanes

Die Verwertung der Ergebnisse bei den einzelnen Projektpartnern ist detailliert in den einzelnen Erfolgskontrollberichten beschrieben. Hervorzuheben sind hier insbesondere:

### 2.4.1 Robert Bosch GmbH

**Ultraschallarray:** Die Arbeiten an der Hard- und Software des Ultraschallsystems haben diese neue Sensortechnologie einen guten Schritt weiter gebracht in Richtung eines Produkts. Insbesondere die neue Fertigungstechnik auf Basis von Mems-Sensoren eröffnet Möglichkeiten einer deutlichen Preisreduktion. Als zukünftige Weiterentwicklung stehen die Integration der Prototypkomponenten in ein kompaktes System mit reduziertem Stromverbrauch an sowie die Sicherheitszertifizierung des Sensors. Die im Rahmen des Projekts gesammelten Erfahrungen mit dem prototypischen Sensorsystem werden hierbei einen wichtigen Beitrag für die Weiterentwicklung des Sensors leisten. Geplant ist die Weiterverwendung des Sensorprototyps für die Weiterentwicklung der Sensordatenverarbeitung und Tests mit dem Robotersystem.

**Mensch-Maschine Interaktion:** Die Erkenntnisse aus den durchgeführten Studien zu direkten und indirekten Interaktionskonzepten zwischen Transportrobotern und Menschen werden weiter verwertet, um die Interaktion zukünftiger Roboterprodukte mit gleichem oder ähnlichem Anwendungsfeld zielgerichtet zu gestalten und damit die Akzeptanz solcher Systeme zu verbessern und Kaufanreize zu bieten. Ebenso bilden sie die Grundlage weiterführender Forschungsarbeit auf dem Gebiet und bei der Konzeption neuer Studien. Ermittelte Erwartungen an die äußere Gestaltung von Transportrobotern ermöglichen darüber hinaus die Entwicklung realitätsnaher Prototypen, welche bereits frühzeitig im Entwicklungsprozess hinsichtlich relevanter

technischer Lösungen sowie übertragbarer Reaktionen des menschlichen Umfeldes untersucht und entsprechend weiterentwickelt werden können.

**Navigation:** Die entwickelte Hinderniserkennungs- und Navigationskomponente soll weiter für Forschungs- und Entwicklungszwecke verwendet werden. Speziell die weiterführende Einbringung der Erkenntnisse aus der Mensch-Maschine Interaktionsforschung ist der Fokus der Entwicklungen. Dies soll zu einem Robotersystem führen, das natürlich mit seiner Umgebung interagiert und vorausschauend handelt.

**Roboter Aufbau:** Die gesammelten Erfahrungen während des Aufbaus der Robotersysteme als auch während deren Einsatzes geben viele Hinweise für mögliche Verbesserungen bei der Konzeption eines Produkts. So könnte beispielsweise der Schwerpunkt und das Gewicht des Robotersystems verbessert werden. Ebenso gibt es Hinweise für die Verbesserung der Hüllenform, um sie noch ansprechender für Benutzer und Passanten zu gestalten. Die aufgebauten Robotersysteme und die Teststrecke werden weiterhin für Forschungs- und Entwicklungszwecke genutzt werden.

## 2.4.2 Technische Universität Clausthal

- Die Entwicklung des Laufzeitverifikationsverfahrens wird am Lehrstuhl für Software Systems Engineering in Rahmen einer Dissertation weiter durchgeführt.
- Der Robotino 3 wird für den Aufbau von Fallstudien innerhalb aktueller Forschungsarbeiten am Lehrstuhl benutzt.
- Das in diesem Projekt vertiefte Fachwissen wird derzeit in einem Projekt mit VW zum Thema Anforderungsmanagement einbezogen.
- Die TU Clausthal hat seinen eigenen Demonstrator aufgebaut und stellt diesen in unterschiedliche Veranstaltungen dar. Es ist vorgesehen den Demonstrator in zukünftige Fachtagungen, z.B. CeBIT und Hannover Messe, vorzuführen.

## 2.4.3 Hochschule Ulm

- Die im Zusammenhang mit AP6 von der HSU entwickelten modellgetriebenen Methoden sowie die von REC im Projekt erarbeiteten Methoden zur Flottenkoordination führten zu einem bewilligten Anschlussprojekt LogiRob (BMBF, KMU-innovativ Juni 2016 - Mai 2019). Die Partner REC und die HSU setzen gemeinsam mit der Festo Didactic SE in LogiRob auf die in iserveU erarbeiteten Grundlagen auf. Der Fokus von LogiRob liegt auf einem Multi-Robot-Transportsystem im mit Menschen geteilten Arbeitsraum.
- Die HSU konnte sich durch die erfolgreiche Einführung eines durchgängigen modellgetriebenen S/W-Entwicklungsansatzes für die Robotik (AP6) nachhaltig in der Robotik-Community in diesem Themenkomplex verankern (zusammen mit der TU

München Koordinator der Topic Group „Software Engineering, System Integration, System Engineering“ der euRobotics AISBL; Organisator der DSLRob Workshop-Serie, etc.). Aus dieser Einbindung resultierten zwei Beteiligungen an Anträgen im EU-H2020 ICT Programm, die sich aktuell in Begutachtung befinden.

- Desweiteren fließen die Ergebnisse der Forschungsarbeiten in die Lehre an der HSU ein (Bachelor Informatik Vorlesung „Autonome mobile Systeme“; Master Informationssysteme „Autonome Systeme“) und sichern die Aktualität der Lehre im Bereich der autonomen Systeme und der Servicerobotik, der Grundlagen von Industrie 4.0 sowie der S/W-Engineering-Ansätze (modellgetriebene S/W-Entwicklung). Dies führte zudem zur Förderung eines kooperativen Promotionskollegs der HSU zusammen mit der Uni Ulm zum Thema „Cognitive Computing in Socio-Technical Systems“ ([http://www.hs-ulm.de/KPK\\_Cognitive\\_Computing](http://www.hs-ulm.de/KPK_Cognitive_Computing)).
- Die von der HSU im Projekt (weiter)entwickelten Softwarekomponenten (Navigation, Lokalisierung, Koordination, etc.) stehen als Open Source der Community zur Verfügung. Besonders hervorzuheben sind die auf das Robotik System Robotino abgestimmten und einfach zu verwendenden (Debian Pakete) SmartSoft Komponenten, welche auf der openrobotino.org (<http://wiki.openrobotino.org/index.php?title=Smartsoft>) Plattform veröffentlicht sind.
- Die Nachhaltigkeit der Entwicklung der Robotik-IDE SmartMDS Toolchain in iserveU wird durch die fortschreitende Weiterentwicklung über Projektgrenzen hinaus, die Verwendung der Toolchain von extern Partnern und in anderen Forschungsprojekten (z.B. ITEA2 Projekt FIONA [FIO]) sowie die kontinuierliche quellenoffene Veröffentlichung der SmartMDS Toolchain sichergestellt. Die freie Verfügbarkeit von Werkzeugen für die modellgetriebene Softwareentwicklung ist ein wichtiger Schritt in Richtung eines Software Business Ecosystems mit einer in anderen Industrien bereits üblichen Rollenverteilung in Wertschöpfungsnetzen.

#### 2.4.4 Symeo GmbH

Die technischen Ergebnisse der Ortung sind sehr positiv. Symeo wird diese Ergebnisse für die Produktentwicklung eines speziell auf Robotik und Ortung dafür zugeschnittenen 2D Ortungssystems nutzen. Dabei liegt der Schwerpunkt auf Produkten mit hochgenauer Ortung und sparse Infrastructure für Industrie- und Logistikanwendungen. Für weitere Bereiche werden Partner gesucht.

#### 2.4.5 FAU

- Das Vorhaben diente der Ausbildung hoch-qualifizierter Ingenieure/-innen und Promotionsstudenten/-innen. Mehrere Studentische Abschlussarbeiten der Studiengänge Elektrotechnik, Medizintechnik, und Mechatronik wurden im Rahmen des

Projektes betreut. Die Forschungsergebnisse fließen bereits in Lehrveranstaltungen ein.

- Die vom Projektpartner Symeo GmbH entwickelten Funk-Sensorknoten werden zu Forschungszwecken im Rahmen von Dissertationen, sowie studentischer Abschlussarbeiten weiterverwendet. Von der FAU entwickelter Quellcode wird von der Symeo GmbH in das eigene Software Framework „Fusion Engine“ integriert.
- Die innerhalb des Projekts entwickelten Algorithmen zur hochauflösenden Winkel-schätzung werden bereits im Rahmen neuer Projekte weiterentwickelt.

#### **2.4.6 RWTH Aachen**

- In einem Praktikum zum Thema Robotik werden die Erfahrungen und Ergebnisse des Projekts, sowie die entwickelte Software von Studenten verwendet, weiterentwickelt und evaluiert. Im Rahmen von verschiedenen Seminaren wird zudem die Anwendbarkeit der entwickelten Modellierungssprachen in anderen Domänen, zum Beispiel autonomem Fahren, untersucht.
- Auf der Informationsveranstaltung “Aachen 2025 - Digitalen Wandel erleben” wird das Projekt im Themenpark Gesundheit einer breiten Öffentlichkeit präsentiert. Der Kontext ist es, eine Lösung für die Zukunft der Servicerobotik als Assistenz von klinischem Personal zu demonstrieren. Vor allem junge, technikinteressierte Menschen sollen so nachhaltig für das Thema der modellgetriebenen Entwicklung von Robotikanwendungen begeistert werden.
- Das entwickelte Metamodell zur Beschreibung von Aufgaben und deren Ausführungskontexten soll generalisiert werden und dessen Anwendbarkeit in anderen Domänen geprüft werden. Auch die Übersetzung von Zielen nach Aktionen mithilfe einer allgemeinen Planungssoftware wird so wiederverwendet.

#### **2.4.7 REC**

- In dem Projekt ist ein weit fortgeschrittener Prototyp eines mobilen Roboters für den Einsatz als Transportroboter in Krankenhäusern entwickelt worden. Viele Entwicklungen an der Hardware können zukünftig in der Standardversion von Robotino Verwendung finden.
- Das entwickelte Software-Framework wird zukünftig genutzt, um Robotino als flexibles Transportsystem in Fabrikumgebungen zu positionieren.

## 2.5 Während der Durchführung des Vorhabens bekannt gewordener Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

Der Einsatz mobiler Robotersysteme als flexible Transportsysteme ist ein Gebiet zahlreicher Entwicklungen. Im Laufe des Projekts wurden einzelne Initiativen bekannt, die Roboter für Transportaufgaben entwickeln wollen. Zu nennen wäre hier beispielsweise *Starship*<sup>4</sup>. Diese adressieren jedoch andere Anwendungsdomänen mit anderen Eigenschaften. Vergleichbare Arbeiten bei anderen Stellen parallel zu den im Projekt durchgeführten Aufgaben mit vergleichbaren Resultaten sind keine bekannt geworden.

## 2.6 Erfolgte oder geplante Veröffentlichungen des Ergebnisses

### 2.6.1 Robert Bosch GmbH

- Mathis Lauckner. Räumliche Interaktion zwischen Mensch und Roboter im Gang. Dissertation, Technische Universität Berlin, 2016.  
DOI: <http://dx.doi.org/10.14279/depositonce-5084>

### 2.6.2 FAU

- Sergio Flores, Martin Vossiek. A Robust Air-Coupled Ultrasonic Piezoelectric-Carbon-Fiber-Reinforced-Polymer Transducer for Mobile Robot Navigation. In: *International Ultrasonics Symposium*, Chicago, September, 2014.
- Yassen Dobrev, Martin Vossiek, Denys Shmakov. A Bilateral 24 GHz Wireless Positioning System for 3D Real-Time Localization of People and Mobile Robots. In: *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, Heidelberg, April, 2015.
- Yassen Dobrev, Sergio Flores, Martin Vossiek. Multi-modal sensor fusion for indoor mobile robot pose estimation. In: *Proceedings of the IEEE/ION Position Location and Navigation Symp. (PLANS)*, Savannah, April, 2016, S. 553-556
- Sergio Flores, Johanna Geiss, Martin Vossiek. An Ultrasonic Sensor Network for High-Quality Range-Bearing-Based Indoor Positioning. In: *Proceedings of the IEEE/ION Position Location and Navigation Symp. (PLANS)*, Savannah, April, 2016, S. 572-576

### 2.6.3 Symeo GmbH

- Vorstellung der Ergebnisse im Rahmen von Kundengesprächen mit ausgewählten Endkunden z.B. im Bereich Fassadenrobotik oder automatische Hafenfahrzeuge.

---

<sup>4</sup><https://www.starship.xyz/>

- Teile des HF-Designs werden auch in einer Promotion eines Symeo Mitarbeiters publiziert werden, voraussichtlich März 2017.

#### 2.6.4 RWTH Aachen

Die Ergebnisse des iserveU Projektes sind in mehreren Publikationen der RWTH Aachen und der Dissertation “An Extensible Component & Connector Architecture Description Infrastructure for Multi-Platform Modeling” eingegangen.

- Robert Heim, Pedram Mir Seyed Nazari, Jan Oliver Ringert, Bernhard Rumpe, Andreas Wortmann. Modeling Robot and World Interfaces for Reusable Tasks. In: Intelligent Robots and Systems Conference (IROS’15), pp. 1793-1789. IEEE, 2015.
- Jan Oliver Ringert, Alexander Roth, Bernhard Rumpe, Andreas Wortmann. Language and Code Generator Composition for Model-Driven Engineering of Robotics Component & Connector Systems. In: Journal of Software Engineering for Robotics (JOSER), 6(1):33-57, 2015.
- Jan Oliver Ringert, Bernhard Rumpe, Andreas Wortmann. Tailoring the MontiArc-Automaton Component & Connector ADL for Generative Development. In: MORSE/VAO Workshop on Model-Driven Robot Software Engineering and View-based Software-Engineering, pp. 41-47, ACM, 2015.

Zwei weitere Publikationen sind noch in Begutachtung. Die Veröffentlichung ist in 2016 vorgesehen.

- Kai Adam, Arvid Butting, Robert Heim, Oliver Kautz, Jérôme Pfeiffer, Bernhard Rumpe, Andreas Wortmann. Modeling Robotics Tasks for Better Separation of Concerns, Platform-Independence, and Reuse. Technischer Report.
- Kai Adam, Arvid Butting, Robert Heim, Oliver Kautz, Bernhard Rumpe, Andreas Wortmann. Model-Driven Separation of Concerns for Service Robotics.

Zudem präsentiert die RWTH Aachen die Projektergebnisse am 24.09.2016 auf der Fachmesse “Aachen 2025” im Themenpark “Gesundheit”.

#### 2.6.5 TU Clausthal

Die TU Clausthal hat die erarbeiteten Ergebnisse in unterschiedlicher Form der Öffentlichkeit zugänglich gemacht. Dies ist sowohl durch drei wissenschaftlichen Arbeiten als auch durch zwei Veröffentlichungen auf internationalen Fachkonferenzen erfolgt.

- A. Aniculaesei, D. Arnsberger, F. Howar, A. Rausch. Towards the Verification of Safety-critical Autonomous Systems in Dynamic Environments. In: *First International Workshop on Verification and Validation of Cyber-Physical Systems (co-located with iFM 2016)*, Juni 2016, Reykjavik, Island.

- T. Warnecke. Sichere Integration von Teilsystemen zu System-of-Systems durch formale Verifikation. In: *Tag des Systems Engineering (TdSE 2014)*, 299-308, November 2014, Bremen, Deutschland.
- Daniel Arnsberger. Entwurf und Prototypische Implementierung für das Fahren und die Zielfindung eines Serviceroboters, Bachelorarbeit, Dezember 2014, Clausthal-Zellerfeld, Deutschland.
- Tim Warnecke. Konfiguration und Integration von Teilsystemen zu System-of-Systems mit Regelüberwachung und -sicherung, Masterarbeit, März 2014, Clausthal-Zellerfeld, Deutschland.
- Daniel Arnsberger. Towards the Verification of Safety-critical Autonomous Systems in Dynamic Environments, Masterprojekt, Mai 2016, Clausthal-Zellerfeld, Deutschland.

Eine Doktorarbeit zum Thema Absicherung von sicherheitskritischen autonomen Systemen in heterogenen Umgebungen ist gerade in Arbeit. Eine weitere Publikation zum Thema Formale Sprachen zur Beschreibung der Schnittstellen zwischen autonomen Systemen und ihren Umgebungen ist derzeit noch in Arbeit und die Veröffentlichung wird demnächst geplant.

### 2.6.6 REC

Die REC GmbH plant zusammen mit der HS Ulm Veröffentlichungen zu der entwickelten Flottenkoordinierung.

### 2.6.7 Hochschule Ulm

#### Publikationen

- Christian Schlegel, Alex Lotz, Matthias Lutz, Dennis Stampfer, Juan F. Inglés-Romero, Cristina Vicente-Chicote. Model-Driven Software Systems Engineering in Robotics: Covering the Complete Life-Cycle of a Robot. Workshop Roboter-Kontrollarchitekturen, Informatik 2013, Springer LNI der GI, Koblenz, September 2013. ISBN 978-3-88579-614-5
- Dennis Stampfer, Christian Schlegel. Dynamic State Charts: composition and coordination of complex robot behavior and reuse of action plots. *Journal of Intelligent Service Robotics*, Volume 7, Number 2, pages 53-65, ISSN 18612776, Springer, 2014.
- Matthias Lutz, Dennis Stampfer, Alex Lotz, Christian Schlegel. Service Robot Control Architectures for Flexible and Robust Real-World Task Execution: Lessons Learned and Open Challenges. Workshop Roboter-Kontrollarchitekturen, Informatik 2014, Stuttgart, September 2014.

- Alex Lotz, Juan F. Inglés-Romero, Dennis Stampfer, Matthias Lutz, Cristina Vicente-Chicote, Christian Schlegel. Towards a Stepwise Variability Management Process for Complex Systems – A Robotics Perspective. *International Journal of Information System Modeling and Design (IJISMD)*, IGI Global, DOI:10.4018/ijismd.2014070103, September, 2014
- Christian Schlegel, Alex Lotz, Matthias Lutz, Dennis Stampfer, Juan F. Inglés-Romero, and Cristina Vicente-Chicote. Model-driven software systems engineering in robotics: Covering the complete life-cycle of a robot. *Journal IT - Information Technology: Methods and Applications of Informatics and Information Technology*, Volume 57, Issue 2, Pages 85–98, ISSN (Online) 2196-7032, ISSN (Print) 1611-2776, DOI: 10.1515/itit-2014-1069, DE GRUYTER, March 2015.
- Alex Lotz, Arne Hamann, Ingo Lütkebohle, Dennis Stampfer, Matthias Lutz and Christian Schlegel. Modeling Non-Functional Application Domain Constraints for Component-Based Robotics Software Systems. 6th International Workshop on Domain-Specific Languages and models for ROBotic systems (DSLRob-15), <http://arxiv.org/abs/1601.02379>, Hamburg, October, 2015.
- Dennis Stampfer, Alex Lotz, Matthias Lutz and Christian Schlegel. The SmartMDSO Toolchain: An Integrated MDSO Workflow and Integrated Development Environment (IDE) for Robotics Software. Special Issue on Domain-Specific Languages and Models in Robotics, *Journal of Software Engineering for Robotics (JOSER)*, 7(1), 3-19 ISSN: 2035-3928, July 2016.

## Dissemination

- Projektvorstellung iserveU in Vortragsprogramm des “7. Ulmer Robotertag” (18.02.2014 Ulm, öffentlich für Fachpublikum)
- Christian Schlegel, Dennis Stampfer. “Tutorial on Managing Software Variability in Robot Control Systems”, *Robotics Science and Systems (RSS)*, University of California, Berkeley, CA, USA (12.-16.07.2014)
- Dennis Stampfer, Alex Lotz, Matthias Lutz, Christian Schlegel. Systematische Wiederverwendung von modularen Software-Komponenten für intelligente Serviceroboter. *Sonderforum Robotik. Motek 2014*, Stuttgart, 9. Oktober 2014.
- Christian Schlegel, Alex Lotz, Matthias Lutz, Dennis Stampfer. Supporting Separation of Roles in the SmartMDSO-Toolchain: Three Examples of Integrated DSLs. 5th International Workshop on Domain-specific Languages and Models for Robotic Systems (DSLRob) in conjunction with the 2014 International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAN 2014), Bergamo, Italy, October 20th 2014. Christian Schlegel. Model-driven software development in robotics: Supporting run-time adaptation by managing variability in software development.

- Christian Schlegel. Invited talk at Workshop System Engineering Human-Centered Intelligent Vehicles, IEEE Int. Conf. on System, Man and Cybernetics (SMC), San Diego, USA, 2014.
- Alex Lotz, Matthias Lutz, Dennis Stampfer and Christian Schlegel. “SmartSoft communication patterns proven in industry”, in Workshop Networking and Communication for Robots. euRobotics Forum 2015, Vienna, Austria.
- Projektvorstellung iserveU im Vortragsprogramm des “8. Ulmer Robotertag” (19.02.2015 Ulm, öffentlich für Fachpublikum) Vorstellung der Projektaktivitäten in einer Vortragsreihe der Aussteller der „InnoRobo“ in Lyon (01.07. - 03.07.2015)

# Literaturverzeichnis

- [ASK<sup>+</sup>05] N. Ando, T. Suehiro, K. Kitagaki, T. Kotoku, and Woo-Keun Yoon. RT-Component Object Model in RT-Middleware Distributed Component Middleware for RT (Robot Technology). In *IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '05)*, pages 457–462, June 2005.
- [Bö01] H.J. Böhme. Serviceroboter und intuitive Mensch-Roboter-Interaktion. *Habilitationsschrift, Technische Universität Ilmenau*, 2001.
- [BGHS04] Howard Barringer, Allen Goldberg, Klaus Havelund, and Koushik Sen. *Rule-Based Runtime Verification*, pages 44–57. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [BLS05] Mike Barnett, K. Rustan M. Leino, and Wolfram Schulte. The spec# programming system: An overview. In *Proceedings of the 2004 International Conference on Construction and Analysis of Safe, Secure, and Interoperable Smart Devices, CASSIS'04*, pages 49–69, Berlin, Heidelberg, 2005. Springer-Verlag.
- [BRH07] Howard Barringer, David Rydeheard, and Klaus Havelund. *Rule Systems for Run-Time Monitoring: From Eagle to RuleR*, pages 111–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [BRI] BRICS. Best practice in robotics. EU FP7 Project. <http://www.best-of-robotics.org>.
- [BS09] D. Brugali and P. Scandurra. Component-Based Robotic Engineering (Part I) [Tutorial]. *IEEE Robotics Automation Magazine*, 16(4):84–96, December 2009.
- [BS10] D. Brugali and Azamat Shakhimardanov. Component-Based Robotic Engineering (Part II). *IEEE Robotics Automation Magazine*, 17(1):100–112, March 2010.
- [CMG05] Toby H.J. Collett, Bruce A. MacDonald, and Brian P. Gerkey. Player 2.0: Toward a Practical Robot Programming Framework. In *Proc. of the Australasian Conf. on Robotics and Automation (ACRA)*, Sydney, Australia, 2005.
- [CP05] Henrik I Christensen and Elena Pacchierotti. Embodied social interaction for robots. *AISB-05*, pages 40–45, 2005.

- [CR07] Feng Chen and Grigore Roşu. Mop: An efficient and generic runtime verification framework. In *Proceedings of the 22Nd Annual ACM SIGPLAN Conference on Object-oriented Programming Systems and Applications, OOPSLA '07*, pages 569–588, New York, NY, USA, 2007. ACM.
- [DKFW10] Klaus Dräger, Andrey Kupriyanov, Bernd Finkbeiner, and Heike Wehrheim. *SLAB: A Certifying Model Checker for Infinite-State Concurrent Systems*, pages 271–274. Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
- [DWW<sup>+</sup>06] Kerstin Dautenhahn, Michael Walters, Sarah Woods, Kheng Lee Koay, Chrystopher L Nehaniv, A Sisbot, Rachid Alami, and Thierry Siméon. How may i serve you?: a robot companion approaching a seated person in a helping context. In *Proceedings of the 1st ACM SIGCHI/SIGART conference on Human-robot interaction*, pages 172–179. ACM, 2006.
- [euR15] euRobotics aisbl. Robotics 2020 Multi-Annual Roadmap, Dec 2015.
- [FIO] FIONA - Framework for Indoor and Outdoor Navigation Assistance. Website. <http://www.fiona-project.eu/>.
- [FL03] Maria Fox and Derek Long. PDDL2. 1: An Extension to PDDL for Expressing Temporal Planning Domains. *Journal of Artificial Intelligence Research*, 20:61–124, 2003.
- [FLC<sup>+</sup>03] WK Fung, YY Leung, MK Chow, YH Liu, Y Xu, W Chan, TW Law, SK Tso, and CY Wang. Development of a hospital service robot for transporting task. In *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on*, volume 1, pages 628–633. IEEE, 2003.
- [FND02] T. Fong, I. Nourbakhsh, and K. Dautenhahn. A survey of socially interactive robots: Concepts, design, and applications. Technical report, CMU-RI-TR-02-29, 2002.
- [FSS] Sandra Frank, Dennis Stampfer, and Christian Schlegel. SmartMDSD Toolchain User Manual. <http://servicerobotik-ulm.de/toolchain-manual/html/>.
- [GB04] K. G. Larsen G. Behrmann, A. David. *A Tutorial on Uppaal*, pages 200–236. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
- [GBWK09] Michael Geisinger, Simon Barner, Martin Wojtczyk, and Alois Knoll. *A Software Architecture for Model-Based Programming of Robot Systems*, pages 135–146. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.

- [HMSNR<sup>+</sup>15] Robert Heim, Pedram Mir Seyed Nazari, Jan Oliver Ringert, Bernhard Rumpe, and Andreas Wortmann. Modeling Robot and World Interfaces for Reusable Tasks. In *Intelligent Robots and Systems Conference (IROS'15)*, pages 1793–1798. IEEE, 2015.
- [Hof03] Jörg Hoffmann. The metric-FF Planning System: Translating “Ignoring Delete Lists” to Numeric State Variables. *J. Artif. Int. Res.*, 20(1):291–341, December 2003.
- [Hol97] Gerard J. Holzmann. The model checker spin. *IEEE Trans. Softw. Eng.*, 23(5):279–295, May 1997.
- [JLJ<sup>+</sup>16] Choulsoo Janga, Seung-Ik Lee, Seung-Woog Jung, Byoungyoul Song, Rockwon Kim, Sunghoon Kim, and Cheol-Hoon Lee. OPRoS: A New Component-Based Robot Software Platform. *ETRI Journal*, 32(5):646–656, Oct. 2016.
- [JSO] JSON. JavaScript Object Notation. <http://json.org/>.
- [KRV08] Holger Krahn, Bernhard Rumpe, and Steven Völkel. Monticore: Modular Development of Textual Domain Specific Languages. In *Conference on Objects, Models, Components, Patterns (TOOLS-Europe'08)*, volume 11 of *LNBIP*, pages 297–315. Springer, 2008.
- [KRV10] Holger Krahn, Bernhard Rumpe, and Steven Völkel. Monticore: a framework for compositional development of domain specific languages. In *International Journal on Software Tools for Technology Transfer (STTT)*, volume 12, pages 353 – 372, 2010.
- [KSF09] Rachel Kirby, Reid Simmons, and Jodi Forlizzi. Companion: A constraint-optimizing method for person-acceptable navigation. In *RO-MAN 2009-The 18th IEEE International Symposium on Robot and Human Interactive Communication*, pages 607–612. IEEE, 2009.
- [Lau15] Mathis Lauckner. *Human-Robot Spatial Interaction in a Hallway*. PhD thesis, Technische Universität Berlin, 2015.
- [LBR06] Gary T. Leavens, Albert L. Baker, and Clyde Ruby. Preliminary design of jml: A behavioral interface specification language for java. *SIGSOFT Softw. Eng. Notes*, 31(3):1–38, May 2006.
- [LIRS<sup>+</sup>14] Alex Lotz, Juan F. Inglés-Romero, Dennis Stampfer, Matthias Lutz, Cristina Vicente-Chicote, and Christian Schlegel. Towards a Stepwise Variability Management Process for Complex Systems: A Robotics Perspective. *International Journal of Information System Modeling and Design (IJISMD)*, 5(3):55–74, July 2014.

- [LSLS14] Matthias Lutz, Dennis Stampfer, Alex Lotz, and Christian Schlegel. Service Robot Control Architectures for Flexible and Robust Real-World Task Execution: Best Practices and Patterns. In *Informatik 2014, Workshop Roboter-Kontrollarchitekturen*. Stuttgart, Germany, Springer LNI der GI, September 2014. ISBN 978-3-88579-626-8.
- [LV01] Flavio Lerda and Willem Visser. Addressing dynamic issues of program model checking. In *Proceedings of the 8th International SPIN Workshop on Model Checking of Software*, SPIN '01, pages 80–102, New York, NY, USA, 2001. Springer-Verlag New York, Inc.
- [Mey92] Bertrand Meyer. Applying “Design by Contract”. *Computer*, 25(10):40–51, October 1992.
- [MF08] B. Mutlu and J. Forlizzi. Robots in Organizations: Workflow, Social, and Environmental Factors in Human-Robot Interaction. In *HRI*, 2008.
- [MFN06] Giorgio Metta, Paul Fitzpatrick, and Lorenzo Natale. YARP: Yet Another Robot Platform. *International Journal on Advanced Robotics Systems*, 2006.
- [MGH<sup>+</sup>98] D. McDermott, M. Ghallab, A. Howe, C. Knoblock, A. Ram, M. Veloso, D. Weld, and D. Wilkins. PDDL-The Planning Domain Definition Language. Tech report cvc tr-98-003/dcs tr-1165, Yale Center for Computational Vision and Control, 1998.
- [Mic] Microsoft Robotics Studio. Project website. <http://www.microsoft.com/robotics>.
- [Opea] Open Robot Control Software. The Orocos Project Website. <http://www.orocos.org>.
- [Opeb] OpenRTM. OpenRTM-aist official website. <http://www.orocos.org>.
- [QCG<sup>+</sup>09] Morgan Quigley, Ken Conley, Brian P. Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
- [RGV08] Sven Roehr, Peter Gulden, and Martin Vossiek. Precise distance and velocity measurement for real time locating using a frequency modulated continuous wave secondary radar approach. *IEEE Transactions on Microwave Theory and Techniques*, vol. 56:2329–2339, October 2008.
- [RRRW15] Jan Oliver Ringert, Alexander Roth, Bernhard Rumpe, and Andreas Wortmann. Language and Code Generator Composition for Model-Driven Engineering of Robotics Component & Connector Systems. *Journal of Software Engineering for Robotics (JOSER)*, 6(1):33–57, 2015.

- [RSRV15] Christoph Reustle, Denys Shmakov, Sven Röhr, and Martin Vossiek. Precise and robust crane boom tip localization using a 24 ghz radar tachymeter. In *2015 German Microwave Conference*, pages 190–193. IEEE, 2015.
- [RTC08] RTC, OMG. OMG RTC: Robotics Technology Component (RTC), 2008. <http://www.omg.org/spec/RTC>.
- [Rum11] Bernhard Rumpe. *Modellierung mit UML*. Xpert.press. Springer Berlin, 2nd edition, September 2011.
- [RW08] Silvia Richter and Matthias Westphal. The LAMA planner. Using landmark counting in heuristic search. In *Procs. IPC-6*, 2008.
- [Sch04] Christian Schlegel. *Navigation and Execution for Mobile Robots in Dynamic Environments: An Integrated Approach*. Dissertation, University of Ulm, 2004.
- [Sch06] Christian Schlegel. Communication Patterns as Key Towards Component-Based Robotics. *Int. Journal of Advanced Robotic Systems*, pages 49–54, 2006.
- [SDW<sup>+</sup>06] Dag Sverre Syrdal, Kerstin Dautenhahn, Sarah Woods, Michael L Walters, and Kheng Lee Koay. 'doing the right thing wrong'-personality and tolerance to uncomfortable robot approaches. In *ROMAN 2006-The 15th IEEE International Symposium on Robot and Human Interactive Communication*, pages 183–188. IEEE, 2006.
- [Ser] Service Robotics Ulm. Website. <http://www.servicerobotik-ulm.de>.
- [SHLS09] C. Schlegel, T. Hassler, A. Lotz, and A. Steck. Robotic software systems: From code-driven to model-driven designs. In *Proc. of the International Conference on Advanced Robotics*, pages 1–8, June 2009.
- [SLLS16] Dennis Stampfer, Alex Lotz, Matthias Lutz, and Christian Schlegel. The SmartMDSD Toolchain: An Integrated MDSD Workflow and Integrated Development Environment (IDE) for Robotics Software. In *Journal of Software Engineering for Robotics (JOSER)*, Special Issue on Domain-Specific Languages and Models in Robotics, 2016.
- [SSL12a] Christian Schlegel, Andreas Steck, and Alex Lotz. Model-Driven Software Development in Robotics: Communication Patterns as Key for a Robotics Component Model. In Daisuke Chugo and Sho Yokota, editors, *Introduction to Modern Robotics*. iConcept Press, 2012. ISBN:978-0980733068.
- [SSL12b] Christian Schlegel, Andreas Steck, and Alex Lotz. Robotic Software Systems: From Code-Driven to Model-Driven Software Development. In Ashish Dutta, editor, *Robotic Systems - Applications, Control and Programming*. InTech, 2012. ISBN:978-953-307-941-7.

- [TFB97] D Fox W Burgard S Thrun, D Fox, and W Burgard. The dynamic window approach to collision avoidance. *IEEE Transactions on Robotics and Automation*, 4:1, 1997.
- [Vea96] Martin Vossiek and et al. Novel fmcw radar system concept with adaptive compensation of phase errors. In *26th European Microwave Conference, EuMC*, pages 135–139, Prague, Czech Republic, October 1996.
- [VHB<sup>+</sup>03] Willem Visser, Klaus Havelund, Guillaume Brat, Seungjoon Park, and Flavio Lerda. Model checking programs. *Journal of Automated Software Engineering*, 10(2):203–232, April 2003.
- [Wal08] M.L. Walters. *The Design Space for Robot Appearance and Behavior for Social Robot Companions*. PhD thesis, University of Herfortshire, 2008.
- [War96] T. Warnecke. J. Bengtsson, K. Larsen, F. Larsson, P. Pettersson and W. Yi. Uppaal in 1995. In *Tools and Algorithms for the Construction and Analysis of Systems*, pages 431–434, November 1996.
- [WDK<sup>+</sup>07] Sarah Woods, Kerstin Dautenhahn, Christina Kaouri, René te Boekhorst, Kheng Lee Koay, and Michael L Walters. Are robots like people?: Relationships between participant and robot personality traits in human–robot interaction studies. *Interaction Studies*, 8(2):281–305, 2007.
- [You] Youtube Channel: Robotics@HS-Ulm.  
<http://www.youtube.com/roboticsathsum>.

## Berichtsblatt

1. ISBN oder ISSN geplant	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel iserveU - Intelligente modulare Serviceroboter-Funktionalitäten im menschlichen Umfeld am Beispiel von Krankenhäusern	
4. Autor(en) [Name(n), Vorname(n)] 1. Gindele, Tobias 2. Rumpe, Bernhard 3. Rausch, Andreas 4. Vossiek, Martin 5. Gulden, Peter 6. Schlegel, Christian 7. Verbeek, Christian	5. Abschlussdatum des Vorhabens 29.02.2016
	6. Veröffentlichungsdatum
	7. Form der Publikation
8. Durchführende Institution(en) (Name, Adresse) 1. Robert Bosch GmbH, Robert Bosch Campus 1, 71272 Renningen 2. RWTH Aachen, Lehrstuhl für Software Engineering, Ahornstraße 55, 52074 Aachen 3. TU Clausthal, Institut für Informatik - Software Systems Engineering Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld 4. FAU Erlangen-Nürnberg, Lehrstuhl für Hochfrequenztechnik, Cauerstr. 9, 91058 Erlangen 5. Symeo GmbH, Professor-Messerschmitt-Str. 3, 85579 Neubiberg / München 6. Hochschule Ulm, Institut für Angewandte Forschung, Prittwitzstraße 10, 89075 Ulm 7. REC GmbH, Robert-Koch-Str. 2, 82152 Planegg	9. Ber. Nr. Durchführende Institution
	10. Förderkennzeichen 01IM12008A
	11. Seitenzahl 88
	12. Fördernde Institution (Name, Adresse)  Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn
13. Literaturangaben 63	
14. Tabellen 0	
15. Abbildungen 45	
16. Zusätzliche Angaben	
17. Vorgelegt bei (Titel, Ort, Datum)	
18. Kurzfassung  Aktuell finden sich kaum Service-Roboter-Produkte in menschlichen Umgebungen aufgrund ihrer Komplexität. Existierende Systeme zeigen bisher wenig intelligentes Verhalten. Ziel des Projekts iserveU war die Entwicklung eines prototypischen intelligenten Transportsystems (ITA) und der dafür notwendigen Technologien für die Ausführung von Logistikaufgaben in Krankenhäusern. Im Rahmen des Projekts wurden die folgenden drei Aufgaben bzw. Szenarien bearbeitet und näher untersucht: 1.) geplante repetitive Transportaufgaben, 2.) ungeplante (ad-hoc) Transportaufgaben, 3.) Begleitdienste und kooperative Transportaufgaben. Die Forschung und Entwicklung adressierte dabei eine neue 3D-Ultraschallarray-Technologie für die Hinderniserkennung und Navigation, eine Radar-basierte Lokalisierung, Missions- und Bewegungsplanung für die Navigation im menschlichen Umfeld, Mensch-Maschine-Interaktion, Roboterplattformaufbau, modellgetriebene Entwicklung für Software und domänenspezifische Beschreibungssprachen sowie Absicherung durch Remote-Management. Der entwickelte Prototyp konnte erfolgreich im Rahmen einer Evaluation im Katharinenhospital in Stuttgart seine Fähigkeiten demonstrieren. Die im Projekt entwickelte Technologie stellt einen wichtigen Schritt dar, um Service-Robotern den Weg zu ebnen in Umgebungen, die von Menschen geprägt sind.	
19. Schlagwörter Service Robotik, Transportassistentz, menschliches Umfeld, Navigation, modellgetriebene Entwicklung	
20. Verlag	21. Preis

## Document Control Sheet

1. ISBN or ISSN planned	2. type of document (e.g. report, publication) report
3. title iserveU – intelligent modular service robot functionalities in human environments exemplified at hospitals	
4. author(s) (family name, first name(s)) <ol style="list-style-type: none"> <li>1. Gindele, Tobias</li> <li>2. Rumpe, Bernhard</li> <li>3. Rausch, Andreas</li> <li>4. Vossiek, Martin</li> <li>5. Gulden, Peter</li> <li>6. Schlegel, Christian</li> <li>7. Verbeek, Christian</li> </ol>	5. end of project 02/29/2016
	6. publication date
	7. form of publication
8. performing organization(s) (name, address) <ol style="list-style-type: none"> <li>1. Robert Bosch GmbH, Robert Bosch Campus 1, 71272 Renningen</li> <li>2. RWTH Aachen, Lehrstuhl für Software Engineering, Ahornstraße 55, 52074 Aachen</li> <li>3. TU Clausthal, Institut für Informatik - Software Systems Engineering Julius-Albert-Str. 4, 38678 Clausthal-Zellerfeld</li> <li>4. FAU Erlangen-Nürnberg, Lehrstuhl für Hochfrequenztechnik, Cauerstr. 9, 91058 Erlangen</li> <li>5. Symeo GmbH, Professor-Messerschmitt-Str. 3, 85579 Neubiberg / München</li> <li>6. Hochschule Ulm, Institut für Angewandte Forschung, Prittwitzstraße 10, 89075 Ulm</li> <li>7. REC GmbH, Robert-Koch-Str. 2, 82152 Planegg</li> </ol>	9. originator's report no.
	10. reference no. 01IM12008A
	11. no. of pages 88
12. sponsoring agency (name, address)  Bundesministerium für Bildung und Forschung (BMBF) 53170 Bonn	13. no. of references 63
	14. no. of tables 0
	15. no. of figures 45
16. supplementary notes	
17. presented at (title, place, date)	
18. abstract  <p>Today, service robots are barely present in human environments due to its complexity. Existing systems do not exhibit very intelligent behavior so far. Goal of the project iserveU was the development of a prototypical intelligent transport assistant (ITA) and the required technologies for the execution of logistic tasks in hospitals. During the project, three tasks respectively scenarios were addressed and investigated closely: 1.) Planned repetitive transportation tasks, 2.) Unplanned (ad-hoc) transportation tasks, 3.) Accompanying services and cooperative transportation tasks. The research and development addressed thereby a new 3d ultrasonic array sensor for obstacle detection and navigation, a radar-based localization, mission and motion planning for navigation in human environments, human robot interaction, robot platform construction, model-based development of software and domain specific languages as well as verification and safeguarding through remote management. The developed prototype successfully demonstrated its abilities in course of an evaluation in the Katharinenhospital in Stuttgart. The technology developed in this project makes an important step towards service robots in human environments.</p>	
19. keywords Service robotics, transportation assistance, human environments, navigation, model-based development	

20. publisher

21. price