



**Hochschule  
Bonn-Rhein-Sieg**  
University of Applied Sciences

Fachbereich  
Informatik

## **Abschlussarbeit**

im Studiengang Master Informatik

# **Automatisierte Erkennung von ungewöhnlichem Verhalten im Privathaushalt und Reaktion in Notsituationen**

von

**Peter Manheller**  
(9016802)

Erstprüfer:  
Zweitprüfer:

Prof. Dr. Karl Jonas  
Prof. Dr. Sascha Alda

Zeitraum:  
Eingereicht am:

14.09.2015 - 11.05.2016  
04.05.2016

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Problemstellung	1
1.2	Fragestellung	2
1.3	Zielsetzung	3
1.4	Vorgehensweise	3
<b>2</b>	<b>Stand der Forschung</b>	<b>5</b>
2.1	Zielgruppe: Generation 65+	6
2.2	Arten und Modellierung der ADL	9
2.3	Sensor-, Kommunikations- und Messdatentypen	17
2.3.1	Restriktion: Schutz der Privatsphäre	21
2.3.2	Resultierende Grenzen für die Entwicklung	22
2.4	Ambient Assisted Living - Systeme	23
2.4.1	Varianten von AAL-Systemen	24
2.4.2	Kategorien von Algorithmen für AAL-Systeme	26
2.4.3	Existente AAL-Systeme	27
2.4.4	Vorgehensweise der Systeme und Fehlerquellen in den ADL	42
2.5	Referenz in Themengebiete der KI	46
2.5.1	Grundgedanken des maschinellen Lernens	47
2.5.2	Einführung in das Agentenmodell	49
2.6	Von der Theorie zur Praxis	52
<b>3</b>	<b>Automatisierte Verhaltensanalyse</b>	<b>57</b>
3.1	Konzept der Datenanalyse	57
3.2	Anforderungsanalyse und Entwurf des Systems	64
3.3	Eingesetzte Hard- und Software	71
3.4	Installation, Konfiguration und Aufbau	73
3.5	Realisierung des Systems	78
3.6	Funktionstests und Evaluation	91
<b>4</b>	<b>Ergebnisse und Ergänzungen</b>	<b>94</b>
4.1	Ergebnisse	94
4.2	Ergänzende Ansätze	99
<b>5</b>	<b>Zusammenfassung und Fazit</b>	<b>102</b>
<b>6</b>	<b>Anhang</b>	<b>104</b>
6.1	Tabellarischer Vergleich ausgewählter AAL-Systeme	104
6.2	Tabellarischer Vergleich ausgewählter Bibliotheken	107

---

6.3	Java-Quellcode für k-Pattern-Ansatz . . . . .	108
6.4	Visualisierungen der Anforderungen FA, QA, RB . . . . .	110
6.5	Auszüge der openHAB-Konfiguration . . . . .	115
6.6	Eventstreams, Validierung und Evaluierung . . . . .	116
6.7	SHLM: Klassendiagramme . . . . .	123
	<b>Literaturverzeichnis</b>	<b>125</b>
	<b>Digitaler Anhang</b>	<b>141</b>

# Tabellenverzeichnis

3.1	Ausgewählte Konzepte der Messdatenverarbeitung . . . . .	58
3.2	Ausschnitt der gemessenen Daten . . . . .	59
3.3	Mögliche Abbildung von Sensorevents auf Zahlen . . . . .	62
3.4	Steckbrief der Persona "Hildegard" . . . . .	66
3.5	Steckbrief der Persona "Alexandra" . . . . .	68
4.1	Anzahl explorierter Zeitfenster $t_{\Delta}$ : Zweiwöchiger Lernzeitraum ( $(iii) t_{diff}, d$ ) . . . . .	95
4.2	Anzahl explorierter Zeitfenster $t_{\Delta}$ : Vierwöchiger Lernzeitraum ( $(iii) t_{diff}, d$ ) . . . . .	95
4.3	Genauigkeitstabelle für Compact Prediction Tree Plus (CPT+): zweiwöchiger Lernzeitraum - Validierung (a) . . . . .	96
4.4	Genauigkeitstabelle für CPT+: zweiwöchiger Lernzeitraum - Evaluierung (b) . . . . .	97
4.5	Genauigkeitstabelle für All-k-Order Markov Chains (AKOM): zweiwöchiger Lernzeitraum - Validierung (a) . . . . .	98
6.1	Vergleich ausgewählter Projekte: Kriterien für Hard- und Softwarekomponenten . . . . .	105
6.2	Vergleich ausgewählter Projekte: Kriterien für Anforderungen der Personas . . . . .	106
6.3	Vergleich ausgewählter Software-Bibliotheken: "Data-Mining" . . . . .	107
6.4	Genauigkeitstabelle für AKOM: zweiwöchiger Lernzeitraum - Evaluierung (b) . . . . .	120
6.5	Genauigkeitstabelle für AKOM: vierwöchiger Lernzeitraum - Validierung (a) . . . . .	121
6.6	Genauigkeitstabelle für AKOM: vierwöchiger Lernzeitraum - Evaluierung (b) . . . . .	122

# Abbildungsverzeichnis

2.1	Kategorien von Eigenschaften der Aktivitäten [i.A.a. Mun15, S.177]	11
2.2	Tagesablauf (Montag) für Persona Hildegard - Ausschnitt	12
2.3	Kategorien von Context-Aware Applications [i.A.a. May04, S.11 f.]	14
2.4	Teilauswahl von Sensoren für O-C-M-Typisierung [i.A.a. Mun15, S.130; Ras+13, S.580]	18
2.5	Ausgewählte Kommunikationsstandards für Sensoren [Ele+15, i.A.a. Mun15, S.173; Ras+13, S.580; al15; AG15]	19
2.6	Messdatentypen für vorgestellte Sensortypen [i.A.a Ras+13, S.580]	20
2.7	Kategorien der AAL-Systeme [i.A.a. Ras+13, S.580 f.]	24
2.8	Algorithmen der AAL-Systeme [i.A.a. Ras+13, S.581 ff.]	26
2.9	Gesamtsystem - SAMDY [Mun15, S.145]	29
2.10	Wohnungsumgebung und Auswertungseinheit - safe@home [AAL15d]	31
2.11	CASAS: Smart Home Komponenten und Architektur [Coo+13, S.63]	34
2.12	Genutzte Techniken im MavHome-System [i.A.a. Coo15, S.6]	37
2.13	Architektur für das Aufspüren von Aktivitäten mit dem $k$ -pattern Algorithmus [i.A.a. Bou+15, S.11959]	39
2.14	Eruierte Grundarchitektur und Vorgehensweise für AAL-Systeme (Smart Home)	43
2.15	Eruierte Teilkomponente "(vor) Inbetriebnahme" der Grundarchitektur [i.A.a May04, S.11 ff.]	44
2.16	Eruierte Teilkomponente "Systemnutzung" der Grundarchitektur	45
2.17	Hard- und Software-Agent mit Benutzerinteraktion [i.A.a Rus+12, S.61; Ert09, S.13]	49
2.18	PEAS-Beschreibung der Aufgabenumgebung für ein AAL-System (Smart Home) [i.A.a Rus+12, S.67]	50
2.19	Allgemeiner lernender Agent [Rus+12, S.83]	52
3.1	Exempl. Matrix für Event $E_1$ ( $M01=1$ )	60
3.2	Lebenszyklus des AAL-Systems	63
3.3	Konzept der Datenanalyse	64
3.4	System- und Kontextgrenze des AAL-Systems (Proof-of-Concept) [i.A.a. Poh+15, S.15 ff.]	65
3.5	Use-Case-Diagramm: FA1	68
3.6	UML-Sequenzdiagramm - Kommunikation zwischen Personas, openHAB und SHLM	70
3.7	UML-Verteilungsdiagramm - Verteilungssicht des AAL-Systems	72
3.8	Hardwareaufbau im "Multimedia Communication Laboratory" H-BRS Raum C060	73
3.9	Eventstream für Bewegungsmelder "M02" (Freitags 11-12Uhr)	75
3.10	AAL-System im Smart Home - Lokalisiert in einer Stadt in Rheinland Pfalz	76

---

3.11	Exemplarische Anbringung der HomeMatic Bewegungsmelder und Magnetkontakte . . . . .	77
3.12	Absolute Häufigkeit der erkannten Bewegung für Bewegungsmelder 'M01' (tagesspezifisch, einstündiges Zeitfenster) . . . . .	78
3.13	UML-Komponentendiagramm - Bausteinsicht des AAL-Systems . . . . .	79
3.14	ERD: Datenbankschema "openhab" . . . . .	79
3.15	UML-Klassendiagramm EventStream: Data Access Object (DAO) für Java-Klasse "ItemData" . . . . .	83
3.16	Übersicht der openHAB-GUI für das Smart Home Learning Module (SHLM) . . . . .	86
3.17	openHAB-GUI für realisierte Sitemap (Auszug der Sensoren) . . . . .	87
3.18	UML-Sequenzdiagramm - Lernphase: Das SHLM lernt. . . . .	88
6.1	Use-Case-Diagramm: FA2+FA7 . . . . .	110
6.2	Use-Case-Diagramm: FA3-FA4 . . . . .	110
6.3	Use-Case-Diagramm: FA5-FA6 . . . . .	111
6.4	Use-Case-Diagramm: FA8 . . . . .	111
6.5	Use-Case-Diagramm: FA9-FA10 . . . . .	112
6.6	Use-Case-Diagramm: FA11-FA12 . . . . .	112
6.7	Use-Case-Diagramm: FA13 . . . . .	113
6.8	Use-Case-Diagramm: FA14 . . . . .	113
6.9	Use-Case-Diagramm: QA1 . . . . .	114
6.10	Eventstream: Bewegungsmelder 'M01' Wohnzimmer - Sonntags: 4 Wochen (1. Teil) . . . . .	117
6.11	Eventstream: Bewegungsmelder 'M01' Wohnzimmer - Sonntags: 4 Wochen (2. Teil) . . . . .	118
6.12	Ausschnitt Validierung der Woche "23-29.01-2016" - vierwöchiger Lernzeitraum ( $t_{diff} = 20$ ) . . . . .	119
6.13	UML-Klassendiagramm des SHLM (alle Klassen - 1. Teil) . . . . .	123
6.14	UML-Klassendiagramm des SHLM (alle Klassen - 2. Teil) . . . . .	124

# Quelltextverzeichnis

3.1	Maven-Konfiguration "pom.xml" (Ausschnitt) . . . . .	80
3.2	Hibernate-Konfiguration "hibernate.cfg.xml" (Ausschnitt) . . . . .	81
3.3	Ausschnitt openHAB-Rules-Artefakt: "Learn Behavior"-Regel . . . . .	88
6.1	Weka: k-Pattern-Ansatz - Vorverarbeitung und FP-growth (Ausschnitt) . . . . .	108
6.2	Auszug der openHAB-Config: MySQL- und HomeMatic-Parameter . . . . .	115
6.3	Auszug der openHAB-Sitemap . . . . .	115
6.4	Auszug der openHAB-Items . . . . .	115
6.5	Auszug der openHAB-MySQL-Persistence . . . . .	116

# Abkürzungsverzeichnis

<b>AAL</b>	Ambient Assisted Living .....	1
<b>AGALZ</b>	Autonomous aGent for monitoring ALzheimer patients .....	39
<b>AGPL</b>	GNU Affero General Public License .....	53
<b>AKOM</b>	All-k-Order Markov Chains .....	IV
<b>AL</b>	Aktivitäten des Lebens .....	10
<b>ALZ</b>	Active LeZi .....	37
<b>ANNs</b>	Artificial Neural Networks .....	16
<b>ANN</b>	Artificial Neural Network .....	38
<b>ARFF</b>	Attribute-Relation File Format .....	57
<b>ATR</b>	Allen's Temporal Relations .....	38
<b>ADL</b>	Activit{y ies} of Daily Living .....	3
<b>AEDL</b>	Aktivitäten und existentiellen Erfahrungen des Lebens .....	27
<b>AHCS</b>	Ambient [Intelligence based] Home Care Systems .....	5
<b>Aml</b>	Ambient Intelligence .....	5
<b>API</b>	Application Programming Interface .....	53
<b>AR</b>	Activity Recognition .....	26
<b>ATL</b>	Aktivitäten des täglichen Lebens .....	10
<b>BDSG</b>	Bundesdatenschutzgesetz .....	22
<b>BI</b>	Business Intelligence .....	53
<b>BLOB</b>	Binary Large Object .....	84
<b>BMBF</b>	Bundesministerium für Bildung und Forschung .....	27
<b>BRD</b>	Bundesrepublik Deutschland .....	21
<b>CORBA</b>	Common Object Request Broker Architecture .....	36
<b>CRFs</b>	Conditional Random Fields .....	35
<b>CSV</b>	Comma-seperated values .....	53
<b>CPT+</b>	Compact Prediction Tree Plus .....	IV
<b>CTT</b>	Concur Task Tree - Metamodell zu Beschreibung von Aktivitäten ..	16
<b>DAO</b>	Data Access Object .....	VI
<b>DCJ</b>	Daily Care Journal .....	27
<b>Destatis</b>	Statistisches Bundesamt .....	1
<b>DECT</b>	Digital Enhanced Cordless Telecommunications .....	56
<b>DG</b>	Dependency Graph .....	62



<b>DHCP</b>	Dynamic Host Configuration Protocol .....	74
<b>DT</b>	Decision Tree .....	38
<b>DTs</b>	Decision Trees .....	16
<b>ED</b>	Episode Discovery .....	36
<b>ELKI</b>	Environment for DeveLoping KDD-Applications Supported by Index-Structures .....	55
<b>EM</b>	Expectation-Maximization .....	38
<b>ERD</b>	Entity-Relationship-Diagramm .....	80
<b>FNs</b>	False Negatives .....	2
<b>FN</b>	False Negative .....	46
<b>FPs</b>	False Positives .....	2
<b>FP</b>	False Positive .....	45
<b>FSC</b>	Frequent Subsequence Compression .....	97
<b>FSM</b>	Finite State Machine .....	37
<b>GUI</b>	Graphical User Interface .....	53
<b>GPL</b>	GNU Public License .....	53
<b>GPS</b>	Global Positioning System .....	25
<b>HAP</b>	Human Activity Pattern .....	26
<b>HIS</b>	Habitat Intelligent pour la Sante .....	37
<b>HMM</b>	Hidden Markov Modell .....	16
<b>IDE</b>	Integrated Development Environment .....	54
<b>ILSA</b>	Independent LifeStyle Assistant .....	40
<b>ISP</b>	Internet Service Provider .....	71
<b>JADE</b>	Java Agent Development Framework .....	40
<b>JDBC</b>	Java Database Connectivity .....	54
<b>JDK</b>	Java Development Kit .....	53
<b>JPA</b>	Java Persistence API .....	55
<b>JRE</b>	Java Runtime Environment .....	53
<b>JRI</b>	Java/R Interface .....	55
<b>JUTTA</b>	JUsT-in-Time Assistance .....	30
<b>JVM</b>	Java Virtual Machine .....	80
<b>KEEL</b>	Knowledge Extraction based on Evolutionary Learning .....	55
<b>KI</b>	Künstliche Intelligenz .....	47
<b>KNX-RF</b>	Konnex-Bus Radio-Frequency .....	20
<b>LA</b>	Lebensaktivitäten .....	10
<b>LAN</b>	Local Area Network .....	20
<b>LE</b>	Learning Environment .....	53
<b>LOS</b>	Line-Of-Sight .....	19
<b>MDP</b>	Markov Decision Prozess .....	37

<b>MavHome</b>	Managing an adaptive versatile Home .....	35
<b>MCLab</b>	Multimedia Communication Laboratory .....	73
<b>MCMC</b>	Markov Chain Monte-Carlo .....	40
<b>MDL</b>	Minimum Description Length .....	36
<b>MXML</b>	Mining eXtensible Markup Language .....	29
<b>NB</b>	Naive Bayes .....	38
<b>NUC</b>	Next Unit of Computing .....	71
<b>ODS</b>	Open Distributed System .....	32
<b>openHAB</b>	open Home Automation Bus .....	2
<b>ORM</b>	Object Relational Mapping .....	71
<b>OS</b>	Operating System .....	36
<b>OSGi-Framework</b>	Softwareplattform mit Komponentenmodell auf Basis von Java ....	32
<b>PAP</b>	Programmablaufplan .....	30
<b>PASW</b>	Predictive Analysis SoftWare - ehem. IBM SPSS Statistics .....	55
<b>PBX</b>	Private Branch Exchange .....	56
<b>PEAS</b>	Performance, Environment, Acutators, Sensors .....	50
<b>PERSONA</b>	PERceptive Spaces pr0moting iNdependent Aging .....	31
<b>PN</b>	Petri Netze .....	16
<b>PNR</b>	Prediction with improved Noise Reduction .....	97
<b>ProM</b>	Process Mining Framework .....	29
<b>POJOs</b>	Plain Old Java Objects .....	71
<b>PPM</b>	First order Markov Chains .....	62
<b>PSG</b>	Polysomnographie .....	33
<b>QoS</b>	Quality of Service .....	2
<b>RDF</b>	Resource Description Framework .....	32
<b>REST</b>	Representational State Transfer .....	86
<b>RFID</b>	Radio-Frequency Identification .....	21
<b>RTL-Modell</b>	Roper-Logan-Tierney-Modell .....	10
<b>RWD</b>	Responsive Webdesign .....	72
<b>SAMDY</b>	Sensorbasiertes adaptives Monitoringsystem für die Verhaltensanalyse von Senioren .....	28
<b>SAX</b>	Symbolic Aggregate approXimation .....	38
<b>SBC</b>	Simple Branches Compression .....	97
<b>SGB</b>	Sozialgesetzbuch .....	11
<b>SHiB</b>	Smart Home in a Box .....	34
<b>SHLM</b>	Smart Home Learning Module .....	VI
<b>SMILEY</b>	SMart and Independent Living for the ElderLY .....	32
<b>SOM</b>	Self-Organizing Map .....	38
<b>SPMF</b>	Sequential Pattern Mining Framework .....	55

---

<b>SQL</b>	Structured Query Language .....	53
<b>SVM</b>	Support Vector Machine .....	35
<b>TDAG</b>	Transition Directed Acyclic Graph .....	62
<b>TSM</b>	Transition System Miner .....	29
<b>UCD</b>	User Centered Design .....	32
<b>UML</b>	Unified Modeling Language .....	67
<b>UPnP</b>	Universal Plug and Play .....	32
<b>UUID</b>	Universally Unique Identifier .....	80
<b>UX</b>	User Experience .....	5
<b>VSP</b>	Virutal Support Partner .....	41
<b>WAN</b>	Wide Area Network .....	31
<b>WAP</b>	Wireless Access Point .....	28
<b>WAPs</b>	Wireless Access Points .....	20
<b>WBAN</b>	Wireless Body Area Network .....	20
<b>WPAN</b>	Wireless Personal Area Network .....	20
<b>WEKA</b>	Waikato Environment for Knowledge Analysis .....	54
<b>XMPP</b>	Extensible Messaging and Presence Protocol .....	34
<b>XML</b>	Extensible Markup Language .....	38
<b>XP</b>	Extreme Programming .....	91
<b>XSD</b>	XML Schema Definition .....	80
<b>ZeroConf</b>	Zero Configuration Networking .....	36

# 1 Einleitung

## 1.1 Problemstellung

Aus den vom Destatis (Statistisches Bundesamt) im Jahr 2015 veröffentlichten demografischen Grunddaten geht hervor, dass im Jahr 2014 exakt 20,8% der in Deutschland lebenden Menschen zur Generation 65+ gehören. Dieser Anteil soll bis zum Jahr 2060 auf 33% ansteigen [Bun15b, S.5f]. Unter Annahme der in [Bun15a, S.15] kalkulierten Bevölkerungszahl von 73,5 Millionen, steigt die Anzahl der Menschen der Generation 65+ von etwa 17 auf 25,5 Millionen an. Da im Alter das Risiko, auf Pflege angewiesen zu sein, steigt, ist bei der prognostizierten Bevölkerungszahl auch mit fast doppelt so vielen pflegebedürftigen Personen zu rechnen [Bun15c]. Dabei findet die Pflege überwiegend im häuslichen bzw. privaten Umfeld statt [Bun15b, S.39]. Unter Berücksichtigung der Vorlieben von pflegebedürftigen Personen, wird eine Pflege meist von Angehörigen aus dem direkten sozialen Umfeld oder von denen mit rechtlichen Verhältnis bevorzugt angenommen. Eine Dienstleistung von ambulanten Pflegediensten bzw. examinierten Pflegekräften wird auch durch die aufzubringenden Geldmittel als zweitrangig eingestuft, außerdem ist eine Pflegekraft nur ca. 4% der Wachzeit der pflegebedürftigen Person anwesend [Mun15, S.6].

Die dem Angehörigen auferlegte Bürde stellt ihn nicht nur vor physische Herausforderungen sondern vielmals auch vor psychische und emotionale Belastungen:

*Pflegende Angehörige leben im ständigen Bereitschaftsdienst. Sie erfüllen einen Fulltime-Job und sind nicht selten 24 Stunden am Tag im Einsatz. [onl15, Brandt A.]*

Der immense Leistungsdruck und die andauernde Belastung, die der Angehörige für die Pflege aufbringen muss, führen zu Einbußen seiner eigenen Lebensqualität. Um der psychischen Entbehrung der Angehörigen entgegenzuwirken und zudem das Wohlbefinden und die Gesundheitsbedingungen für die Lebenserwartung im Alter steigern zu können, ist die technische Entwicklung von Assistenzsystemen aus dem Bereich des AAL (Ambient Assisted Living) notwendig [Ras+13, S.579]: AAL-Systeme als "unsichtbare Pfleger" können angemessen auf Notsituationen reagieren und somit für die Sicherheit im Alltag assistieren [Gmb15h].

## 1.2 Fragestellung

Für die Assistenz wertet der "unsichtbare Pfleger" die Daten von einzelnen Sensoren oder Sensornetzwerken aus. Die vorherrschende Inkompatibilität der herstellerabhängigen Sensorik zwingt die Hersteller jedoch zur Entwicklung von unterschiedlichen Hard- und Softwarelösungen, um die unterschiedlichen Standards zu vereinen. In jüngster Entwicklung sieht der Trend zur Überbrückung dieser Inkompatibilität ein einzelnes Softwareprodukt auf dem Leitrechner der Managementebene der Gebäude- bzw. Heimautomatisierung vor.

Die quelloffene openHAB (open Home Automation Bus) Automatisierungssoftware implementiert bereits 160 herstellerabhängige Standards [K.15, 26. Mai 2015] und gewährleistet somit eine Kommunikation zwischen den Technologien bzw. Sensoren [Mas]. Die openHAB-Software stellt 12 Dienste zur Verfügung, um die von den Sensoren gemessenen Daten zu persistieren [Uh15b]. Dabei ist die daran anknüpfende Fragestellung inwieweit die openHAB-Software genutzt werden kann, um die der Persistenz anknüpfende Datenanalyse im AAL-System in angemessener Zeit durchzuführen, um ein ungewöhnliches Verhalten der pflegebedürftigen Personen im Privathaushalt zu erkennen und auf evtl. Notsituationen zu reagieren, derzeit ungelöst.

Für die Fragestellung sollen folgende Teilanforderungen gelöst werden:

- Die eingesetzte Hardware und die gemessenen bzw. aufgezeichneten Daten sollen den Schutz der Privatsphäre der pflegebedürftigen Personen achten.
- Die aus dem Schutz der Privatsphäre resultierenden Grenzen für die Entwicklung der AAL-Systeme sind aufzuweisen.
- Die verwendete Sensorik ist unauffällig in den Alltag einzubinden, um die Akzeptanz des Systems zu steigern.
- Neben einer initialen Konfiguration der Systemkomponenten erfordert das AAL-System keine weitere Bedienung.
- Um Abweichungen vom "Normalverhalten" der Personen zu erkennen, sind die persistierten Daten eingangs in einer Lernphase anhand etablierter Verfahren zu analysieren. Eine heuristische Abschätzung der recherchierten Verfahren ist obligatorisch.
- Für die in der Lernphase anschließende Erkennungsphase ist die QoS (Quality of Service) der eingesetzten theoretischen Verfahren und Softwareprodukte zu beurteilen: Wie aussicheres ist das AAL-System, existieren FNs (False Negatives) und können FPs (False Positives) auf ein Minimum reduziert werden? (FN: Ein ungewöhnliches Verhalten wird nicht erkannt.). Mit welcher Genauigkeit können Verhaltensprognosen gestellt werden?

## 1.3 Zielsetzung

Das Ziel der anstehenden Arbeit ist die Entwicklung eines AAL-Systems, das der automatisierten Erkennung von ungewöhnlichem Verhalten im Privathaushalt dient und in angemessener Zeit auf erkannte Abweichungen reagiert. Als Kernkomponente für die Automatisierungslogik ist die openHAB-Software einzusetzen. Für ein Proof-of-Concept ist neben der Entwicklung des AAL-Systems auch eine reale Installation vorgesehen. Die daran anschließenden Testszenarien simulieren eine Systemlaufzeit des AAL-Systems und evaluieren das gemessene Verhalten im Hinblick auf Verhaltensabweichungen und -prognosen. Das AAL-System ist in zwei Hauptkomponenten, die Lern- und Erkennungsphase, unterteilt. Für die Lernphase sind theoretische Methodiken zu erforschen, die die gemessenen Daten analysieren und aufbereiten. Für die anknüpfenden Ergebnisse der Erkennungsphase soll der Zusammenhang zwischen verwendeter Methodik der Lernphase und QoS des Gesamtsystems untersucht werden. Die QoS wird davon abhängen, ob ein ungewöhnliches Verhalten jederzeit bemerkt wird, die Anzahl der fälschlicherweise erkannten Abweichungen gering ist und die gestellten Verhaltensprognosen eine hohe Genauigkeit aufweisen.

## 1.4 Vorgehensweise

In der anstehenden Literaturrecherche ist der Stand der Forschung hinsichtlich existierender AAL-Systeme und verwendeter Verfahren und Algorithmen zur Erkennung von Abweichungen der ADL (Activities of Daily Living) zu erarbeiten. Die AAL-Systeme sollen unter den in Sektion 1.2 gelisteten Teilanforderungen kritisch betrachtet werden. Für die anschließende prototypische Entwicklung eines AAL-Systems bei Verwendung der openHAB-Software sind die aus der Recherche eruierten Algorithmen im Hinblick auf FNs, FPs und Nutzungskontext zu analysieren. Für die Nutzung des AAL-Systems ist eine zeitnahe Auswertung der von den Sensoren gemessenen Daten obligatorisch, um in angemessener Zeit auf Notsituationen reagieren zu können. Um eine hohe Allgemeingültigkeit und eine realitätsnahe Situation widerzuspiegeln, soll der entwickelte Prototyp zusammen mit einem Sensornetzwerk in einem Feldexperiment eingesetzt werden. Um die QoS des Prototyps zu messen, werden die im Feldexperiment aufgezeichneten Sensordaten für ein initiales Normalverhalten genutzt und mögliche Abweichungen der ADL sowie Verhaltensprognosen evaluiert.

Um zu zeigen, dass die in Sektion 1.3 genannten Ziele unter den in Sektion 1.2 erwähnten Fragestellungen erreicht werden, wird die Realisierung (Installation, Konfiguration und Entwicklung) des AAL-Systems durchgeführt. Im Anschluss an die Entwicklung wird für das realisierte Konzept der Datenanalyse die Systemlaufzeit simuliert. In einer abschließenden Dis-

kussion sind die ausgewerteten Ergebnisse im Zusammenhang mit verwendeten Algorithmen im Prototyp und QoS des Systems zu analysieren.

## 2 Stand der Forschung

Unter das Paradigma der AAL fallen alle kontextsensitiven Anwendungen, Technologien und Assistenzsysteme, deren Zielsetzung eine unterstützende Umsetzung der ADL befürworten. Die AAL-Systeme finden also überall dort Anwendung, wo kranke, ältere oder pflegebedürftige Personen in ihrem täglichen Leben und Aktivitäten unterstützt werden wollen bzw. müssen [Hri+08, S.1]. Kategorisiert werden die AAL-Systeme sowohl anhand der verwendeten Sensortypen, genutzten Verfahren zur Erfassung von Umgebungsparametern, Analyse und Erkennung von Abweichungen der ADL als auch Art der Reaktion bei unterstützenden Maßnahmen und angebotenen altersgerechten System-Schnittstellen [AAL15a]. Die Anwendungsfelder von Systemen der Aml (Ambient Intelligence) sind nicht, wie bei den AAL-Systemen, auf das Anwendungsfeld der Pflege bzw. Gesundheitsvorsorge im häuslichen Umfeld begrenzt, sondern betrachten z. B. auch den Arbeitsbereich und die Erkennung oder Prävention von Angriffen beim Einsatz von ausgebauten Sensornetzwerken. In diesem Zusammenhang ist der Begriff der AHCS (Ambient [Intelligence based] Home Care Systems) [vgl. Hri+08, S.1; Bec+06, S.1 f.] anzutreffen. AAL kann somit als Teilgebiet von Aml angesehen werden.

Um den Stand der Forschung von AAL-Systemen in diesem Kapitel aufzugreifen, grenzt der Abschnitt 2.1 die Nutzer der Systeme auf die Zielgruppe der Generation 65+ ein und analysiert die dem Personenkreis zuzuordnenden Arten von ADL, um anschließend eine Aussage über Kontextinformationen und mögliche Modellierungen für die jeweiligen ADL zu treffen. Der Folgeabschnitt behandelt die Typen von Sensoren die in einem AAL-System verwendet werden können und zeigt die Variationen von Formaten der gemessenen Daten und die unterschiedlichen Kommunikationsstandards auf. Zudem wird aufgeschlüsselt, inwieweit die Restriktion zur Einhaltung der Privatsphäre der überwachten Personen Auswirkungen auf die Dimensionen der Entwicklung eines AAL-Systems hat.

Der Abschnitt 2.4 zeigt die Klassifizierung der Systeme und Algorithmen der AAL. Der Forschungsschwerpunkt für bereits existierende AAL-Systeme liegt in der Analyse des Nutzungskontextes, der eingesetzten Hardware (Sensoren, Aktoren, Leitrechner, etc.), der aufzubringenden Konfiguration des Systems durch den Nutzer, die UX (User Experience) bei Interaktion mit dem System, wie z. B. die Gebrauchstauglichkeit, Akzeptanz und Barrierefreiheit, und die angewendeten Verfahren zur Erkennung oder Vorhersage von ungewöhnlichem Verhalten (Verhaltensmonitoring).



Den Abschluss des Kapitels bildet eine kurze Einführung zu angrenzenden Themen der künstlichen Intelligenz, deren Schwerpunkt ein maschinelles Lernen bzw. das Agentenmodell beschreibt - das AAL-System soll irgendwann von den gemachten Fehlern (FPs und FNs) lernen können. In Vorbereitung zur automatisierten Verhaltensanalyse in Abschnitt 2.6, ist in abschließenden Worten der Übergang von der Theorie zur praktischen Umsetzung beschrieben.

## 2.1 Zielgruppe: Generation 65+

Von den im Jahr 2014 ca. 17 Millionen in Deutschland registrierten Menschen der Generation 65+, leben 34% (5,78 Mio.) alleine in ihrem Privathaushalt [Bun15b, S.5]. Wenn auch 2,2 Millionen Menschen in dieser Altersgruppe auf Pflege angewiesen sind (davon 67% zu Hause versorgt) [Bun15b, S.39], ist die Anzahl der Personen, die eine Unterstützung in den ADL durch Assistenzsysteme benötigen, durchaus höher: Denn als pflegebedürftig werden nur solche Personen eingestuft, die nach dem Sozialgesetzbuch (§ 64 SGB XII) in die Pflegestufen 1-3 eingruppiert werden. Die Einstufung erfolgt durch die Menge der Verrichtungen, die zur Verringerung der Defizite bei den ADL von Körperpflege, Ernährung, Mobilität und hauswirtschaftlicher Versorgung notwendig sind [Was15].

Alle Personen, die nicht pflegebedürftig sind, und aufgrund von gesundheitlichen Beeinträchtigungen ihre ADL nur unzureichend erfüllen können, zählen nicht zu den genannten 2,2 Millionen: Ab einem Alter von 65 Jahren ist mindestens jeder Fünfte krank oder unfallverletzt und kurzfristig auf die Unterstützung seiner Mitmenschen oder auf Pflegepersonal angewiesen, um weiterhin im eigenen Haushalt leben zu können [Bun15b, S.35]. Demnach sind der für dieser Arbeit relevante Zielgruppe auch die Menschen zugeordnet, deren geminderte Selbstständigkeit bzw. Lebensqualität auf kognitive und körperliche Einschränkungen zurückzuführen ist [Gmb15g, S.9].

Nicht nur weil die Akzeptanz zur Nutzung der Informations- und Kommunikationstechnologie bei älteren Menschen stetig ansteigt [Bun15b, S.29 f.], sondern auch weil aktuelle Studien zeigen, dass die Bereitschaft älterer Menschen steigt, ihre soziale Umgebung, Wahrnehmung und Einstellung den neuen Technologien anzupassen [Zaa+15, S.68], ist eine (Weiter-)Entwicklung von intelligenten Assistenzsystemen notwendig.

Getreu dem bevorzugten Prinzip *[..] Lieber daheim als im Heim [..]* [Mun15, S. 1] kann den Personen durch die Nutzung der AAL-Systeme ein möglichst langes und selbstbestimmtes Leben in der gewohnten Umgebung geboten werden. Das Nachfragepotenzial der Generation 65+ wird dabei auf 2,8 Mrd. Euro im Jahr geschätzt [Gmb15g, S.20 ff.].

Aber auch aus technischer Sicht steht durch die anhaltende Strukturverkleinerung der Halbleiter, die eine kompaktere Bauweise von Hardware ermöglicht [KG15], einer Weiterentwicklung der AAL-Systeme nichts im Wege.

Um den Leser die Personen der Zielgruppe noch eingehender zu verdeutlichen, damit er schlussendlich auf evtl. nahe Bekannte, Freunde und Verwandte als mögliche Nutzer des AAL-System schließen kann, wird für die folgende Personenbeschreibung auf die Methodik der "Ad-Hoc Personas" zurückgegriffen. In Abgrenzung zu der Persona-Methode nach Nielsen [Nie12, S.2 ff.], der "Goal-Directed"-Methode nach Cooper [Nie12, S.14 f.] und der "Role-Based"-Methode nach Adlin & Pruitt [Nie12, S.15], beschreibt die "Fiction-Based"-Methode zur Erzeugung von "Ad-Hoc Personas" nach Norman [Nor15], ein Verfahren, bei dem der Autor auf eigene extensive Erfahrung und Intuition zurückgreifen kann.

### Die Persona "Hildegard":

*Zitat:* "Im Seniorenheim fühle ich mich nicht daheim und von den Betreuern nicht verstanden. Außerdem schmeckt das Essen nicht. Deshalb möchte ich weiterhin zu Hause wohnen."

*Empfohlene Priorität:* Primär

Hildegard ist genau die Person, der wir helfen wollen. Die Erfüllung ihrer Ziele steht als Anwendungszweck des AAL-Systems an erster Stelle. Wenn das AAL-System in angemessener Zeit auf Notsituationen reagiert, ermöglichen wir Hildegard weiterhin in ihrem gewohnten Umfeld leben zu dürfen.

*Lerne Hildegard kennen:* Hildegard ist 79 Jahre alt, pensionierte Reinigungskraft und Haushaltshilfe und lebt alleine in ihrer Privatwohnung. Nach dem Tod ihres Mannes vor 3 Jahren, mit dem sie über 50 Jahre verheiratet war, geht es ihr psychisch und physisch zunehmend schlechter. Wenn auch die große Trauerphase überstanden ist, zeigt sie neuerdings ihren Verwandten und Bekannten gegenüber ein boshaftes Verhalten: Ihrer Meinung abweichende Ansichten werden grundsätzlich nicht in Betracht und teilweise ins Lächerliche gezogen. Hildegard leidet aber nicht an einer Demenzerkrankung und ist für ihr Alter geistig noch sehr fit. Das äußert sich dadurch, dass sie es trotz ihrer Vorerkrankung, der Diabetes, schafft, eine ausgewählte Ernährung einzuhalten und ihr Insulin regelmäßig zu spritzen. Sie achtet akribisch auf ihre Körperpflege und ein gepflegtes Äußeres. Leider wollen - wie sie selbst sagt - "ihre Knie nicht mehr so richtig" und so ist Hildegard schon das ein oder andere mal auf der Treppe und im Wohnraum gestürzt und konnte selbst aus eigener Kraft nicht mehr aufstehen. Im letzten Sommer, bei dem Versuch den Ventilator vom Dachboden zu holen, ist sie von der Dachbodentreppe abgerutscht und hat sich beim Aufprall den Arm gebrochen. Glücklicherweise konnte "**Alexandra**", Hildegard's einzige Tochter, die ihren Besuch am Vorabend angekündigt hatte, ihre Mutter nach mehr als drei Stunden auffinden und den so dringend benötigten Notruf absetzen.

*Hildegard's Ziele:*

- Mir ist bewusst, dass ich nicht mehr die Jüngste bin: Ich möchte trotz meiner körperlichen Einschränkungen weiterhin zu Hause wohnen bleiben.
- Ich möchte nicht mehr wie im Sommer solange auf Hilfe warten müssen.

*Hildegard's Fragen:*

- Ich bin bereit dazu, dass meine Wohnung mit intelligenten Computern ausgestattet wird, aber inwieweit wird dadurch meine Privatsphäre eingeschränkt?
- Mit den ganzen neuen Computern kann ich nicht umgehen, kann die Bedienung jemand für mich übernehmen?

Die Persona " **Alexandra** ":

*Zitat:* " Nach dem Tod meines Vaters ist meine Mutter einfach nicht mehr dieselbe. Ich mache mir ständig Sorgen um sie."

*Empfohlene Priorität:* Sekundär bis Tertiär

Alexandra ist die Person, die wir von der Anschaffung eines AAL-Systems überzeugen wollen. Die Zuverlässigkeit des AAL-Systems entscheidet darüber, ob Alexandra nicht mehr in ständiger Sorge um ihre Mutter leben muss und somit ihren Ballast des 24 Stunden " Bereitschaftsdienst " ablegen kann.

*Lerne Alexandra kennen:* Alexandra ist 54 Jahre alt und besetzt eine 75% Stelle im Verkauf bei einem namhaften deutschen Handels- und Touristikkonzern. Sie ist mit ihrem Mann " Manfred " nun 27 Jahre glücklich verheiratet und hat mit ihm zwei Kinder - Markus ist 24 und Marie ist 22 Jahre alt. Da ihre Kinder bereits erwachsen sind, opfert Alexandra einen Großteil ihrer Freizeit damit, ihre Mutter bei körperlich anstrengender Arbeit im Haushalt zu unterstützen (z. B. Einkauf von Lebensmitteln und Getränken) oder, um ab und an nach dem Rechten zu sehen. Seitdem sie ihre Mutter im Sommer schwerverletzt in ihrer Wohnung aufgefunden hat, lebt sie in ständiger Unsicherheit, wenn sie Hildegard an einigen Tagen nicht besuchen kann. Alexandra hat somit den Entschluss gefasst, ihre Mutter während ihres dreiwöchigen Urlaubes nicht alleine zu Hause zu lassen, sondern in einem Seniorenheim unterzubringen. Ohne Alexandra davon in Kenntnis zu setzen, hat sich Hildegard nach wenigen Tagen - so die Pflegerin - mit den Worten " Ich bin jetzt weg! " einfach selbst entlassen. Nach dieser Aktion steht für Alexandra fest, ihre Mutter braucht eine assistierende Pflegekraft, und da die Krankenkasse ihr Vorhaben finanziell nicht unterstützt, fällt die Wahl auf den " unsichtbaren Pfleger " .

*Alexandra's Ziele:*

- Ich möchte meine Freizeit wieder Manfred und meinen Hobbies widmen, mit der Sicherheit, dass es meiner Mutter gut geht.
- Für die Anschaffung des AAL-Systems sind folgende Punkte relevant: Es soll einfach zu bedienen sein und ich möchte lieber einmal zuviel fälschlicherweise alarmiert werden, als einmal zu wenig in einer Notsituation.

*Alexandra's Fragen:*

- Mit der Installation und Konfiguration möchte ich nicht unbedingt eine externe Firma beauftragen. Kann das zur Not auch mein Sohn Markus übernehmen?
- Wenn ich zu Hause bin und ein Notfall vorliegt, reicht ein Anruf auf meiner Festnetz-Rufnummer. Kann das AAL-System mich auch kontaktieren, wenn ich unterwegs bin?

## 2.2 Arten und Modellierung der ADL

Die Personen der Zielgruppe möchten weiterhin ihrem alltäglichen Leben in ihrer gewohnten sozialen Umgebung nachgehen. Damit ein AAL-System eine Abweichung im alltäglichen Leben, also Abweichungen in den einzelnen ADL, erkennt und als Notsituationen kennzeichnen kann, müssen die ADL als wesentlicher Bestandteil des Alltags wohldefiniert sein.

Ausgehend von [PH13, S.114] ist der Alltag jedoch als ...

*"[...] Teil der sozialen Lebenswelt, der 'keiner formalen Steuerung' unterliegt' [...]"  
und "[...] alltägliches Handeln ein relativ schwach formalisiertes Handeln [...] was  
regelmäßig und andauernd, eben 'alle Tage' geschieht."*

... zu betrachten. Weiterhin teilt [PH13, S.122 f.] dem Alltag alle Arten "praktischer Lebensbewältigung" bzw. "empirisch-kausale Handlungsprobleme" zu, welche durch "technische Artefakte" mit den "Zweck-Mittel-Schemata" bewältigt werden können. Die Lösung der Handlungsprobleme ist mit einer korrekten Ausführung der ADL gleichzusetzen und die Artefakte und deren Schemata als Hardware des AAL-Systems und dessen Software-Dienste anzusehen.

Um ein ungewöhnliches Verhalten aufzudecken, bedarf es für die Definition einer ADL hinreichender Informationen zur Ausführungsfolge, der Permutationen und der jeweiligen Dauer für die Teilaktivitäten. [Mun15, S.8] äußert sich dazu wie folgt:

*"Bislang existieren keine formalen Beschreibungen dieser Zusammenhänge [...],  
dennoch ist dieses Wissen zwingend erforderlich [...] für die computerbasierte Nutzung [...]"*

An dieser Stelle sei darauf hingewiesen, dass [Mun15] zwar einen möglichen Lösungsansatz vorstellt, dieser aber nicht mit den in Sektion 1.2 Teilanforderungen und den Zielen der Persona, hinsichtlich einfacher Inbetriebnahme des AAL-Systems, vereinbar ist - dazu mehr in Sektion 2.4.3 für das Projekt "SAMDY".

Nach jetzigem Kenntnisstand existiert demnach kein generisches Modell oder eine computerbasierte Sprache, die den Alltag und vor allem die ADL im Detail kennzeichnet und somit direkt einer personenspezifischen Verhaltensanalyse dienen könnte.

Historisch gesehen, wurde der Begriff ADL bzw. ATL (Aktivitäten des täglichen Lebens) bereits im Jahre 1977 von der Sr. Liliane Juchli geprägt. In ihrem Modell kennzeichnet sie insgesamt 12 ATL, die neben Psychischen- auch Umgebungsfaktoren betrachten [AIG15]. Die 12 ADL lauten [BB15]:

- 1 Wach sein und schlafen
- 2 Sich bewegen
- 3 Sich waschen und kleiden
- 4 Essen und trinken
- 5 Ausscheiden
- 6 Körpertemperatur regulieren
- 7 Atmen
- 8 Sich sicher fühlen und verhalten (für Sicherheit sorgen)
- 9 Raum und Zeit gestalten (Beschäftigungen nachgehen)
- 10 Kommunizieren
- 11 Kind, Frau, Mann sein (geschlechtsspezifisches Fühlen)
- 12 Sinn finden (Werden, Sein, Vergehen)

Im Jahre 1980 erscheint das Lebens- und Pflegemodell "RTL-Modell (Roper-Logan-Tierney-Modell)" in dem Buch "The Elements of Nursing" [MK12, S.30], in dem von der Pflegewissenschaftlerin Nancy Roper die Begriffe AL (Aktivitäten des Lebens) bzw. LA (Lebensaktivitäten) geprägt werden.

[MK12, S.32] fasst dabei die zentrale Motivation zur Entwicklung des RTL-Modells zusammen:

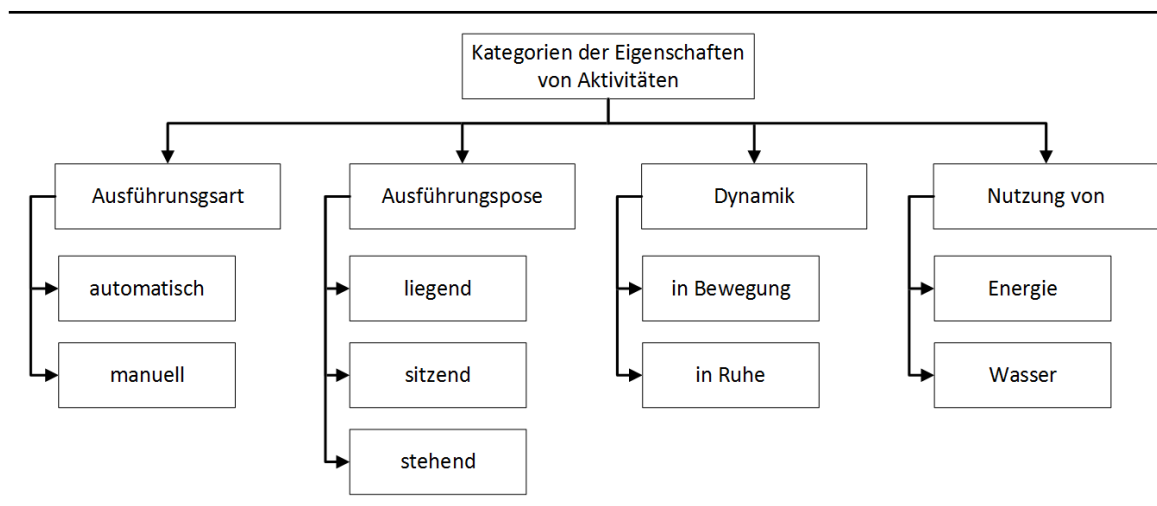
*"Roper, Logan und Tierney [...] heben hervor, dass es nicht möglich ist, die Komplexität des Lebens in einem Modell zusammenzufassen [...]" und das "[...] unser Wissen über das Leben immer noch höchst unvollständig ist [...]"*

Das RTL-Modell ist nicht nur eine reine Aufzählung von Aktivitäten, sondern beschreibt in seinen Konzepten die Abhängigkeits- bzw. Unabhängigkeitskontinua und die beeinflussenden Faktoren im Zusammenhang der Lebensspanne eines Individuum [MK12, S.32 f.]. Die Infor-

mationen zur Ausführungsdauer oder Teilaktivitäten der ADL können diesem Modell nicht entnommen werden.

Neben diesem Modell existieren Indexes, die Bewertungsverfahren für die Einstufung der Pflegebedürftigkeit von Personen darstellen: Der Barthel-Index betrachtet zehn pflege-relevante ADL, die mit einer Ordinalskala  $\{0, 5, 10\}$  bei maximal 100 zu erreichenden Punkten bewerten. Auch wenn eine Person eine hohe Punktzahl erreicht und nicht unbedingt als pflegebedürftig eingestuft wird, heißt dies nicht, dass die Person grundsätzlich keine Hilfe bei den ADL benötigt [Inf15, S.12 f.]. Der Katz-Index beurteilt eine Unabhängigkeit für sechs ADL von 0-6 Punkten [GN12, S.1 f.]. Bei Anwendung der Verfahren für die oben beschriebene Persona "Hildegard" vergibt der Barthel-Index 90 und der Katz-Index 6 Punkte.

[Mun15, S.282] liefert eine Mindmap von Indikatoren für neun ADL, die als Ergebnis aus einem Interview mit examinierten Pflegekräften entstanden ist. Darin enthalten sind auch einige Teilaktivitäten für eine ADL. [Mun15, S.8] erwähnt die Pflegebemessungszeiten aus dem SGB (Sozialgesetzbuch) und verwendet für den ersten Ansatz der ADL-Modellierung die doppelten Werte, um einen Sicherheitspuffer einzubauen [Mun15, S.181]. Um allerdings auch den Wechsel und die Nebenläufigkeit in bzw. von den ADL zu betrachten, sind die Zeiten für Mobilität (Gehen, Stehen, Treppensteigen, Verlassen und Wiederaufsuchen der Wohnung) von zentraler Bedeutung. In den Richtlinien zur Begutachtung von Pflegebedürftigkeit wird explizit darauf hingewiesen, dass eine Angabe von Zeiten für diese ADL jedoch vom individuellen Wohnbereich abhängig ist [Ke09, S.120 f.].



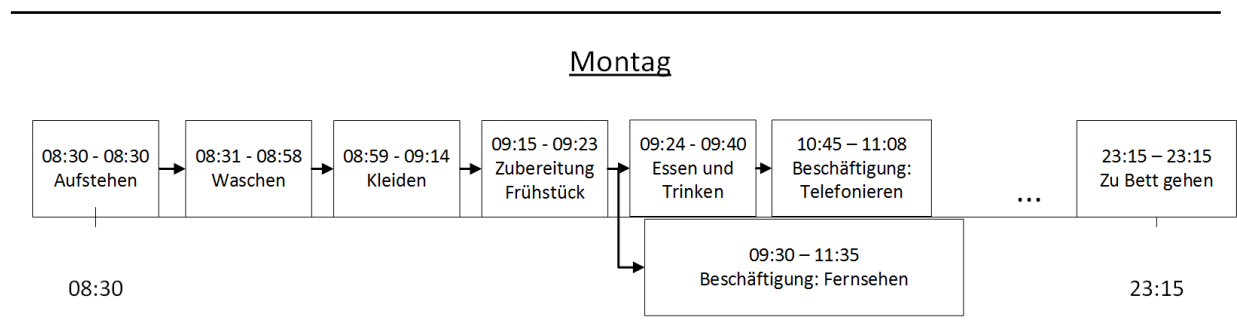
**Abbildung 2.1:** Kategorien von Eigenschaften der Aktivitäten [i.A.a. Mun15, S.177]

Die Abbildung 2.1 zeigt die rudimentären Eigenschaften der Aktivitäten. Beispielsweise wird die ADL "Ausscheiden" manuell und in Ruhe von Hildegard ausgeführt. Ihre bevorzugte Pose ist dabei "sitzend". Beim Betätigen der Toilettenspülung und anschließenden Händewaschen nutzt sie Wasser.

Für unserer zukünftiges AAL-Systems sind die oben genannten Informationen der ADL zu allgemein formuliert, als dass vordefinierte oder aus den Daten der Lernphase erzeugte Aktivitätsmodelle genutzt werden könnten, um in der anschließenden Erkennungsphase auf ein normales- oder abweichendes Verhalten zu schließen. Deshalb ist es zwingend erforderlich, die Parameter der Aktivitäten detaillierter zu beschreiben.

In der Literatur sind synonym zu den Begriffen der Lern- und Erkennungsphase, aus dem Bereich der unterstützenden Pflege, die Begriffe "Aktivitätserkennung", "Kontextgenerierung" und "Verhaltensermittlung" anzutreffen [Mun15, S.7] und für das Fachgebiet von "Context-Aware Applications" mit der "Context transformation" und "Context Awareness" bzw. "Context Prediction" zu vergleichen [May04, S.34]. Die Anwendungsfelder der "Context-Aware Applications" erstrecken sich von einer Rekonfiguration des Systems und der Prävention von Unfällen über die Alarmierung in einer Ablaufplanung bis hin zur Gesundheitsvorsorge [May04, S.15 f.].

Ein erster Gedanke verleitet zu sagen, dass die Aktivitäten sequentiell - also nacheinander - mit geringen zeitlichen Deviationen ausgeführt werden. Die Abbildung 2.2 zeigt einen möglichen Tagesablauf (Montag) für die Persona "Hildegard" in der parallele - also nebenläufige - Aktivitäten ausgeführt werden [Mun15, S.14].



**Abbildung 2.2:** Tagesablauf (Montag) für Persona Hildegard - Ausschnitt

Nachdem Hildegard aufgestanden ist, sich gewaschen und gekleidet hat, bereitet sie das Frühstück zu. Seitdem ihr Mann gestorben ist, isst sie lieber im Wohnzimmer und schaut dabei fern, als alleine in der Küche zu sitzen. Um 9:55 Uhr schaut sie dann "Sturm der Liebe" im Ersten. Neue Geschehnisse der Telenovela werden direkt im Anschluss mit Freundin "Agnes" am Telefon diskutiert - nebenbei lauscht Hildegard den Nachrichten der "Tagesschau". Auch wenn Hildegard die in der Abbildung dargestellten parallelen Aktivitäten nicht unbedingt aktiv gestaltet, sind diese in der ADL-Modellierung für das AAL-System zu berücksichtigen.

Zusammenfassend sind die ADL als Intervalle oder Prozesse mit einem fixen Start- und Endzeitpunkt und einer definierten Anzahl an Teilaktivitäten zu betrachten. In diesem Kontext wird ein Ereignis als ein Satz von Sensordaten für eine konkrete Uhrzeit deklariert, dass ei-

ner (Teil-)Aktivität zugeordnet werden kann. Die Literatur unterscheidet drei Ereignistypen [Mun15, S.73]:

- 1 Das "Start-Ereignis" kennzeichnet den Beginn einer Aktivität.
- 2 Der Ereignistyp "Complete" benennt das Ende einer Aktivität.
- 3 Für die Unterbrechung einer Aktivität steht das "Abort-Ereignis".

Aktivitäten, deren Ausführung nur wenige Sekunden in Anspruch nimmt, werden als atomar bezeichnet, d.h. eine Aktivität besteht aus nur einem Ereignis [Mun15, S.73]:

*"Für zwei atomare Aktivitäten lässt sich nie eine Parallelität ableiten."*

Die ADL "Aufstehen" und "Zu Bett gehen" in der Abbildung 2.2 sind atomar. Für die Güte der ADL-Modellierung ist zudem eine feingranulare Beschreibung der Inter- und Intravariabilität von zentraler Bedeutung, die folgende Fragestellungen beantworten sollte [Mun15, S.14 f.]:

- Kann die ADL parallel oder nur sequentiell ausgeführt werden?
- Aus welchen Teilaktivitäten besteht die ADL und können sich diese gegenseitig unterbrechen?
- Kann ein Ereignis eindeutig einer ADL zugeordnet werden?
- In welchem temporalen Kontext findet die ADL statt?

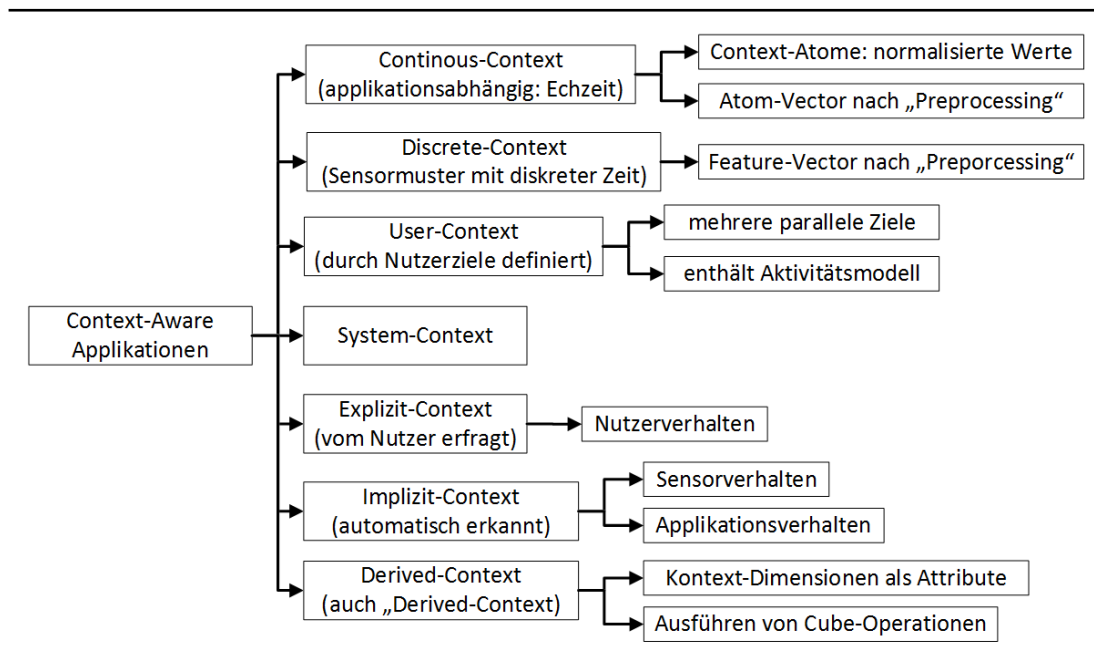
Der Kontext (Hintergrundinformationen) einer ADL ist in Benutzer-, Umgebungs- und Zeitinformationen einzuteilen [Mun15, S.14]. Bis jetzt wird in der Beschreibung der Modellierung davon ausgegangen, dass sich Hildegard zu jeder Zeit alleine in der Wohnung aufhält - Besuche von Agnes oder Alexandra oder das Vorhandensein möglicher Haustiere bleiben unberücksichtigt. Die Benutzerinformationen geben Aufschluss über die Anzahl und Personen oder Haustiere, die an einer ADL beteiligt sind. Da Besuche im Alltag nicht zwingend regelmäßig geschehen, ist für die künftige Implementierung des AAL-Systems zu entscheiden, ob Besuche, die innerhalb der Lernphase und Erkennungsphase auftreten, mit aufgezeichnet und ausgewertet oder gekennzeichnet und vernachlässigt werden sollen - denn wenn eine Notsituation vorliegt, könnte der Besuch entsprechende Hilfemaßnahmen einleiten. Die Umgebungsinformationen beschreiben den Ausführungsort der ADL. Ein "Einkleiden" findet im Schlaf- oder Badezimmer statt, hingegen das Zubereiten von Mahlzeiten in der Küche. Neben einer stündlichen und minütlichen Auswertung der Ereignisse ist eine Auswertung der Zeitinformationen auf täglicher Basis denkbar: Denn Hildegard schaut nur am Montag die Telenovela "Sturm der Liebe", an anderen Tagen geht sie vormittags auf dem Friedhof das Unkraut jäten oder ihre Freundin besuchen. Für den Begriff "Context-Awareness", aus dem Fachgebiet der "Context-Aware Applications", sieht die Beschreibung eines Kontextes die Beantwortung von vier Fragen vor [vgl. Mun15, S.46 ff.; May04, S.10 f.]:

- 1 Wann? - Die temporale Einordnung (Ausführungszeit der ADL)



- 2 Wer? - Die Identifizierung der Bezugsperson, die im Systemmittelpunkt steht.
- 3 Wo? - Die Lokalisierung der Bezugsperson während der Aktivitätsausführung.
- 4 Was? - Die Erkennung der Aktivität.

In [May04, S.11 f.] werden die Kontextinformationen der "Context Aware Applications", wie in Abbildung 2.3 dargestellt, aufgeschlüsselt.



**Abbildung 2.3:** Kategorien von Context-Aware Applications [i.A.a. May04, S.11 f.]

Exemplarisch für die Applikationen bzw. Systeme mit "Continous-Context" arbeiten diese die gemessenen Sensordaten in kontinuierlichen Zeitintervallen ab. Die Systeme mit "Discrete-Context" werden bei eingehenden Sensordaten getriggert und werten diese als diskrete Ereignisse aus. Für die Entwicklung der "Context-Prediction" Architektur erwähnt [May04, S.12 f.] zehn rudimentäre Kontext-Aspekte: Geografisch, physikalisch, organisatorisch, sozial, emotional, Nutzer, Tätigkeit, Aktion, technisch und Zeit. Demnach kann jeder Typ von Information genutzt werden, der dazu beiträgt die Sensordaten-Entität feingranularer zu charakterisieren bzw. zu labeln: Durch die "Context Transformation" erhält das System die "High-Level Context Information", mithilfe deren die "Context Prediction" vollzogen wird [May04, S.34 f.].

Ausgehend von den Kontextinformationen aus [Dey00] beschreibt [Mun15, S.59 f.] den Formalismus der Assistenzfunktionen. Der Formalismus ist nicht zu verwechseln mit den Assistenzfunktionen der VDI-Richtlinie 3812, die für eine Bedarfsermittlung für Elektroinstallation und Gebäudeautomation bestimmt sind [GG10, S.1]. Die Assistenzfunktion  $f_{assist}(p, x, a, t)$  erhält als Eingabe ein Kontext-Quadrupel: Eine Person  $p \in beteiligtePersonen$ , die Lokalisierung  $x \in Raum$ , die derzeitige ausgeführte Aktivität  $a \in Activities$  und  $t \in Time$  für die temporale Einordnung. In [Mun15, S.59 f.] wird nicht beschrieben, worauf die Assistenzfunktion des

Formalismus abbildet. Zwar wird in diesem Zusammenhang das Erkennen von Ausführungen für Aktivitäten des AAL-Systems erwähnt, einer Abbildung auf Aktivitäten widerspricht die Eingabe  $a$ , sodass davon auszugehen ist, dass die Funktion auf eintretende Ereignisse abbildet.

Im Idealfall soll ein und dasselbe Aktivitätsmodell für mehrere Zielpersonen verwendet werden [Mun15, S.8f]. Allerdings sind in diesem Zusammenhang die Aspekte von Over- und Underfitting zu beachten [Mun15, S.86 f.]: Beim Overfitting ist das Aktivitätsmodell zu stark an den Kontext gebunden und verändert die QoS des AAL-Systems dahingehend, dass vermehrt FPs auftreten. Das System erkennt fälschlicherweise ein abweichendes Verhalten. Umgekehrt treten beim Underfitting mehr FNs auf, d.h. das Modell ist zu generisch gestaltet. Das System erkennt mögliche Notsituationen nicht. Im Hinblick für die Realisierung des AAL-Systems ist das über-angepasste Aktivitätsmodell vorteilhafter. Dadurch alarmiert das AAL-System Alexandra unnötigerweise, anstatt Hildegard in einer Notsituation nicht zu helfen.

Die oben genannten Informationen geben einen Überblick über die verschiedenen ADL, deren Eigenschaften und zugehörige Kontextinformationen. Der Folgeabschnitt greift Ansätze auf, die eine ADL-Modellierung vollziehen. Für das Proof-of-Concept, der Realisierung eines AAL-Systems auf Basis von openHAB, sind für die Auswahl zur geeigneten ADL-Modellierung die in Sektion 1.2 gegebene Teilanforderung zur initialen Konfiguration der Systemkomponenten (keine weitere Bedienung) und die Ziele der Persona Hildegard und Alexandra zu beachten.

Für die Aktivitätserkennung bzw. die Erzeugung von Aktivitätsmodellen zeigt ein bewerteter Systemvergleich neun unterschiedlich entwickelte Systeme für fünf Kriterien [Mun15, S.23]: (1) Einfache Inbetriebnahme, (2) verständliche Ausgabe, (3) Komplexität der Aktivitätsmodelle, (4) praktische Realisierung und (5) Evaluationsergebnis.

Dabei sind unter (1) vier unterschiedliche Kategorien von Verfahren gelistet:

- 1 *Gelabelter Datensatz (Ground-Truth)*: Bei diesem Verfahren der Aktivitätserkennung werden die Daten eingangs in der Lernphase aufgezeichnet und gleichzeitig gelabelt. Für die Inbetriebnahme des AAL-Systems müsste Hildegard (oder Alexandra), jede durchgeführte ADL im System mit Aktivitätenlabel, Start- und Endzeitpunkt und evtl. Deviationen eintragen. Solche Anforderungen sind für die Persona unvorstellbar und deshalb auch zurecht in diesem Systemvergleich negativ bewertet [Mun15, S.10].
- 2 *Datenbank & Mining* (neutrale Bewertung): Für diese Kategorie der Aktivitätserkennung werden Verfahren aus dem Bereich des "Data-Mining" eingesetzt. Die gemessenen Sensordaten werden zuerst diskretisiert und in einer Datenbank persistiert. Anschließend wird dann z. B. mit dem PPS (Position Pairs Set) Algorithmus des (Frequent-)Episode-Mining die Ereignissequenzen erzeugt, welche anschließend in einer Baumstruktur festgehalten werden [Lin+14]. Die Persona müsste bei diesem Verfahren lediglich einen Lernzeitraum angeben und die Lernphase starten (Betätigen einer Schaltfläche). Das Problem beim

"Umgang mit großen Datenmengen" wird durch ein "Sliding Window" gelöst [Mun15, S.18], d.h. bei der Aktivitätsmodellierung und für die Verhaltensanalyse in der Erkennungsphase wird nur ein begrenztes zeitliches Fenster betrachtet.

- 3 *Aktivitäts-Mining* (positive Bewertung): In Abgrenzung zur Kategorie des "Datenbank & Mining" werden für die Verfahren der Kategorie des "Aktivitäts-Mining" meist keine DTs (Decision Trees) eingesetzt. In erster Instanz werden Pattern durch Mining-Verfahren extrahiert und anschließend in Clustern zusammengefasst. Die generierten Cluster werden dann z.B. durch ein HMM (Hidden Markov Modell) klassifiziert [Mun15, S.22 ff.]. In der Literatur wird darauf hingewiesen, dass der Einsatz von HMM bei großen Datenmengen keine Auswertung in Echtzeit garantieren kann. Stattdessen verwenden die Autoren ANNs (Artificial Neural Networks) [vgl. Suk+15, S.120; Bou+15, S.15]. [Mun15, S.136] erwähnt, dass der Abbruch einer Aktivität bei Verwendung von HMM nicht sicher erkannt wird. Die Inbetriebnahme des AAL-Systems für die Persona ist mit Kategorie (2) zu vergleichen.
- 4 *A-priori-Wissen* (positive Bewertung): Für den Ansatz der "A-Priori-Modellierung" versucht [Mun15, S.131 ff.] aus dem bekannten Wissen über die ADL und den Kontextinformationen eine Modellierung ohne konkrete Lernphase (sofortige Einsatzbereitschaft des Systems) zu extrahieren, d.h. vor Inbetriebnahme des Systems findet eine personenspezifische Auswahl von ADL statt, die dann in CTT (Concur Task Tree - Metamodell zu Beschreibung von Aktivitäten) modelliert und anschließend in PN (Petri Netze) überführt werden. Das AAL-System müsste demnach einen reichhaltigen Satz an vor-modellierten ADL bereitstellen, was zudem die Defizite von Under- und Overfitting beachtet. Falls eine Aktivität für unsere Persona fehlen sollte, ist diese nachträglich zu modellieren und im System einzutragen. Es ist aber zu hinterfragen, ob diese Vorgehensweise eine Missachtung der Anforderungen und Ziele der Persona ist.

Der obenstehende Abschnitt stellt eine Kategorisierung der existierenden Verfahren zur Aktivitätsmodellierung (Nutzerprofile) dar und nennt Exemplare. In Anbetracht der oben genannten Defizite der Kategorien (1) und (4) kann die anstehende Realisierung des AAL-Systems (Proof-of-Concept) nur Verfahren der Kategorien (2) und (3) umsetzen. Die Verfahren der (1) und (4) Kategorie schränken die Ziele bzw. Anforderungen der Personas an ein AAL-Systems derart ein, dass nicht nur die Gebrauchstauglichkeit nicht zufriedenstellend ist, sondern auch die Vorerfahrung, also z.B. eine Inbetriebnahme des Systems nicht ohne externe Dienstleister möglich ist.

Die Anforderung zur Nutzung der quelloffenen Heimautomatisierungssoftware openHAB für das AAL-System und die Kontextsensitivität triggert dabei folgende Idee: Die initial konfigurierte openHAB-Sitemap enthält hierarchische Kontextinformationen zum Installationsort der Sensoren. Das Verarbeiten dieser Informationen könnte dazu dienen, die Aktivitäten eindeuti-

ger zu kennzeichnen und der in einer Notsituation alarmierten Person eine Auskunft über den Aufenthaltsort (Raum-genau) der hilfebedürftigen Person zu liefern.

## 2.3 Sensor-, Kommunikations- und Messdatentypen

Damit die Akzeptanz zur Nutzung der AAL-Systeme gesteigert und die gewohnte Umgebung der Personen so wenig wie möglich verändert wird, sind die Sensoren unauffällig in den Alltag einzubinden [Mun15, S.176 f.].

*"Im Allgemeinen sollten Sensoren möglichst klein, leicht und einfach in der Handhabung sein."* [Mun15, S.36]

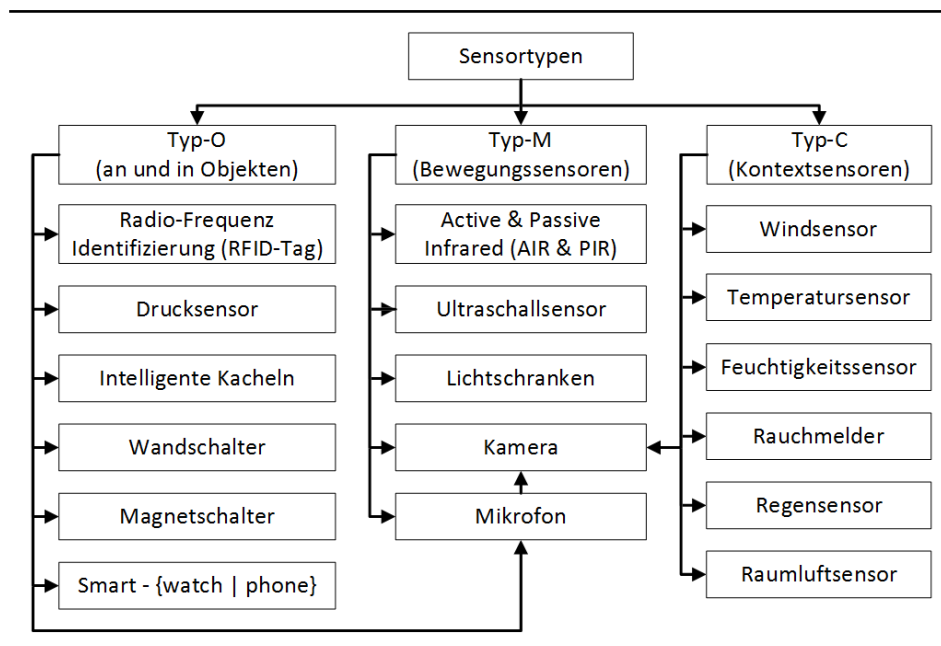
Für eine einfache Handhabung bzw. Nutzung der Sensoren ist sowohl der Aufwand für eine initiale Installation, Konfiguration und Steuerung als auch eine effiziente Energiezufuhr, Lebensdauer, Fehleranfälligkeit und Sicherheit zu beachten. Diese Aspekte sind nicht nur von den eingesetzten Sensortypen, sondern auch von den herstellerabhängigen Hard- und Softwarekomponenten abhängig. Laut der Fraunhofer IGD ist das Potenzial der heutigen Sensoren als autonome Bausteine noch lange nicht ausgeschöpft [UNNG15].

In [al15, S.149] ist eine Aufstellung für gängige Hausautomatisierungssysteme und deren verfügbare Hardware- und Softwarekomponenten gegeben. Unterteilt sind die Komponenten in die Kategorien (1) Sicherheit, (2) Komfort, (3) Steuerung und (4) Anbindung:

- (1) In die Kategorie der Sicherheitskomponenten fallen sowohl Öffnungs-, Bewegungs- und Rauchmelder als auch Türöffner, Fensterantriebe, Magnetkontakte und die klassische Alarmanlage.
- (2) Neben Umwelt- und Wettersensoren, Heizungs- und Rolladensteuerung sind für den Komfort auch Schalt- und Dimm-Aktoren, die in "Auf Putz" (AP) und "Unter Putz" (UP) eingeteilt sind, vorgesehen.
- (3) Die Nichtverfügbarkeit von Komponenten der Steuerung stellen zugleich ein "K.O.-Kriterium" dar, denn wenn unsere Persona das System nicht über eine zentrale Steuerung, über einen PC oder Browser bedienen kann, stellt die initiale Konfiguration der Komponenten eine zu große Hürde dar.
- (4) Eine Anbindung der Systeme ist entweder drahtgebunden oder drahtlos realisiert. Da für drahtgebundene Komponenten zumeist eine UP-Vernetzung verwirklicht wird, bevorzugen unsere Persona, neben AP-Aktoren, den Einsatz von funkvernetzten Komponenten mit einem eigenen Energiespeicher (batterie- oder solarbetrieben).

In der Aufstellung zeigt das "Homematic"-System die höchste Verfügbarkeit der oben genannten Hard- und Software-Komponenten.

[Mun15, S.130 + 173] teilt die Sensoren in drei Typen ein. In Abgrenzung zu obiger Aufstellung, liegt der Schwerpunkt der Einteilung in der Informationsart, also Kontextinformationen, und dem Installationsort der Sensoren: Die Sensoren vom Typ-O werden an oder in Objekten angebracht, die der Systemnutzer bei Ausführung einzelner ADL nutzt (z. B. Möbel und Haushaltsgeräte). Die Sensoren vom Typ-C liefern Umgebungsinformationen, ohne einen direkten Einfluss auf die ausgeführte Aktivität zu haben - vergleichbar mit den Umwelt- und Wettersensoren aus (2). Für eine Lokalisierung von Personen sind für Sensoren vom Typ-M z. B. Bewegungsmelder und Lichtsensoren vorgesehen.

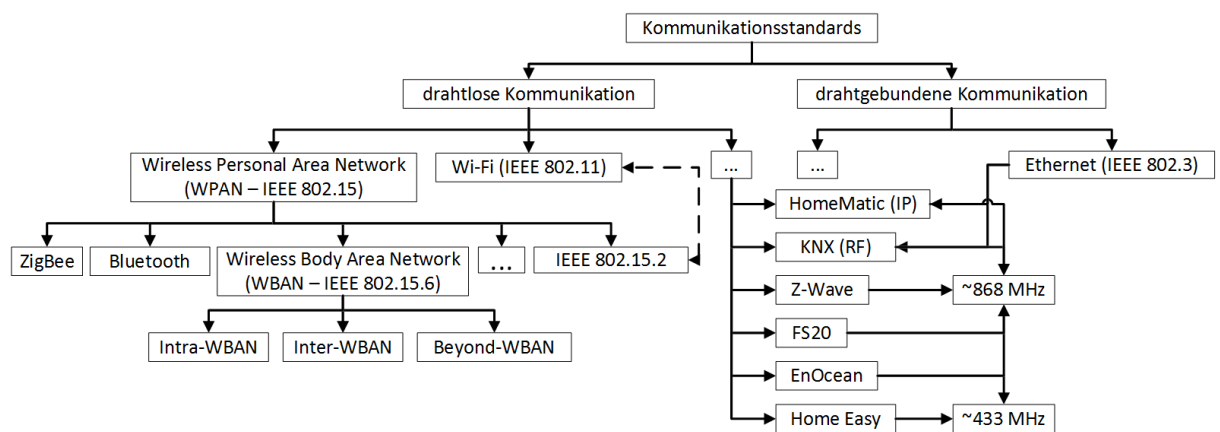


**Abbildung 2.4:** Teilauswahl von Sensoren für O-C-M-Typisierung [i.A.a. Mun15, S.130; Ras+13, S.580]

Die Abbildung 2.4 zeigt eine Teilauswahl von Sensoren für die O-C-M-Typisierung. Dabei können Kameras und Mikrofone nicht uneindeutig zugeordnet werden. Sie können in der Räumlichkeit, an und in Objekten oder Personen angebracht sein und dabei Informationen zu den ausgeführten ADL liefern: Für eine Aktivitätserkennung können sowohl mobile als auch stationäre Kameras eingesetzt werden [Son+11]. Bei der "First-Person" Aktivitätserkennung ist die Kamera um die Brust des Nutzers geschnallt [Pir+12]. Die Auswertung der Aktivitäten ist dabei so präzise, dass z.B. zwischen "Telefonieren", "Banane essen" und "Wasser trinken" differenziert werden kann [Avg+15] - Das impliziert gleichzeitig eine Bewegungserkennung. Ob ein abweichendes Verhalten bzw. eine Notsituation vorliegt, hängt von Körperhaltung, dem Erscheinungsbild oder der momentanen Bewegung ab [Mun15, S.21]. Durch die Videoauswertung kann eine Kamera auch als Rauchmelder fungieren [Tor+05]. Eine Audioauswertung kann eine Person raum-genau lokalisieren und auf deren Aktivitäten (z. B. Geschirrspülen und Fernsehen) schließen [Sto+12].

Für den Anwendungszweck der Lokalisierung bzw. Bewegungserkennung weisen die dargestellten Sensoren auch Defizite in der Reichweite und Einschränkung in der LOS (Line-Of-Sight) auf. Bei der Nutzung von Kameras oder Bewegungsmeldern ist bei der Anbringung auf Hindernisse für eine Sichtausbreitung zu achten (z. B. tote Winkel, Möbel, Raumtrenner, Dunkelheit). Falls die pflegebedürftige Person ein Haustier besitzt, muss auf tierimmune Bewegungsmelder zurückgegriffen werden [Gmb15a]. Von dem Einsatz der Bewegungsmelder in Räumen mit Fußbodenheizung ist abzuraten, da sonst fälschlicherweise Bewegungen erkannt werden. Die Anzahl der notwendigen Sensoren hängt somit stark von dem individuellen Aufbau der Wohnung und der einzelnen Räume ab. Eine raumübergreifende Bewegungserkennung durch die Auswertung von Differenzen der gemessenen Signalstärken handelsüblicher Router gilt als plausibel [Mas], d.h. eine Einschränkung der LOS stellt für dieses Verfahren kein Hindernis dar. Eine geräte-freie Lokalisierung und Detektion von Personen in Raumkorridoren mittels redundanter RF-Transceiver wird in [Wic+14, S.136] gezeigt. In [Dep+15] wird ein verwandtes Verfahren angewendet, bei dem eine Abschätzung für die Anzahl der in der untersuchten Region befindlichen Personen ermöglicht wird: Das AAL-System könnte dann auch differenzieren, ob sich unsere Persona Hildegard allein in der Wohnung aufhält oder gerade von Alexandra oder Agnes besucht wird.

Die anstehende Auswahl der Komponenten zur Umsetzung des AAL-Systems hängt nicht nur von der Beantwortung der vier Kontextfragen und Assistenzfunktion (Sektion 2.2) ab, mittels derer wir auf die Sensortypen schließen und eine Entscheidung treffen können, sondern auch von den unterschiedlichen Kommunikationsstandards und Messdatentypen, die im Folgenden berücksichtigt werden. Die Abbildung 2.5 stellt eine Selektion der Standards dar und teilt diese in eine drahtlose und drahtgebundene Kommunikation ein.



**Abbildung 2.5:** Ausgewählte Kommunikationsstandards für Sensoren [Ele+15, i.A.a. Mun15, S.173; Ras+13, S.580; al15; AG15]

Für die drahtgebundene Kommunikation sei nur die Ethernet-Technologie genannt. Eine drahtlose Kommunikation kann mit Wi-Fi Sensorik, den Technologien der WPAN (Wireless Perso-

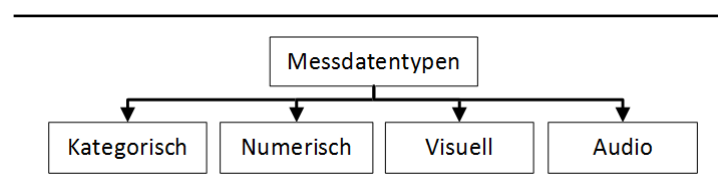
nal Area Network) oder aber auch mit firmeneigenen Lösungen wie z. B. HomeMatic, FS20, EnOcean oder KNX-RF (Konnex-Bus Radio-Frequency) umgesetzt werden. Den WPAN sind ZigBee, Bluetooth und die WBAN (Wireless Body Area Network) zugeteilt. Die IEEE-Norm "802.15.2" regelt die Koexistenz von WPAN und Wi-Fi Netzen. Die WBAN ist in "intra", "inter" und "beyond" eingeteilt, und beschreibt - allgemein formuliert - die unterschiedlichen Entfernungen von Sensorik und menschlichem Körper. Bei Intra-WBAN sind alle Sensoren, die miteinander kommunizieren, am Körper der Person angebracht. Exemplarisch für die Inter-WBAN steht die Kommunikation zwischen Smartphone und Wireless Access Points (WAPs). Bei Beyond-WBAN werden zusätzlich LAN (Local Area Network) -fremde Netze miteinbezogen [Ras+13, S.581]. Der Einsatz von Sensorik der WBAN in einem AAL-System, schränkt die Persona in ihrem täglichen Leben ein - schließlich sind die Sensoren zu jeder Tageszeit an der Person angebracht und es muss genügend Energie für die Systemnutzung zur Verfügung stehen:

*"Es ist sogar ziemlich wahrscheinlich anzusehen, dass von der Zielgruppe gelegentlich vergessen wird, den Sensor aufzuladen bzw. anzulegen."* [Mun15, S.10]

In der vorliegenden Arbeit sind demnach die nicht-WBAN Technologien von Bedeutung und "ambiente" Sensorik einzusetzen.

Für die anderen drahtlosen Kommunikationsstandards sind für eine Messdatenauswertung die Regularien der Frequenzbänder zu beachten. Neben dem 868 MHz wird in Europa auch das 433 MHz Frequenzband für die drahtlose Kommunikation eingesetzt [Gmb15c]. Die Bundesnetzagentur sieht das 433 MHz Frequenzband und dessen Subbänder für industrielle, wissenschaftliche oder häusliche Anwendungen und das 868 MHz Frequenzband zur Nutzung bei Alarmfunkanlagen vor [Bun15e, S.2 ff.]. Die maximale relative Frequenzbelegungsdauer ist für die Komponenten auf unter 1% begrenzt [Bun15e, S.4] d.h., um einer Fehlkommunikation durch Interferenzen der Frequenzen vorzubeugen, ist das Senden auf ca. 36 Sekunden pro Stunde pro Komponente eingeschränkt. [Wil15] weist zudem auf Engpässe bei der langsamen Übertragung bei dem FS20 Protokoll hin.

Unterschiedliche Sensoren erzeugen unterschiedliche Messdaten und für die anstehende Modellierung der Aktivitäten sind diese auch unterschiedlich zu interpretieren und zu transformieren [May04, S.33]. Die Abbildung 2.6 zeigt für AAL-Systeme vier Messdatentypen.



**Abbildung 2.6:** Messdatentypen für vorgestellte Sensortypen [i.A.a Ras+13, S.580]

Die Messungen von Bewegungsmeldern, Magnetschaltern oder RFID (Radio-Frequency Identification) liegen als kategorische (nominale) Daten vor [Ras+13, S.580]: {1|Wahr} eine Bewegung erkannt, die Schublade ist geöffnet und die Person hält den getaggten Gegenstand in der Hand oder {0|Falsch} keine Bewegung erkannt, die Schublade ist geschlossen und die Person hält nicht diesen Gegenstand in der Hand.

Bei Drucksensoren oder intelligenten Kacheln werden numerische Daten gemessen, um die Intensität des ausgeübten Druckes zu differenzieren.

Kameras zeichnen visuelle Messdaten auf: Zu unterscheiden sind Auswertungen von Momentaufnahmen oder eines kontinuierlichen Videostreams - eine Audioaufnahme ähnelt dieser Vorgehensweise.

### 2.3.1 Restriktion: Schutz der Privatsphäre

Für die Auswahl der geeigneten Sensoren ist nicht nur der Kommunikationsstandard oder der Messdatentyp zu beachten, sondern auch das resultierende Ausmaß zur Einschränkung der Privatsphäre der Personen: Die Wahrung der Privatsphäre ist mit den Artikeln 1, 2 und 13 der Grundrechte der BRD (Bundesrepublik Deutschland) zu begründen [Bun15f]:

*„Die Würde des Menschen ist unantastbar [...], jeder hat das Recht auf die freie Entfaltung seiner Persönlichkeit [...] und [...] die Wohnung ist unverletzlich.“*

Je nach eingesetzter Sensorik im AAL-System weisen die gemessenen Daten einen weitaus reichhaltigeren Informationsgehalt als die "einfachen" personenbezogenen Daten auf - schließlich gilt es, ein Verhalten zu analysieren und Abweichungen zu deklarieren.

Um den Schutz der Privatsphäre dennoch zu gewährleisten, sind vier Fragen kritisch zu betrachten:

- 1 Welche Arten von Daten werden gemessen oder aufgezeichnet?
- 2 Wo werden die Daten gespeichert?
- 3 Wie lange werden die Daten gespeichert?
- 4 Wer hat Zugriff auf die Daten?

Für die Beantwortung der ersten Frage bietet die Beschreibung der Messdatentypen (Sektion 2.3) bereits hinreichende Informationen. Den Einsatz von Kameras oder Mikrofonen stufen die Personen der Zielgruppe als aufdringlich ein [Mun15, S.10]. Aktuelle Studien zeigen, dass die Personen den Schutz der Privatsphäre grundsätzlich als wichtig einstufen, in Notsituationen aber auch Einbußen in Betracht ziehen: Eine gute vertragliche Vereinbarung ist zwingend erforderlich [Zaa+15, S.69], d.h. ein AAL-System könnte erst mit Ton- und Videoaufnah-



men beginnen, wenn eine Notsituation vorliegt - bei FPs zeichnet das System unnötigerweise auf und missachtet die Wahrung der Privatsphäre. Nach Meinung von [Mun15, S.12] bietet der Einsatz von binärer Sensorik (vgl. mit kategorischem Messdatentyp) den größtmöglichen Schutz der Privatsphäre.

Die gemessenen Daten können entweder auf dem zentralen Leitrechner in der Wohnung der Persona Hildegard oder aber auf einem externen Server beim Servicedienstleister des AAL-Systems persistiert werden - für den Fall, dass die Infrastruktur der Wohnung nicht für die Öffentlichkeit zugänglich ist, sollte die erste Option bevorzugt realisiert werden.

Wie lange die Daten dann gespeichert werden, hängt von der individuellen Realisierung des AAL-Systems ab. Auch wenn die Persona Hildegard einer langfristigen Speicherung der Daten zustimmen würde, ist es für die Sicherung des BDSG (Bundesdatenschutzgesetz) von Vorteil ein "geplanten Vergessen" der Daten vorzusehen [Sch15a]. Zudem wird die QoS des AAL-Systems gesteigert: Alte Verhaltensweisen unserer Persona werden nicht mehr berücksichtigt und neue in der Verhaltensanalyse stärker gewichtet.

Wer Zugriff auf die Daten hat, hängt stark vom jeweiligen Anwendungsfall und dem genutzten Speicherort der Daten ab. Wenn ein Dienstleister (z. B. Betreuer, Pflegepersonal) die analysierten Aktivitäten oder Verhaltensmuster eingehender betrachten möchte, kann er die persistierten Daten über eine Schnittstelle im lokalen Netzwerk erfragen oder aber auch an einem externen Server abrufen. Für die Realisierung in der vorliegenden Arbeit soll das AAL-System ein abweichendes Verhalten als Notsituation kennzeichnen, für unsere Peronas Hildegard und Alexandra sind die aufgezeichneten Daten nicht relevant und betrachten den "unsichtbaren Pfleger" als "Black Box System", welches keine öffentlichen Schnittstellen anbietet.

### 2.3.2 Resultierende Grenzen für die Entwicklung

Das AAL-System soll in angemessener Zeit auf Notsituationen reagieren können - jede Minute zählt. Zeitgleich möchten unsere Persona von der Präsenz des "unsichtbaren Pfleger" so wenig wie nötig in ihrer gewohnten Umgebung gestört werden, d.h. neben der ambienten Sensorik kann der zentrale Leitrechner nicht die Dimensionen eines PCs und schon gar nicht die eines Servers einnehmen. Die heutigen "Mini-PCs" sind für eine Balance zwischen Energieverbrauch und benötigter Rechenleistung optimiert, die verbauten Prozessoren streben eine maximale Rechenkapazität an [LLC15].

Für die anstehende Wahl zur Nutzung von Sensoren für das AAL-System ist es deshalb nicht nur wichtig den Kommunikationsstandard oder Messdatentyp zu betrachten, sondern auch die Limitierungen der Hardware des Leitrechners für maximale Rechen- und Speicherkapazität.

[Mun15, S.15] äußert sich in diesem Kontext über die Auswertung der Daten bei Verwendung von Kameras:

*"Zudem ist die digitale Verarbeitung von Videomaterial äußerst rechenintensiv und folglich kostspielig in der Umsetzung."*

Die Verarbeitung von großen Datenmengen mit reichhaltigen Informationen kann also durchaus auch kontraproduktiv sein [Mun15, S.108]. Eine Erkennungsgenauigkeit der Aktivitäten, wie sie in [Avg+15] beschrieben ist, kann, wenn keine Kamera verwendet wird, derzeitig nur annähernd durch eine Kombination von anderen Sensoren erzielt werden, steigert aber dann auch die Akzeptanz zur Nutzung des Systems.

Wenn ein AAL-System auf Basis von drahtlosen Komponenten realisiert wird, die das 868 MHz Frequenzband nutzen, ist zudem die 1% Frequenzbelegungsdauer zu beachten: Es kann kein zeit-kontinuierlicher Kontext (>1 Messungen pro Sekunde) verarbeitet werden, die gemessenen Daten sind in der Implementierung als eintretende Ereignisse und deren Abfolge als Ereignissequenzen zu werten. Für die von der Bundesnetzagentur geforderte maximale Sendeleistung von 25mW (14dBm) bedarf es vor Installation der Sensoren einer Reichweitenplanung [Bun15d, S.3].

## 2.4 Ambient Assisted Living - Systeme

Die vorherigen Abschnitten grenzen die Zielgruppe ein (vgl. Persona Hildegard und Alexandra), beschreiben die Arten und Modellierung der ADL, weisen auf unterschiedliche Sensoren, Kommunikationsstandards und Messdatentypen hin. Der Schutz der Privatsphäre und die daran anknüpfenden Einschränkungen in der Entwicklung sind zudem berücksichtigt. Der vorliegende Abschnitt dient der Klassifizierung der AAL-Systeme und deren genutzten Kategorien von Algorithmen. Im Anschluss daran werden real existierende AAL-Systeme beschrieben: Der Fokus der Analyse liegt sowohl im Nutzungskontext, der eingesetzten Hard- und Software als auch der UX (Usability, Akzeptanz und Barrierefreiheit) und den Verfahren zur Modellierung der ADL und Verhaltensanalyse.

Die Anwendungsfelder der AAL-Systeme, oder auch altersgerechte Assistenzsysteme, erstrecken sich von dem Erhalt der Selbständigkeit über die Förderung von Mobilität und Kommunikation bis hin zur Erhöhung von Sicherheit und Wohnkomfort [Gmb15g, S.6].

Die Assistenzsysteme sind auf oberster Ebene in drei Generationen einzuteilen [Gmb15g, S.5]:

1. Als Systeme bzw. Geräte der ersten Generation sind einfache Hilfen für das Sehen, Hören und die Mobilität anzusehen (z. B. Rollatoren, Fernrohrbrillen, Leuchtlupen,

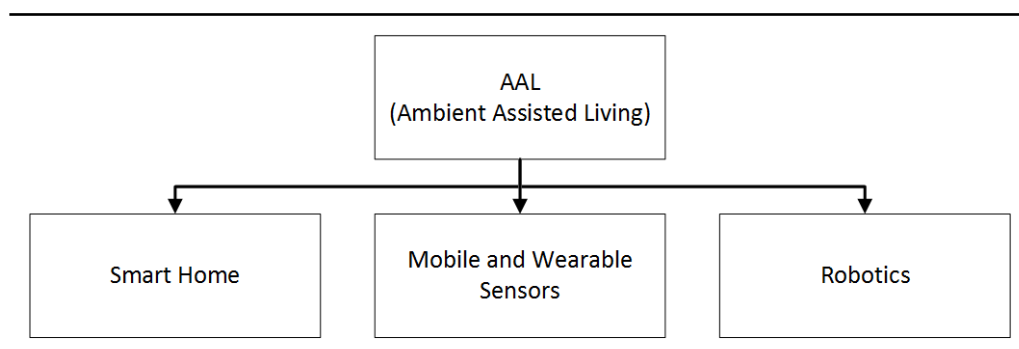
Hörverstärker oder Blutzuckermessgeräte). Die Geräte sind nicht vernetzt und tauschen keine Daten aus.

2. Die Systeme der zweiten Generation sind miteinander vernetzt und transferieren Daten an Dienstleister (z. B. Hausnotrufsysteme oder Telemonitoringgeräte).
3. Die Systeme der dritten Generation basieren auf ambienten Technologien und sind "intelligent" vernetzt. In Abgrenzung zu den Hausnotrufsystemen, bei denen der Nutzer aktiv einen Notruf ausführen muss, erkennen die Systeme der dritten Generation die Notsituation eigenständig und leiten - ohne Nutzerinteraktion- hilfeleistende Maßnahmen ein.

Zu den Systemen der zweiten und dritten Generation gehören alle die in den Sektionen 2.3 - 2.3.2 genannten Sensoren, Elemente zur Interaktion und Aktoren [Gmb15g, S.8]. Das AAL-System, der vorliegenden Arbeit, ist in die dritte Generation einzuordnen.

### 2.4.1 Varianten von AAL-Systemen

Die Literatur teilt die AAL-Systeme (3. Generation), wie in Abbildung 2.7 dargestellt, in drei Kategorien ein. Der Kategorie "Smart Home" sind alle Systeme eingruppiert, die einen Haushalt durch die Installation von Aktoren und Sensoren erweitern. Zum Einsatz kommen, wie oben partiell dargestellt, sowohl passive und aktive Infrarot-Sensoren (PIR, AIR) und Ultraschallsensoren zur Bewegungserkennung, RFID-Tags zur Objektidentifizierung, Drucksensoren für z.B. Matten und Stühle, Magnetschalter als Signalgeber beim Öffnen und Schließen von Schubladen oder Türen als auch Mikrofone und Kameras zur detailgetreuen Aktivitätserkennung [Ras+13, S.580].



**Abbildung 2.7:** Kategorien der AAL-Systeme [i.A.a. Ras+13, S.580 f.]

Es existieren fünf Subkategorien von Smart Homes, die nicht eindeutig zu trennen sind, sondern auch aufeinander aufbauen können [Suk+15, S.115 f.]:

1. *Event-tracking Home*: Die einzelnen intelligenten Komponenten können, die in der Umgebung auftretenden Ereignisse, nachhalten. Einige der Komponenten können funktional

eigenständig sein - vergleichbar mit dem Leitrechner auf der Managementebene der Automatisierung.

2. *Component-meshed Communication Home*: Im Vordergrund steht die Interaktion der Komponenten untereinander: Jede Komponente kann einen Broadcast an alle anderen senden.
3. *Remote Network-Access Home*: Die Smart Homes der dritten Kategorie besitzen nicht nur Zugriff auf lokale oder externe Netzwerke, sondern garantieren auch von außerhalb den Zugriff auf das eigene Netzwerk.
4. *Learning Home*: Die "Learning Homes" zeichnen die Aktivitäten von Nutzern auf und überwachen diese.
5. *Attentive Home*: Für eine automatische Steuerung der Komponenten werden in einem "Attentive Home" die Aktivitätsmuster aufgezeichnet.

In [Gmb15g, S.12 f.] sind, für das jeweilige Anwendungsfeld, die Teilsysteme eines Smart Homes und für das Szenario von "*Länger selbstbestimmt [...] zu Hause leben*" die Produkte einer Musterwohnung aufgelistet. Die im Kontext dieser Arbeit wichtigsten Punkte sind die breitbandfähige Grundverkabelung, eine intelligente Wohnungsausstattung inklusive Alarm und eine Homestation zum Empfangen, Messen und Auswerten von Daten.

Zu der Kategorie der "Mobile and Wearable Sensors" gehören Systeme mit Sensorik die Personen mit sich führen müssen. Zumeist werden die im Smartphone verbauten Sensoren (z.B. Global Positioning System (GPS), Beschleunigungs-, Lage- und WiFi-Sensoren) genutzt. Sind die Sensoren in der Kleidung vernäht (E-textiles) oder direkt am Körper angebracht, können diese in der Architektur der WBAN (s. Abbildung 2.5) zusammengefasst werden. Mögliche Messdaten beschreiben Herzaktivitäten, Blutzucker- oder Blutdruckwerte oder Körpertemperaturen der pflegebedürftigen Personen [Ras+13, S. 580f].

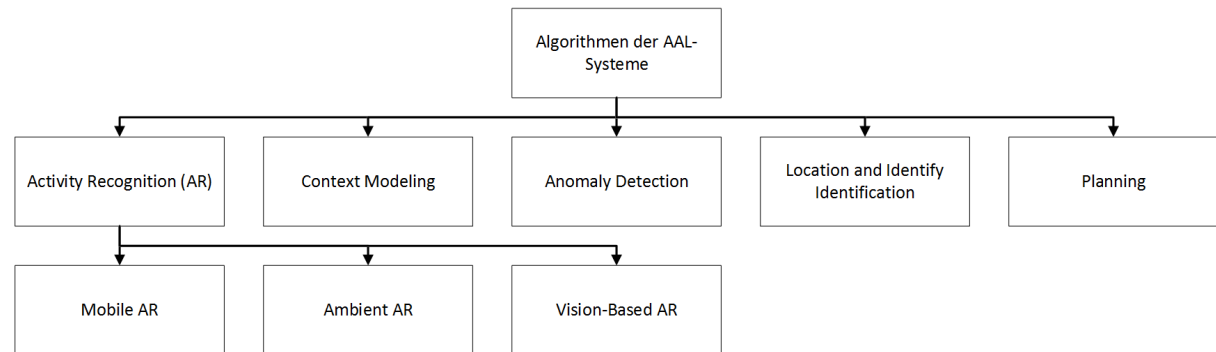
Der dritten Kategorie "Robotics" sind die AAL-Systeme zugewiesen, die eine Entwicklung von Robotern für die Unterstützung der alltäglichen Aktivitäten (z.B. Füttern, Anziehen, Körperpflege oder Telefonieren) vorsehen [Ras+13, S. 581].

Neben der dieser Kategorisierung weißt [Mun15, S.42] auf eine abweichende Eingruppierung von AAL-Systemen mit vier Klassen hin: (1) General Support of Elderly, (2) Medical Systems, (3) Intelligent Environments und (4) Technical Infrastructures.

Das AAL-System der vorliegenden Arbeit ist als Kombination aus Event-tracking und Learning Home anzusehen.

## 2.4.2 Kategorien von Algorithmen für AAL-Systeme

Damit ein AAL-System die Abweichungen von den ADL erkennen kann, unterscheidet die Literatur die in Abbildung 2.8 gelisteten Kategorien von Algorithmen.



**Abbildung 2.8:** Algorithmen der AAL-Systeme [i.A.a. Ras+13, S.581 ff.]

Die Algorithmen der AR (Activity Recognition) erkennen ein ungewöhnliches Verhalten durch Abgleich bewährter HAP (Human Activity Pattern) nach der Lernphase des Systems oder vor Nutzung des Systems mit den modellierten Aktivitäten z. B. nach A-priori-Wissen (s. Sektion 2.2). Die Subkategorien der AR sind Anhand der Verwendung von spezifischen Sensoren eingruppiert. Die Algorithmen der "Context Modeling" Kategorie beziehen neben Nutzerprofilen auch räumliche Informationen zu Aufbau und Struktur des Haushalts oder der Umgebung mit in den Verarbeitungsprozess ein. In der Kategorie der "Anomaly Detection" sind clusterbasierte, statistische und theoretische Methoden angesiedelt. Anwendung finden demnach sowohl regelbasierte und ähnlichkeitsbasierte Techniken als auch die zeitliche Zusammenhangserkennung [Ras+13, S. 581ff] - also eine Auswertung von Zeitserien.

Für die Algorithmen der "Location and Identify Identification" steht die feingranulare Positionsbestimmung der pflegebedürftigen Personen im Vordergrund. Dabei können in Abgrenzung zu GPS im Freien, die Sensoren der PIR oder RFID-Systeme auch eine Lokalisierung in Gebäuden ermöglichen. Beim "Planning" werden zuvor definierte Tagesabläufe dazu genutzt, um z. B. dem "Vergessen" bei demenzkranken Personen vorzubeugen [Ras+13, S. 583f].

Eine Trennung von Aktivitätserkennung (oder Kontextgenerierung, hier: Lernphase) und Verhaltensermittlung (oder Musterermittlung, hier: Erkennungsphase) ist zwar sinnvoll [Mun15, S.22], aber die Algorithmen bzw. Verfahren können nicht immer eindeutig einem spezifischen Einsatzort im Gesamtsystem zugeordnet werden. Der in Sektion 2.2 genannte Ansatz des "Aktivitäts-Mining" stellt die Aktivitäten nach der Lernphase als Ereignissequenzen in einer Baumstruktur dar. In der Erkennungsphase wird die zuvor erstellte Baumstruktur traversiert, um ein abweichendes Verhalten zu kennzeichnen: Die Erkennung und Ermittlung nutzen in diesem Verfahren gleiche programmatische Strukturen.

### 2.4.3 Existente AAL-Systeme

Der vorliegende Abschnitt betrachtet AAL-Systeme für die in Sektion 2.4.1 benannten Kategorien. Aufgrund der Vielzahl an Systemen wird nur auf öffentlich etablierte und in der Literatur verbreitete AAL-Systeme eingegangen. Die Systeme bzw. Projekte der "Smart Home"-Kategorie werden ausführlicher erörtert, da das AAL-System der vorliegenden Arbeit dieser Kategorie zugehörig ist. Die Projekte der "Mobile and Wearable Sensors"- und "Robotics"-Kategorie werden als zweitrangig eingestuft und sind nur namentlich genannt.

Das BMBF (Bundesministerium für Bildung und Forschung) förderte in den Jahren 2009-2012 exakt 17 AAL-Projekte mit einem Etat von insgesamt 45 Millionen Euro [For15]. Die Fraunhofer-Allianz AAL ist ein Zusammenschluss von elf Fraunhofer-Instituten, die an 50 Projekten im Kontext des AAL arbeiten [Liv15c]. Sechs dieser Fraunhofer-Projekte werden durch das BMBF gefördert: Daily Care Journal (DCJ), SAMDY, sens@home, SmartSenior, WebDa und WohnSelbst [Liv15b].

In [alni] werden insgesamt 24 Projekte für technische Assistenzsysteme präsentiert, wobei der Fokus der Kurzvorstellung der Projekte in der Beschreibung von Technik und Dienstleistungen, den Finanzierungskonzept und den "Nächsten Schritten" gesetzt ist.

#### Smart Home

(1) Das DCJ-Projekt beschreibt in der Phase der fachlichen Erhebung und Konzeption eine AAL-Plattform zur Unterstützung von Pflegenetzwerken für den Austausch von Pflegeinformationen. Ziel des Projektes ist es, die Pflegeprozesse zu optimieren und dabei Angehörige, Freunde und professionelle Dienstleister in einem gemeinsamen Pflegenetzwerk als Teil der koordinierten Pflege miteinzubeziehen. Die Dokumentation und der Austausch von Informationen zur Pflegeplanung, Pflegemaßnahmen und Wohlbefinden des Nutzers (Klient) sind anwendungsorientiert konzipiert [Woj+15, S.1 ff.]. Eine Einschätzung des Zustandes der pflegebedürftigen Person erfolgt in einem 4-6 wöchigen Rhythmus neben den AEDL (Aktivitäten und existentiellen Erfahrungen des Lebens) anhand der handschriftlich festgehaltenen Formulare, die durch die Nutzung des Eingabestiftes "ePEN" [Gmb15b] im Pflegenetzwerk digitalisiert werden können [Woj+15, S.3 f.]. Die Architektur des Projektes sieht vor, die aufgezeichneten Daten an externe Netzwerke weiterzuleiten. Eine Rechtevergabe und die Datensicherheit bei externen Zugriffen sind rudimentär behandelt [Woj+15, S.7]. Zwar ist in der Architektur für die "Context"-Komponente auch eine Kontexterkenkung, Zuordnung zu einem Kontextmodell und eine aktive Überwachung und Benachrichtigung für einen Hausnotruf geplant, aber hinsichtlich zu verwendbaren Algorithmen oder anwendbaren Verfahren nicht näher erläutert [Woj+15, S.4 f.].

(2) Das SAMDY (Sensorbasiertes adaptives Monitoringsystem für die Verhaltensanalyse von Senioren) - Projekt dient sowohl der Gewinnung von Informationen für den Pflege- und Betreuungsprozess, der frühzeitigen Erkennung von Verhaltensabweichungen als auch der Überwachung von Vitalparametern im Schlaf- und Wachverhalten der betroffenen Personen, um diesen weiterhin ein Leben in der eigenen Wohnung zu ermöglichen [Wyb+15, S.1]. Zum Einsatz kommen Bewegungsmelder, Magnetkontakte und ein Bettsensor-System [Wyb+15, S.2 f.] - durch den modularen Aufbau kann das System an den Assistenzbedarf des Individuums angepasst werden. Das Projekt ist in mehrere Teilvorhaben gegliedert, und wie in [Liv15a] gezeigt, auf unterster Ebene in "Bettsensor-Box" und "SAMDY-Homestation" zu trennen. Für das Teilvorhaben der "Bettsensor-Box" sind mobile Finsor-Bettsensormatten eingesetzt, deren gemessene Daten zur weiteren Verarbeitung über einen Wireless Access Point (WAP) an einen externen Server übermittelt werden [Sch15b, S.2 f.]:

*"Eine 'Echtzeit-Online-Darstellung', die eine direkte Überwachung der gescreenten Patienten ermöglicht hätte, war zu diesem Zeitpunkt bei Verwendung der Finsortechnik leider nicht realisierbar."*

Nach den Schlussfolgerungen des Teilvorhabens ist bei eingesetzter Hard- und Software eine "verlässliche" Interpretation des Schlafverhaltens nicht möglich, lediglich die Bewegungshäufigkeit und ein Aufstehen kann detektiert werden [Sch15b, S.1+6]:

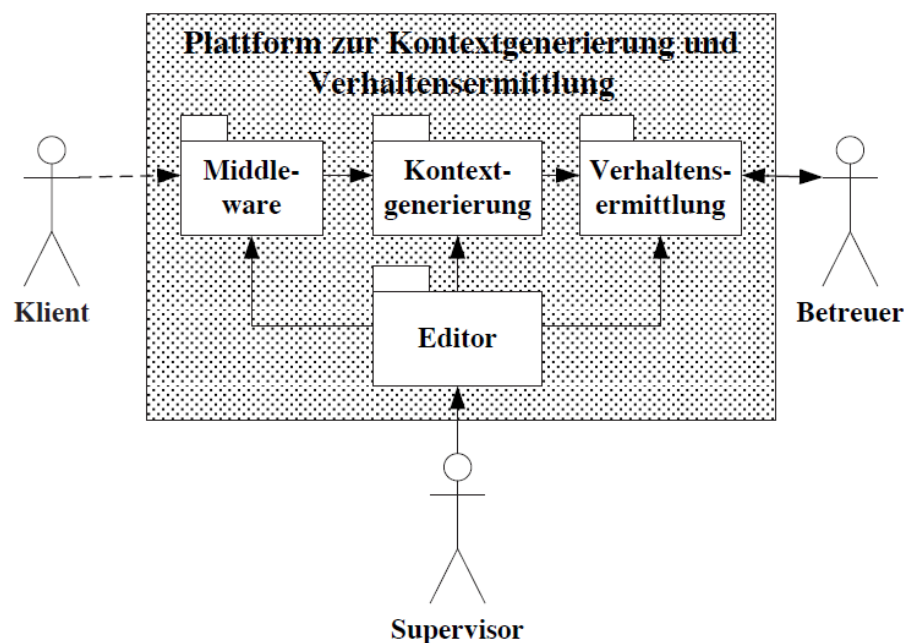
*Es "[...] bestand die Hoffnung, mittels eines solchen Sensorsystems eine kabellose Überwachung von Vitalparametern [...] zu erfassen, um so Notfallsituationen rasch erkennen zu können."*

Die Komponenten von "Erkennung, "Ereignisse" und "Verhaltensmonitoring und Analyseplattform" des Teilvorhabens "SAMDY-Homestation" [Liv15a] sind in [Mun15] beschrieben. Die dargestellte "Vorhersage"-Komponente ist "[...] bisher nicht implementiert worden.[...] Das System arbeitet retrospektiv." [Mun15, S.171], d.h. die Homestation zeichnet auf und analysiert die Daten hinsichtlich ausgeführter ADL, um dem Betreuer oder Angehörigen eine tägliche Verhaltensermittlung zu liefern [Mun15, S.183 f.]. Eine Echtzeitauswertung und Alarmierung bei Notsituationen ist während der Systemnutzung demnach nicht vorgesehen bzw. realisiert [Mun15, S.111] - lediglich ein visualisierter Ampelzustand, bei der täglichen Ermittlung, weist auf abweichende Verhaltensmuster hin.

*"Eine Erweiterung in diese Richtung mit zusätzlicher Alarmierungsfunktion in Echtzeit bei Abweichungen vom Modell wäre aber durchaus realisierbar." [Mun15, S.171]*

Für den Vorgang der Kontextgenerierung und abschließender Verhaltensermittlung werden vor Inbetriebnahme der Homestation die ADL, welche bei einem Interview mit examinieren Pflegekräften analysiert wurden [Mun15, S.61], mit der Tasknotation CTT modelliert und für die

weitere Vorgehensweise automatisch in PN transformiert [Mun15, S.38], obwohl der Autor erwähnt, das PN nicht "optimal geeignet" sind und bereits in der Bewertung der Grundlagen auf kausale Netze (C-Netze) hinweist [Mun15, S.95]. Für die Verifikation der modellierten Aktivitäten - also der PN - wird die quelloffene Software TAPAAL eingesetzt [Mun15, S.213]. Als Ergebnis der Kontextgenerierung werden die Aktivitätsinstanzen im Format der MXML (Mining eXtensible Markup Language) festgehalten [Mun15, S.234]. Bei der Inbetriebnahme der Homestation werden für das Aufzeichnen der Daten, die Sensoren aus allen drei Bereichen der O-C-M-Typisierung verwendet [Mun15, S.174]. Die Middleware der Homestation erzeugt aus den Sensorrohdaten logische Ereignisse die mit den aus Sektion 2.2 beschriebenen Ereignistypen gekennzeichnet werden [Mun15, S.214 f.]. Anschließend wird auf Basis des MXML-Formates mithilfe des ProM (Process Mining Framework) [Gro+15] und mit dem Verfahren des TSM (Transition System Miner) [Mun15, S.290] auf vorherrschende Aktivitäten und Abweichungen hin analysiert [Mun15, S.158]. Ein globaler Time-out-Mechanismus sorgt dafür, dass die Aktivitäten, die ihre maximale Ausführungszeit überschritten haben, von anderen Aktivitätsinstanzen unterschieden werden können [Mun15, S.213].



**Abbildung 2.9:** Gesamtsystem - SAMDY [Mun15, S.145]

Für die tägliche Einsicht der Daten bietet SAMDY, wie in Abbildung 2.9 dargestellt, einen Editor an, der von einem Supervisor bedient wird. Neben dem Erstellen und Bearbeiten von Profilen und der Auswahl von ADL, können weiterführende personenbezogene Daten im System hinterlegt werden (z. B. Name, Geschlecht, Geburtsdatum oder Auswahl von bekannten Krankheiten) [Mun15, S.153]. Der Betreuer der pflegebedürftigen Person legt dabei fest, welche Aktivitäten überwacht werden sollen [Mun15, S.183]. Für die Verhaltensanalyse bietet der Editor eine zeitorientierte Aufstellung der ausgewählten Aktivitäten (Intervall >15min), die



mit "Start" bzw. "Complete" gekennzeichnet sind [Mun15, S.240]. Dem PAP (Programmablaufplan) der Verhaltensermittlung mit dem TSM ist zu entnehmen, dass für die Aufstellung überprüft wird, ob die derzeitig betrachtete Aktivitätssequenz "neu" ist und einem Normalverhalten zugeschrieben werden kann, oder als abweichendes Verhalten gewertet werden muss [Mun15, S.216].

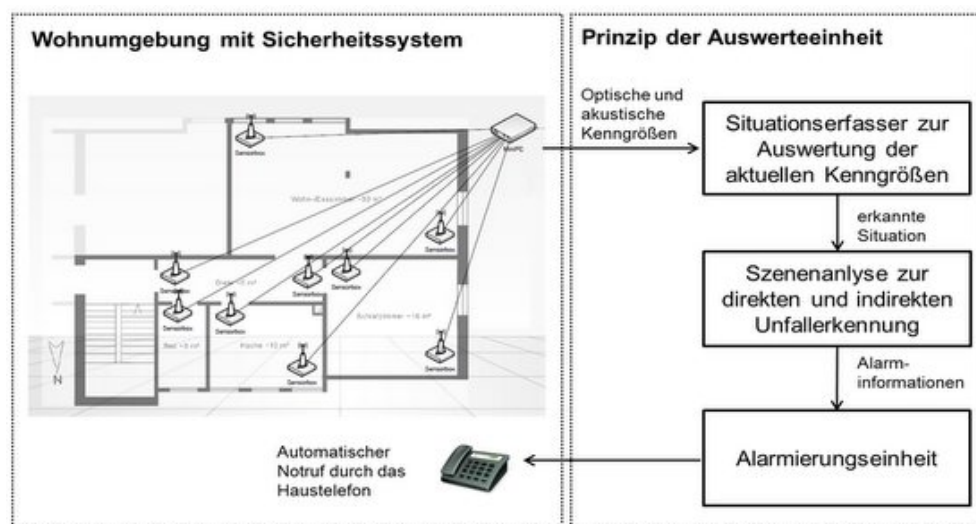
Die in SAMDY angewandte Kombination aus Kontextgenerierung und Verhaltensermittlung stellt der Autor als Novum dar [Mun15, S.27] und bei Anwendung der PN und des TSM, nutzt das System die Algorithmen aus den Kategorien der "Activity Recognition" und "Context Modeling".

**(3)** Das JUTTA (JUSt-in-Time Assistance) - Projekt hat sich, wie auch bei oben genannten Projekten, zum Ziel gesetzt, den Assistenzbedarf für ein unabhängiges und möglichst langes Leben in der eigenen Wohnung zu decken. Für die Realisierung der Zielsetzung baut JUTTA auf der SOPHIA- und Vitaphone-Basisstation, der Fraunhofer-inHaus-Homestation [AAL15b] und bestehender Mikrosystemtechnik auf [Shi+12, S.325]. Für die Aktivitätserkennung und Vitaldatenerfassung werden die Sensoren des sogenannten "Assistenzkoffers" und der einzelnen Basisstationen eingesetzt: Neben einem EKG-Messgerät, den Bewegungsmeldern und Wassersensoren werden auch Drucksensoren und Türkontakte verwendet [Shi+12, S.334]. Für eine rechtzeitige Reaktion werden dem Betreuer, nach Auswertung der Daten auf dem Server der Pflegezentrale, die individuellen Gesundheitszustände der Person als Ampelzustände (Rot, Gelb, Grün) visualisiert und in der Notsituation der nächstgelegene Mitarbeiter mittels GPS zur Person geführt [Shi+12, S.332]. Zwar treffen die beiden Basisstationen die eigenverantwortlichen Entscheidungen und Maßnahmen während der Verhaltensanalyse, in der vorliegenden Literatur sind jedoch keine genauen Verfahren oder Algorithmen genannt. Als Ergebnis der Evaluation ist aus Sicht der Pflegekräfte eine Vorkonfiguration mit den wohnungsspezifischen Parametern für die einzelnen Systemkomponenten notwendig [Shi+12, S.341].

**(4)** Das sensorbasierte Sicherheitssystem "safe@home" ist aus der Entwicklung des BMBF-geförderten "sens@home"-Projektes entstanden und detektiert sowohl aktive als auch passive Ausprägungen von Notfällen im privaten Wohnumfeld [Lin+15; IPA15]. Für die Reaktion auf evtl. Notsituationen ist keine Nutzerinteraktion erforderlich. In der deckenmontierten Notfall-Sensorbox (kabellos oder kabelgebunden) sind akustische und optische Sensoren verbaut. Für den Einsatz bei schlechten Lichtverhältnissen bzw. Dunkelheit ist die Sensorbox zudem mit einer aktiven "Low-Power-IR-Beleuchtung" ausgestattet [Lin+15, S.1 f.]. Wie in Wohnungsumgebung der Abbildung 2.10 zu sehen, sind, für die Prävention von möglichen Einschränkungen der LOS beim Einsatz von optischen Sensoren, einige Räume mit zwei Notfall-Sensorboxen ausgestattet. Die Alarmierungseinheit (Mini-PC oder auch Carebox) kommuniziert mit und überprüft regelmäßig den Status der Notfall-Sensorboxen [Lin+15, S.3]. Es ist anzunehmen, dass die Carebox als zentraler Leitrechner dient und neben der Alarmierungsfunktion auch die Hauptanalyse der gemessenen Daten vornimmt [Lin+15, S.4]:

*"Sämtliche Auswertungen finden direkt in den Sensorboxen oder im übergeordneten Alarmierungsrechner statt."*

Da die zugrundeliegenden Algorithmen für die Auswertung nicht genannt sind, kann "safe@home" in seinem Anwendungskontext als Kombination von "Vision-Based AR" und "Location and Identify Identification" angesehen werden. Über die Dimensionen der Carebox ist in der vorliegenden Literatur keine Aussage getroffen - sie ist an das WAN (Wide Area Network) angebunden. Eine Auswertung von akustischen Kenngrößen kann Hilfeschreie identifizieren. Die Verwertung von optischen Kenngrößen erkennt zuvor definierte Notgesten, Sturzphasen oder eine Reglosigkeit der betroffenen Person. Welche Sturzphasen als Notfall gewertet werden und ob ein einfaches "zu Knie gehen" (vgl. Persona Hildegard) auch detektiert wird, bleibt unbeantwortet.



**Abbildung 2.10:** Wohnungsumgebung und Auswertungsprinzipien - safe@home [AAL15d]

Um FPs auszuschließen geht das System, bevor es einen Notfall ausruft, in den Voralarmzustand und verifiziert einen notwendigen Alarm durch einen Anruf bei der betroffenen Person: Das Nichtbeantworten kennzeichnet den Notfall [Lin+15, S.2]. Obwohl zuvor definierte Gesten einen Notfall ausrufen können, setzt das Projekt nicht auf Gesten, die das System aus dem Voralarmzustand in den "Erkennungszustand" zurückführen. Auch wenn die gemessenen Bild- und Tondaten nicht außerhalb der Wohnung und Aktivitätsprofile grundsätzlich nicht gespeichert werden, verweist der Autor auf eine hohe Akzeptanz der Probanden und des Betreuungspersonals beim Einsatz des Systems (inkl. Kameras und Mikrofone) [Lin+15, S.4] - Der notwendige Schutz der Privatsphäre (Sektion 2.3.1) ist dennoch zu beachten.

**(5)** Das PERSONA (PERceptive Spaces promoting iNdependent Aging) - Projekt beschreibt ein Rahmenwerk um "Context-Awareness" in verteilten Systemen zu unterstützen. Das Rahmenwerk bietet konzeptuelle Lösungsansätze, um die Hürde der Kontextmodellierung bzw.

Kontextgenerierung zu überwinden und eine dynamische Kopplung der Systemkomponenten zu ermöglichen [FV+15, S.91]. Eine Anwendung von PERSONA ist, aufgrund der einzelnen physischen Knoten mit eigenen logischen Softwareeinheiten im Open Distributed System (ODS), in den Aml-Umgebungen zu sehen. In ihrem Konzept versuchen die Autoren zwei grundlegende Aspekte zu berücksichtigen: (1) Die Akquisition und das Veröffentlichen der Kontextdaten (z. B. Menschen, Lokationen und Endgeräte). (2) Die Auswahl von geeigneten Verfahren für das Interpretieren der Kontextdaten und die Reaktion bei Kontextänderung [FV+15, S.92]. Wie auch bei den Architekturen der oben erörterten Projekte (vgl. SAMDY, JUTTA) ist die "Middleware" als Kernkomponente von PERSONA zu betrachten. Die "Middleware" verfügt über vier Bussysteme, mit denen eine nahtlose Konnektivität und Interoperabilität der physischen Knoten des ODS erzeugt wird [FV+15, S.95 f.]: Input-, Output-, Context- und Service-Bus. Implementiert ist die "Middleware" als Satz von "Bundles" des OSGi-Framework (Softwareplattform mit Komponentenmodell auf Basis von Java) und in drei logische Schichten (Bottom-Up-Sicht) gruppiert [FV+15, S.96 f.]:

- 1 *Abstract Connection Layer (ACL)*: In der ACL wird die peer-to-peer Konnektivität der Instanzen geregelt. Für die Kommunikation der Instanzen ist die "P2PConnector"-Schnittstelle und für das "Bridging" der "PeerDiscoveryListener" der OSGi-Service Plattform implementiert. Neben Universal Plug and Play (UPnP), Bluetooth und R-OSGi sind keine weiteren Kommunikationsstandards für die anstehende Realisierung angedacht.
- 2 *SodaPop Layer*: Als Kernelement der "Middleware" agiert diese Schicht als Discovery-Service und basiert auf den event- und call-basierten Buskonzepten für die Kommunikation der Peer-Instanzen.
- 3 *PERSONA-specific Layer*: Durch die Verwendung des RDF (Resource Description Framework) - Serialisierers erweitert die PERSONA-specific Schicht die vorherigen Schichten um die vier genannten Bussysteme.

Die Kommunikationsablauf zwischen dem Context-Bus und Service-Bus - genauer "Context-Consumer" und "Context-Provider" - unterstützt die Context-Awareness im Rahmenwerk [FV+15, S.103 f.].

**(6)** Das SMILEY (SMart and Independent Living for the ElderLY) - Projekt soll Probleme von älteren Personen im Alltag durch die Unterstützung von Technologien bewältigen. Das Projekt ist in zwei Teilvorhaben einzuteilen und kann deshalb auch nicht eindeutig einer Kategorie von AAL-Systemen zugeordnet werden, sondern ist eine Kombination von "Smart Home" und "Mobile and Wearable Sensors" System. Das erste Teilvorhaben beschreibt die prototypische Entwicklung von Applikationen für die mobilen Betriebssysteme Android und iOS. Der Fokus der Entwicklung liegt in der Visualisierung und Zugang zu Informationen für die eigene Gesundheit, der Umgebung, den Kontakten und Erinnerungen. Für die Realisierung der Nutzungsschnittstelle verfolgen die Autoren den UCD (User Centered Design) - Ansatz [vgl. SI15, S.24 ff.; Sen13, S.227]. Damit ältere Personen den Umgang mit der Applikation einfach erler-

nen können, wird für die Teilrealisierung ein Recommender-System eingesetzt: Dem Benutzer werden individuelle Interaktionen mit dem System empfohlen [SI15, S.81]. Das zweite Teilvorhaben ist die "Mikrosystemtechnische Unterstützung von AAL-Dienstleistungsnetzwerken" und umfasst in den sechs Arbeitsanforderungen auch eine Entwicklung für die präventive Sturzerkennung und die Integration von hausinterner Sensorik [SI15, S.40]. Der Schwerpunkt in der Entwicklung des Sturzmonitoring Systems ist jedoch nur die Detektion der gestürzten Person:

*"Ziel dieser Entwicklung ist es nicht, den Sturz an sich zu erkennen, da der Sturz per Definition sehr abstrakt sein kann."* [SI15, S.42]

Da die Autoren den Einsatz von "bildgebenen Sensoren" auch als Eingriff in den Schutz der Privatsphäre deuten, wird für das Sturzmonitoring zwischen dem Passiv-Infrarot und dem Aktiv-Infrarot-Verfahren (Thermopile) unterschieden. Es wird darauf hingewiesen, dass das Passiv-Infrarot-Verfahren langsam bewegende Personen nicht zuverlässig erkennt und das Thermopile-Verfahren deshalb bevorzugt einzusetzen ist [SI15, S.42]. Da die Idee das eine nicht erkannte Bewegung das System, wie beim sens@home-Projekt, in einen Voralarmzustand versetzt, nicht berücksichtigt wird, ist ein grundsätzlicher Ausschluss von passiven Bewegungsmeldern für die Notfallerkennung nicht plausibel. Eine 1,2m hohe horizontale Wandmontage der Sensoren könnte Störeinflüsse, durch eine vorhandene Fußbodenheizung, entgegenwirken [SI15, S.45]. Bei der Implementierung der Detektionsalgorithmen sollen die alltäglichen Verhaltensmuster und Bewegungen von Haustieren erkannt und von einem Sturz abgrenzt werden [SI15, S.52 f.]. Weitere Informationen zu den Algorithmen sind in [SI15] nicht angegeben.

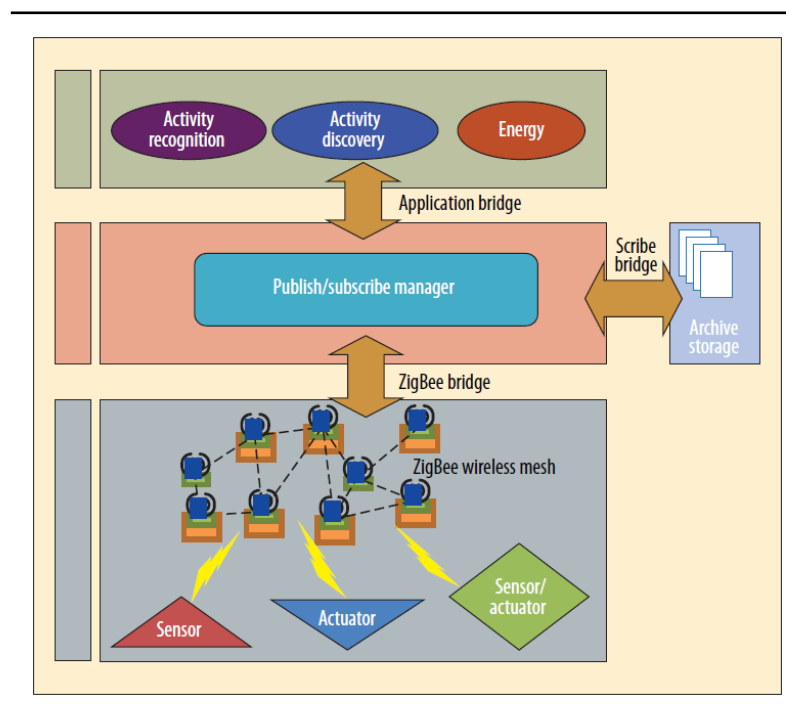
**(7)** Das im SELBST-Projekt entwickelte System ist nicht auf die Erkennung von Notsituationen ausgelegt [Ham+13, S.157]. Die Fokus des Systems liegt in Verbesserung der Kommunikation der hilfebedürftigen Person mit seinen Angehörigen und in der rudimentären Erkennung von Abweichungen für einen routinierten Tagesablauf [Ham+13, S.158]. Für die realisierten Anwendungen (z.B. Aktivität durch Geräte-ein und -abschaltung, Herdstatus, Herdabschaltung oder Erinnerung zur Tabletteneinnahme) werden die Sensoren der EnOcean-Technologie eingesetzt [Ham+13, S.158 ff.].

**(8)** Aus dem vom BMBF geförderten "SensFloor"-Projekt entstehen neben der "SensFloor"- auch die "SensFloor-Matten"- und "SensBed"-Technologie, die auf kapazitiven Näherungssensoren basieren [vgl. Ste+13b, S.517; Ste+13a, S.521 f.; Ste+12, S.2]. Die "SensFloor"-Technologie ist flächendeckend unter alle Bodenbeläge der Wohnung zu verlegen und ermöglicht dadurch ein Aktivitätsmonitoring, eine Sturzdetektion, eine Energiesparfunktion (An- und Abschalten von Geräten) oder auch einen Einbruchalarm. Die "SensFloor-Matten"-Technologie kann auch für eine Zugangskontrolle oder für die Detektion des Aufstehen aus dem Bett genutzt werden [Ste+13b, S.517]. Mit der "SensBed"-Technologie können Schlafbewegungen aufgezeichnet werden. Die Diagnostik von Schlafstörungen und z. B. Herzrhythmus oder Atmungsbewegung wie bei der PSG (Polysomnographie) ist derzeit noch nicht möglich [Ste+13a,

S.523 f.]. Der Einsatz von "SensFloor" ist während der Wohnungsplanung vorzusehen, eine nachträgliche Installation in ausgebaute Wohnungen ist deshalb mit größeren Kosten verbunden. "SensFloor-Matten" und "SensBed" könnten, in Kombination mit Systemen die Bewegungsmelderverwenden, eine Bereicherung für das Verhaltensmonitoring darstellen.

(9) Das Ziel des PareSense-Projektes ist die Entwicklung eines Sprachsteuerungssystem, mit dem die Gebäude- und Sicherheitstechnik von pflegebedürftigen Personen gesteuert werden kann [AAL15c].

(10) Das CASAS-Projekt der Washington State University entwickelt ein SHiB (Smart Home in a Box) - System, das keiner Kundenanpassung oder eines Trainings bedarf und aufgrund der umgesetzten "Activity-Discovery" (Aktivitätserforschung) und AR, zusammen mit den initial gelabelten Hardwarekomponenten, ein System der "Activity-aware"-Applikationen bildet [Coo+13, S.62 ff.]. CASAS nutzt neben Infrarot-, Licht-, Tür-, Relais- und Temperatur-Sensoren einen Rechner (ITX-Formfaktor) auf dem die "Middleware", die Datenbank und alle weiteren Softwarekomponenten abgelegt sind [Coo+13, S.64].



**Abbildung 2.11:** CASAS: Smart Home Komponenten und Architektur [Coo+13, S.63]

Wie in Abbildung 2.11 dargestellt, sind die Aktoren und Sensoren auf der physischen Schicht, die über ein ZigBee-Netz mit dem in darüberliegenden "Middleware"-Schicht befindlichen "Publish-Subscribe"-Manager kommunizieren, angesiedelt. Für die Kommunikation mit dem Manager wird das XMPP (Extensible Messaging and Presence Protocol) eingesetzt. Neben der Funktion als zentrale Broadcast-Einheit, für das Senden von Nachrichten an Aktoren und Empfangen von Nachrichten von Sensoren, stellt die "Middleware" die Funktionen für

die Transformation der gemessenen Daten bereit (z. B. Hinzufügen von Zeitstempeln oder das eindeutige Adressieren durch Vergabe von Kennungen). Auf der "Application"-Schicht sind die Softwarekomponenten der Aktivitätserkennung und Verhaltensermittlung anzutreffen [Coo+13, S.62 f.].

Für eine "out of the box" Aktivitätserkennung sind zum einen die Sensoren bereits vorkonfiguriert und mit einem Schriftzug für ihren geplanten Einsatzort gekennzeichnet und zum anderen wurden die gemessenen Sensordaten von insgesamt 18 separaten Smart Homes mit einbezogen. Für die Zuordnung der Sensor- bzw. Ereignissequenzen zu einer Aktivität (am Einsatzort) verwendet das CASAS-Team die  $k$ -häufigste Beschriftung der neuesten Sensordaten in einem definierten Zeitfenster ( $k$ -event sensor window) [Coo+13, S.64 f.]. Für ein Verhaltensmonitoring, während der Systemnutzung, verwendet CASAS eine Support Vector Machine (SVM) [Ert09, S.277 ff.]. Andere Verfahren, wie z. B. der Bayes-Klassifikator [Keo16], ein HMM oder Conditional Random Fields (CRFs) [Laf+01], sind vom CASAS-Team getestet worden, der Vergleich zu einer SVM ist jedoch in einem Satz abgetan:

*"However, we found that SVMs achieve consistently stronger performance than other approaches."* [Coo+13, S.64]

DTs bleiben bei den Tests unberücksichtigt. Als Eingabe für die Klassifikatoren dient ein Datenpunkt, der einer Event-Sequenz entspricht. Die einzelnen Events werden als Vektor angegeben und sowohl mit Start- und Endzeitpunkt als auch Zeitspanne und die Anzahl der aufgetretenen sensor-spezifischen Events für das definierte Zeitfenster angereichert. Die Beschriftung des zuletzt auftretenden Events - der Zeitspanne - dient der Zuordnung zu einer eindeutigen Aktivität [Coo+13, S.64 f.].

**(11)** Das Team des MavHome (Managing an adaptive versatile Home) - Projektes, der Universität von Texas (Arlington), forscht an einem agentenbasierten Smart Home System für die Gesundheitsüberwachung und Unterstützung der ADL im häuslichen Umfeld [Coo15, S.1]. Neben dem Identifizieren von Tendenzen der Lebensführung und dem Entwerfen von Erinnerungsfunktionen, steht die Anomaliedetektion der ADL für die Reaktion in Notsituationen und die Verhaltensvorhersage für das Assistieren bei Aktivitäten (z. B. Licht an- und ausschalten) im Mittelpunkt [Coo15, S.1 ff.]. Das MavHome-System ist in sechs Schichten eingeteilt (Bottom-Up-Sicht) [Coo15, S.4 ff.]:

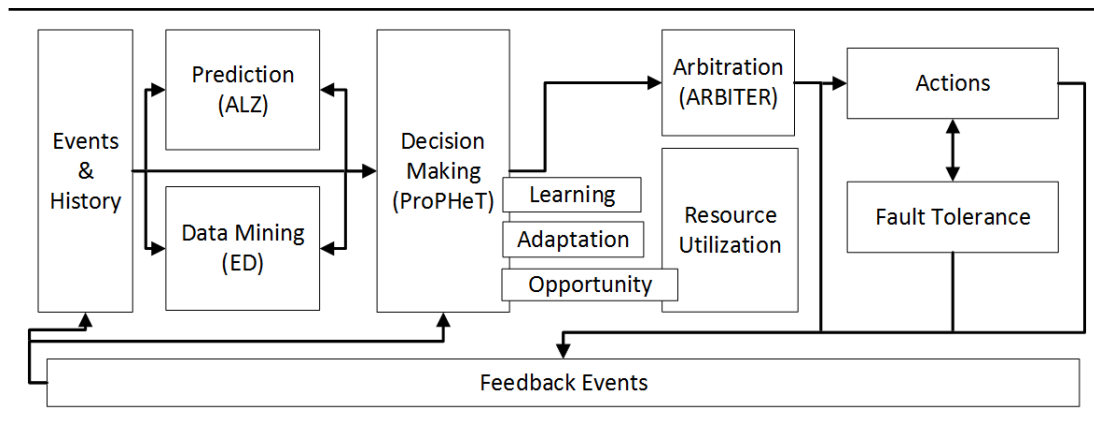
- 1 *Physical Components*: Auf der physischen Schicht sind alle Sensoren und Aktoren angesiedelt. Eingesetzt werden Umgebungssensoren für das Messen der Temperatur und Luftfeuchtigkeit im Raum, Reed- bzw. Magnetschalter für Tür- und Fensterkontakte, Lichtquellen, Rauch-, Gas- und Bewegungsmelder. Zur Kommunikation mit der darüber liegenden Schicht, werden die Hardwareschnittstellen der Dongles, "Super Slaves", "Motion Masters", "Mini-Blind Master" und des "Masters" genutzt. Für die Kommunikation

wird das Netzwerkprotokoll X10 eingesetzt, bei den "Masters" und "Slaves" ist von X10-Modulen auszugehen: Ein redundanter Netzwerkaufbau mit drahtlosen und drahtgebundenen Hardwarekomponenten. In dem aufgestellten Szenario einer Zweizimmerwohnung werden insgesamt 25 Controller verwendet [Coo15, S.5].

- 2 *Computer Interface*: Mit den Funktionen des OS (Operating System) wird der Übergang von Hard- und Softwareschnittstellen realisiert.
- 3 *Logical Interface*: Auf dieser Schicht befinden sich logische Proxies. Es ist anzunehmen, dass es sich bei den Proxies um eine Realisierung des Strukturmusters aus der Softwareentwicklung handelt, damit die Steuerung delegiert werden kann.
- 4 *Middleware*: Für eine objektorientierte Middleware-Spezifikation wird CORBA (Common Object Request Broker Architecture) eingesetzt. Für eine Namensauflösung wird ZeroConf (Zero Configuration Networking) eingesetzt. Die Funktionalität von CORBA und ZeroConf kann auf der fünften und sechsten Schicht genutzt werden.
- 5 *Services*: Auf der fünften Schicht sind unter anderem die Dienste der Verhaltensvorhersage, der Datenbank und des Data-Minings vorzufinden.
- 6 *Applications*: Die Applikationsschicht verwaltet die Module die eine Entscheidung treffen (Decision Maker), den Handlungsprozess des System begutachten (Arbitration) und auf ein Fehlverhalten des Systems (Fault Tolerance) reagieren.

Für das Detektieren der Anomalien (Verhaltensabweichungen) der gemessenen Daten ist im MavHome-System der ED (Episode Discovery) Algorithmus implementiert [Coo15, S.5 f.]. Der ED-Algorithmus [Hei+03] dient dem Erkennen von sequenziellen Mustern (Episoden von Events) und ist vorzugsweise dort anzuwenden, wo der Datenbestand aus zeitlich geordneten Transaktionen (Events) besteht. Ein Event ist ein Triplett aus der Zeit des Auftretens, der Kennung des Sensors und die Werteänderung. Als Eingabe erhält der Algorithmus einen kontinuierlichen Strom an Events, der im ersten Schritt in maximale Episoden partitioniert wird. Die Episoden werden wiederum in eigenständige Untermengen "Itemsets" geteilt, für die dann ein signifikantes Event zusammen mit weiteren Kenngrößen errechnet wird. Eine Evaluierung von nennenswerten maximalen Episoden findet anhand der signifikanten Events der Itemsets statt. Die Kenngrößen für die Evaluierung werden mit dem Prinzip der Minimum Description Length (MDL) [Ris98, S.80] berechnet, d.h. es werden die Eventsequenzen gesucht, die eine gute Balance zwischen Anzahl ihrer Vorkommnisse (Frequenz) und Sequenzlänge besitzen. Die Itemsets, die am wenigsten Abweichungen in ihrer Regularität und somit nur einen geringen Teil an zusätzlichen Events beinhalten, sind ausschlaggebend für die QoS der Aktivitätserkennung [Coo15, S.6 f.].

Die Abbildung 2.12 zeigt die schichtenübergreifende Kommunikation der Softwarekomponenten. In Abgrenzung zu den "Feedback Events", die zusammen mit den "Actions" für eine aktive bzw. passive Interaktion mit dem betroffenen Person und Sensoren stehen, stellt die Komponente "Events & History" den aktuellen persistierten Datenbestand in den Datenbank



**Abbildung 2.12:** Genutzte Techniken im MavHome-System [i.A.a. Co015, S.6]

dar. Nachdem der ED-Algorithmus angewendet wurde, kann der ALZ (Active LeZi) - Algorithmus, eine Erweiterung des LZ78 [Ziv+78], die Aktivitäten aus den Mustern vorhersagen. Der ALZ-Algorithmus iteriert über die Symbolketten, welche die Eventsequenzen repräsentieren, und speichert diese in einem Baum ab. Die Wahrscheinlichkeiten für die Knoten des Baumes (Symbole der Events) werden aus einer Kombination von Frequenz und weiteren Kontextgrößen errechnet. Für die Aktivitätsvorhersage arbeitet der ALZ-Algorithmus auf einer  $n$ -langen Symbolkette und interpretiert das  $n+1$ -Symbol mit der höchsten Wahrscheinlichkeit als Aktivität, die die Person als nächsten ausführen wird.

Das MavHome-System verfolgt den agentenbasierten Ansatz des negativen bestärkenden Lernens (negative Reinforcement Learning) [Coo15, S.8 ff.], d.h. wenn das System eine künftige Handlung der betroffenen Person vorhersagt und dementsprechend unterstützende Maßnahmen einleitet, die Person diese Maßnahme aber widerruft, wird die Vorhersage negativ bewertet: Für die Modellierung der Aktivitäten arbeitet der "ProPHeT" mit den Sequenzen des ED-Algorithmus und erzeugt ein Markov-Modell, wobei die Knoten auf unterster Hierarchieebene, die einzelnen Events darstellen. Das "Decision Making" des "ProPHeT"s wird durch einen MDP (Markov Decision Prozess) realisiert. Der "ARBITER" sieht Sicherheit- und Komfortrichtlinien vor, die neben dem Widerruf der Person dafür sorgen, dass die vom "ProPHeT" getroffene Entscheidung nicht ausgeführt wird [Coo15, S.8 ff.].

**(12)** Für das HIS (Habitat Intelligent pour la Sante) - Projekt steht die Überwachung der Aktivitäten von Krankenhauspatienten im Vordergrund und ist deshalb für 1-2 Zimmerwohnungen ausgelegt. Betrachtet und evaluiert werden die ADL der Mobilität, des Ausscheidens und der Hygiene [LeB+06, S.92+98]. Zum Einsatz kommen Türkontakte und Infrarotsensoren. In einem Raum sind die Infrarotsensoren redundant verbaut und leuchten nur das Teilgebiet des Raumes, welches maßgeblich für eine ADL steht (z. B. auf das Waschbecken oder die Toilette, die Sitzgruppe im Wohnbereich oder auf das Bett gerichtet), aus. Der Zyklus der Aktivitäten des Patienten wird in einem endlichen Automaten (Finite State Machine (FSM) [Art62]) festgehalten: Die Räume stellen die Zustände und die Sensorevents die Transaktionen



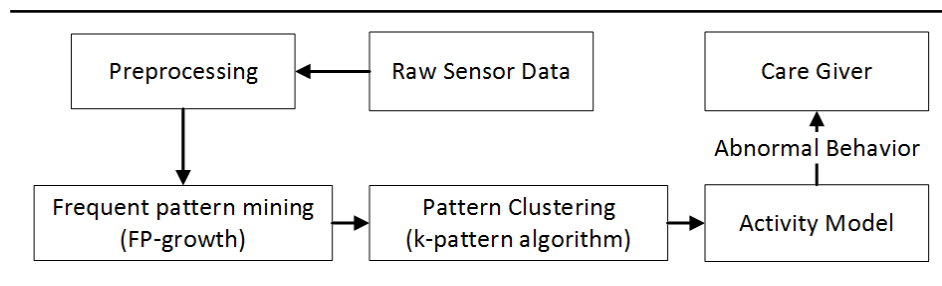
dar [LeB+06, S.92 f.]. Die gemessenen Events werden in einer Datei in der XML (Extensible Markup Language) persistiert und zur weiteren Verarbeitung in die "MatLab"-Software importiert [LeB+06, S.93 ff.]. Eine feingranulare Auswertung von komplexen Aktivitäten mit einer hohen Anzahl von Teilaktivitäten ist bei dem genutzten Verfahren nicht angedacht.

**(13)** In [Bou+15, S.11953 f.] ist ein Ansatz für ein unüberwachtes Lernen (unsupervised Learning) für das Erkennen und die Vorhersage von Nutzeraktivitäten in Smart Homes erörtert. Der Ansatz ist eine Kombination aus dem  $k$ -pattern Clustering-Algorithmus [Suk+15, S.119] und einem ATR (Allen's Temporal Relations) basierten Artificial Neural Network (ANN). In den verwandten Arbeiten ist erwähnt, dass die Self-Organizing Map (SOM) bei ansteigender Anzahl von Clustern und bei einem großen Datenbestand zu schlechten Resultaten in der Performance neigen. Die Effizienz des Expectation-Maximization (EM)-Algorithmus, den der Autor als einfach und schnell tituliert, ist von der Anzahl der Eingabetypen und Iterationen abhängig. Der gewählte  $k$ -pattern Clustering-Algorithmus soll neben diesen Defiziten zudem auch mit im Datenbestand vorhandenen Rauschdaten umgehen bzw. diese extrahieren können [Bou+15, S.11956 f.]. Letztendlich suggerieren die Autoren den Einsatz der beiden Algorithmen als beste Kombination mit der höchsten Genauigkeit bei der Aktivitätserkennung und der Detektion von unerwarteten Verhalten für das Anwendungsfeld der "Smart Home"-Umgebungen, weisen jedoch gleichzeitig auf die ineffiziente Laufzeit der Algorithmen hin: Wird zudem der "J48"-Decision Tree (DT) eingesetzt, kann die Laufzeit, bei Auswertung eines zweiwöchigen Datensatzes, von 314s auf 25s reduziert werden. Im Vergleich zum NB (Naive Bayes) - Klassifikator (0,03s) und einem HMM (0,089) dennoch deutlich ineffizienter [Bou+15, S.11968 f.]. Im Vergleich zu einer Aktivitätserkennung mit HMM, NB und dem C4.5-Algorithmus zeigt der konzipierte Ansatz der Autoren für jede Aktivität die höchste Genauigkeit.

Wie in Abbildung 2.13 zu sehen, sind vor Anwendung des  $k$ -pattern Clustering-Algorithmus drei Stationen zu erreichen bzw. zwei Prozesse durchzuführen [Bou+15, S.11958 f.]:

- (1) Die rohen Sensordaten, deren Entität aus dem Vierer-Tupel (Datum, Zeit, Sensor-ID und Sensor-Status) bestehen, werden im "Preprocessing" mit dem SAX (Symbolic Aggregate appRoXimation) - Algorithmus von der Zeitserie in eine symbolische Darstellung transformiert, d.h. aus der Menge der Tupel wird eine Symbolkette erzeugt.
- (2) Im zweiten Prozess wird der FP-growth Algorithmus auf den Datenbestand der Symbolketten angewandt: Es werden nur solche Muster wahrgenommen, deren Anzahl ihrer Vorkommnisse einen Grenzwert überschreitet. Damit der Algorithmus möglichst effizient die Symbolketten iterieren kann, wird ein Präfixbaum eingesetzt.

Die durch den FP-growth Algorithmus erkannten Muster, dienen dem  $k$ -pattern Algorithmus, neben der Anzahl von Clustern und dem Cluster-Center, als Eingabe und errechnet bzw. gruppiert ähnliche Muster in Clustern: Initial ist die Clusteranzahl auf null gesetzt. Um ein Muster einem Cluster zuzuordnen, wird überprüft, ob das Muster den cluster-spezifischen Grenzwert



**Abbildung 2.13:** Architektur für das Aufspüren von Aktivitäten mit dem *k*-pattern Algorithmus [i.A.a. Bou+15, S.11959]

unterschreitet und errechnet das Cluster-Center neu. Wird der Grenzwert überschritten, gehört das Muster zu einem neuen bzw. anderen Cluster [Bou+15, S.11959 ff.].

Um aus den entstandenen Clustern ein "Activity Model" zu erzeugen und dann während einer Systemnutzung auf abweichendes Verhalten zu schließen oder Verhalten vorherzusagen, nutzen die Autoren einen "J48"-DT in Verbindung mit dem ANN, das auf den ATR basiert. Durch die Verwendung des "J48"-DT können Fehler der Clusterbildung, irrelevante oder redundante Eigenschaften reduziert werden, um der schlechten Laufzeiteffizienz entgegenzuwirken. Die Algebra der ATR kann die Nebenläufigkeit von Aktivitäten abbilden [Bou+15, S.11963 ff.].

Die Autoren erwähnen, dass die Daten für die Auswertung des gewählten Ansatzes dem CASAS-Projekt entnommen sind. Die Frage, ob eine Realisierung in einer realen Smart Home Umgebung getestet wurde, bleibt unbeantwortet [Bou+15, S.11966] und unklar ist demnach auch, inwieweit ein reales System nach der Persistierung der gemessenen Sensordaten in angemessener Zeit auf eine Verhaltensabweichung reagieren kann. Zwar wird in der Abbildung 2.13 die Pflegekraft bzw. der pflegende Angehörige dargestellt, dennoch ist nicht erörtert, wie eine Reaktion auf das abnormale Verhalten zu gestalten ist - was letztendlich auch auf ein fehlendes Feldexperiment zurückzuführen ist.

## Weitere Smart Home

- Der AGALZ (Autonomous aGent for monitoring ALZheimer patients) ist ein System, das in einem Multiagentensystem (ALZ-MAS) integriert ist. Das Ziel des Agenten ist sowohl die Planung und Optimierung der Pflege als auch die Sicherheit für Personen mit Alzheimer in betreutem Wohnen. Um personenspezifische Aktivitätsmodelle zu erzeugen, wird die RFID-Technologie eingesetzt: Jede Person trägt einen RFID-Transponder an Hand- oder Fußgelenken. In der Wohnung selbst sind RFID-Lesegeräte verbaut, die eine Aktivität bzw. Bewegung der Person aufzeichnen und deren Identifikation an einen zentralen Leitrechner übermitteln. Durch den Einsatz von AGALZ können Pflegekräfte eingespart werden [Baj+07, S.1090 ff.].

- Das im "FOOD"-Projekt entwickelte AAL-System ist erprobt für die frühzeitige Erkennung von altersbezogenen Gesundheitsrisiken in der Küchenumgebung. Für die Messdaten der Aktivität der Nahrungsaufnahme ist ein Bewegungsmelder an der Küchenwand und ein Temperatur-/Lichtsensoren an der Kühlschranktür angebracht. Die Erzeugung von Verhaltensmustern wird über einen gerichteten azyklischen Graph (Bayes'sches Netz) realisiert. Um Abweichungen in den Mustern zu erkennen (Rhythmus der Nahrungsaufnahme und Essstörungen) wird ein Algorithmus aus der Klasse MCMC (Markov Chain Monte-Carlo) - Verfahren eingesetzt [Nic+16, S.1 ff.].
- Die Forschungsgruppe "House\_n" am Massachusetts Institute of Technology (MIT) zeichnet sich durch die Forschungsgebiete "The Home", "The Workplace" und "The City" in ihrer realitätsnahen Lebensumgebung "Living Lab" aus. Die Herausforderungen bei "The Home" liegen in der Entwicklung von selbstorganisierenden Netzwerken, dem Entwerfen von digitalen Schnittstellen zu physischen Umgebungen, um die Mensch-Computer-Interaktion zu erweitern, und einer proaktiven Unterstützung für ein gesundes Leben [Tec16, S.1 ff.].
- Der ILSA (Independent LifeStyle Assistant) ist ein agentenbasiertes System, um Senioren ein selbstbestimmtes längeres Leben in der eigenen Wohnung zu ermöglichen. Die Anwendungsszenarien erstrecken sich von einer passiven Überwachung (z.B. der Medikation) über die kognitive Unterstützung durch Erinnerungen an Termine und die Alarmierung von Angehörigen bis hin zu Abschlussberichten über Nutzerverhalten. Der ILSA ist auf Basis der quelloffenen Middleware JADE (Java Agent Development Framework) realisiert und nutzt die Hardwarekomponenten der Honeywell-Hausautomation [KZ+16, S.1 ff.].

Die AAL-Systeme der Kategorien "Mobile and Wearable Sensors" und "Robotics" sind, nicht nur weil das AAL-System für das Proof-of-Concept als Smart Home anzusehen, sondern auch weil die Nutzung von ambienten Sensoren als Teilanforderungen zu betrachten ist, für den Kontext der vorliegenden Arbeit in geringem Maße von Bedeutung und deshalb nur aufgelistet.

### Mobile and Wearable Sensors

- Das WebDA-System bietet sechs webbasierte Dienste für ältere Menschen und deren Angehörige an (<http://webda.info/index.html>).
- Bei dem MOBECS (MOBility and Emergency Call System), das auf die automatische Ereignis- und Notfallerkennung ausgerichtet ist (<http://www.mobecs.eu/>), schlägt für die der Nutzerbefragung abgeleiteten Anforderungen an die zu verwendenden Endgeräte den Einsatz von Uhren oder Mobiltelefon vor [Wel+13, S.507].
- Das KNOTS (KNOWledge-Transfer System) ist ein Konzept für eine serverbasierte, mobile Dokumentations- und Informationsplattform (<http://www.knots-project.eu/>).

- Im SmartSenior-Projekt steht die sichere Mobilität, organisierte medizinische Vorbeugung und Versorgung und eine längere Selbständigkeit für Senioren im Vordergrund. Neben den Assistenzfunktionen im häuslichen Umfeld, wird auch eine sichere Mobilität zu Fuß oder im Auto durch erweiterte Ortungssysteme und dem Nothalteassistent gewährleistet (<http://www.smart-senior.de>).
- Das Ziel des MiraculousLife-Projektes ist die Entwicklung eines VSP (Virtual Support Partner) (<http://www.miraculous-life.eu/>).
- Für die Überwachung von Herz-Kreislaufpatienten, in ihrer eigenen Wohnung, wird im SmartVital-Projekt ein erweitertes Monitoring, das über die Methoden des Data-Clustering realisiert ist, geboten (<http://publica.fraunhofer.de/dokumente/N-272106.html>).
- Das WohnSelbst-Projekt beschreibt eine Service-Plattform, die die Datenflüsse von den plattform-beteiligten Dienstleistern steuert und kontrolliert. Eingesetzt werden Sensoren und Aktoren der Beurer GmbH und Continua Allianz (<http://publica.fraunhofer.de/documents/N-190090.html>).

## Robotics

- ACCOMPANY: Das System basiert auf einem Serviceroboter, der ältere Menschen auf physische, kognitive und soziale Art bei häuslichen Aktivitäten unterstützen soll. (<http://accompanyproject.eu/>)
- Die Entwicklung im ALIAS (Adaptable Ambient Living ASsistent) - Projekt bringt ein mobiles Robotersystem vor, das mit älteren Menschen, zum Ziel der kognitiven Assistenz, interagiert (<http://www.aal-alias.eu/>).
- CIN-Reha ein mobiler Roboter für das Assistieren bei Rehabilitationsübungen und Elevon ein Lifter für den Transport oder die Positionierung von Personen (<http://www.aal.fraunhofer.de/projects.html>).
- Im MobiNa-Projekt ist ein Serviceroboter realisiert, der als mobiles Kommunikationssystem in der eigenen Wohnung angedacht ist. Zusammen mit der Sensorbox (s. Smart Home Projekt (4) "sens@home"), die einen Sturz detektiert, kann der Roboter daraufhin mit der gestürzten Person kommunizieren (<http://w3t.ipa.fraunhofer.de/2210.98.html>).
- Das SRS-Projekt ist auf die Entwicklung von ferngesteuerten, semi-automatischen robotergestützten Lösungen spezialisiert (<http://srs-project.eu/>).

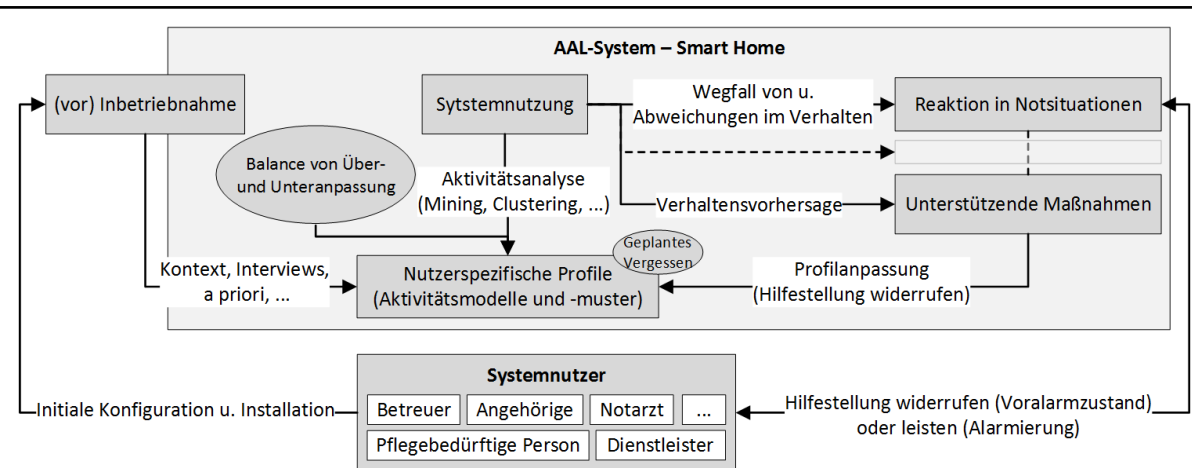
Die Analyse der in diesem Abschnitt (2.4.3) erörterten Projekte hat gezeigt, dass die Entwicklung bzw. Realisierung der AAL-Systeme nicht auf die Auswahl eines einzelnen Algorithmus (Sektion 2.4.2), einer Middleware oder spez. Hardwarekomponenten gestützt ist, sondern vielmehr von einer Kombination aus unterschiedlichen Algorithmen bzw. Softwarearchitekturen

abhängt. Letztendlich ist der Entscheidungsprozess zu geeigneten Hard- und Softwarekomponenten für ein AAL-System abhängig vom geforderten Anwendungsbereich: Die AAL-Systeme, die bei der Planung von Tagesabläufen unterstützen oder eine Lokalisierung oder Identifizierung von Personen bewerkstelligen, zeigen durchschnittliche Analysen der ADL im Vergleich zu solchen Systemen, die eine Aktivitätserkennung, Kontextmodellierung oder Anomaliedetektion umsetzen. Im Anhang 6.1 ist ein tabellarischer Vergleich von ausgewählten AAL-Systemen (in diesem Kapitel vorgestellt) gegeben. Aufgestellt sind nur solche Systeme, die ein Mindestmaß an Informationen zu verwendeten Hard- und Softwarekomponenten (Tab. 6.1) und Anforderungen (Tab. 6.2) der Personas liefern. Der Vergleich zeigt, dass die Systeme bzw. Lösungsansätze der Projekte (11)-MavHome und (13)-k-Pattern, die eine hohe Genauigkeit in der Aktivitätenerkennung aufweisen, alle möglichen Fehlerklassen der ADL (Abschnitt 2.4.4) beachten. Dadurch, dass Event-Logs als Eventsequenzen bzw. Streams mit zeitlicher Referenz verarbeitet werden, bietet das minütliche bzw. variable Zeitfenster eine Auswertung und Reaktion auf evtl. Verhaltensabweichungen in Echtzeit: Eine Evaluation für die QoS eines AAL-Systems bei unterschiedlichen Zeitfenstern ist noch nicht erbracht. Für das AAL-System (Proof-of-Concept) bei Verwendung von der openHAB-Software und den Persistenz-Bindings liegt eine ähnliche Datenbasis (Zeit-Werte-Paare) vor [Mas, S.25 f.]. Sowohl die Autoren des MavHome-Projektes als auch die des k-pattern Ansatzes verweisen auf den Einsatz von Sensoren aus allen drei Bereichen der O-C-M-Typisierung (Abschnitt 2.3) - eine Einschränkung zu Komponenten einer spez. Hausautomationslösung ist nicht gegeben. Wegen diesen Gegebenheiten und der Tatsache, dass für das MavHome-Projekt lediglich ein Experiment in einer realitätsähnlichen Umgebung (Student als Testperson) und beim k-Pattern-Ansatz noch kein Feldexperiment erfolgt ist (s. unzureichende Abdeckung: Anforderungen der Personas Tab. 6.2), hängt die Auswahl zu einer Data-Mining-Bibliothek (Abschnitt 2.6) von den Algorithmen und Datenstrukturen der beiden Verfahren ab.

Die folgenden Unterkapitel stellen die generische Vorgehensweise und Struktur der AAL-Systeme vor und beschreiben unterschiedliche Fehlerquellen in den ADL. Zudem wird für die anstehende Implementierung des AAL-Systems auf den agentenbasierten Ansatz, im Zusammenhang mit den "Machine Learning" Begriffen, hingewiesen.

#### **2.4.4 Vorgehensweise der Systeme und Fehlerquellen in den ADL**

Die Vorstellung der existenten AAL-Systeme (Abschnitt 2.4.3) hat gezeigt, dass die verwendete Hardware, Software und Algorithmen und die daran anknüpfende Systemarchitektur von dem Anwendungskontext, den Nutzeranforderungen und gewünschten Assistenzfunktion abhängt. Die Abbildung 2.14 stellt die generische Vorgehensweise und Architektur von AAL-Systemen der Smart Home - Kategorie dar, die als Teilergebnis aus der Recherche entstanden ist.



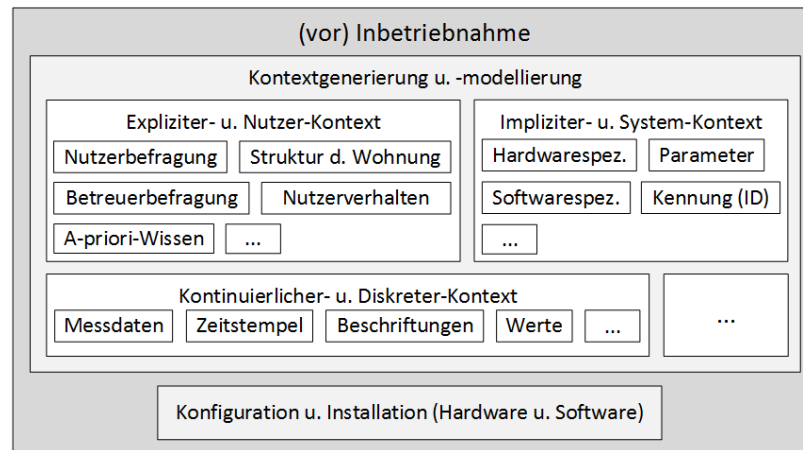
**Abbildung 2.14:** Eruierte Grundarchitektur und Vorgehensweise für AAL-Systeme (Smart Home)

Ausgehend von den Systemnutzern sind es die Angehörigen und pflegebedürftigen Personen, die direkt von der Systemnutzung profitieren: Das Smart Home unterstützt die Ausführung der ADL durch die Steuerung der Aktoren und reagiert, nach Auswertung der Messdaten, auf Notsituationen indem z. B. Angehörige alarmiert werden. Neben einer initialen Konfiguration und Installation von Hard- und Software, deren korrekte Umsetzung neben den Betreuern oder Angehörigen zumeist von externen Dienstleistern gewährleistet wird, sind für die Inbetriebnahme auch nutzungsspezifische Kontextgenerierungen und -modellierungen durchzuführen. Soll das AAL-System dem Betreuer der pflegebedürftigen Person eine tägliche oder wöchentliche Auswertung der erkannten ADL liefern und auf Abweichungen hinweisen, sieht der Ansatz von [Mun15, S.32] vor der eigentlichen Systemnutzung neben den Interviews mit den Pflegekräften auch Gespräche mit Angehörigen oder Betreuern vor. Als Ergebnis soll eine Teilauswahl von vordefinierten ADL-Modellen erreicht werden, um ein personenbezogenes Profil zu erstellen - jeder Aktivität wird ein Handlungsablauf zugeordnet [Mun15, S.154 f.].

*„Der Betreuer legt bei der Initialisierung des Systems fest, welche ADL bei einer bestimmten Person zu beobachten sind. Nach der eigentlichen Festlegung erfolgt die Ausführung durch den Supervisor.“ [Mun15, S.32]*

Für die Kontextgenerierung wird der Nutzer bzw. Betreuer demnach explizit gefragt. Eine Einsicht der Wohnung vor Ort kann weiteren expliziten Kontext formulieren. Wenn Sensoren oder Aktoren vor der Inbetriebnahme beschriftet (s. (10) CASAS-Projekt), in Konfigurationsdateien oder als Teil der Entität gekennzeichnet werden [Mun15, S.154], ist von impliziten- oder auch System-Kontext die Rede.

Je nach Sensortyp und zugrundeliegendem Kommunikationsstandard sind Messdaten in kontinuierlichen oder unregelmäßigen Intervallen zu erwarten, die auf Kontextinformationen für die Inbetriebnahme oder auch während der Systemnutzung schließen lassen. Aus den Kontexten ließe sich somit noch vor der ersten Systemnutzung ein Profil (Aktivitätsmodell) erzeugen.



**Abbildung 2.15:** Eruierte Teilkomponente "(vor) Inbetriebnahme" der Grundarchitektur [i.A.a May04, S.11 ff.]

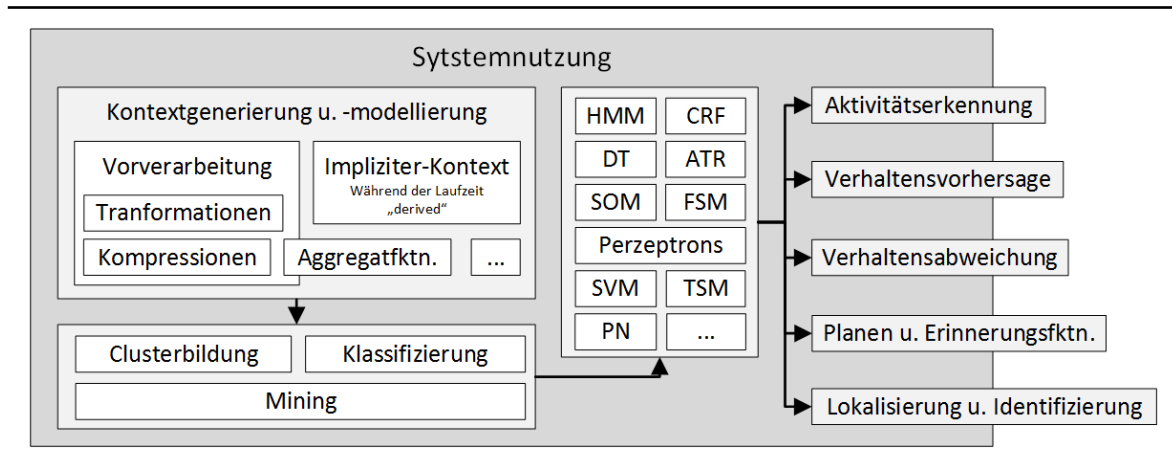
Die in Echtzeit gemessenen Daten bei der Systemnutzung werden in Formate transformiert und komprimiert, um diese anschließend den Aktivitätsmodellen zuzuordnen bzw. zu klassifizieren. Wird von expliziten- bzw. Nutzer-Kontext (z. B. Nutzerverhalten und Befragungen) abgesehen, um die Systemnutzer (s. Persona Hildegard und Alexandra in Abschnitt 2.1) mit der Inbetriebnahme des AAL-System nicht zu überfordern, kann ein Profil auch im Anschluss an eine Lernphase erzeugt werden: Die Muster werden aus den vor-verarbeiteten Daten "gemined" und anschließend in Clustern abgebildet.

Eine Kombination von beiden Herangehensweisen zur Profilerzeugung ist auch möglich, vereint aber auch die Defizite beider Ansätze. Letztendlich hängt die Qualität der Profile, und somit auch ein Teil der QoS des Gesamtsystems, von einer Balance von Über- und Unteranpassung, einem geplanten Vergessen und den Umgang mit Fehlern in den ADL ab: Bei einem über-angepassten Profil neigt das AAL-System zu FPs, ist das Profil zu generisch gestaltet, steigt die Anzahl der FNs. Verhaltensweisen die in der Vergangenheit aufgezeichnet und in der Profilerstellung berücksichtigt wurden, aber nicht mehr zum Alltag der pflegebedürftigen Person gehören, führen zu unter-angepassten Aktivitätsmodellen bzw. -mustern: Ein geplantes Vergessen könnte demnach nicht nur den Schutz der Privatsphäre begünstigen, sondern auch die QoS positiv beeinflussen.

Es konnte kein Konsens eruiert werden, wie ein AAL-System die Profile im Hinblick auf unregelmäßige Besuche von Angehörigen oder Betreuern anpassen soll: [Mun15, S.104+143] schlägt vor, dass Besucher ein "elektronisches Ausweisen" an der Eingangstür mithilfe von RFID-Tags vornehmen - vergisst der Besucher sich zu erkennen zu geben, ist diese Methodik hinfällig. Denkbar ist, dass für das Ausweisen an der Eingangstür, eine intelligente, selbst-versorgende Türüberwachung auf Basis von bildgebender Sensorik eingesetzt wird [Lin15]. Das AAL-System könnte dann selbst entscheiden, ob ein besucherspezifisches Profil erzeugt

oder das System für die Zeit des Besuches pausiert wird - schließlich kann der Besucher die Hilfemaßnahmen in Notsituationen leisten.

Sind die Profile vor der Inbetriebnahme oder nach Abschluss der Lernphase erzeugt, können die Sensordaten eines Zeitfensters während der Systemnutzung in das Datenformat der Aktivitätsmodelle bzw. -muster überführt und mit den in Abbildung 2.16 gelisteten Konstrukten im AAL-Systems ausgewertet werden.



**Abbildung 2.16:** Eruierte Teilkomponente "Systemnutzung" der Grundarchitektur

Die Systemfunktionen für das Planen und die Erinnerungsfunktionen können pflegebedürftige Personen z. B. auf Termine oder die Tabletten-Einnahme hinweisen - Eine Unterscheidung von Wochentagen steigert die Genauigkeit und ist demnach zwingend erforderlich [Mun15, S.30]. Mit der Funktionalität der Lokalisierung und Identifizierung können Betreuer oder Pflegekräfte in Verbindung mit der Aktivitätserkennung, die täglichen, wöchentlichen oder monatlichen Aktivitätsablaufpläne der betroffenen pflegebedürftigen Person kreieren, um anschließend pflegerelevante Prognosen aufzustellen. Wenn das AAL-System Verhalten prognostiziert, können Aktoren gesteuert werden, noch bevor der Systemnutzer die Aktion selbst ausführen muss. Für die Systemfunktion der Verhaltensabweichung ist nicht nur die Interpretation von Wegfallen, sondern auch das Hinzukommen von Verhaltensweisen wichtig [Mun15, S.137].

In diesem Zusammenhang betrachtet [Mun15, S.244 ff.] bei der Verhaltensanalyse vier verschiedene Klassen von Fehlern, die bei Ausführung der ADL auftreten und folgend exemplarisch für die Persona Hildegard (Sektion 2.1) erörtert werden:

- 1 *Auslassungen:* Das AAL-System erkennt, dass Hildegard nicht wie gewohnt Montag vormittags ihre Telenovela fern sieht, da diese aufgrund tagespolitischer Themen nicht ausgestrahlt wird, sondern stattdessen direkt mit ihrer Freundin Agnes telefoniert - es liegt also keine Notsituation vor und eine Alarmierung ist als False Positive (FP) zu werten. Folgt dem Auslassen der Aktivität kein alternativer Handlungsablauf, bedarf



- Hildegard evtl. Hilfe und eine ausbleibende Alarmierung von Angehörigen ist ein False Negative (FN).
- 2 *Wiederholungen*: Statt eine tages- und uhrzeit-typische ADL wegzulassen, werden diese vermehrt wiederholt. Falls Hildegard vergessen haben sollte, ob sie ihre Medikamente bereits eingenommen hat und daraufhin erneut einnimmt, ist eine Verständigung von Alexandra erforderlich, um eine Überdosis bei Hildegard auszuschließen bzw. zu vermeiden.
  - 3 *Verzögerungen*: Hildegard steht in der Woche immer gegen 8.30 Uhr auf. Falls sie bis zu einem gemittelten Grenzwert nicht aufgestanden ist, sollten Technologien wie "SensBed" (s. Projekt Nr. 8) die Vitalparameter unserer Persona überprüfen.
  - 4 *Vertauschungen*: Hildegard bereitet das Frühstück vor noch bevor sie sich gewaschen und gekleidet hat. Augenscheinlich benötigt Hildegard keine Hilfe von Angehörigen und das AAL-System muss diese Vertauschung der Aktivitäten vorsehen, um keinen FP auszurufen.

Des Weiteren weißt [Mun15, S.244 f.] auf Kombinationen der oben erörterten Fehlerklassen hin und greift zudem Unterkategorien von Fehlerklassen in der Literatur auf, die eingesetzte Kraft, Bewegungstyp und Geschwindigkeit der ADL beachten.

Anschließend an die Systemfunktionen zeigen die analysierten AAL-Systeme zwei unterschiedliche Reaktionen: Entweder wird die unterstützende Maßnahme sofort eingeleitet oder das System schaltet zuerst in einen Voralarmzustand, worauf eine Interaktion vom Nutzer mit dem System verlangt wird (z. B. das Beantworten eines Anrufes beim Sicherheitssystem "safe@home"). Falls eine Interaktion ausbleibt, ist eine Notsituation anzunehmen. Signalisiert die Person jedoch ein Wohlbefinden, war die Systementscheidung falsch (FP) und die Aktivitätsmodelle und -muster des Profils müssen angepasst werden: Das System lernt aus seinen Fehlern!

## 2.5 Referenz in Themengebiete der KI

Ziel dieses Unterkapitels ist es nicht, die oben genannten Verfahren der künstlichen Intelligenz zu erörtern und an Beispielen zu verdeutlichen - das würde schlichtweg den Rahmen der vorliegenden Arbeit überschreiten. Im Hinblick auf die anstehende Konzipierung, den Entwurf, Aufbau aber auch die Realisierung des AAL-Systems für den "unsichtbaren Pfleger" zeigt die Analyse der existenten AAL-Systeme, dass es ratsam ist, die anstehende Implementierung auf etablierte Architekturen der künstlichen Intelligenz zu stützen (z. B. Projekt Nr.11 MavHome: agentenbasierter Ansatz mit negativen bestärktem Lernen). Die folgenden Abschnitte stellen demnach die Grundgedanken des maschinellen Lernens und eine Einführung in das Agentenmodell und dessen Arten vor.

Die Begriffe "Datenbank & Mining" und "Aktivitäts-Mining" (s. Sektion 2.2), die in [Mun15, S.23] geprägt sind, stehen unter dem Oberbegriff des Data-Minings. Dabei steht das Data-Mining für die Anwendung von etablierten Verfahren der künstlichen Intelligenz, um aus großen Datenmengen neues Wissen zu gewinnen. Das Data-Mining bringt also selbst keine neuen Verfahren hervor, sondern ist als eigene Teildisziplin der KI (Künstliche Intelligenz) anzusehen [Ert09, S.9]. Für eine detaillierte Erläuterung der Verfahren, die in Abschnitten von 2.2 bis 2.4.4 erwähnt und beim Data-Mining eingesetzt werden können, sei auf etablierte Literatur verwiesen:

In [Alp10, S.185 ff.] ist ein Kapitel den DTs gewidmet. Vorgestellt werden sowohl Klassifikations- und Regressions-Bäume als auch Univariate- und Multivariate-Bäume. Zudem ist das Verfahren des Pruning (Beschneiden von Bäumen) und die Regelextraktion und das Lernen von Regeln aus Daten beschrieben. Beispiele, wie von Entscheidungsbäumen gelernt werden kann, sind in [Ert09, S.203 ff.] gegeben. Obgleich für die ANN, wie z.B. das Perzeptron als linearer Klassifizierer, eine verständliche Beschreibung in [Ert09, S.187 ff.] geboten wird, erläutert [Rus+12, S.845 ff.] zudem die Multilayer-Perzeptrons und das Lernen in solchen Netzen. Ein kurze Einführung in den Klassifikator der SVM ist in [Ert09, S.277 ff.] und [Rus+12, S.863 ff.] wiederzufinden. Das HMM wird detailliert beschrieben und mit Beispielen und Übungen belegt [vgl. Rus+12, S.675 ff.; Alp10, S.363 f.]. Die TSM und deren verwandte Process-Mining-Algorithmen, welche zusammen mit A-priori-Wissen im SAMDY-Projekt eingesetzt werden, sind in [Mun15, S.86 ff.] erörtert und in einem bewerteten Vergleich hinsichtlich Eignung für den Einsatz in Systemen für die technisch unterstützte Pflege abgewägt.

## 2.5.1 Grundgedanken des maschinellen Lernens

Lernen ist der Prozess aus Erfahrungen neues Wissen zu induzieren. Für eine Person ist der Lernprozess aufgrund seiner individuellen Erlebnisse einzigartig und für Außenstehende nicht unbedingt nachvollziehbar, da sie z. B. nicht über die gleiche Wissensbasis verfügen.

Durch die heutige Computer-Technologie sind wir aber durchaus in der Lage große Datenmengen zu persistieren, zu jeder Zeit darauf zuzugreifen und nach Belieben zu verarbeiten. Deshalb werden beim maschinellen Lernen Muster und Regelmäßigkeiten aus den Datenbeständen extrahiert, von denen wir annehmen, dass diese die Prozesse das menschliche Handeln und Denken am besten approximieren [Alp10, S.1 f.]. Dabei sind die Prozesse für die smarten AAL-Systeme im häuslichen Umfeld mit den Handlungsabfolgen der ADL und deren Teilaktivitäten, die der Systemnutzer in seinem Alltag ausführt, gleichzusetzen.

Die Form des maschinellen Lernens hängt davon ab, welche Komponenten verbessert, welcher Wissensstand vorliegt, wie die Daten von der Komponente dargestellt werden und welche Rückmeldung verfügbar ist [Rus+12, S.809]. Welche Komponenten und Darstellung der Daten

vorliegt, beruht auf dem verwendeten Agentenkonzept (s. Abschnitt 2.5.2). Bei den Rückmeldungen kann zwischen (1) keine explizite Rückmeldung, (2) Belohnungen oder Bestrafungen und (3) gelabelte positiv oder negativ Beispiele unterschieden werden. Aus den Feedback-Typen inferieren drei Haupttypen des maschinellen Lernens [vgl. Rus+12, S.810 f.; Alp10, S.11 ff.], die anhand der eruierten Vorgehensweise der AAL-Systeme und der Personas (Abschnitt 2.1) vorgestellt werden:

- (1) *Unsupervised Learning*: Beim unüberwachten Lernen wird dem AAL-System kein Feedback gegeben. Für die Inbetriebnahme werden Daten über einen definierten Zeitraum (1-2 Wochen) gemessen. Nach Abschluss der Messphase werden aus den Daten z. B. Muster extrahiert und in Clustern abgebildet (Lernphase), welche repräsentativ für die Aktivitäten der pflegebedürftigen Person stehen. Während der Systemnutzung können aktuelle Messdaten die Modelle dann zusätzlich anpassen. Verzichtet man auf die Messphase, um eine sofortige Einsatzbereitschaft des Systems zu ermöglichen, ist vermehrt mit Fehlern zu rechnen. Das AAL-System im SAMDY-Projekt versucht dem Defizit der Untereinpassung durch einen reichhaltigen Satz vor-modellierte Aktivitäten zu begegnen, der allerdings für die pflegebedürftige Person vom Betreuer (mit Sachverstand) festgelegt und konfiguriert wird [Mun15, S.127 f.]. Wie das AAL-System letztendlich ohne explizites Feedback entscheidet, ob die gestellte Hilfemaßnahme oder die Alarmierung bei vermuteter Notsituation notwendig war, hängt von den Handlungen und Aktivitäten ab, die als Reaktion von den betroffenen Personen getroffen werden: Ist Hildegard in Not, wird Alexandra z. B. mit ihr ins Krankenhaus fahren oder alarmiert einen Notarzt der Hildegard in ihrer Wohnung versorgt. Liegt ein FP vor, würde Alexandra sich im ersten Moment ärgern, aber dann doch froh sein, dass es Hildegard gut geht und nach kurzer Zeit ihrer unterbrochenen Tätigkeit nachgehen. In beiden Fälle sind unterschiedliche Messdaten zu erwarten.
- (2) *Reinforcement Learning*: Für das bestärkende Lernen sind Belohnungen und Bestrafungen des Systems vorgesehen. Wie auch beim unüberwachten Lernen, kann eine Messphase durchgeführt oder eine sofortige Einsatzbereitschaft gewährleistet werden. Vermutet das System eine Notsituation und geht in den Voralarmzustand und signalisiert (audio/visuell) die anstehende Alarmierung von Alexandra, könnte Hildegard diese Entscheidung durch z. B. eine bloße Bewegung zurückziehen oder einen Anruf beantworten, worauf die systeminternen Routinen die fälschlicherweise interpretierte Situation im System "bestrafen". Für die Systemfunktion der Verhaltensvorhersage, bei der Aktivitäten prognostiziert und Aktoren gesteuert werden (Licht ausschalten), könnte eine innerhalb kürzester Zeit ausgeführte Aktion (Licht wieder anschalten) vom Systemnutzer einen Hinweis auf eine Fehlinterpretation liefern. Bleibt eine Aktion aus, kann das System belohnt werden.
- (3) *Supervised Learning*: Beim überwachten Lernen liegen dem AAL-System positiv und negativ Beispiele für die ADL und deren notwendige Reaktionen vor, woraus eine Funktion

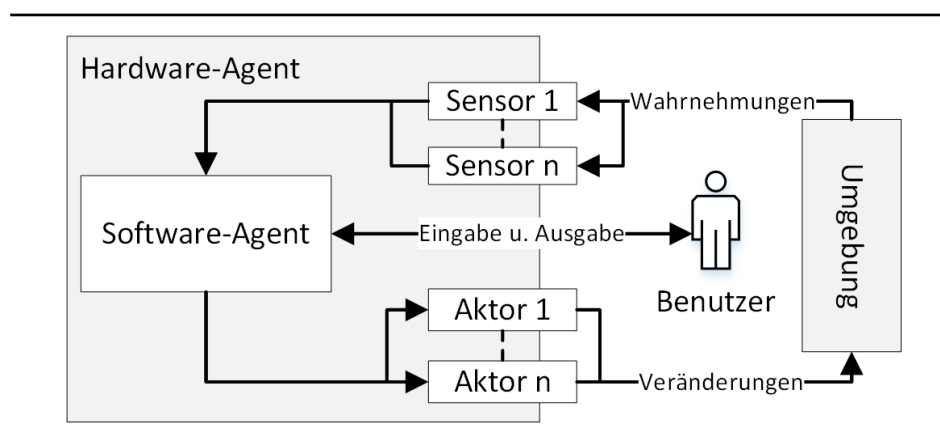
erlernt wird, die Messdaten bzw. Muster während der Systemnutzung auf eine entsprechende Reaktion abbildet.

- (4) *Semi-Supervised Learning*: In [Rus+12, S.811] ist zudem das halb überwachte Lernen, eine Kombination aus Supervised- und Unsupervised- Learning, erwähnt, dessen Existenz auf nicht klare Unterscheidungen von Handlungsweisen in der Praxis zurückzuführen ist.

Wenn die Lernphase für die Inbetriebnahme und die Erkennungsphase während der Systemnutzung differenziert betrachtet werden, ist für den anstehenden Entwurf des AAL-Systems in erster Instanz ein unüberwachtes mit anschließendem bestärkenden Lernen zu berücksichtigen.

## 2.5.2 Einführung in das Agentenmodell

In Abgrenzung zum einfachen Computerprogramm bzw. Algorithmus erwartet man vom einem Agenten, das er seine Umgebung wahrnimmt, dementsprechend notwendige Änderungen vornimmt, eigenständig handelt und dabei eigene Ziele formuliert und verfolgt. Für den Einstieg in das Thema der intelligenten Agenten beschreibt [Rus+12, S.24 ff.] das menschliche Denken als Ansatz der kognitiven Modellierung, rationales Denken als Ansatz für Denkregeln (Logiken) und den rationalen Agenten, der sein Handeln an logischem Schließen ausrichtet. Darauf aufbauend konzipiert [Rus+12, S.60 f.] Entwurfsgrundlagen, die der Erstellung von intelligenten Agenten dienen: In Abbildung 2.17 sind die Grundkomponenten eines intelligenten Agenten dargestellt.



**Abbildung 2.17:** Hard- und Software-Agent mit Benutzerinteraktion [i.A.a Rus+12, S.61; Ert09, S.13]

Die Abbildung greift dabei das einfache Agentenkonzept von [Rus+12, S.61] und die Konzepte von Hardware- und Software-Agenten von [Ert09, S.12 f.] auf. Für den Software-Agent, der vorwiegend in der klassischen Informatik und den Hardware-Agent, der in der Robotik, anzutreffen ist, sieht das ursprüngliche Konzept eine klare Trennung beider Agenten vor. Die Analyse der existenten AAL-Systeme zeigt auch, dass ein Zusammenspiel von hard- bzw. soft-

waregetriebenen Projekten, um ergänzende Hilfemaßnahmen zu bieten, durchaus sinnvoll ist - z. B. Serviceroboter des MobiNa-Projektes und Sicherheitssystem "safe@home". Zudem ist das Grundkonzept von Smart Homes in dieser Kombination der Agenten wiederzufinden, denn der Leitrechner auf der Managementebene koppelt die Sensoren und Aktoren durch die auf ihm befindliche Software, welche intelligente Entscheidungen trifft.

Die Folge an Beobachtungen, die der Agent im gesamten Zeitverlauf den Sensordaten aus der Umgebung entnimmt, sind als Wahrnehmungsfolge definiert - was einem Wissenstand von Anfang der Agentenlaufzeit bis dato gleicht. Um mit diesem Wissenstand die Aktoren anzusteuern, bildet die Agentenfunktion, welche vom Agentenprogramm im Software-Agenten implementiert wird, die Wahrnehmungen auf eine Aktion ab [Rus+12, S.60 f.].

"Agent = Architektur + Programm" [Rus+12, S.74] (+ Wahrnehmungsfolge)

In [Ert09, S.13] ist dieses Konzept auch als "Agent mit Gedächtnis" bezeichnet. Der Agent ohne Gedächtnis (Reflex-Agent) basiert auf einem deterministischen Programm, dessen Funktion die Eingaben auf bekannte Ausgaben abbildet. Die Wahl zur passenden Architektur für einen Agenten hängt stark von der Aufgabenumgebung bzw. dem Bild der Umgebung und deren Eigenschaften ab, die von den Messdatenarten der Sensoren abgeleitet werden. Für die Spezifizierung der Aufgabenumgebung eignet sich die PEAS (Performance, Environment, Actuators, Sensors) -Beschreibung [Rus+12, S.66 ff.], die exemplarisch für AAL-Systeme (Smart Home) in der Abbildung 2.18 visualisiert ist.

Agententyp	Leistungsbewertung	Umgebung	Aktuatoren	Sensoren
Altersgerechtes Assistenzsystem (Smart Home)	Reaktion in angemessener Zeit, benutzerfreundliche Inbetriebnahme u. Interaktion, wenige FPs, keine FNs	Wohngebäude, Wohnung, Haushalte	Leuchten, Lautsprecher, Bildschirme, etc.	Bewegungsmelder, Lichtschranken, Drucksensoren, Magnetschalter, etc.

**Abbildung 2.18:** PEAS-Beschreibung der Aufgabenumgebung für ein AAL-System (Smart Home) [i.A.a Rus+12, S.67]

Die Eigenschaften der Aufgabenumgebungen sind wie folgt zu sehen [vgl. Rus+12, S.69 ff.; Ert09, S.14 f.]:

- { *nicht* | *teilweise* | *vollständig* } beobachtbar: Agenten die keine Sensoren besitzen, können ihre Umgebung nicht beobachten. Eine Umgebung ist teilweise beobachtbar, wenn die Sensordaten ungenau, gestört oder nicht zu jedem Zeitpunkt den gesamten Umgebungszustand aufgreifen. Die Umgebung des Agenten ist vollständig beobachtbar, wenn die Sensordaten zu jedem Zeitpunkt den Umgebungszustand abbilden.
- Ein Multiagent, der nicht auf einer einzelnen sondern auf mehreren Umgebungen aufbaut, ist *konkurrierend* wenn beide Agenten ihre eigene Leistungsbewertung maximieren

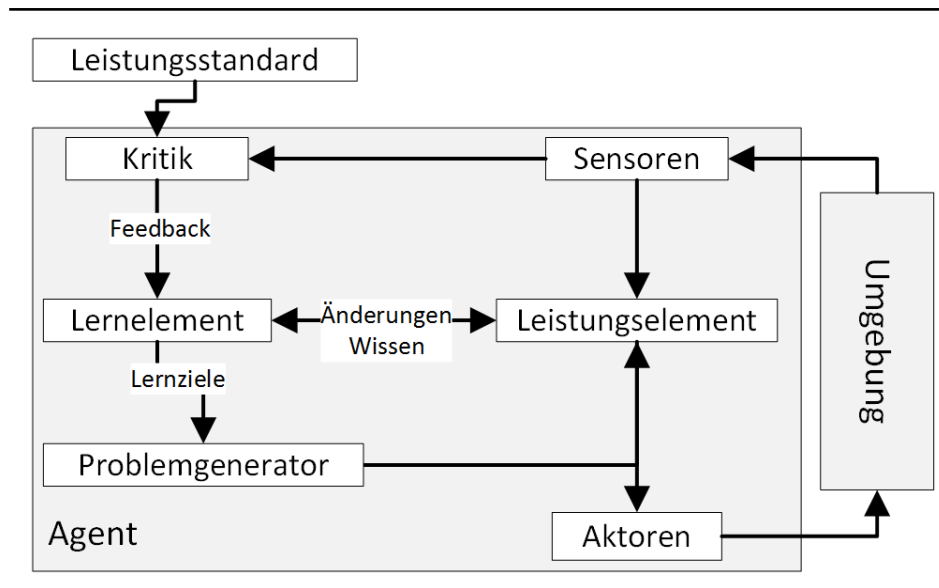
möchten, oder *kooperativ*, um eine gemeinsame maximale Leistungsbewertung zu erzielen.

- Eine Aufgabenumgebung ist *deterministisch*, wenn durch den aktuellen Umgebungszustand und die vom Agentenprogramm berechneten Aktionen auf einen Folgezustand der Umgebung geschlossen werden kann und *stochastisch* wenn das Verhalten der Umgebung nicht vorhersehbar ist.
- Für Agenten die auf einer *episodischen* Aufgabenumgebung agieren, beeinflussen die vergangenen Entscheidungen nicht die gegenwärtigen Aktionen einer Episode - in einer *sequenziellen* Aufgabenumgebung sind derzeitige von vorherigen Entscheidungen abhängig.
- Ein Agent kann auf *statischen* Umgebungen, die nicht während der Programmausführung geändert werden, oder *dynamischen* Umgebungen basieren.
- Existieren in der Aufgabenumgebung des Agenten nur eine endliche Anzahl von unterschiedlichen Zuständen, ist diese *diskret* - andernfalls *stetig*.

Für die anstehende Realisierung des AAL-Systems ist bei Verwendung von Sensoren, die auf drahtlosen Kommunikationsstandards basieren und die geforderte relative maximale Frequenzbelegungsdauer einhalten müssen, darauf zu achten, dass die Aufgabenumgebung gegebenenfalls nur teilweise beobachtbar ist und der Agent einen internen Zustand verwalten muss. Eine *diskrete* und *deterministische* Aufgabenumgebung ist gerade für Systemfunktionen, die ein Verhalten vorhersagen möchten, existenziell.

Der Entwurf der Agentenprogramme ist an die vorliegende Architektur bzw. Art des Agenten gebunden. [Rus+12, S.74 ff.] erörtert neben den einfachen und modellbasierten Reflexagenten auch zielbasierte, nutzenbasierte und lernende Agenten. Im Gegensatz zu einfachen Reflexagenten speichert der modellbasierte Reflexagent den aktuellen Zustand seiner Umgebung ab. Zusätzlich zu dem Zustand verwaltet der zielbasierte Agent eine Menge an Zielen, die im bei der Auswahl zur geeigneten Aktion verhelfen. Der nutzenbasierte Agent bewertet die einzelnen Zustände seiner modellierten Umgebung mit einer Funktion für den besten zu erwartenden Nutzen. Der allgemeine lernende Agent (Abbildung 2.19) enthält neben dem Leistungselement, das bei den zuvor genannten Agenten mit der Komponente, die Aktionen anhand der Wahrnehmungen auswählt, zu vergleichen ist, den Problemgenerator, das Lernelement und die Kritik. Das Lernelement verbessert das Leistungselement durch Rückmeldungen aus der Kritik, welche an einen festgelegten Leistungsstandard gebunden sind. Der Problemgenerator schlägt dem Leistungselement bereits von ihm sondierte Aktionen vor, die auf lange Sicht bessere Ergebnisse erzielen.

Die Darstellung der Zustände und deren Übergänge kann (1) atomar, (2) faktorisiert oder (3) strukturiert erfolgen [Rus+12, S.85 f.]. Für die erste Darstellung wird der Agent als Blackbox betrachtet. Bei (2) liegen die inneren Zustände als Vektoren von Attributwerten vor. In der strukturierten Darstellung umfasst ein Zustand mehrere Objekte, die untereinander verknüpft sind.



**Abbildung 2.19:** Allgemeiner lernender Agent [Rus+12, S.83]

Schlussendlich ist die Wahl zu einer geeigneten Agentenarchitektur für ein AAL-System nicht einfach, denn [Rus+12, S.1207] weist darauf hin, dass erst eine Kombination aus den Architekturen einen vollständigen Agenten ausmacht:

*"Eine wichtige Eigenschaft hybrider Architekturen ist, dass die Grenzen zwischen den verschiedenen Entscheidungskomponenten nicht feststehend sind."*

## 2.6 Von der Theorie zur Praxis

Die Analyse der existenten AAL-Systeme und deren zugrundeliegenden Methodiken hat gezeigt, dass die Umsetzung der Systemfunktionen nicht an einen allgemeingültigen Lösungsansatz gebunden ist, sondern sowohl von den Modellierungskonzepten der ADL, den Sensor- als auch den Messdatentypen der eingesetzten Hausautomationslösung abhängt.

Für die anstehende Realisierung des AAL-Systems werden die Hardwarekomponenten der Homematic-Hausautomationslösung eingesetzt, da diese die höchste Verfügbarkeit in der Machbarkeitsbetrachtung für die Planung einer AAL-Lösung aufweist [al15, S.149]. Für das Data-Mining, also den Prozess um aus den Messdaten mittels der Verfahren der KI die Aktivitätsmodelle- bzw. Muster zu gewinnen, soll eine existente Software-Bibliothek eingesetzt werden. Die Auswahl zur geeigneten Data-Mining-Bibliothek ist an folgende Restriktionen gebunden:

- 1 Die Wahl der Programmiersprache ist grundsätzlich freigestellt. Vorzugsweise ist die Hochsprache "Java" zu selektieren, da diese nicht nur plattformunabhängig ist, sondern

die notwendigen Entwicklungspakete - das Java Development Kit (JDK) und die Java Runtime Environment (JRE) - bereits für die Inbetriebnahme der openHAB-Software installiert sind.

- 2 Die Data-Mining-Bibliothek wie auch alle anderen Softwarekomponenten zur Realisierung des AAL-Systems soll quelloffenen und nicht proprietär sein.
- 3 Eine GUI (Graphical User Interface) ist für den Endanwender obligatorisch, für den Entwickler nicht notwendig und somit ein optionales Kriterium.
- 4 Die Bibliothek muss über eine dokumentierte API (Application Programming Interface) verfügen, um heuristische Methoden wie z. B. "Trial and Error" auszuschließen.
- 5 Für eine Auswertung in Echtzeit müssen die persistierten Sensordaten zur weiteren Verarbeitung in einem minütlichen Zeitfenster aus der Datenbank abgefragt werden. Die Bibliothek sollte einen Query-Manager zur Verfügung stellen, der diese Aufgabe übernimmt, und die Daten in ein von der Bibliothek unterstütztes Format übertragen.
- 6 Welche Formate unterstützt die Bibliothek neben dem nativen SQL (Structured Query Language) - Format?
- 7 Die Bibliothek muss die im MavHome-Projekt und k-Pattern-Ansatz erörterten Algorithmen und Datenstrukturen implementieren, um die Konzepte ((11)+(13) Abschnittes 2.4.3) für das Proof-of-Concept (AAL-System mit openHAB) zu verwirklichen.

### Data-Mining-Bibliotheken

**(1)** Das ProM, das im SAMDY-Projekt eingesetzt wird, um aus den Event-Logs die Transitionssysteme zu konvertieren [Mun15, S. 205], ist grundsätzlich für das Erschließen von Wissen aus Protokollen von Ausführungsvorgängen über Business-Prozesse geeignet. ProM ist in Java geschrieben und dessen Kernkomponente unter der GPL (GNU Public License) veröffentlicht. Die Dokumentation für die GUI ist ausreichend beschrieben [Gro+16b], die API ist eine reine Aufzählung von Java-Klassen,-Schnittstellen und -Methoden [Gro+16a]. Mit der "RapidProm"-Erweiterung können die Funktionalitäten des Frameworks in dem Learning Environment (LE) "RapidMiner" genutzt werden [Gro+15], das selbst unter der GNU Affero General Public License (AGPL) steht.

**(2)** "RapidMiner" ist auf die Anwendungsfelder der Business Intelligence (BI), des Data- und Text-Mining und maschinellen Lernens spezialisiert. Für die Visualisierung und Analyse der Daten können diese in einer GUI geladen, transformiert, modelliert und anschließend evaluiert werden. Der Import der Daten kann als Dateien in Formaten, wie z. B. Excel, Comma-separated values (CSV) oder auf Basis von Datenbanken erfolgen [Gmb15e]. Die Konzepte und Funktionen von "RapidMiner" können in bestehende Java-Applikationen integriert werden [i15]. Um in "RapidMiner" eigene Operatoren zugänglich zu machen, können Erweiterungen als Skripte in der Programmiersprache "Groovy" verfasst werden [Gmb15d, S.4]. Es existiert eine Extension



für die Software-Bibliothek der WEKA (Waikato Environment for Knowledge Analysis) von Version 5.3.1 [Gmb15f].

**(3)** Die WEKA selbst, ohne deren Komponenten der Bibliothek als Erweiterung im "RapidMiner" zu nutzen, bietet eine GUI an, mit der vier Applikationen der Softwareumgebung ausgeführt werden:

- 1 *Explorer*: Im Explorer können die Funktionen der fünf Kernpakete der Bibliothek on-the-fly angewendet werden. Neben den Filtern, der Klassifizierung und der Clusterbildung von Daten, können Assoziierungsregeln extrahiert und eine Auswahl von Attributen getroffen werden. Die Ergebnisse werden in 2D visualisiert [Scu+15, S.3 ff.].
- 2 *Experimenter*: In dieser Applikation kann der Anwender einzelne Experimente erstellen, um mehrere Ansätze der Datenanalyse statistisch zu vergleichen.
- 3 *KnowledgeFlow*: Die Komponenten der Bibliothek werden in grafischen Elementen miteinander verknüpft [Hal+15, S.2 f.].
- 4 *Simple CLI*: In der Kommandozeile können die Funktionen der Bibliothek auch genutzt werden, ohne die GUI von Explorer oder Experimenter zu verwenden.

Die Daten können z. B. aus MySQL-, PostgreSQL- und Oracle-Datenbanken mittels der JDBC (Java Database Connectivity) Schnittstelle importiert werden [Wai15a]. Für den Einsatz in einer IDE (Integrated Development Environment) ist darauf zu achten, dass ab der Version 3.7.2 das Konstrukt der Pakete, als zusätzliche Funktionserweiterung und der Paket-Manager für das De- und Installieren der Pakete eingeführt wurde [Wai15b]. Eine Analyse von Zeitserien ist mit den Paketen "timeSeriesFilters" und "timeseriesForecasting" möglich (erst ab Version 3.7.3) [AHGC15] - die letzte stabile Version von WEKA ist 3.6.13.

**(4)** Neben dem RapidMiner und der WEKA erwähnt [Ert09, S.232 f.] das "KXEN"-Framework, welches von SAP erworben wurde [SE15]. Die GMDH Shell ist genauso wie die KXEN eine proprietäre Software, deren Expertise in der Analyse, Optimierung und Vorhersage von Lagerbestand, Verbrauch und Zeitserien liegt [She15].

**(5)** Die Orange-Software (GPL) ist spezialisiert auf die Daten-Visualisierung und -Analyse in interaktiven und grafischen Arbeitsabläufen. Die Komponenten der Orange-Software können neben der GUI auch über die API der Bibliothek verwendet werden, die in Python geschrieben ist. Der Import der Daten kann über Dateien oder Referenzen zu Datenbanken erfolgen [Min15]. Eine Auswertung von Zeitserien scheint derzeit nicht möglich.

**(6)** "R" ist eine Programmiersprache und eine Softwareumgebung zugleich. Ihr Anwendungsschwerpunkt sind statistische Methodiken, Werkzeuge für die Datenanalyse (Zeitreihenanalyse, Klassifizierung und Clusteranalyse) und die graphische Visualisierung der Ergebnisse [Fou16d]. Mit dem "CRAN-Package Repository" kann der Funktionsumfang von "R" ergänzt bzw. er-

weitert werden [Fou16c] und listet derzeit über 8000 Pakete. Mit dem JRI (Java/R Interface) - Paket kann "R" innerhalb Java-Anwendungen ausgeführt werden [RFo16].

(7) Das SPMF (Sequential Pattern Mining Framework) implementiert 113 Algorithmen aus den Bereichen der Clusteranalyse, Klassifikation und des "Pattern-Mining". Das Framework kann über die Kommandozeile oder eine einfache Benutzerschnittstelle genutzt werden [FV16b].

(8) Weitere quelloffene Data-Mining-Bibliotheken sind KEEL (Knowledge Extraction based on Evolutionary Learning) [WS15] und ELKI (Environment for DeveLoping KDD-Applications Supported by Index-Structures) [Mü15], die unter der GPL bzw. AGPL Version 3 nutzbar sind. Eine Alternative zur PASW (Predictive Analysis SoftWare - ehem. IBM SPSS Statistics) [Inc15] bildet die PSPP-Software, die unter der GPL Version 3 verfügbar ist [Fou15]. Die PSPP-Software, KEEL und ELKI streben den Funktionsumfang der WEKA und RapidMiner an.

In Anhang 6.3 ist ein tabellarischer Vergleich der vorgestellten Data-Mining-Bibliotheken mit den oben genannten Kriterien gegeben. Der Vergleich zeigt, dass keine der eruierten Bibliotheken die notwendigen Algorithmen und Datenstrukturen des (11)-MavHome-Projektes (ED, MDL, ALZ und MDP) besitzt. Für den (13)-k-Pattern-Ansatz weist lediglich die WEKA den SAX und FP-growth auf. Der notwendige k-Pattern-Algorithmus ist in keiner der Bibliotheken implementiert, jedoch als Pseudocode vom Autor angegeben [Suk+15, S.119]. Zudem erwähnen die Autoren, dass der von ihnen vorgestellte Ansatz und der Vergleich zu Algorithmen wie z.B. EM und K-means (Erkennungsrate von Clustern) und dem HMM, NB und C4.5 (DT) (Aktivitätserkennung) die besten Ergebnisse erzielt und mit der WEKA erfolgt ist [Bou+15, S.11961]: Somit begründet, wird für die anstehende Realisierung des AAL-Systems auf die Weka-Bibliothek gesetzt.

### Weitere notwendige Softwarekomponenten

Die openHAB-Software verwendet die Eclipse-Equinox, eine Implementierung der OSGi Service Plattform. Die Automatisierungslogik der openHAB-Software ist als einfacher Reflex-Agent anzusehen, die den Status der gekoppelten Sensoren überwacht und auf Veränderungen mit den in den Regeln definierten Aktionen reagiert. Eine Beschreibung der openHAB-Komponenten im Zusammenhang mit dem OSGi-Rahmenwerk ist in [Mas, S.4 ff.] gegeben. Die Persistenz der Daten wird in der Software über die Persistence-Bindings, eine Implementierung von OSGi-Bundles, realisiert [Uh15b]. Die Daten können unter anderem in einer MySQL-Datenbank, MongoDB, InfluxDB oder mit der JPA (Java Persistence API) in der Apache Derby oder PostgreSQL gespeichert werden. Das Datenbankmanagementsystem des Datenbanksystems sollte neben einem SQL-Dialekt auch über prozedurale Sprachkonstrukte verfügen, um die Daten nach der Speicherung in kontextspezifische Formate zu transformieren, da die Software-

Bibliothek diesen Vorgang nicht ausreichend unterstützt. Für die Reaktion in Notsituationen können Anrufe getätigt oder Kurznachrichten und E-Mails versendet werden: openHAB bietet Aktionen, um E-Mails oder Android-Kurznachrichten zu verschicken und Bindings, um Anrufe über die PBX (Private Branch Exchange) Lösung "Asterisk" oder die "FritzBox" via WLAN oder DECT (Digital Enhanced Cordless Telecommunications) zu tätigen, an [Uh15a].

Die Anforderungsanalyse und der Entwurf des AAL-Systems im Kapitel 3 stützen sich auf die Agenten-Konzepte des maschinellen Lernens und auf standardisierte Gliederungen für Softwarearchitekturen von namhaften Templates: Der internationale Standard (ISO/IEC/IEEE 42010) mit dem Namen "Systems and Software engineering - Architecture description" ist als Ergebnis eines Zusammenschlusses aus der ANSI/IEEE 1471-2000 Norm entstanden [(Hr15)]. Ein weiteres Template für die Beschreibung von Software- und Systemarchitekturen ist das "arc42", welches auf praktischen Erfahrungen von internationalen Architekturprojekten basiert [Sta+15]. In Abgrenzung zum Template der ISO/IEC/IEEE 42010 Norm liegt der Fokus beim arc42-Template auf den Randbedingungen, Kontextabgrenzungen, Systemsichten, zentralen Entwurfsentscheidungen, Qualitätsszenarien und den damit verbundenen Risiken.

# 3 Automatisierte Verhaltensanalyse

Im vorliegenden Kapitel werden alle Fragmente erörtert, die in ihrer Gesamtheit eine automatisierte Verhaltensanalyse umsetzen: Das Konzept der Datenanalyse beschreibt, wie die Messdaten ausgewertet werden, um eine Verhaltensabweichung zu erkennen bzw. eine Verhaltensprognose zu stellen. Die aus der anschließenden Anforderungsanalyse eruierten Anforderungen dienen als Basis für den Entwurf des Systems. Die Abschnitte 3.3 und 3.4 geben dem Leser sowohl über verwendete Hard- und Softwarekomponenten als auch die durchgeführte Installation und Konfiguration des AAL-Systems Auskunft. Nach der Vorstellung der genuinen Wohnung, in der das AAL-System eingesetzt ist, wird die umgesetzte Realisierung des Systems anhand der Implementierungen der Anforderungen erläutert. Die abschließende Durchführung der Evaluation und Beschreibung der Funktionstests stellt die Wissensbasis für die in Kapitel 4 beschriebenen Ergebnisse dar.

## 3.1 Konzept der Datenanalyse

Im vorgestellten (11)-k-Pattern-Ansatz (Abschnitt 2.4.3) erwähnen die Autoren die Umsetzung der Verfahren (Abbildung 2.13) mit der WEKA-Bibliothek. Für die Vorverarbeitung der Rohdaten ist der SAX-Filter eingesetzt. Dieser kann mit der WEKA nur durch eine Filterkette (sog. MultiFilter) auf die Rohdaten angewendet werden und ist erst mit dem "timeSeriesFilters"-Paket ab der WEKA-Bibliothek Version 3.7 erhältlich [Fra16b]. Die gefilterten Daten können jedoch nicht direkt als Eingabe für den FP-Growth-Algorithmus genutzt, sondern müssen zuerst in ein ARFF (Attribute-Relation File Format) transformiert werden, indem die Daten binär aufgelistet sind. In der offiziellen Dokumentation gibt es keine Hilfestellung für notwendige Ein- oder Ausgabeformate des Algorithmus [Wai16] - lediglich die WEKA-Mailingliste greift die notwendige Information auf [Fra16c]. Die Dokumentation zum FP-Growth-Algorithmus beschreibt Parameter und Optionen, die ein Endanwender bei Verwendung der WEKA-Software (GUI) verwenden kann. Wenige Java-Quellcode Beispiele zur Datenbankabfrage, Klassifizierung, dem Filtern, der Clusteranalyse und dem Erzeugen von ARFF-Dateien sind in [LLC16] gegeben, aber für die WEKA-Bibliothek Versionen 3.7.x teilweise veraltet. Der Anhang 6.3 zeigt einen Ausschnitt der Implementierung von der Datenabfrage (Datenbank) über die Datenkonvertierung und dem Erstellen einer ARFF-Datei bis hin zur Ausgabe der Assoziationen durch den

FP-Growth-Algorithmus. Da die notwendigen Informationen der ARFF-Konvertierung für die aktuellen Versionen nicht verfügbar sind, ist er im Anhang aufgeführte Java-Quellcode mit der "Trial and Error"-Methode erzeugt worden. Die Ausgabe des FP-Growth soll der Eingabe des k-Pattern-Algorithmus dienen, um die Eventmuster einem neuen oder einem bestehenden Cluster hinzuzufügen [Bou+15, S. 11959 f.]. In diesem Zusammenhang schreiben die Autoren:

*Two patterns belong to the same cluster if the distance between them is less than a specified threshold. [Bou+15, S.11960]*

Das Attribut bzw. Kriterium, welches für die Distanzmessung von zwei Mustern angewendet wird, ist nicht beschrieben und konnte nicht vom Autor der vorliegenden Arbeit eruiert werden. Die Nachfrage beim betreuenden Professor Younghwan Yoo konnte keinen Aufschluss liefern:

*Actually, my student was in charge of the implementation of K-pattern, so I think he can answer your questions.*

Der Student "Serge T. M. Bourobou" hat bis dato (Stand: April 2016) nicht auf die Anfrage reagiert (Mailverkehr: November 2015). Aufgrund der fehlenden Informationen, die eine Umsetzung des in [Bou+15] beschriebenen Ansatzes einer unüberwachten Aktivitätserkennung für Smart Home unmöglich machen, wird auf ein neues Konzept der Messdatenverarbeitung gesetzt.

Konzepte					
	Doku. u. Beschr.	AlgoDat	Implementierung	Features	
(11)-MavHome	n.a.	ED, MDL, ALZ, MDP	✗, grober Pseudocode für ED	minütliches Zeitfenster (ungenau Beschr.), Wochentage (geplant?)	
(13)-k-Pattern	n.a.	SAX, FP-growth, k-Pattern, ANN, ATR, DT	WEKA > 3.7.x (teilweise)	variable Zeitfenster	
Smart Home Learning Module (SHLM)	✓, eigenes Konzept	eigene Algorithmen (Anomalie), SPMF [CPT+ (Prognose)]	eigene Implementierung, SPMF	variable Zeitfenster, Unterscheidung von Wochentagen	

**Tabelle 3.1:** Ausgewählte Konzepte der Messdatenverarbeitung

**Abkürzungen:**

AlgoDat: Algorithmen und Datenstrukturen | ✓ verfügbar | ✗ nicht verfügbar | n.a. nicht ausreichend

Wie in Tabelle 3.1 visualisiert, kann für die Implementierung des Proof-of-Concept der vorliegenden Arbeit das (11)-MavHome- oder ein eigenes Konzept realisiert werden. Die Beschreibung und Dokumentation von dem (11)-MavHome-Konzept sind für eine sofortige Implementierung ungenügend. Zudem scheint das Konzept nicht die notwendige Unterscheidung von Wochentagen, die in [Mun15, S.30+137] als obligatorisch gewertet wird, zu verfolgen:

*„Ein weiteres bedeutsames Ergebnis der Arbeit besagt, dass die Genauigkeit der Vorhersage durch die Unterscheidung von Wochentagen weiter gesteigert werden kann. Verhaltensweisen an den Wochenenden unterscheiden sich zum Teil deutlich von denen in der Woche.“* [Mun15, S.30]

Um diesem Ergebnis zu genügen, kann folglich nur das eigene Konzept des SHLM umgesetzt werden: Die Erkennungsrate von Aktivitäten bzw. Anomalien wird durch die Features eines variablen Zeitfensters und der Unterscheidung von Wochentagen gesteigert bzw. durch Begünstigung nicht von vornherein beeinträchtigt.

Datum u. Uhrzeit	Sensor	Wert
2016-01-02 12:56:49	M04	1
2016-01-02 12:56:51	C01	1
2016-01-02 12:56:59	M05	0
2016-01-02 12:57:08	C01	0
2016-01-02 12:57:11	M01	0
2016-01-02 12:57:20	C02	0
2016-01-02 12:57:21	M05	1
2016-01-02 12:57:23	M01	1
2016-01-02 12:57:24	M04	0

**Tabelle 3.2:** Ausschnitt der gemessenen Daten

**Abkürzungen:**

M0X: Bewegungsmelder Nr.X | C0X Kontaktsensor Nr.X | 1 Bewegung erkannt bzw. offen | 0 Bewegung nicht erkannt bzw. geschlossen

Das eigene Konzept der Messdatenverarbeitung beantwortet somit zwei Fragen:

- (1) Liegt eine Verhaltensabweichung zum Zeitpunkt  $t = \text{jetzt}$  vor? (Anomalie?!). Hierzu sind die vier Fehlerklassen der ADL zu beachten (Abschnitt 2.4.4): Auslassungen, Wiederholungen, Verzögerungen und Vertauschungen.
- (2) Welches Ereignis erwarte ich nach einer erfolgten Ereignisfolge? (Prognose!)

(1) Um die Frage zu klären, ob zum Zeitpunkt  $t = \text{jetzt}$  eine Anomalie vorliegt, wird für jedes Event  $E_i$  (Kombination aus Sensorkennung und Wert) eine  $m \times n$  Matrix  $M(E_i^w)$  mit  $1 \leq i \leq \#Events$  und  $w = \#betrachteteWochen$  erstellt.  $m$  steht dabei für die Anzahl der Wochentage ( $1 = \text{Sonntag}$  bis  $7 = \text{Samstag}$ ) und  $n$  ist die Anzahl der Minuten für einen

ganzen Tag ( $24 \cdot 60 = 1440$ ) - mit  $n = 0$  für 0:00 Uhr beginnend. Der in einer Zelle  $M(E_i^w)_k^j$  mit  $1 \leq j \leq m$  und  $1 \leq k \leq n$  hinterlegte Wert ist die absolute Häufigkeit für das Eintreten des Events  $E_i$  für die  $k$ -Minute am  $j$ -ten Wochentag für einen Beobachtungszeitraum von  $w$ -Wochen. Eine durchschnittliche Häufigkeit ist mit  $\frac{M(E_i^w)_k^j}{w}$  zu errechnen. Der Gesamtbestand der Matrizen stellt ein Normalverhalten über  $w$ -Wochen dar (Lernphase).

$$M(E_1^5) = \begin{matrix} & \dots & 685 & 686 & 687 & 688 & 689 & 690 & 691 & 692 & 693 & 694 & \dots \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{matrix} & \left( \begin{matrix} \dots & 0 & 0 & 0 & 1 & 1 & 2 & 2 & 1 & 1 & 0 & \dots \\ \dots & 0 & 0 & 1 & 1 & 2 & 1 & 1 & 0 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 2 & 2 & 0 & 1 & 1 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & \dots \\ \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\ \dots & 1 & 1 & 2 & 0 & 1 & 0 & 1 & 0 & 2 & 0 & \dots \end{matrix} \right) \end{matrix}$$

**Abbildung 3.1:** Exmpl. Matrix für Event  $E_1$  (M01=1)

Die Matrix 3.1 zeigt exemplarische Häufigkeiten für das Event  $E_1$  (M01=1) über einen Zeitraum von fünf Wochen für 11:25-11:34 Uhr: Der Bewegungsmelder mit der Kennung "M01" hat eine Bewegung erkannt.

Bezugnehmend auf den Tagesablauf 2.2 der Persona "Hildegard" und die Montags zwischen 09:30-11:35 Uhr vorherrschende ADL (Raum und Zeit gestalten: Fernsehen) ist eine Auswertung der absoluten bzw. durchschnittlichen Häufigkeit für ein einminütiges Zeitfenster augenscheinlich nicht relevant und verursacht viele FPs (Anomalie fälschlicherweise erkannt!).

Wenn auch [Mun15, S.8] eine initiale Näherung der Ausführungszeit einer ADL an den doppelten Pflegebemessungszeiten SGB XI orientiert, sind die Zeitfenster für die Verhaltensauswertung vielmehr aus den gemessenen Daten in der ausführenden Umgebung und an dem Individuum bzw. den Individuen, welche in dieser Umgebung leben, als Teil der Lernphase zu explorieren.

Die Zeitfenster  $t_\Delta$  werden in einem sensor- und tagesspezifischen Zeilenvektor  $T(S_i^d)$  festgehalten - mit  $i = \text{Sensorerkennung}$  und  $d = \text{Wochentag}$ . Die Anzahl der  $n$ -Spalten von  $T(S_i^d)$  ist abhängig von den explorierten Zeitfenster  $t_\Delta$ , d.h. eine Spalte beschreibt den Start- und Endzeitpunkt von  $t_\Delta$ . In einer Zelle  $T(S_i^d)_k^1$  mit  $1 \leq k \leq n$  ist die für das Zeitfenster  $t_\Delta$  kumulierte Häufigkeit der sensor-zugehörigen Eventmatrizen hinterlegt.

Die Zeitfenster  $t_\Delta$  werden mit einem Algorithmus exploriert, dessen Semantik mit folgenden Schritten erörtert ist:

- 1 Deserialisiere den Gesamtbestand der Eventmatrizen.
- 2 Iteriere über die persistierten Eventmatrizen  $M(E_i^w)$ .

- 3 Deserialisiere den Datenbestand einer Eventmatrix und sortiere aufsteigend nach Wochentag.
- 4 Iteriere über jeden Zeitpunkt im Datenbestand.
- 5 Setze den Startpunkt vom derzeitigen Zeitfenster  $t_\Delta$ .
- 6 Überprüfe die Zeitdifferenz  $t_{diff}$  vonzeitigem Zeitpunkt und Folgezeitpunkt.
- 7 Ist  $t_{diff}$  überschritten, lege Folgezeitpunkt als Startpunkt für neues Zeitfenster  $t_\Delta$  fest. Ansonsten erweitere das Zeitfenster  $t_\Delta$  um Folgezeitpunkt und wiederhole (6) bis Ende des Datenbestandes.
- 8 Serialisiere die explorierten Zeitfenster einer Eventmatrix.
- 9 Nächste Iteration in (2) - ansonsten Ende.

In Kapitel 4 sind die Ergebnisse der Evaluierung für die Zeitdifferenz  $t_{diff}$  erörtert.

Für eine automatisierte Erkennung von ungewöhnlichen Verhalten sind die Routinen, eine Kombination aus der Automatisierungslogik von openHAB und den Java-Klassen des SHLM, zu implementieren, die in der Erkennungsphase am Endzeitpunkt eines explorierten Zeitfensters  $t_\Delta$  prüfen, ob das aufgestellte Normalverhalten aus der Lernphase mit den in  $t_\Delta$  eingegangenen Events übereinstimmt bzw. keine große Abweichung aufweist. Die Routinen beachten dabei die vier Fehlerklassen der ADL:

- a *Auslassungen*: Hildegard schaut nicht wie gewohnt "Sturm der Liebe". Was macht sie stattdessen? Oder liegt sie regungslos auf dem Boden?
- b *Wiederholungen*: Hildegard hält sich länger im Wohnzimmer auf und schaut mehr fern als gewohnt.
- c *Verzögerungen*: Hildegard liegt ungewohnt lange im Bett. Ist alles in Ordnung?
- d *Vertauschungen*: Anstatt nach dem Aufstehen sich zu Waschen bereitet Hildegard erstmal das Frühstück zu.

Ob eine Verhaltensabweichung vorliegt, ist abhängig von den gemessenen Häufigkeiten in der Lernphase und den aufgezeichneten Häufigkeiten des Eventstreams in der Erkennungsphase. Zur Abgrenzung der durchschnittlichen Häufigkeiten des Normalverhaltens  $N(d, t_\Delta, S_i) = (T(S_i^d)_k^1)_{norm}^{t_\Delta}$  von den Häufigkeiten für das aktuelle Verhalten  $A(d, t_\Delta, S_i) = (T(S_i^d)_k^1)_{akt}^{t_\Delta}$  wird ein manueller Schwellwert eingesetzt, die der Persona "Alexandra" oder ihrem Sohn "Markus" als Abstufungen von "Das AAL-System ist {träge, ausgewogen, sensibel} eingestellt." präsentiert wird und einer einmaligen Auswahl bedarf. Für eine Produktreife sind detaillierte Beschreibungen anzufertigen, welche die Persona für diese Auswahl ausreichend sensibilisieren. Die Einstellungen "träge", "ausgewogen" und "sensibel" sagen aus, dass  $A(d, t_\Delta, S_i)$  maximal 75%, 50% und 25% von  $N(d, t_\Delta, S_i)$  abweicht:  $abs\_diff(A, N) > \text{Schwellwert}$ . Ein "träges" System erzeugt mehr FNs, hingegen ein "sensibles" System erzeugt mehr FPs: Die Ergebnisse in Kapitel 4 schlüsseln die Zusammenhänge der Abstufungen auf. Für die oben genannten Fehlerklassen der ADL bedeutet dies:



- (a): Der in der Erkennungsphase erwartete Eventstream bleibt aus. Die Routine iteriert  $\forall i$  über die Vektoren  $T(S_i^d)$  und errechnet, dass für  $N(d, t_\Delta, S_i)$  "normalerweise" ein Verhalten mit gegebener Häufigkeit vorliegt: Verhaltensabweichung detektiert.
- (b)+(d): Der eingehende Eventstream zeigt unterschiedliche Events. Die Routine iteriert  $\forall i$  über die Vektoren und errechnet für das derzeitige Zeitfenster  $t_\Delta$  ob  $abs\_diff(A, N)$  den Schwellwert überschreitet.
- (c): Falls die am oder im Bett verbaute Sensorik keine Polysomnographie umsetzt, ist der Eventstream der Fehlerklasse (a) zuzuordnen. Andernfalls zeigt der Eventstream gleiche Charakteristik wie bei Fehlerklasse (b)+(d).

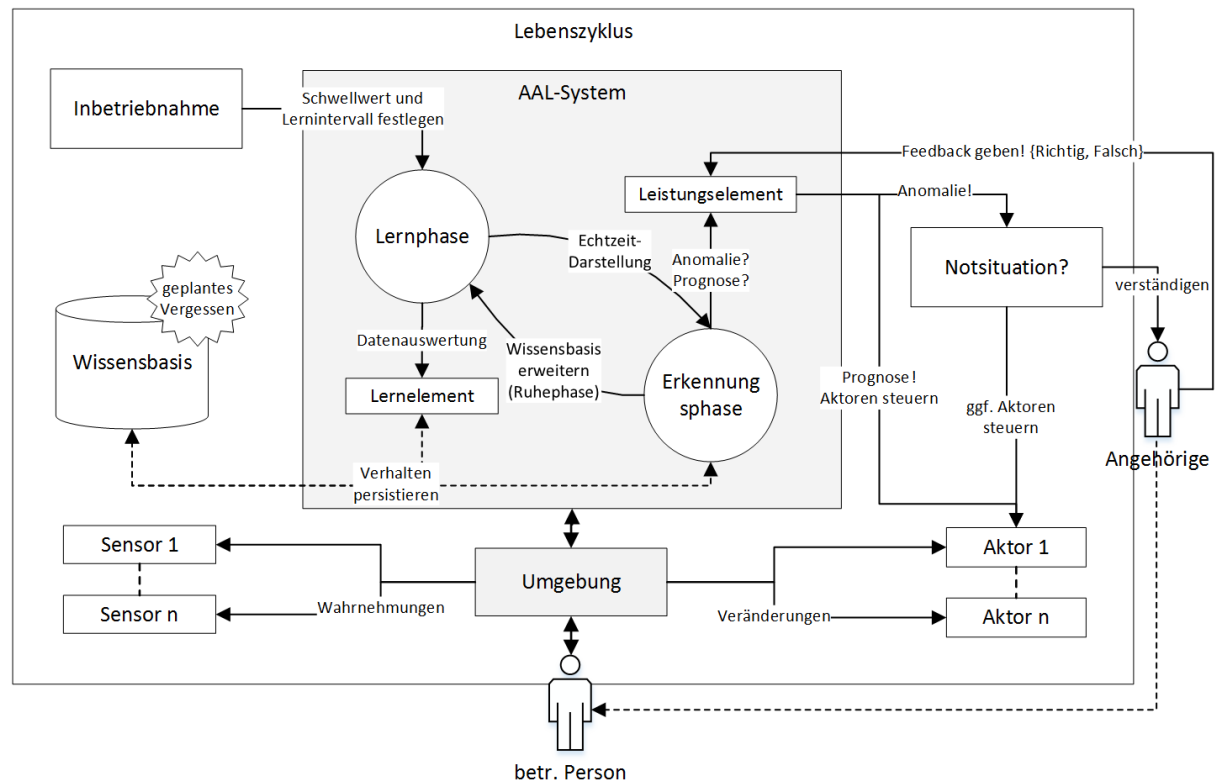
(2) Zur Beantwortung der zweiten Frage wird der CPT+- Algorithmus eingesetzt. Der CPT+ ist eine Weiterentwicklung vom CPT-Algorithmus, ist bis zu 4,5 mal schneller und erzielt eine höhere Genauigkeit (+5%) [Ted+16, S.12]. Im Vergleich zu anderen Prognose-Modellen, wie den First order Markov Chains (PPM), AKOM, dem Dependency Graph (DG), Transition Directed Acyclic Graph (TDAG) (LZ78-Algorithmus) zeigt der CPT+ bei Simulationen mit sieben unterschiedlichen Datensätzen mitunter die höchste Genauigkeit [Ted+16, S.11]. Der interessanteste verwendete Datensatz ist die Bibel: Jeder Satz entspricht einer Sequenz. Für ein Satzfragment wird der nächste Buchstabe bzw. das nächste Satzzeichen prognostiziert [Ted16]. Interessant ist der Datensatz nicht nur deshalb weil der CPT+ die besten Resultate liefert, sondern auch weil die mit der openHAB-Software persistierten Eventsequenzen als zeitliche Abfolge von Sensorevents vorliegen und für ein Zeitfenster als Satz bzw. Satzfragment gewertet werden können. In der Evaluation (3.6) soll gezeigt werden, ob die in [Ted+16, S.11] getroffene Aussage, dass der CPT+ - Algorithmus im Vergleich zu AKOM eine höhere Genauigkeit liefert, auch für die vorliegenden persistierten Eventsequenzen gültig ist. Die in Abschnitt 2.6 gelistete Bibliothek (7)-SPMF verfügt über eine Implementierung des CPT+-Algorithmus. Als Eingabe erwartet der CPT+ ein bibliothek-eigenes Format, das die einzelnen Events einer Eventsequenz als Zahl benötigt [FV16b]. Für die Ausführung des Algorithmus mit den gemessenen Daten sind die Events auf Zahlen abzubilden:

<b>Sensor=Wert:</b>	M01=0	M01=1	...	M04=0	M04=1	M05=0	M05=1	C01=0	...
<b>Zahl:</b>	1	2	...	7	8	9	10	11	...

**Tabelle 3.3:** Mögliche Abbildung von Sensorevents auf Zahlen

Die Teilsequenz "M04=1,C01=1,M05=0,C01=0,M01=0,C02=0,M05=1" aus Tabelle 3.2 kann somit auf "8,12,9,11,1,13,9" abgebildet werden: Der CPT+-Algorithmus prognostiziert dann "2" bzw. "M01=1", also eine Bewegung im Sichtbereich vom Bewegungsmelder "M01" (Erkennungsphase). Die notwendige Wissensbasis - das "Buch", das der CPT+ benötigt um eine Prognose zu stellen, wird aus den gemessenen Daten für unterschiedliche Zeitdifferenzen  $t_{diff}$  exploriert (Lernphase).

Die Abbildung 3.2 zeigt den Lebenszyklus des künftigen AAL-Systems. Nach Installation und Konfiguration der Sensoren (herstellerabhängig) legt die Persona zur Inbetriebnahme des AAL-System den Schwellwert und das Lernintervall fest: "träge, ausgewogen oder sensibel" und Anzahl der Wochen, die das System als initiale Wissensbasis nutzen soll.

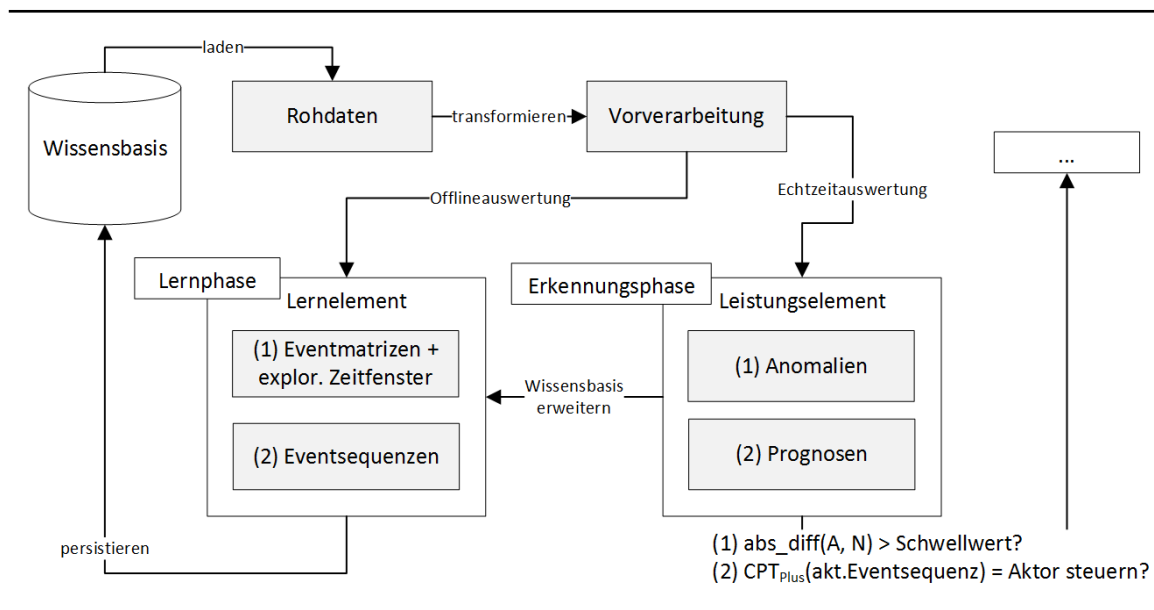


**Abbildung 3.2:** Lebenszyklus des AAL-Systems

In der Abbildung 3.3 ist das oben beschriebene Konzept der Datenanalyse mit Lern- und Leistungselement visualisiert. Als Abschluss der Lernphase wird das vom Lernelement erzeugte Verhalten in der Datenbank persistiert und das AAL-System geht automatisch in die Erkennungsphase bzw. Echtzeitanalyse über - der Angehörige sollte über diesen Zustandswechsel benachrichtigt werden.

In der Erkennungsphase wird der aktuelle Eventstream mit dem Leistungselement auf Anomalien hin untersucht und eine Prognose für ein zukünftiges Event gestellt. Bei Verhaltensabweichungen wird der Angehörige informiert. Auf die Benachrichtigung vom System kann der Angehörige ein Feedback geben, um den Lernprozess positiv zu beeinflussen: Die Anomalie wurde richtig erkannt oder ist nicht korrekt. Abhängig vom gegebenen Feedback kann sowohl das zum Zeitpunkt  $t$  hinterlegte Zeitfenster  $t_{\Delta}$  verringert oder erhöht als auch der genutzte Schwellwert angepasst werden. Um eine Historie der ausgerufenen Anomalien und Prognosen zu gewährleisten, bietet es sich an, diese zu persistieren. Bei Prognosen kann ein Aktor gesteuert werden, um der betroffenen Person als Unterstützung im alltäglichen Leben zu fungieren.

Ist an einem Tag oder innerhalb einer Woche keine Verhaltensabweichung gegeben, kann die Wissensbasis in einer tagesspezifischen Ruhephase erweitert bzw. ein Normalverhalten erneut erlernt werden. Eine tagesspezifische Ruhephase ist dort anzutreffen, wo das initial erlernte Verhalten keinen Eventstream verzeichnet: Exemplarisch nachts wenn die Sensoren mit kategorischen Messdaten keine Events erzeugen oder tagsüber wenn die Persona außer Haus ist. Ein geplantes Vergessen sorgt dafür, dass alte Verhaltensweisen nicht in den aktuellen Lernprozess einfließen, um somit die Verhaltensveränderungen der Persona über die Zeit zu berücksichtigen.



**Abbildung 3.3:** Konzept der Datenanalyse

Insgesamt stellt das AAL-System einen Agenten dar, der seine Umgebung teilweise beobachtet, kooperativ mit der openHAB-Software als einfacher reflex Agent, deren zugrundeliegender Architektur und Automatisierungslogik agiert und eine stochastisch episodische Aufgabenumgebung besitzt. Für die (Offline-)Lernphase verfolgt das AAL-System den Ansatz des unüberwachten Lernens. Durch das Feedback, das der Angehörige dem AAL-System auf ausgerufene Anomalien geben kann, stellt die (Online-)Erkennungsphase ein bestärkendes Lernen mit Belohnungen und Bestrafungen dar.

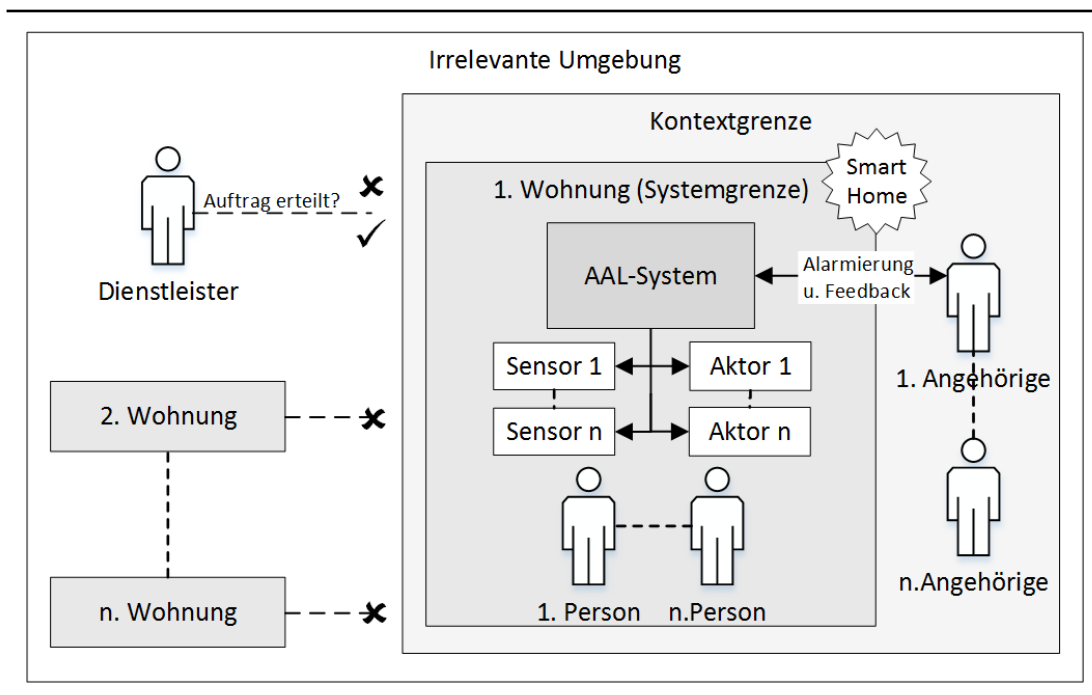
## 3.2 Anforderungsanalyse und Entwurf des Systems

Die Anforderungsanalyse (Requirements Engineering) ist in [Poh+15, S.4 f.] in vier grobe Schritte aufgeteilt:

- (A) *Ermitteln* der Anforderungen aus verschiedenen Quellen.
- (B) *Dokumentieren* der Anforderungen durch natürliche Sprache oder Modelle.

- (C) *Prüfen und abstimmen* der manifestierten Anforderungen durch definierte Qualitätskriterien.
- (D) Das Schritte übergreifende *Verwalten* von Anforderungen, um diese zu strukturieren, zu bearbeiten oder umzugestalten.

Vor der Anforderungsermittlung muss die System- und Kontextabgrenzung erfolgen, für deren Dokumentation unterschiedliche Modellierungssprachen (z. B. UML) bzw. Diagramme eingesetzt werden können [Poh+15, S.20]. Die folgende Abbildung 3.4 zeigt die Abgrenzung für das AAL-System:



**Abbildung 3.4:** System- und Kontextgrenze des AAL-Systems (Proof-of-Concept) [i.A.a. Poh+15, S.15 ff.]

Innerhalb der Systemgrenze sind die Sensoren und Aktoren, die Wohnung (Smart Home), das AAL-System und die betroffene Person (hier: Hildegard) anzutreffen. Die Angehörigen (hier: Alexandra und evtl. Markus), die alarmiert werden oder Feedback geben, fallen bereits in den Systemkontext. Weitere Systemkomponenten oder Stakeholder sind innerhalb der Kontextgrenze nicht vorgesehen. Alles außerhalb des Systemkontextes ist der irrelevanten Umgebung zuzuschreiben, d.h. andere Wohnungen stehen nicht im direkten Kontakt und haben keinen Zugang auf die Systemgrenze. Sollten die Stakeholder einen Auftrag an einen externen Dienstleister stellen (z. B. Installation der Aktoren), erhält dieser Zugriff auf das System.

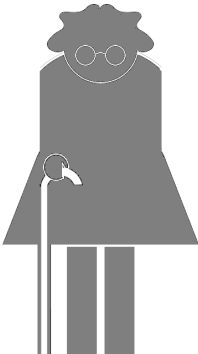
### **(A)** Anforderungen ermitteln

[Poh+15, S.8 f.] unterscheidet insgesamt drei Arten von Anforderungen:

- (1) *Funktionale Anforderungen (FA)* definieren Funktionalitäten und Dienste, die von den Systemkomponenten bereitgestellt werden sollen.
- (2) *Qualitätsanforderungen (QA)* (oder *nichtfunktionale Anforderungen*) beschreiben sich als Anforderungen, die qualitative Merkmale von Funktionen oder Komponenten des Systems definieren (z. B. Performanz, Zuverlässigkeit und Portabilität).
- (3) *Randbedingungen (RB)* schränken "den Lösungsraum im Entwicklungsprozess" ein und sind unabhängig von Nutzerzielen durch organisatorische oder technologische Gegebenheiten des Auftrages gestellt.

Um die Anforderungen zu ermitteln, können drei Quellen dienen [Poh+15, S.21]. Die Stakeholder (z. B. Nutzer, Betreiber, oder Auftraggeber) können befragt werden oder formulieren und priorisieren die Anforderungen zielgerichtet. Allgemeingültige Dokumente (z. B. das arc42-Template), vorhandene Anforderungsdokumente oder Fehlerberichte aus existierenden branchenspezifischen Systemen können als Strukturstütze und als Beispiel dienen. Die dritte Quelle beschreibt Systeme die bereits in Betrieb sind und den Stakeholdern als direkten Vergleich zu dem künftigen System dienen.

Die Personas sind Nutzer, Betreiber und Tester (UX), der Autor der vorliegenden Arbeit ist Entwickler, Architekt und Tester (Quellcode) des Systems. Abgeleitet von den Personabeschreibungen folgen Steckbriefe (Tab. 3.4 u. 3.5) für die Stakeholder "Hildegard" und "Alexandra", um die Zusammenhänge zwischen Nutzerzielen und notwendigen Anforderungen zu verdeutlichen:

<b>Hildegard</b> ( <i>primäre Persona</i> )	
	<p><i>Allgemein:</i></p> <p>79 Jahre alt, pensionierte Reinigungskraft, Familienstand: verwitwet, mental fit jedoch starrsinnig und körperlich gebrechlich.</p>
	<p><i>Wünsche &amp; Ziele:</i></p> <p>Möchte weiterhin zu Hause wohnen bleiben und auf Hilfe in Notsituationen nicht lange warten müssen. Ist auf den Schutz ihrer Privatsphäre bedacht. Kann nicht mit heutiger Technik mithalten.</p>

**Tabelle 3.4:** Steckbrief der Persona "Hildegard"

Als Quellen für die Analyse dienen die Wünsche & Ziele der Personas und das eruierte Wissen von bestehenden AAL-Systemen (State-of-the-Art). Für das Ermitteln und die Dokumentation der Anforderungen wird das arc42-Template und der IEEE-42010-Standard zu Rate gezogen.

Ausgehend von den Quellen sind die Anforderungen wie folgt spezifiziert:

FA: Funktionale Anforderungen

- FA1 Messdaten in Echtzeit in der Datenbank abspeichern.
- FA2 Aufgezeichnete Messdaten auswerten und initiales Normalverhalten erzeugen.
- FA3 Echtzeitauswertung des Eventstreams und Abgrenzung zum persistierten Normalverhalten.
- FA4 Detektierte Anomalien in der Datenbank abspeichern.
- FA5 Alarmierung der Angehörigen bei erkannter Verhaltensabweichung.
- FA6 Feedback-Funktion für Angehörige, um FPs zu kennzeichnen oder das System zu stärken.
- FA7 Eventsequenzen in der Datenbank abspeichern.
- FA8 Prognose-Modell für Eventsequenzen von unterschiedlichen Zeitfenstern erzeugen.
- FA9 Echtzeitauswertung des Eventstreams und Eventsequenzen identifizieren.
- FA10 Eventprognose aus identifizierten Eventsequenzen stellen.
- FA11 Tagesspezifische Ruhephase explorieren.
- FA12 Abgespeichertes Normalverhalten in der Ruhephase erweitern bzw. aktualisieren.
- FA13 Geplantes Vergessen von älteren Sensordaten.
- FA14 Umsetzung einer simplen GUI, die technologischen Wissensstand der Persona aufgreift und eine zufriedenstellende Mensch-Computer-Interaktion begünstigt.

#### QA: Qualitätsanforderungen

- QA1 Eine Verhaltensabweichung in der Echtzeitauswertung wird innerhalb weniger Sekunden erkannt.
- QA2 Die Anzahl der eingesetzten kategorischen Sensoren ist nicht begrenzt.
- QA3 Das System ist nicht an einen spezifischen Wohnraum gebunden.

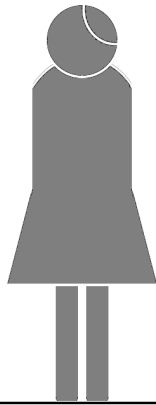
#### RB: Randbedingungen

- RB1 Der Einsatz von audio-visuellen Sensoren wird strikt abgelehnt. Ein höheres Akzeptanzniveau wird durch unauffällige Sensorik gewährleistet.
- RB2 Die rigorose Kontextgrenze wahrt den Schutz der Privatsphäre: Aufgezeichnetes Verhalten verweilt lokal und ein Zugriff von außerhalb ist nicht möglich.
- RB3 Die openHAB-Software und deren Automatisierungslogik bietet die zeitbasierte Ausführung von Prozessen an.

### (B) Anforderungen dokumentieren

Für die Dokumentation der Anforderungen kann jede "mehr oder weniger" formale Darstellung genutzt werden, welche die Kommunikation zwischen den Stakeholdern (hier: Leser und Autor) begünstigt [Poh+15, S.35]: Möglich sind demnach Formulierungen in natürlicher Sprache (z. B. das Use-Case-Template nach Bruegge und Dutoit [Ber+16]) oder Visualisierungen mit Modellierungssprachen wie z. B. die Unified Modeling Language (UML) und deren Use-Case-Diagramme [Sco16].

Die Abbildung 3.5 zeigt das Use-Case-Diagramm für die funktionale Anforderung "FA1". Die Darstellungen der Anforderungen "FA2-FA14" und "QA1" sind im Anhang 6.4 gelistet. So-

**Alexandra** (sekundäre Persona)*Allgemein:*

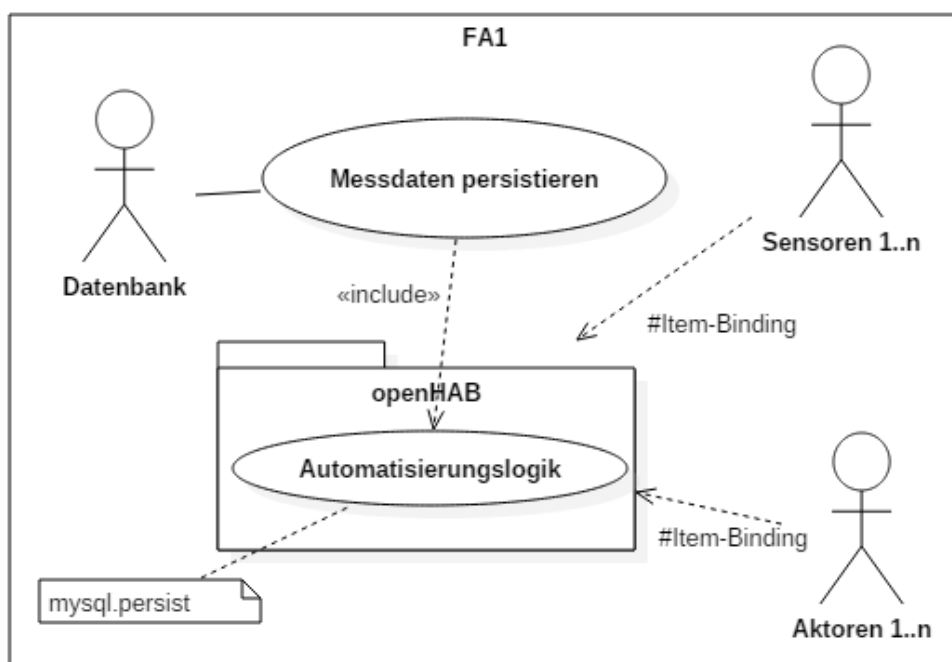
54 Jahre alt, ambitionierte Kauffrau, Familienstand: verheiratet mit Manfred, Kinder: Markus(24) und Marie(22), agil, engagiert und opferungsvoll.

*Wünsche & Ziele:*

Sie ist um ihre Mutter besorgt, möchte ihre Freizeit wieder mit ihrem Mann gestalten. Steht neuer Technik weltoffen gegenüber. Nimmt eine Fehlalarmierung vom System in Kauf - auch unterwegs.

**Tabelle 3.5:** Steckbrief der Persona "Alexandra"

wohl die Qualitätsanforderungen "QA2-QA3" als auch die Randbedingung "RB2" sind in der Abbildung 3.4 berücksichtigt. Die Randbedingungen "RB1+RB3" grenzen den Lösungsraum in der Entwicklung ein (Abschnitt 3.5).



**Abbildung 3.5:** Use-Case-Diagramm: FA1

Die Darstellung der "FA1" ist wie folgt zu interpretieren: Die in der Wohnung verbauten Sensoren und Aktoren sind über das hardware-spezifische openHAB-Binding (OSGi-Bundle) mit der Hausautomationslösung gekoppelt und als "Items" in der Laufzeitumgebung verfügbar. Die Automatisierungslogik kann mithilfe des MySQL-Bindings die zur Laufzeit eintretenden Werteänderungen der Items in der Datenbank ablegen.

### (C+D) Prüfen, abstimmen und verwalten

Sowohl der Schritt "Prüfen und abstimmen" als auch der Schritt "Verwalten" sind in [Poh+15, S.95 f.+ S.119 ff.] als Prozesse beschrieben, die im Verlauf der Anforderungsanalyse iterativ wiederholt werden. Geprüfte Anforderungen werden zur Abstimmung den Stakeholdern vorgelegt, um Fehler (z. B. Mehrdeutigkeit oder Widersprüche) zu entdecken oder eine Freigabe zu erhalten. Zudem werden Anforderungen anhand ihrer Attribute in Sichten mit unterschiedlichen Selektionen oder in Priorisierungsmatrizen verwaltet. Die oben definierten Anforderungen können keinen fiktiven Personen vorgelegt werden. Eine Verwaltung mit "Requirements-Management-Werkzeugen" ist für das Proof-of-Concept überdimensioniert.

Die visualisierten Diagramme teilen die eruierten Anforderungen in Anwendungsfälle ein, welche die Herausforderung an die Realisierung des AAL-Systems in Teilaufgaben separiert: Die Daten und das Verhalten sind klar getrennt. Für eine hinreichende Kontextsicht des Systems sind zudem Teilsysteme in den Diagrammen verzeichnet. Um den Entwurf des Systems abzuschließen, ist die Softwarearchitektur genauer zu beschreiben. In [Hru+14, S.31 ff.] ist erwähnt, dass erfolgreiche Softwarearchitekten vier unterschiedliche Sichten auf ihre Systeme darlegen:

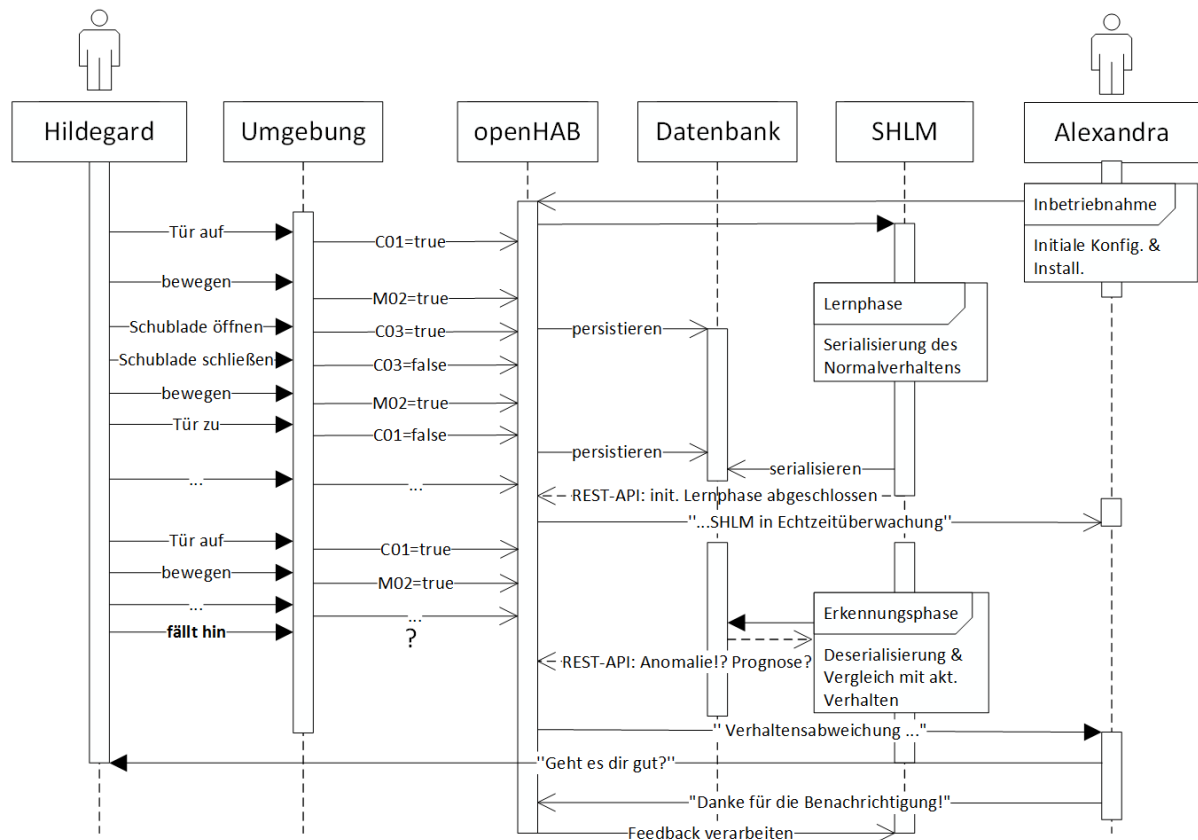
- 1 Die *Bausteinsicht* zeigt die Architekturelemente in unterschiedlichen Abstraktionsebenen (Zoomstufen) - vom Blackbox-System bis hin Klassen und Operationen. Die Abbildung einer Zoomstufe ist dem Softwarearchitekten überlassen.
- 2 Die *Verteilungssicht* oder *Deployment-Sicht* legt dar, welche Software auf welcher Hardware abläuft.
- 3 Die *Laufzeitsicht* verdeutlicht, wie Architekturelemente zur Laufzeit miteinander agieren - z. B. welche Methoden aufgerufen und ob Parameter übergeben werden.
- 4 Zu der Sicht der *technischen Konzepte* gehören Logging-Konzepte oder Details der GUI-Implementierung.

Eine Entwurfsentscheidung, welche die Beschaffenheit der Softwarearchitektur maßgeblich beeinflusst, ist der Einsatz der openHAB-Software: openHAB nutzt die Eclipse-Equinox. Die Equinox ist eine Realisierung der OSGi Service Plattform - ein Standard, der ein dynamisches (serverseitiges) Komponentensystem spezifiziert. In diesem Zusammenhang ist eine Komponente ein openHAB-Binding, das eine OSGi-Bundle-Implementierung darstellt. Da ein OSGi-Bundle Hot-Deployment-fähig ist und somit auch ein openHAB-Binding, kann der Funktionsumfang der Hausautomationslösung folglich während der Laufzeit erweitert werden [Mas, S.10+21].

Das UML-Sequenzdiagramm 3.6 zeigt den Entwurf der Laufzeitsicht mit hohem Abstraktionslevel. Die Personas Hildegard und Alexandra sind als Akteure und die Umgebung, openHAB-Instanz, Datenbank und das SHLM sind als Objekte dargestellt. Nachdem Alexandra (ggf. auch Markus) das AAL-System durch eine initiale Installation und Konfiguration in Betrieb genom-



men hat, veranlasst openHAB die Instanziierung der Lernphase im SHLM für ein gegebenes Zeitintervall.



**Abbildung 3.6:** UML-Sequenzdiagramm - Kommunikation zwischen Personas, openHAB und SHLM

Während der Lernphase geht Hildegard ihrem alltäglichen Leben nach und führt wie gewohnt die ADL aus: Sie bewegt sich z. B. in den unterschiedlichen Räumen ihrer Wohnung, öffnet oder schließt Schubladen und Türen. Die dadurch verursachten Werteänderungen der Sensoren werden in der openHAB-Instanz als Statusänderung der openHAB-Items registriert und zeitnah in der Datenbank persistiert. Als Abschluss der Lernphase wertet das SHLM die gespeicherten Daten aus und serialisiert die verwerteten Daten als tagesspezifische Verhaltensstrukturen ab. Als Überleitung zur Erkennungsphase kann Alexandra auch ergänzend benachrichtigt werden. In der Erkennungsphase wird die tagesspezifische Verhaltensstruktur deserialisiert und mit den aktuellen Statusänderungen der openHAB-Items - also mit den Aktivitäten die Hildegard gerade ausführt - verglichen. Neben möglichen Verhaltensprognosen, welche die openHAB-Instanz dazu veranlassen Aktoren zu steuern, ruft das SHLM die Verhaltensabweichungen aus und befiehlt der openHAB-Instanz Alexandra zu benachrichtigen, die dann wiederum mit Hildegard Kontakt aufnimmt, um sich über ihren Gesundheitszustand zu informieren. Auf die ausgerufene Anomalie kann Alexandra dem System positives oder negatives Feedback geben: In diesem Fall positiv, denn Hildegard ist gestürzt und konnte nicht mehr aus eigener Kraft aufstehen. Das Feedback wird vom SHLM in den Verhaltensstrukturen begünstigt.

Die Verteilungssicht, Bausteinsicht und weitere technische Konzepte werden in den Folgeabschnitten erbracht, um die durch die Verwendung von Teilsystemen und weiteren Software-Bibliotheken, welche insbesondere eine Kommunikation der Architekturelemente gewährleisten, entstehenden Entwurfsentscheidungen mit in das Systemdesign einfließen zu lassen.

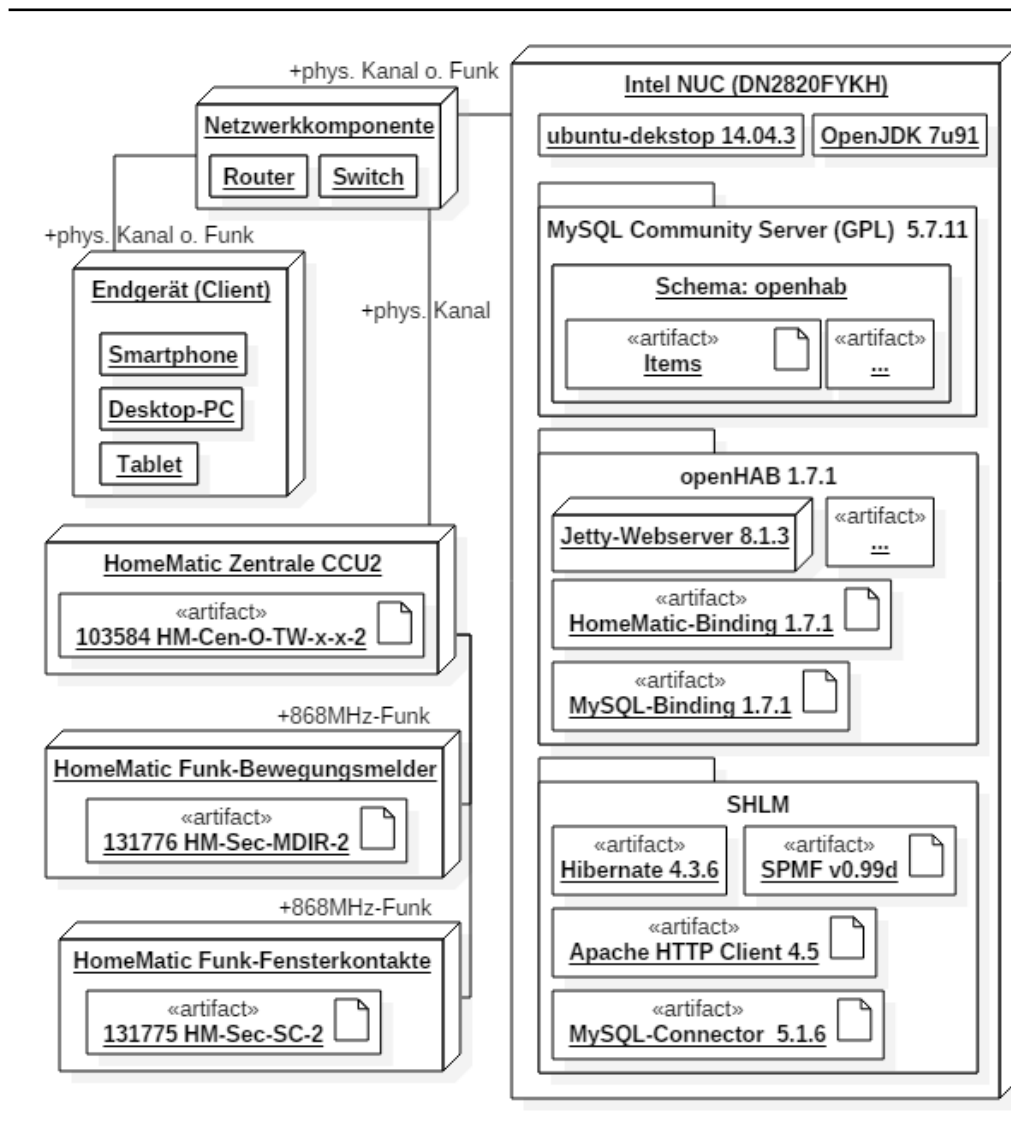
### 3.3 Eingesetzte Hard- und Software

Ausgehend von der Verteilungssicht (Abb. 3.7) sind es sechs verschiedene Hardwarekomponenten, die im AAL-System eingesetzt werden:

- 1 Der Mini-PC NUC (Next Unit of Computing) (Kit: DN2820FYKH) [Cor16b] von Intel.
- 2 Optionale Hardware, die durch den ISP (Internet Service Provider) gestellt ist - z. B. ein Router oder Switch.
- 3 Mögliche Endgeräte, die von den Persona zur Interaktion mit dem AAL-System genutzt werden können.
- 4 Die HomeMatic-Zentrale CCU2, die als Kopplungseinheit von Sensoren und Mini-PC dient, kann über die Homematic Benutzerschnittstelle "WebUI" konfiguriert werden [3 A16a].
- 5 Der HomeMatic Funk-Bewegungsmelder [3 A16c] ist durch den Batteriebetrieb frei platzierbar. Die Reichweite der Bewegungserkennung beträgt bei einem 90° Winkel bis 12m. Durch eine zweite Linse mit 360° Abdeckung ist ein Unterkriechschutz gewährleistet.
- 6 Der Funk-Fensterkontakt wie von HomeMatic als unauffälliger, batteriebetriebener Sensor deklariert, dessen Anbringungsort nicht nur auf Fenster festzulegen ist [3 A16b].

(1) Der NUC verfügt über einen Intel Celeron Prozessor (Dual-Core) mit bis zu 2,4GHz Taktfrequenz, 4GB RAM, eine 128GB HDD und über kabelgebundene bzw. kabellose Netzwerkschnittstellen. Durch das mitgelieferte VESA-Montageset kann der NUC unauffällig an Wohnwände und auch hinter Mobiliar angebracht werden. Auf dem NUC ist das quelloffene Ubuntu-Desktop (14.04.03) Betriebssystem und das Java-Entwicklungskit OpenJDK (7u91) installiert. Neben dem MySQL Community Server (5.7.11) [Cor16a], der für die Persistenz der Daten eingesetzt wird, ist auf dem NUC auch die openHAB-Software (1.7.1) und das SHLM vorzufinden. Das HomeMatic-Binding realisiert die Verbindung mit der HomeMatic Zentrale und das MySQL-Binding die Verbindung mit dem MySQL Server. Mit dem Mail-Binding kann ein möglicher Kommunikationskanal zu den Angehörigen verwirklicht werden. Das SHLM nutzt Hibernate (4.3.6 - LGPL V2.1) [JBo16] für das Object Relational Mapping (ORM), also den Vorgang, um die modulinternen Klassen bzw. deren Plain Old Java Objects (POJOs) während der Laufzeit auf die gespeicherten Daten abzubilden. Die Datenbankverbindung ist mit dem JDBC-Treiber (5.1.6) umgesetzt. Eine Implementierung des CPT+-Algorithmus ist durch die eingebettete SPMF-Bibliothek (0.99d) gegeben. Für die Kommunikation des SHLM mit der openHAB-

Instanz wird die Komponente "HttpClient" (4.5) der Apache "HttpComponents"-Bibliothek eingesetzt [Fou16a]. Die Apache Maven Software (3.3.9) ermöglicht ein Build-Management des SHLM als Uber-JAR [Fou16b].



**Abbildung 3.7:** UML-Verteilungsdiagramm - Verteilungssicht des AAL-Systems

(2) Die vom ISP gestellte Hardware (z. B. ein Router) kann als Netzwerkkomponente für die Kommunikation zwischen dem NUC, der CCU2 und einem Endgerät fungieren. Optional kann eine Kommunikation auch über ISP-fremde Hardware oder direkt zwischen den einzelnen Komponenten physisch gebunden oder per Funk realisiert werden. Die von den Personas geforderte Wahrung der Privatsphäre wird durch den Einsatz ISP-Hardware nicht verletzt - das gespeicherte Verhalten verbleibt im lokalen Netzwerk auf dem NUC.

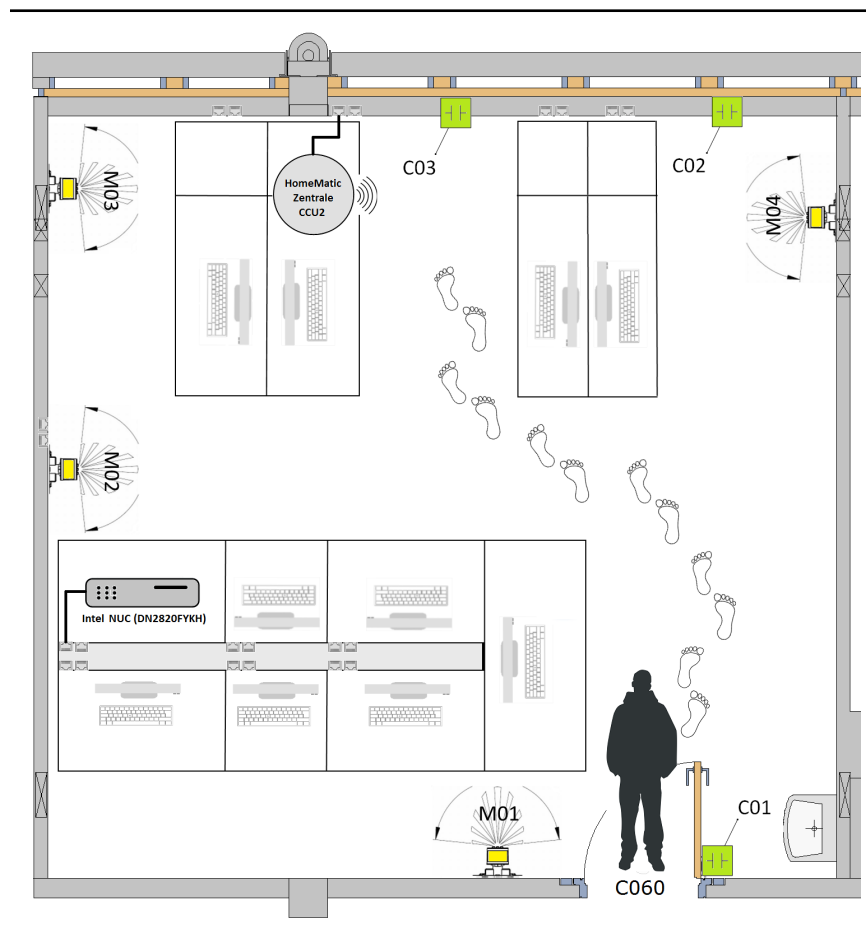
(3) Die GUI der openHAB-Software verfolgt den Ansatz des Responsive Webdesign (RWD), d.h. der Webserver "Jetty" erzeugt die Websites flexibel, sodass der Angehörige zur Steuerung des AAL-Systems sowohl ein mobiles als auch ein stationäres Endgerät verwenden kann.

(4-6) Für die Inbetriebnahme der HomeMatic Zentrale, Bewegungsmelder und Fensterkontakte ist keine weitere Software nötig.

### 3.4 Installation, Konfiguration und Aufbau

Der vorliegende Abschnitt erörtert die Installations- und Konfigurationsschritte, welche notwendig sind, um das AAL-System in Betrieb zu nehmen. Diese Schritte sind jedoch nicht zu vergleichen mit dem Aufwand, den die Persona für die Inbetriebnahme des AAL-Systems aufbringen muss, denn für eine zukünftige Produktreife können Betriebssysteme vorinstalliert und die Verbindungen von Sensoren, Aktoren und Zentralen bereits dynamisch vorkonfiguriert sein.

Ein initialer Testaufbau des Systems - visualisiert in Abbildung 3.8 - ist im Multimedia Communication Laboratory (MCLab) der Hochschule Bonn-Rhein-Sieg erfolgt. Die für den Aufbau notwendigen Arbeitsschritte sind bis auf (a)+(b) unabhängig von den gewählten Räumlichkeit und im Folgenden beschrieben:



**Abbildung 3.8:** Hardwareaufbau im "Multimedia Communication Laboratory" H-BRS Raum C060

- a Die Bewegungsmelder an die Wände montieren und Magnetkontakte an Fenster, Türen oder Schubläden anbringen.
- b Die HomeMatic Zentrale und den Intel NUC anschließen und in das Netzwerk integrieren.
- c Das OS, OpenJDK und die openHAB-Software auf dem NUC installieren.
- d Ein Pairing zwischen HomeMatic Zentrale und den Sensoren durchführen.
- e Den MySQL Server aufsetzen und notwendige Schemata anlegen.
- f Die Items-, Sitemap-, Persistence-, und etwaige Rule-Dateien mit dem openHAB-Designer für das gegebene Szenario entwerfen.
- g Das System nach erfolgreicher Installation und Konfiguration in Betrieb nehmen.

(a)+(b) Sowohl die Bewegungsmelder "M01-M04" als auch die Magnetkontakte "C01-C03" können mit den mitgelieferten Schrauben oder mit Kabelbindern und Klebesockeln montiert werden. Eine horizontale Wandmontage der Bewegungsmelder auf 1,2m Höhe, wie sie in [SI15, S.45] praktiziert wird, um Aktivitäten im Blindschatten zu detektieren, ist durch den Unterkriechschutz der verwendeten Melder hinfällig. Auch Hindernisse der Verkabelung zur Spannungsversorgung der Sensoren an der Decke treten durch den Batteriebetrieb nicht auf. Der NUC und die Zentrale sind im bestehenden Labornetzwerk angeschlossen und erhalten bei Anmeldung eine IP-Adresse vom existierenden Dynamic Host Configuration Protocol (DHCP)-Server.

(c) Das Ubuntu-Desktop OS und OpenJDK werden nach der in [ubu16b] bzw. [Cor16c] gestellten Anleitung auf der Festplatte des NUC's installiert. [Uh16] beschreibt eine plattform-spezifische Installation der openHAB-Software. Für die Kopplung der HomeMatic-Hardware an openHAB und die Persistenz der Messdaten werden die openHAB-Bindings "org.openhab.binding.homematic-1.7.1" und "org.openhab.persistence.mysql-1.7.1" eingesetzt.

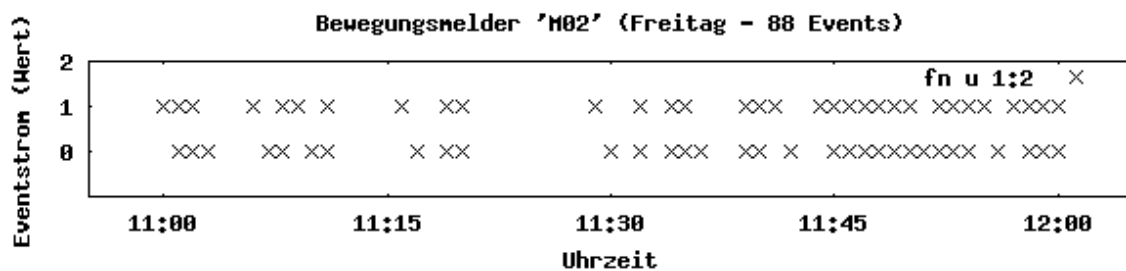
(d) [3 L16, S.148 ff.] zeigt das Anlernen von Geräten (hier: Bewegungsmelder und Magnetkontakt) an der HomeMatic Zentrale. Die Seriennummern und der eingestellte Funkkanal dienen als eindeutige Kennzeichner der Sensoren und sind für die Konfiguration der Items-Datei in (f) notwendig.

(e) Eine Installation des MySQL Servers ist in [ubu16a] geschrieben. Das Erzeugen der Datenbank, der Nutzer und erforderlichen Rechte wird in [gee16] erörtert.

(f) Das Erstellen der Artefakte (s. Abb. 3.13 Teilsystem "openHAB") kann mit dem openHAB-Designer oder einem Texteditor umgesetzt werden. Als eigenständige Desktop-Applikation bietet der openHAB-Designer eine zusätzliche Autovervollständigung, Syntaxhervorhebung und -überprüfung von der ein Endanwender profitieren sollte. Im Anhang 6.5 sind Auszüge von den openHAB-Artefakten gelistet. Die vollständigen Konfigurationsdateien sind im digitalen Anhang hinterlegt. Die Regeln für die automatisierte Analyse der Daten sind in der Beschreibung der Realisierung des Systems (3.5) gegeben.

(g) Nachdem alle Hard- und Softwarekomponenten installiert und konfiguriert sind, kann der openHAB-Server gestartet werden. Der Jetty-Webserver erzeugt die notwendigen Hypertext-Dokumente, die der Endanwender unter der in [Uh16] aufgeführten Adresse erreichen kann.

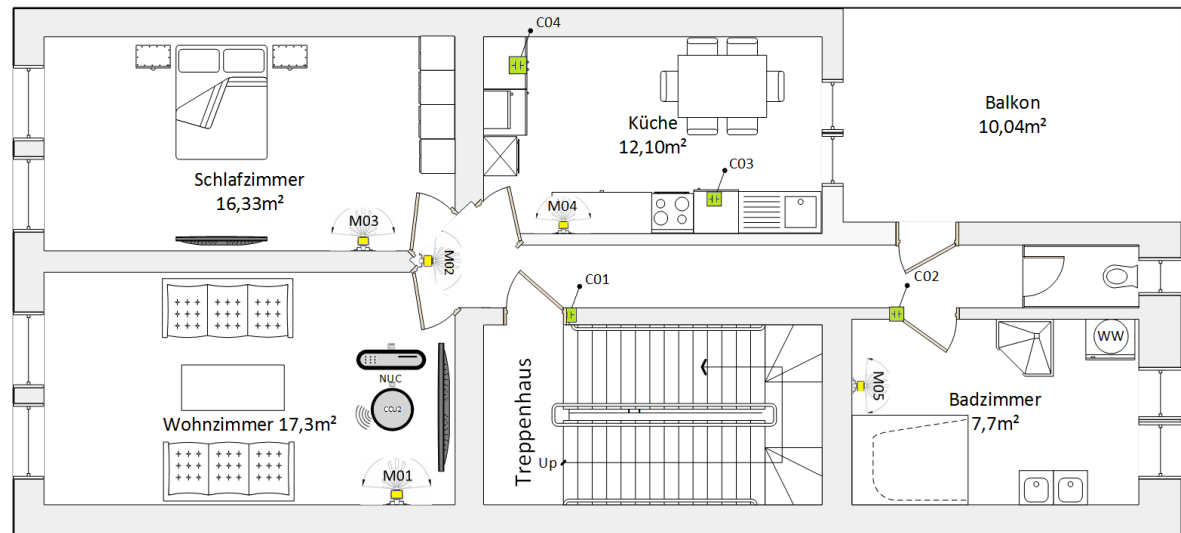
Der redundante Testaufbau im MCLab testet die Funktion der Hardware auf Ausfallsicherheit und mögliche Fehlaufzeichnungen der Sensoren. Das Aktivitätsaufkommen im MCLab ist zu vergleichen mit einem fünf Mann Büro mit Arbeitszeiten von 08:00 bis 20:00 Uhr. Für einen Zeitraum von 33 Tagen wurden mehr als 55k Event aufgezeichnet. Der Systemaufbau lief zuverlässig, d.h. es ist keine Hard- oder Softwarekomponente ausgefallen. Unter Betrachtung der nächtlichen Ruhephasen, in denen von den Magnetkontakten und Bewegungsmeldern keine Events erwartet werden, hat lediglich der Bewegungsmelder "M04" einmal eine Bewegung fälschlicherweise erkannt. Bei der gewählten Persistenzstrategie "everyChange" (s. Auszug openHAB-MySQL-Persistence 6.5) wird über den Tag hinweg im Durchschnitt alle 26 Sekunden ein Event gemessen.



**Abbildung 3.9:** Eventstream für Bewegungsmelder "M02" (Freitags 11-12Uhr)

Die Abbildung 3.9 zeigt den Eventstream für den Bewegungsmelder "M02" an einem Freitag zwischen 11:00-12:00 Uhr mit insgesamt 88 verzeichneten Events. Bei einem in der HomeMatic CCU2 eingestellten Mindestsendeabstand von 15 Sekunden sind die Wertigkeiten wie folgt zu interpretieren: Der Bewegungsmelder erkennt eine Bewegung (Wert=1). Findet innerhalb von 15 Sekunden keine Bewegung mehr statt, sendet der Bewegungsmelder, dass keine weitere Bewegung registriert wurde (Wert=0). Die dem Zeitraum (Fr. 11-12 Uhr) zugrundeliegende Aktivität ist das wöchentliche Labormeeting bei dem ca. 5-10 Studenten anwesend sind. Zwischen 11:00-11:45 Uhr präsentiert jeder Student in sitzender Position den Fortschritt seiner Arbeit. Danach ist zur offenen Diskussionsrunde geladen, bei der eine Vielzahl der Studenten ihren Beitrag in stehender Position leisten und somit, wie auch in der Abbildung verzeichnet, einen höher frequentierten Eventstream verursachen.

Ausgehend von den oben beschriebenen Installation und Konfiguration ist der Testaufbau "umgezogen" und wird nun seit dem Frühjahr 2016 in einer genuine Wohnung (Abb. 3.10) in einer Stadt im Bundesland "Rheinland Pfalz" eingesetzt (Anonymisierung auf Wunsch der Probanden).



**Abbildung 3.10:** AAL-System im Smart Home - Lokalisiert in einer Stadt in Rheinland Pfalz

Insgesamt sind in dieser Wohnung fünf Bewegungsmelder (Wohnzimmer, Flur, Schlafzimmer, Küche und Badezimmer) und vier Magnetkontakte (Wohnungstür, Badezimmertür, Küchenschublade und Kühlschranktür) angebracht. Der NUC und die CCU2 sind nur für eine vereinfachte Darstellung mitten im Raum gezeichnet und in der Realität unauffällig im TV-Board platziert. In der Abbildung 3.11 sind exemplarische Montagen der Sensoren gezeigt: Der Magnetkontakt der Badezimmertür (o.l.), der Magnetkontakt der Küchenschublade (u.l.), der Bewegungsmelder im Flur zwischen den Türzargen (o.r.) und der Bewegungsmelder in der Küche oberhalb der Türzarge (u.r.).

Aufgrund der im Rahmen dieser Arbeit limitierten Möglichkeiten zur Probandenfindung entsprechen die verfügbaren Probanden nur annähernd den Persönlichkeitsprofilen der beschriebenen Personas. In der Wohnung gestaltet eine dreiköpfige Familie mit zwei Haustieren (Hunde) ihren Alltag:

- Die *1. Person* ist männlich, Mitte Zwanzig und unregelmäßig an den Wochenenden und Werktags zu Besuch.
- Die *2. Person* ist weiblich, Mitte Zwanzig und regelmäßig Werktags anwesend. Tagsüber ist sie arbeiten und selten in den Mittagspausen vor Ort.
- Die *3. Person* ist weiblich, Mitte Vierzig und tagsüber regelmäßig zu Hause. Für gelegentliche Einkäufe oder Besuche bei Freunden ist sie außer Haus.
- Die Hunde (Höhe: 59cm und 28cm) sind gelegentlich für 1-2 Tage in der Woche anwesend. Abhängig von der Lichtintensität im Raum und Bewegungsgeschwindigkeit der Hunde konnte festgestellt werden, dass die Melder eine Bewegung erkennen.

Der Anhang 6.6 (Teil 1 +2) zeigt die angefallenen Eventstreams des Bewegungsmelder im Wohnzimmer für den Wochentag "Sonntag" an vier aufeinanderfolgenden Wochen. In der



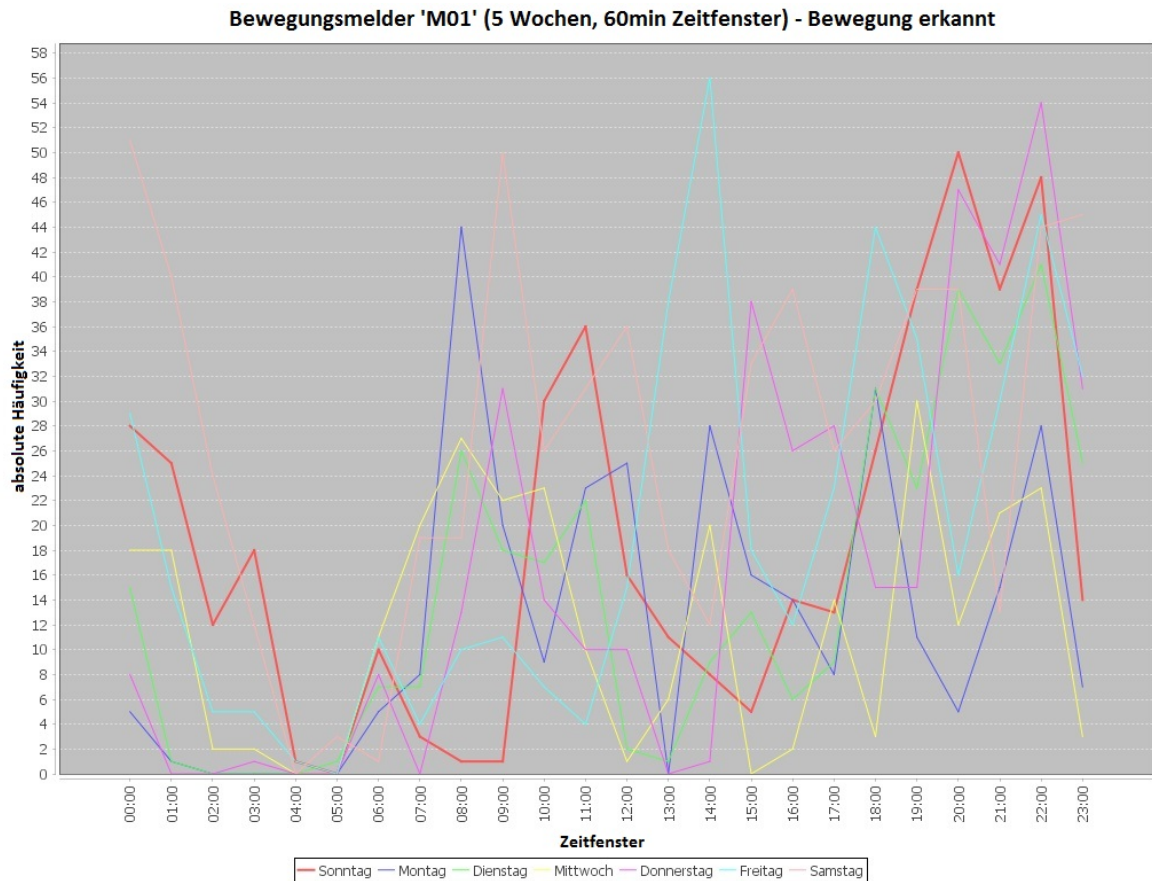
**Abbildung 3.11:** Exemplarische Anbringung der HomeMatic Bewegungsmelder und Magnetkontakte

dritten Woche sind alle drei Probanden am frühen Mittag bis in die späten Abendstunden in der Wohnung anwesend: In Abgrenzung zu den anderen drei Wochen sind die Events in deutlich erkennbar kürzeren Zeitabständen aufgetreten.

Die Abbildung 3.12 zeigt die absoluten Häufigkeiten des Bewegungsmelders im Wohnzimmer über fünf Wochen. Gehäuft sind die Events für einstündige Zeitfenster. Die Zeitfenster sind nicht zu verwechseln mit den explorierten Zeitfenstern  $t_{\Delta}$ , die in Abschnitt 3.1 erörtert sind. Jeder Graph repräsentiert einen einzelnen Wochentag. Die Visualisierung bestätigt das in [Mun15, S.30] erfasste Ergebnis, welches besagt, dass die Wochentage zwingend zu unterscheiden sind, um die Genauigkeit der Vorhersage zu steigern. In den frühen Morgenstunden (0-7 Uhr) und in den späten Abendstunden (21-23 Uhr) scheinen die Wochentage vergleichbar, in den Mittagsstunden zeigt sich jedoch die tagesspezifische Verhaltensweise (ADL "Raum und Zeit gestalten" - Wohnzimmer) der Probanden.

Im digitalen Anhang (Messdaten und Auswertungen) sind sowohl die Visualisierungen der absoluten Häufigkeiten als auch die Eventstreams für die Magnetkontakte und die übrigen Bewegungsmelder hinterlegt.





**Abbildung 3.12:** Absolute Häufigkeit der erkannten Bewegung für Bewegungsmelder 'M01' (tages-spezifisch, einstündiges Zeitfenster)

## 3.5 Realisierung des Systems

Ausgehend von den Anforderungen (Abschnitt 3.2) und der Bausteinsicht (Abb 3.13) kann die Realisierung des Systems in vier Segmente geteilt werden:

- (I) Die (a) Datenvorverarbeitung, (b) Build-Maßnahmen und das (c) ORM.
- (II) Das Lernen von Verhalten ((a) Eventmatrizen u. (b) -sequenzen).
- (III) Die Kommunikation zwischen der Automatisierungssoftware und dem SHLM.
- (IV) Das (a) Erkennen von ungewöhnlichem Verhalten und das (b) Ausrufen von Prognosen.

**(I)** (a) Nachdem die Installation und Konfiguration des AAL-Systems - wie in Abschnitt 3.4 geschildert - durchgeführt wurde, ist die Kopplung zwischen den Magnetkontakten bzw. Bewegungsmeldern und der openHAB-Instanz realisiert. Zudem sind diese in den Artefakten `"*.items"` und `"*.sitemap"` hinterlegt. Die realisierte MySQL-Persistenz legt für jeden Sensor eine eigene Datenbanktabelle an. Der Name der Tabelle ist an der inkrementellen Namensgebung der openHAB-Items orientiert, d.h. ausschlaggebend ist nicht der Bezeichner, der in dem `"*.items"`-Artefakt angegeben ist, sondern der von der Laufzeitumgebung bei erstma-

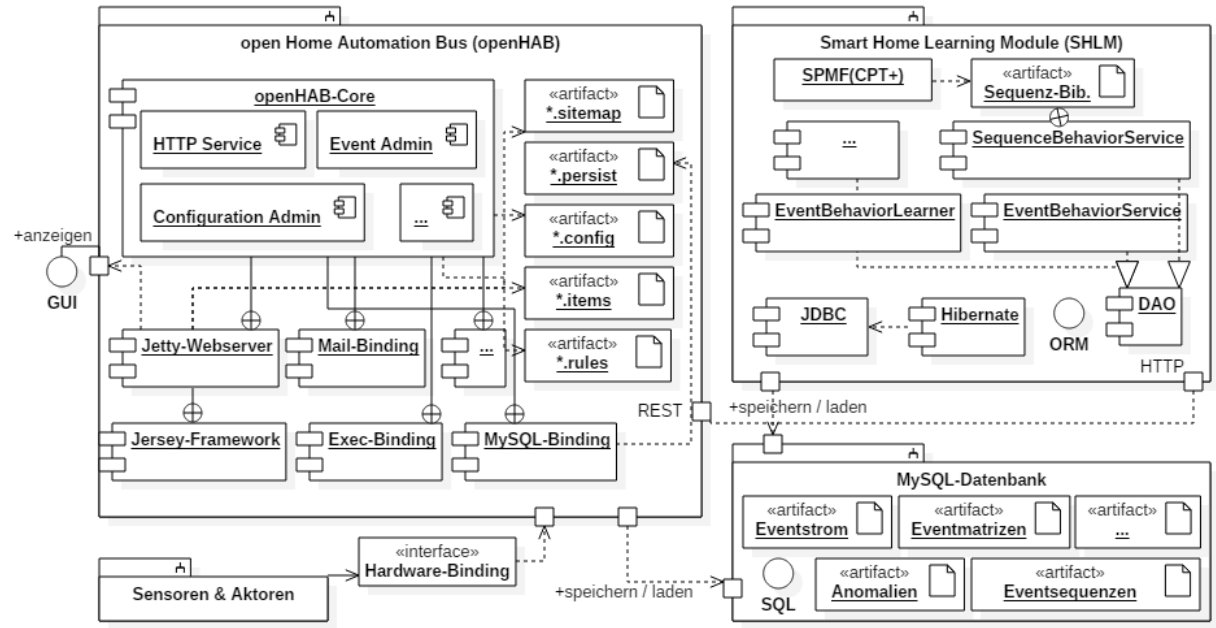


Abbildung 3.13: UML-Komponentendiagramm - Bausteinsicht des AAL-Systems

liger Ausführung vergebene Kennzeichner "item\*" mit  $1 \leq * \leq n$  - wobei  $n$  die Anzahl der im "\*.persist"-Artefakt angegebenen Items ist. In den "item\*" -Tabellen sind die sensor-spezifischen Zeit-Werte-Paare gespeichert. Neben diesen Tabellen legt die Persistenzlogik die Tabelle "items" an, in der das Mapping von Bezeichner der Items und Kennzeichner der Tabellen hinterlegt ist.

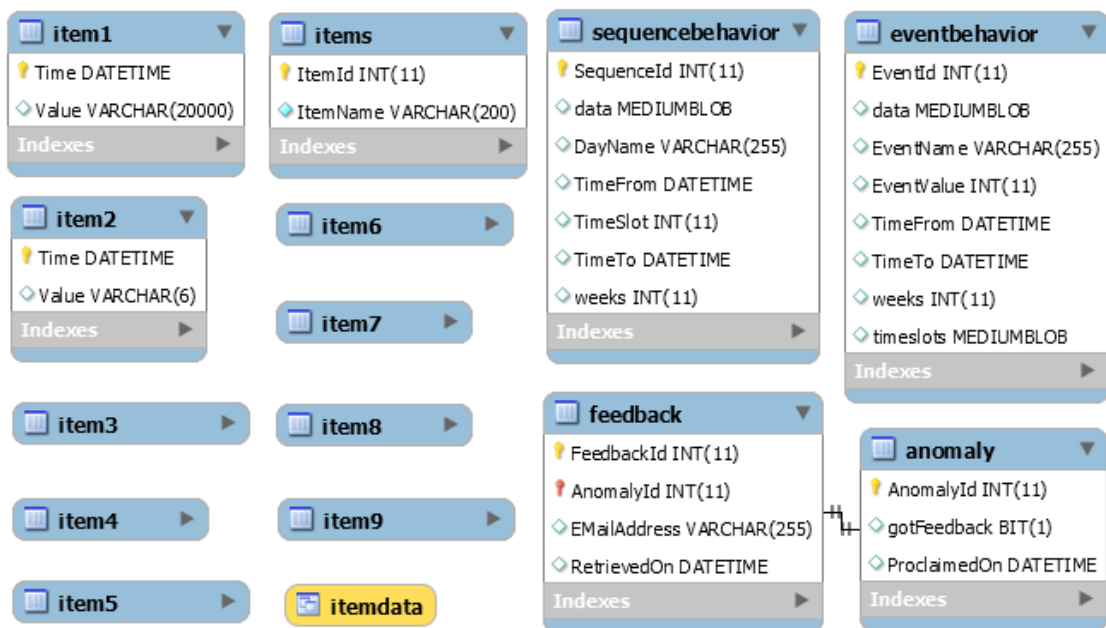


Abbildung 3.14: ERD: Datenbankschema "openhab"

Das Entity-Relationship-Diagramm (ERD) in Abbildung 3.14 visualisiert die automatisch angelegten Tabellen, deren Attribute und Datentypen. Alle weiteren Tabellen sind erst für die in den Abschnitten (II-IV) erörterte Implementierung relevant. Damit die Zeit-Werte-Paare aller Sensoren nicht getrennt betrachtet werden, sondern den erforderlichen Eventstream (angelehnt an Tabelle 3.2) bilden, ist die View "itemdata" angelegt. Die View besteht aus vier Spalten: Für die anstehende ORM-Realisierung benötigt Hibernate für die POJOs einen eindeutigen Kennzeichner. Ein Zeitstempel ist in einer "item\*" -Tabelle als eindeutiger Primärschlüssel ausreichend. Bei der Zusammenlegung der Daten zeigen sich Uneindeutigkeiten für diesen Primärschlüssel, da es durchaus möglich ist, dass Events von unterschiedlichen Sensoren zeitgleich eintreten. Deshalb ist für die erste Spalte "Id" der Universally Unique Identifier (UUID) aus der MySQL uuid()-Funktion entnommen. Fortführend ist dann der Zeitstempel, der Sensorname ("M01-M05" und "C01-C04") und der Sensorwert ("0" und "1") als Spalte in der View deklariert.

(b) Damit das SHLM als eigenständiges Uber-JAR erzeugt und in der Java Virtual Machine (JVM) ausgeführt werden kann, setzt Maven auf die Konfiguration von Projektabhängigkeiten und -zyklen in der "pom.xml" Datei, die im Wurzelverzeichnis des Projektes abgelegt ist. Dabei folgt der Aufbau der "pom.xml" einer XML Schema Definition (XSD), sodass die notwendigen Projektabhängigkeiten zu Fremdressourcen, -Bibliotheken, Plugins und anderen Softwareprojekten im XML-Dialekt festgehalten und sowohl aus lokalen als auch externen Quellen zu beziehen sind. Die für das SHLM in der "pom.xml" eingebunden Abhängigen sind zum einen der Hibernate-Core, die MySQL-Connector-Java (JDBC Treiber) und Apache HttpComponents aus dem externen Maven-Repository, und zum anderen die lokal abgelegte SPMF-Bibliothek. Daneben beschreiben die verwendeten Maven-Plugins, wie die angegebenen Projektabhängigkeiten in den einzelnen Projektzyklen verarbeitet bzw. angelegt werden, sodass diese dem SHLM während der Laufzeit zur Verfügung stehen. Der Quelltext 3.1 zeigt den Ausschnitt der "pom.xml" in dem die lokale SPMF-Bibliothek referenziert ist.

---

```

1: ...
2: <dependencies>
3:   ...
4:   <dependency>
5:     <groupId>spmf-lib </groupId>
6:     <artifactId>spmf</artifactId>
7:     <version>0.99</version>
8:     <type>jar </type>
9:     <scope>system</scope>
10:    <systemPath>\${project.basedir}/libs/spmf.jar</systemPath>
11:   </dependency>
12: </dependencies>
13: ...

```

---

**Quelltext 3.1:** Maven-Konfiguration "pom.xml" (Ausschnitt)

(c) Das Hibernate-Framework bietet eine JPA, die eine Umsetzung des ORM ermöglicht. Die Konfiguration (Quelltext 3.2) von Hibernate zeigt neben den Parameter für die Datenbankverbindung auch das "Mapping" der Java-Klassen (model.\*). Für jede der in der Konfiguration angegebenen Java-Klassen wird während der Laufzeit durch Hibernate geprüft, ob die in den Klassen verzeichneten Notationen (z. B. @Table, @Id und @Column) mit dem Datenbankschema übereinstimmen: Die Datenbanktabellen "sequencebehavior", "eventbehavior", "anomaly" und "feedback" werden automatisch mit den Informationen der Notationen angelegt - jedes POJO entspricht dann einer Entität in der Datenbank.

```

1: <hibernate-configuration >
2:   <session-factory >
3:     <property name="hibernate.dialect">org.hibernate.dialect.
         MySQLDialect</property>
4:     <property name="hibernate.connection.driver_class">com.mysql.jdbc.
         Driver</property>
5:     <property name="hibernate.connection.url">jdbc:mysql://localhost/
         openhab</property>
6:     <property name="hibernate.connection.username">root</property>
7:     <property name="hibernate.connection.password">openhab1!</property>
8:     ...
9:     <mapping class="org.hbrs.mcs.thesis.shlm.model.Item"/>
10:    <mapping class="org.hbrs.mcs.thesis.shlm.model.ItemData"/>
11:    <mapping class="org.hbrs.mcs.thesis.shlm.model.EventBehavior"/>
12:    <mapping class="org.hbrs.mcs.thesis.shlm.model.SequenceBehavior"/>
13:    <mapping class="org.hbrs.mcs.thesis.shlm.model.Anomaly"/>
14:    <mapping class="org.hbrs.mcs.thesis.shlm.model.Feedback"/>
15:   </session-factory >
16: </hibernate-configuration >

```

**Quelltext 3.2:** Hibernate-Konfiguration "hibernate.cfg.xml" (Ausschnitt)

Im Anhang ist das UML-Klassendiagramm (1.Teil 6.13 und 2.Teil 6.14) des SHLM aufgeführt. Aus Komplexitätsgründen ist auf die Angabe von Variablen und Methoden verzichtet worden, sodass im Diagramm lediglich die Paket-, Klassennamen und deren Beziehungen untereinander visualisiert sind. Die Paketstruktur und die Funktionalitäten der inliegenden Java-Klassen sind folgendermaßen aufzufassen:

*org.hbrs.mcs.thesis.shlm.\**

- *.\**: Im Wurzelverzeichnis ist die Java-Klasse "App" hinterlegt. Sie dient als Schnittstelle für Softwarekomponenten, welche die Funktionen des SHLM nutzen möchten, d.h. sie delegiert die eintreten Anfragen und deren Kommandozeilenargumente an die Objekte der anderen Klassen.
- *analyzer.\**: Die Analyzer-Klassen können dazu genutzt werden, um eine Verhaltensabweichung und Eventprognose in der Echtzeitüberwachung zu realisieren.

- *communication.\**: Möchte das SHLM die Automatisierungssoftware über z. B. ein gelerntes Verhalten oder eine eingetretene Anomalie in Kenntnis setzen, wird die PUT-Methode des HTTP unter Zuhilfenahme der "utils.\*" genutzt, um eine Statusänderung der openHAB-Items hervorzurufen.
- *dao.\**: Für jede im ORM vorgesehene Java-Klasse ist das DAO Entwurfsmuster umgesetzt [Fra16a]. Die Objekte sind von der SQL-Logik gekapselt, das Session-Management ist zentral erfasst und ein späterer Austausch der Datenquelle (hier: MySQL-Datenbank) ist vereinfacht möglich.
- *evaluation.\**: Aus der Ausführung der Java-Klassen resultieren die Ergebnisse im Kapitel 4. Die Eingabeparameter und Ausgabeformate der Evaluationen sind im Abschnitt 3.6 thematisiert. Die Implementierungen der Evaluationen verwenden die Algorithmen der "learner"-Klassen.
- *learner.\**: In dem "learner"-Paket sind die Algorithmen sowohl für das Erlernen der Zeitfenster  $t_{\Delta}$ , der Eventmatrizen und -sequenzen als auch das Trainieren der Prognosemodelle (z. B. "CPTPlus" und "AKOM") hinterlegt.
- *model.\**: Das "model"-Paket enthält alle Klassen die beim ORM durch Hibernate behandelt werden.
- *service.\**: Durch das umgesetzte DAO der "model"-Klassen sind Dienste geschaffen deren Methoden das Session-Management nutzen, um einen zentralen Datenzugriffspunkt zu ermöglichen.
- *utils.\**: Neben dem HTTP-Client sind im "utils"-Paket die Initialisierer der Lernstrukturen, Datums- und Zeitfunktionen und die Abbildung der Sensorevents auf die Zahlen (s. Tabelle 3.3) wiederzufinden.

In Vorbereitung für das Lernen der Eventmatrizen und -sequenzen in der Folgesektion soll das DAO der "ItemData"-Klasse erörtert werden, da dieses den Eventstream der "itemdata"-View darstellt und maßgeblich an der Lernphase beteiligt ist (s. abgebildete Assoziationen im Klassendiagramm). In der Abbildung 3.15 sind die Klassen und die Schnittstelle für das "ItemDataDAO" visualisiert.

Angenommen ein Szenario indem nur die Events des Bewegungsmelder "M01" im Wohnzimmer der letzten zwei Wochen von Bedeutung sind, ermöglicht der realisierte "ItemDataService" (IDS) einen simplen Datenzugriff: Nach einer Initialisierung des IDS durch den Konstruktor erhält der Entwickler nach Aufruf der Methode "findBySensorTimeBetween("M01", "Zeitstempel-2Wochen", "heutiger Zeitstempel")" die gewünschten Daten in Form einer Liste mit Einträgen vom Typ "ItemData" über die er iterieren kann, um die "ItemData"-Objekte bzw. die Entitäten der Datenbank zu erhalten und fortführend zu manipulieren. Der Entwickler kann den Dienst bzw. die Dienste (*service.\**) in unterschiedlichsten Szenarien verwenden: Das Session-Management, die Transaktionen und deren SQL-Abfragen werden von dem IDS, "ItemDataDAO" bzw. Hibernate übernommen und müssen nicht bei jedem Szenario von dem

Entwickler abgearbeitet werden. Diese Vorgehensweise beugt redundanten Programmcode und einer Architekturerosion vor.

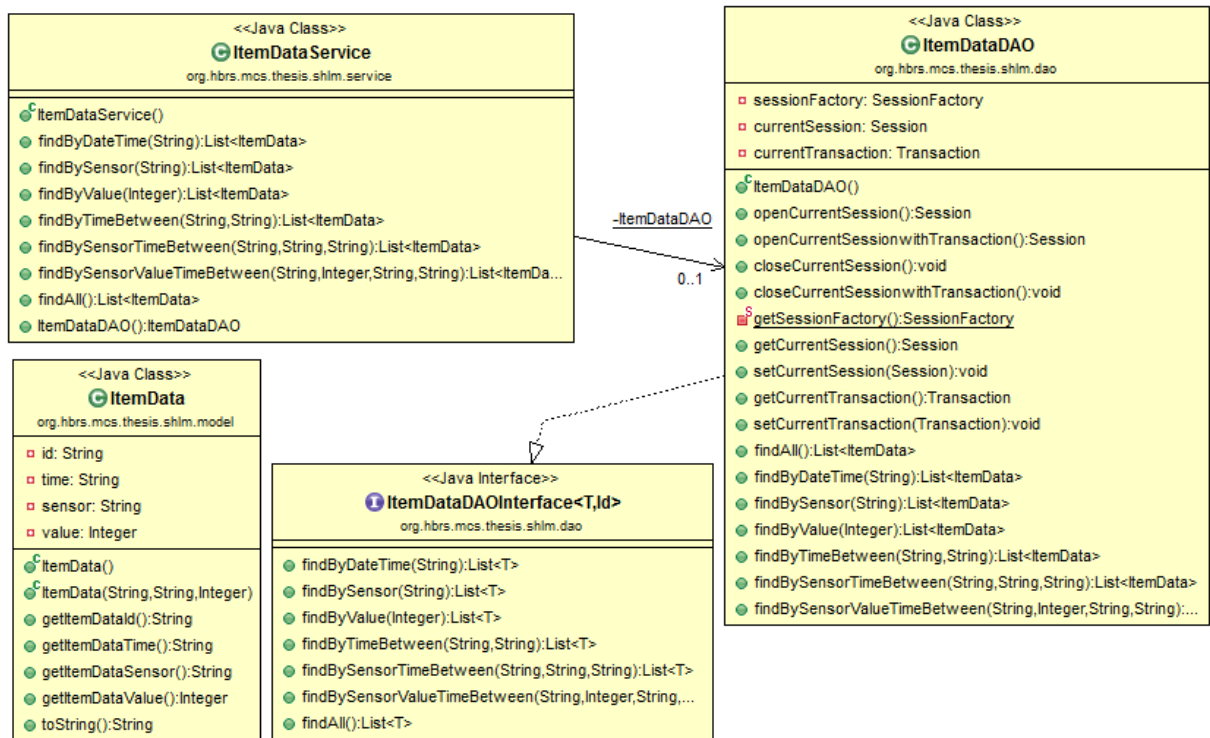


Abbildung 3.15: UML-Klassendiagramm EventStream: DAO für Java-Klasse "ItemData"

(II) (a) Das Lernen der Eventmatrizen  $M(E_i^w)$  und der Zeitfenster  $t_\Delta$  in den Zeilenvektoren  $T(S_i^d)$  ist in den Klassen "EventBehaviorLearner" (EBL) und "EventTimeSlotLearner" (ETSL) im "learner"-Paket realisiert. Der EBL kann die Eventmatrizen für einen konkreten Lernzeitraum oder die letzten  $w$ -Wochen erlernen. Der boolesche Eingabeparameter "deleteold" sagt aus, ob bereits erlerntes Verhalten gelöscht (true) oder beibehalten (false) werden soll. Neben dem EBS nutzt der EBL den "ItemService" (IS) und den IDS. Die semantischen Eigenschaften des EBL sind im Folgenden Pseudocode beschrieben:

- 1 Initialisiere den EBS, IS und IDS.
- 2 Wenn `deleteold = true` lösche das alte Verhalten mit dem EBS, ansonsten erweitere es.
- 3 Iteriere mit dem IS über alle vorhandenen Einträge in der "items"-Tabelle (entspricht den Sensoren), welche von der openHAB-Persistenz angelegt wurden.
- 4 Initialisiere für jeden kategorischen Wert der Sensoren eine "TreeMap<String, Double>" (entspr. Eventmatrix) dessen Schlüssel "w\_hh:mm" (vgl. Zelle  $M(E_i^w)_k^j$ ) eine Konkatination aus  $w$ -Wochentag und Uhrzeit ist. Der Wert der "TreeMap" speichert die absolute Häufigkeit.
- 5 Führe für jeden Sensor die Methode `IDS.findBySensorTimeBetween(...)` aus und iteriere über die Liste der zurückgegebenen "ItemData"-Einträge.

- 6 Analysiere den "ItemData"-Eintrag und füge das Event der "TreeMap" an der Stelle hinzu, an der der Schlüssel das Eintreten des Events kennzeichnet. Ist bereits ein Wert  $\neq 0$  für den Schlüssel verzeichnet, häufe den Wert.
- 7 Falls vorhanden, arbeite den nächsten "ItemData"-Eintrag (6) ab, ansonsten serialisiere die "TreeMap" als Binary Large Object (BLOB) und speichere diese zusammen mit den Informationen zu Eventnamen, Eventwert und Lernzeitraum in der Tabelle "eventbehavior" als Entität ab.
- 8 Gehe über zum nächsten Sensor und wiederhole die Schritte ab (4), andernfalls beende den Algorithmus.

Ausgehend von den gelernten Eventmatrizen können die Zeitfenster  $t_{\Delta}$  mit dem ETSL exploriert werden - dessen semantische Eigenschaften im Abschnitt 3.1 erörtert sind. Es sei angemerkt, dass die Deserialisierung der Eventmatrizen nach dem Aufruf der "findAll()" -Methode des EBS erfolgt. Nach der Exploration werden die "TreeMaps" der Zeitfenster als BLOB serialisiert und die Entität in der Tabelle "eventbehavior" mit dem EBS aktualisiert.

(b) Genauso wie der EBL kann der "SequenceBehaviorLearner" (SBL) ein Verhalten für einen spezifischen Lernzeitraum oder den der letzten  $w$ -Wochen erlernen und mit dem booleschen Parameter "deleteold" selbiges auch wieder vergessen. Der Eingabeparameter "timeslots[ ]" grenzt die zeitlichen Rahmen ein, aus denen die Events entnommen werden sollen, um die Eventsequenzen zu bilden. Für z. B. ein vierminütiges Zeitfenster werden die Eventsequenzen aus immer vier aufeinanderfolgenden Minuten entnommen. Um die Eventsequenzen zu lernen führt der SBL folgende Schritte aus:

- 1 Initialisiere den "SequenceBehaviorService" (SBS) und IDS.
- 2 Wenn *deleteold* = *true* lösche die alten Eventsequenzen mit dem SBS, ansonsten ergänze diese.
- 3 Iteriere über jeden Wochentag und beginne mit Sonntag.
- 4 Iteriere über jedes Zeitfenster aus dem Eingabeparameter "timeslots[ ]" (entspr.  $t_{db}$ ).
- 5 Initialisiere die "TreeMap<String,String>" für den Wochentag und das Zeitfenster der derzeitigen Iteration. Für den Schlüssel "w.hh:mm" sind die Eventsequenzen aus dem Zeitfenster "w.hh:mm" - "w.hh:mm+timeslot" hinterlegt.
- 6 Führe die Methode IDS.findByTimeBetween(...) aus und iteriere über die "ItemData"-Einträge der zurückgegebenen Liste.
- 7 Wenn der "ItemData"-Eintrag dem Wochentag der vorherrschenden Iteration entspricht, füge das Event dem richtigen Zeitfenster hinzu. Sind bereits Sequenzen unter dem Schlüssel vorhanden bzw. liegt eine neue Woche im Eventstream vor, trenne die Sequenzen unterschiedlicher Wochen durch ein Semikolon.
- 8 Sind alle "ItemData"-Einträge abgearbeitet, serialisiere die Eventsequenzen und speichere diese zusammen mit dem Wochentag, Zeitfenster und Lernzeitraum in der Datenbank

als Entität der Tabelle "sequencebehavior" ab. Verbleiben Einträge, führe den (7)-Schritt erneut aus.

- 9 Setze ein neues Zeitfenster aus dem Eingabeparameter "timeslots[ ]", ansonsten gehe zum nächstem Wochentag über. Beende den Algorithmus wenn alle Wochentage abgearbeitet sind (Ende bei Samstag).

Aus den gelernten Eventsequenzen kann der "SequencePredictionLearner" (SPL) die Wissensbasis erzeugen, die benötigt wird, um die Prognosemodelle zu trainieren. Dazu ruft der SPL die Eventsequenzen mit dem SBS ab und formuliert diese in dem Format, das von den Algorithmen der SPMF-Bibliothek interpretiert werden kann: Die Wissensbasis wird als Textdatei gespeichert und die Eventsequenzen der Form "...M01=1 M01=0" werden mit der Abbildung aus dem "utils"-Paket in die numerische "SPMF"-Darstellung "... 2 -1 1 -1 -2" konvertiert. Dabei ist jedes Event mit "-1" und das Ende der Sequenz zusätzlich mit "-2" gekennzeichnet [FV16b]. Der Wissensbasis werden nur eindeutige Eventsequenzen hinzugefügt. Für die Zukunft ist es sinnvoller eine Wissensbasis als die Menge der Entitäten einer Datenbanktabelle anzusehen und auf das Erstellen einer Textdatei zu verzichten - die SPMF-Bibliothek bietet derzeit keine Verarbeitung von Daten aus SQL-Abfragen für das Trainieren der Prognosemodelle an.

**(III)** Das dritte Segment der Realisierung soll zeigen, wie die Kommunikation zwischen der openHAB-Software und dem SHLM verwirklicht ist. Dabei sollen beide Richtungen der Kommunikation exemplarisch verdeutlicht werden. Die Anwendungsszenarien für die Kommunikation sind folgendermaßen:

- (a) Die openHAB-Instanz ruft das SHLM auf, um den Lernvorgang anzustoßen, die Erkennungsphase zu initiieren und auf evtl. Feedback vom Angehörigen hinzuweisen.
- (b) Das SHLM informiert die openHAB-Instanz über die abgeschlossene Lernphase, die aufgetretenen Verhaltensabweichungen und Prognosen.

Die Abbildungen 3.16 (Übersicht) und 3.17 (Bewegungsmelder & Magnetkontakte) zeigen die Benutzeroberflächen, die vom Jetty-Webserver aus der deklarierten openHAB-Sitemap erzeugt werden. Dabei ist die Benutzeroberfläche "Bewegungsmelder & Magnetkontakte" über den ersten Eintrag in der "Übersicht" zu erreichen. Weitere Funktionalitäten der "Übersicht" ermöglichen es dem Angehörigen den Schwellwert für die Verhaltensabweichung, den Zeitraum bis zur nächsten Prognose, das Lernintervall, ein Vergessen von Verhalten und seine E-Mail-Adresse einzustellen bzw. einzutragen. Über die Schaltflächen der Sektion "Lernphase" wird ein Verhalten erlernt und gelöscht. In der Sektion "Erkennungsphase" startet das Betätigen der Schaltfläche die Echtzeitüberwachung für die Zeitpunkte der explorierten Zeitfenster  $t_{\Delta}$  und für die minütig zu stellenden Prognosen.





**Abbildung 3.16:** Übersicht der openHAB-GUI für das SHLM

Die openHAB-Versionen 1.x bieten keine expliziten Sitemap-Elemente für Benutzereingaben an. Das Eingabefeld der E-Mail-Adresse wird über ein Webview-Element angegeben, das ein lokal abgelegtes HTML-Dokument (./webapps/static/) aufruft [Inc16]. Die E-Mail-Adresse wird gespeichert indem ein JavaScript im HTML-Dokument die Representational State Transfer (REST)-API von openHAB nutzt und den Status eines Items auf die getätigte Eingabe setzt.

(a) Nachdem das SHLM die konfigurierten Phasen des Maven-Lebenszyklus durchlaufen hat, wird dieses zusammen mit den Abhängigkeiten im lokalen openHAB-Verzeichnis " ./configura-

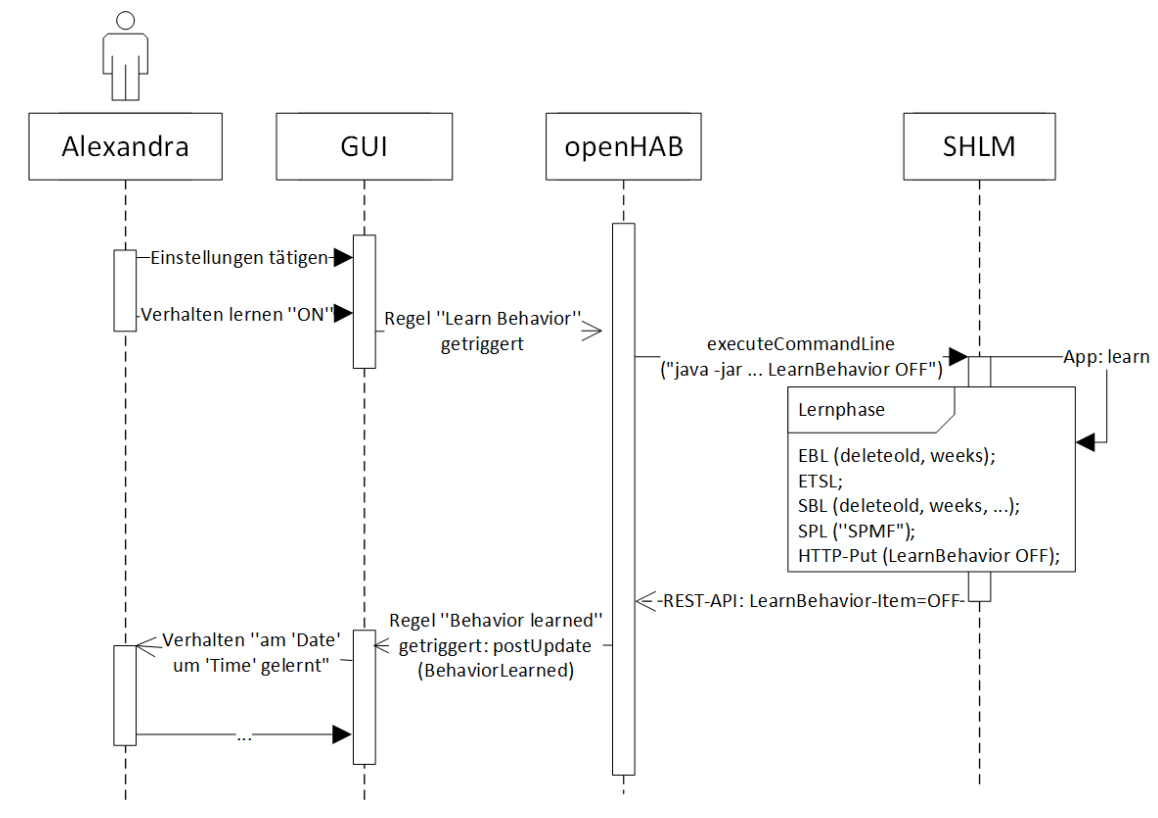


**Abbildung 3.17:** openHAB-GUI für realisierte Sitemap (Auszug der Sensoren)

tions/scripts/" abgelegt. Das SHLM wird dann während der Laufzeit von der openHAB-Instanz nach dem Triggern einer openHAB-Regel mit der Methode "executeCommandLine()" und den anwendungsspezifischen Parametern aufgerufen.

Im Kontext der openHAB-Regeln ist es wichtig die Status der einzelnen GUI-Elemente bei jedem Systemstart von openHAB mit einer Regel zu initialisieren, denn wenn auch die Schaltflächen laut Benutzerschnittstelle auf "OFF" stehen, sind diese im System noch "Uninitialized". Andernfalls ist die Funktion der Schaltflächen es beim zweiten Betätigen verfügbar - für den Angehörigen irreführend. Für das Triggern einer Regel beim Systemstart bietet openHAB die Bedingung "System started" an. Der derzeitige Status eines Items ist mit "Itemname.state" zu erfragen und mit "Itemname.postUpdate("state")" zu setzen.

Das Sequenzdiagramm 3.18 zeigt den Ablauf "Verhalten lernen", der startet sobald der Angehörige den Schalter betätigt: Der Status des openHAB-Item "LearnBehavior" wird auf "ON" gesetzt. Durch den Statuswechsel wird die Regel (Quelltext 3.3) ausgelöst, die dann wiederum das SHLM mit den fünf Parametern ausführt: Dem String "learn", dem Lernintervall (dw), dem booleschen Ausdruck (deleteold), das Item und dessen Status für das die openHAB-Instanz eine Antwort erwartet. Die "App" nimmt die Parameter entgegen, wertet den String-Parameter aus und ruft die Unterklassen auf, welche das Lernen der Eventmatrizen, Zeitfenster und Eventsequenzen, wie in Sektion (II) beschrieben, bewerkstelligen.



**Abbildung 3.18:** UML-Sequenzdiagramm - Lernphase: Das SHLM lernt.

(b) Im Anschluss informiert das SHLM die openHAB-Instanz über den Abschluss der Lernphase indem das "LearnBehavior"-Item über die REST-API (HTTP-PUT) auf den Status "OFF" gesetzt wird. Dieser Statuswechsel triggert die weitere Regel "Behavior Learned", die dem Angehörigen ein gelerntes Verhalten auf der Benutzeroberfläche visualisiert.

```

1: ...
2: rule "Learn_Behavior"
3: when
4:   Item LearnBehavior changed from OFF to ON
5: then
6:   var dw = (DurationWeeks.state as DecimalType).intValue
7:   executeCommandLine("java -jar " + gb_shlm_path + "_learn_" + dw + ".jar"
8:     + true + "_LearnBehavior_OFF")
9: end

```

**Quelltext 3.3:** Ausschnitt openHAB-Rules-Artefakt: "Learn Behavior"-Regel

Eine Kommunikation die nicht zwischen der openHAB-Instanz und dem SHLM, sondern zwischen der openHAB-Instanz und dem Angehörigen erfolgt, bezieht sich auf den Anwendungsfall in dem ein ungewöhnliches Verhalten automatisiert erkannt wurde und eine Reaktion erfolgen muss: Der Angehörige wird per E-Mail über die Verhaltensabweichung informiert. Hierfür ist

das Mail-Binding "org.openhab.action.mail-1.8.1" eingesetzt, dessen Konfiguration in der zentralen openHAB-Config in der Sektion "Mail Action configuration" verwirklicht wird. Hinterlegt wird der SMTP-Server, SMTP-Port, die SMTP-Authentifizierung und die E-Mail-Adresse, von der aus die Nachrichten an die E-Mail-Adresse des Angehörigen versendet werden. Eine Verschlüsselung über das Transport Layer Security (TLS) Protokoll ist möglich. Letztendlich ruft das SHLM eine Verhaltensanomalie aus, indem der Status des Items "SendEMail" über die REST-API auf "ON" gesetzt wird. Die Regel "Send E-Mail Anomaly" führt darauf hin die Methode "sendMail(E-Mail-Adresse, Betreff, Nachricht)" aus und setzt den Status von "SendEMail" wieder auf "OFF". Der Betreff und die Nachricht ist derzeit statisch hinterlegt, kann aber auch dynamisch mit einem Webview-Element realisiert werden.

**(IV)** Vergleichbar mit dem Ablauf bei dem "Verhalten lernen" ereignet sich der Kommunikationsablauf, um eine Verhaltensabweichung auszurufen oder eine Prognose zu stellen. Der große Unterschied besteht darin, dass nach dem Betätigen der Schaltfläche "Echtzeitüberwachung starten" keine weitere Mensch-Computer-Interaktion notwendig ist, denn das AAL-System überprüft das aktuelle Verhalten immer am Ende eines jeden explorierten Zeitfensters  $t_{\Delta}$  und stellt Prognosen in Abständen von 1-5 Minuten. Die Parameter der Items "PredictionTime", "ThresholdValue" werden dem SHLM bei Ausführung übergeben.

Die entwickelten Evaluationen "AnomalyDetectionEvaluation" (ADE) und "EventPredictionEvaluation" (EPE) simulieren eine tagesübergreifende Systemlaufzeit des AAL-Systems und dienen als entwickelte Vorlage für eine Implementierung von Systemlaufzeiten in  $t_{\Delta}$ -Intervallen, die in ergänzenden Arbeiten in den Java-Klassen ("analyzer"-Paket) erfolgen kann.

(a) Der ADE wird der Lernzeitraum, Validierungs- bzw. Evaluierungszeitraum, die maximalen Zeitdifferenzen  $t_{diff}$  und die unterschiedlichen Schwellwerte übergeben. Die ADE simuliert in folgenden semantischen Schritten:

- 1 Lerne mit dem EBL das Verhalten für den angegeben Lernzeitraum.
- 2 Iteriere über die Zeitdifferenzen  $t_{diff}$  und exploriere mit dem ETSL die Zeitfenster in Abhängigkeit von  $t_{diff}$ .
- 3 Erzeuge für jede Woche im Validierungs- bzw. Evaluierungszeitraum eine Ausgabedatei und füge die Kopfzeile mit den Bezeichnern ein.
- 4 Initialisiere den EBS und IBS. Führe EBS.findAll() aus und iteriere über die Liste der "EventBehavior" (EB) - entspr. einer Eventmatrix.
- 5 Serialisiere die Zeitfenster  $t_{\Delta}$  für das EB und erfrage mit dem IDS alle EB-spezifischen "ItemData"-Einträge ab.
- 6 Iteriere über alle Zeitfenster, überprüfe für jeden "ItemData"-Eintrag ob dieser dem derzeitigen Zeitfenster (Wochentag + Uhrzeit) zugehörig ist und häufe ein Zutreffen.
- 7 Sind alle "ItemData"-Einträge abgearbeitet, errechne die Schwellwertüberschreitungen und füge der Ausgabedatei eine neue Zeile mit allen relevanten Informationen hinzu.

- 8 Falls vorhanden, gehe über zum nächsten Zeitfenster, ansonsten wähle den nächsten EB und führe den (6)-Schritt erneut aus.
- 9 Ist kein EB mehr vorhanden, speichere die Ausgabedateien und gehe über zum nächsten  $t_{diff}$  und wiederhole den (2)-Schritt.
- 10 Beende den Algorithmus sobald alle Zeitdifferenzen abgearbeitet sind.

(b) Die EPE erhält für die Berechnung der Eventprognosen neben den gleichen Zeiträumen wie die ADE noch zusätzlich den Pfad zur Wissensbasis, die zeitliche Eingrenzung  $t_{db}$ , das Prognosemodell und dessen Parameter als Eingabeinformationen. Die Verarbeitung der gemessenen Daten ist im folgenden Pseudocode erörtert:

- 1 Wähle das Prognosemodell ("CPT+", "AKOM", "DG" oder "LZ78") und dessen Parameter aus.
- 2 Lerne mit dem SBL die Eventsequenzen aus dem Lernzeitraum.
- 3 Erzeuge mit dem SPL die Wissensbasis und überführe diese in das "SPMF"-Format.
- 4 Iteriere über die Zeiten  $t_{db}$ , erstelle die Ausgabedatei und füge die Kopfzeile mit den Bezeichnern hinzu.
- 5 Erzeuge ein Trainingsset aus der Wissensbasis und trainiere das gewählte Prognosemodell mit dem SPL.
- 6 Initialisiere den IDS und erstelle die Liste mit den "ItemData"-Einträgen aus dem gewählten Validierungs- bzw. Evaluierungszeitraum.
- 7 Iteriere über die "ItemData"-Einträge.
- 8 Erzeuge die  $t_{db}$ -langen Sequenzen aus den "ItemData"-Einträgen. Ist eine neue zeitliche Eingrenzung erreicht, prognostiziere das Folgeevent für die vorliegende Sequenz. Füge die Sequenz und Prognose der Ausgabedatei hinzu.
- 9 Falls vorhanden, arbeite die nächsten "ItemData"-Einträge ab, ansonsten speichere die Ausgabedatei.
- 10 Wähle den nächsten Wert für  $t_{db}$  und fahre mit (4)-Schritt fort, ansonsten beende den Algorithmus.

Der Abschnitt 3.6 listet die verwendeten Werte der Eingabeparameter auf und beschreibt die Ausgabe der Evaluationen.

Die verbleibenden FA (FA6, FA11-12 und FA13), die aus zeitlichen Gründen nicht mehr im Rahmen dieser Arbeit implementiert werden konnten, stehen für einen Arbeitsaufwand von Personenwochen in Folgearbeiten frei. Für die Implementierung in den Folgearbeiten sind die nachstehenden Vorschläge zu beachten: Alle notwendigen DAO-Klassen und deren Dienste sind für die FA6 implementiert. Dabei kann die Routine der FA6 auf der Implementierung der FA5 aufbauen, sodass die Nachricht der E-Mail um HTTP-Links ergänzt wird, welche dem Angehörigen ein Feedback ermöglichen. Die HTTP-Links sind unterteilt in positives oder negatives Feedback und stellen die REST-API Aufrufe mit der Kennung der Anomalie (s.

ERD 3.14) dar. Führt der Angehörige den Link aus, kann eine openHAB-Regel getriggert werden, welche das Feedback mit dem SHLM persistiert und in einem anstehenden Lernen verarbeitet. Die tagesspezifischen Ruhephasen (FA11) können die Zeiträume zwischen den explorierten Zeitfenstern  $t_{\Delta}$  darstellen. Ein Erweitern bzw. Aktualisieren (*deleteold = false*) des abgespeicherten Normalverhaltens (FA12) kann dann mit dem EBL (Eventmatrizen), ETSL (Zeitfenster) und SBS (Eventsequenzen) erfolgen: Der in den Zellen  $M(E_i^w)_k^j$  hinterlegte Wert wird gehäuft, die Zeilenvektoren erweitert und die Wissensbasis nur um eindeutige Sequenzen ergänzt. Für die Umsetzung der FA13 ist zu beachten, dass die Datenquellen grundsätzlich ausgetauscht werden können: Ein After-Insert-Trigger im MySQL-Dialekt kann die Daten, die einen gewünschten Zeitraum überschreiten, löschen. Weitere Datenquellen mit unterschiedlicher Syntax in der Data Definition Language (DDL) dürften auf Probleme beim Anlegen des Triggers stoßen. Die FA13 kann also entweder realisiert werden, indem für jede mögliche Datenquellen ein eigenes Sprachkonstrukt hinterlegt wird, oder das SHLM übernimmt diese Aufgabe mit den implementierten Diensten.

### 3.6 Funktionstests und Evaluation

Der vorherige Abschnitt beschreibt anhand der Realisierung des AAL-Systems die Funktionalitäten, die dem Endanwender bzw. den Personas zur Verfügung stehen. Da die Implementierung des AAL-Systems mit der Methode des XP (Extreme Programming) erfolgt ist, sind die erarbeiteten Softwareteilsysteme, welche mit dem Anwendungskontext der in Abschnitt 3.2 erforschten Anforderungen korrespondieren, iterativ entwickelt und die Teilfunktionalitäten vom Entwickler direkt nach Fertigstellung getestet worden. Aufgrund der zeitlichen Begrenzung der vorliegenden Arbeit ist die Implementierung von Unit-Tests für die einzelnen Komponenten nicht praktiziert. Zudem behandelt die umgesetzte Implementierung keine anwendungsorientierte Ausnahmebehandlung, d.h. auftretende Fehler, die in den abgehaltenen Tests nicht festgestellt werden konnten, werden lediglich auf der Konsole der Laufzeitumgebung dargestellt und sind für die Persona nicht ersichtlich. Für eine Produktreife kann das SHLM im Fehlerfall eine Kommunikation mit der Automatisierungssoftware aufnehmen, sodass diese dann wiederum den Missstand an den Angehörigen über den gewünschten Kommunikationskanal weiterleitet.

Die Evaluation des AAL-Systems ist in zwei Sektionen gegliedert, welche die im Konzept der Datenanalyse aufgeworfenen Fragen betrachten. Da der zeitliche Rahmen der vorliegenden Arbeit keine Systemlaufzeit von vier bis sechs Wochen zulässt, sind eigens für die Evaluation zwei eigenständige Simulationen implementiert:

- (1) Die Simulation, die überprüft, ob zum Zeitpunkt  $t$ =jetzt eine Verhaltensabweichung vorliegt, ist in der Klasse `**.evaluation.AnomalyDetectionEvaluation`

.java" (Digitaler Anhang - Workspace SHLM-Projekt) hinterlegt. Der Zeitpunkt  $t=$ jetzt ist der Endzeitpunkt eines jeweiligen explorierten Zeitfensters  $t_{\Delta}$ .

- (2) In der Klasse "`*.evaluation.EventPredictionEvaluation.java`" ist die Simulation umgesetzt, die ein Ereignis nach einer stattgefundenen Ereignisfolge prognostiziert.

Simuliert wird der Systemablauf des AAL-Systems in der Erkennungsphase für einen Zeitraum von ein bzw. zwei Wochen. Die Datenbasen für die Lern- und Erkennungsphase sind nicht simuliert, sondern stellen die real aufgezeichneten Messdaten aus dem Smart Home (Rheinland Pfalz) dar. Für die Simulationen sind folgende Eingabeparameter gewählt (Zeitstempel: YYYY-MM-DD hh:mm:ss).

Die Eingabeparameter, die für beide Simulationen gültig sind:

- i *zweiwöchiger Lernzeitraum*: 2016-01-02 00:00:00 - 2016-01-15 23:59:00
  - a Validierung mit beiden Wochen aus dem Lernzeitraum
  - b Evaluierung mit zwei lernunabhängigen Wochen: 2016-01-16 00:00:00 - 2016-01-29 23:59:00
- ii *vierwöchiger Lernzeitraum*: 2016-01-02 00:00:00 - 2016-01-29 23:59:00
  - a Validierung mit zwei Wochen des Lernzeitraumes: 2016-01-16 00:00:00 - 2016-01-29 23:59:00
  - b Evaluierung mit zwei lernunabhängigen Wochen: 2016-01-30 00:00:00 - 2016-02-12 23:59:00

Eingabeparameter für die Verhaltensabweichungen der Simulation (1):

- iii Die *maximale Zeitdifferenz*  $t_{diff}$  für zwei aufeinanderfolgende Events zur Exploration der Zeitfenster  $t_{\Delta}$ : 5, 10, 15, 20, 25, 30, 45 und 60 Minuten.
- iv Der *Schwellwert* zur Berechnung von  $abs\_diff(A(d, t_{\Delta}, S_i), N(d, t_{\Delta}, S_i))$ : 25%, 50% und 75%.

Eingabeparameter für die Verhaltensprognosen der Simulation (2):

- v Die ermittelte Wissensbasis aus den gelernten Eventsequenzen für  $t_{ab}$ : 1 - 5 Minuten.
- vi Die Auswahl des Prognosemodells: "CPTPlus", "AKOM", "DG" oder "TDAG". Der DG oder TDAG sind zwar in der Simulation implementiert, werden jedoch nicht in den Ergebnissen behandelt.
- vii Die Standardparameter des CPT+-Algorithmus:  $CCF:true$ ,  $CBS:true$ ,  $CCFmin:1$ ,  $CCFmax:6$ ,  $splitMethod:0$ ,  $minPredictionRatio:1.0$  und  $noiseRation:1.0$  [FV16a]
- viii Für AKOM die Ordnungen:  $order:5$ ,  $order:10$  und  $order:15$  mit separater Trainingsphase.

Die Simulationen geben die eruierten Daten als Textdateien im CSV-Format aus. Bei oben genannten Eingabeparametern sind es für Simulation (1) insgesamt 64 und für Simulation

(2) 160 Textdateien, die erstellt werden. Die kommaseparieren Spalten sind wie folgt zu interpretieren:

(1) Ausgabe der Simulation "Verhaltensabweichung"

- 1 Die Sensorkennung: "M01", "M02", "M03", "M04", "M05", "C01", "C02", "C03" oder "C04".
- 2 Der Eventwert mit "0" für keine Bewegung erkannt bzw. Tür oder Schublade geschlossen und "1" für Bewegung erkannt und Tür oder Schublade geöffnet.
- 3 Der Wochentag am Startzeitpunkt des explorierten Zeitfensters  $t_{\Delta}$ . Beginnend mit "1=Sonntag" und endend bei "7=Samstag".
- 4 Der Startzeitpunkt von  $t_{\Delta}$ .
- 5 Der Wochentag am Endzeitpunkt von  $t_{\Delta}$ .
- 6 Der Endzeitpunkt von  $t_{\Delta}$ .
- 7 Die Anzahl  $N(d, t_{\Delta}, S_i)$  der im Lernzeitraum aufgetretenen Events für  $t_{\Delta}$  pro Woche (durchschnittliche Häufigkeit).
- 8 Die Anzahl  $A(d, t_{\Delta}, S_i)$  der in der Erkennungsphase aufgetretenen Events für  $t_{\Delta}$ .
- 9 Die absolute Differenz  $abs\_diff(N, A)$  von (7) und (8).
- 10 Der boolesche Wert für eine Verhaltensabweichung von mehr als 25% - "sensibel".
- 11 Der boolesche Wert für eine Verhaltensabweichung von mehr als 50% - "ausgewogen".
- 12 Der boolesche Wert für eine Verhaltensabweichung von mehr als 75%. - "träge".

(2) Ausgabe der Simulation "Verhaltensprognosen"

- 1 Die Eventsequenz im numerischen Format, entnommen aus dem Evaluationszeitraum für ein Zeitfenster von  $t_{db}$  Minuten, dient dem Prognosemodell als Eingabe. Die Eventsequenzen sind in der Textdatei als zeitlicher Verlauf abgelegt, d.h. die Sequenz der  $n + 1$ -Zeile ist Folge der  $n$ -Zeile.
- 2 Das vom verwendeten Prognosemodell prophezeite Event in numerischer Darstellung.

Die Ergebnisse (Kapitel 4) der Simulationen behandeln die ADL "Raum und Zeit gestalten", d.h. in der Evaluation werden die Messdaten des Bewegungsmelders "M01" (Wohnzimmer) untersucht. Für die Eventprognosen der unterschiedlichen Prognosemodelle sind tabellarische Genauigkeitsvergleiche aufgestellt.



## 4 Ergebnisse und Ergänzungen

Im vorliegenden Kapitel wird ergründet, welche QoS das im Kapitel 3 realisierte AAL-System beim ausgewählten Konzept der Datenanalyse (3.1) aufweist. Die QoS des AAL-Systems hängt davon ab, wie viele erkannte Verhaltensabweichungen falsch sind, ob eine Verhaltensabweichung nicht erkannt wird und ob das prognostizierte Event mit dem zukünftigen Event übereinstimmt. Für unsere Persona sind keine FPs zwar wünschenswert, werden aber im Austausch für etwaige FNs bevorzugt angenommen bzw. akzeptiert. Die realisierten Routinen, die eine Abgrenzung von gelernten und aktuellem Verhalten vornehmen, arbeiten dabei so restriktiv, dass für jeden gelernten Endzeitpunkt eines explorierten Zeitfensters  $t_{\Delta}$ , gebunden an den jeweiligen Wochentag  $d$  und den Sensor  $S_i$ , ein FN nur dann auftreten kann, wenn initiale Einstellung des manuellen Schwellwertes unbekümmert bzw. nicht sensibel genug ausgewählt ist oder die maximale Zeitdifferenz  $t_{diff}$ , welche genutzt wird, um die Zeitfenster  $t_{\Delta}$  zu erforschen, zu groß ist und einen saloppen Beobachtungszeitraum zulässt. Wenn also die Daten der Lernphase keine Verhaltensabweichungen aufweisen und einen wochenübergreifenden Alltag widerspiegeln, lässt die Erkennungsphase keine FNs zu und es ist vermehrt mit FPs zu rechnen, die der Angehörige dann sukzessive durch sein Feedback terminieren kann. Zwar ließe sich mit dem gelernten Verhalten auch eine Prognose in der Form "Die Persona wird sich in der Zeit von 12-14 Uhr im Wohnzimmer aufhalten." oder "Die Persona wird voraussichtlich in den nächsten 30 Minuten die Küchenschublade öffnen." tätigen, dennoch soll der zweite Aspekt des Konzeptes der erörterten Datenanalyse zeigen, inwieweit etablierte Verfahren (CPT+ und AKOM) dazu geeignet sind, die kategorischen Messdaten der vorliegenden Arbeit zu verarbeiten und welche Prognosegenauigkeit in den aufgestellten Modellen resultiert.

### 4.1 Ergebnisse

Die Ergebnisse der Simulationen sind im Folgenden ausgehend von den bezeichneten Eingabeparameter (Abschnitt 3.6) beschrieben. Dabei sind die Ergebnisse der Simulation (1) beschränkt auf die ADL "Raum und Zeit gestalten" im Wohnzimmer des Smart Home "Stadt in Rheinland Pfalz": Die analysierten Ausgabedaten geben also Aufschluss darüber an welchem Tag und zu welcher Uhrzeit die Probanden das Wohnzimmer betreten, sich dort aufgehalten und es wieder verlassen haben.

Die Tabellen 4.1 und 4.2 listen die Anzahlen der explorierten Zeitfenster  $t_{\Delta}$  für  $t_{diff}$  und den jeweiligen Wochentag bei einem zwei- bzw. vierwöchigen Lernzeitraum auf.

(i) *zweiwöchiger Lernzeitraum*

$t_{diff}$	Sonntag	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	$\Sigma$
5	43	44	56	31	22	20	32	248
10	23	25	27	22	19	13	14	143
15	12	17	17	16	15	13	8	98
20	8	14	12	16	11	11	7	79
25	6	13	9	14	10	11	5	68
30	5	12	7	12	8	10	4	58
45	3	5	3	7	7	8	1	34
60	2	2	2	7	5	5	1	24

**Tabelle 4.1:** Anzahl explorierter Zeitfenster  $t_{\Delta}$ : Zweiwöchiger Lernzeitraum ((iii)  $t_{diff}, d$ )

Wenn auch in Abgleich mit den zuvor gezeigten Eventstreams (z. B. Abbildung 3.9) eine maximale Zeitdifferenz  $t_{diff}$  größer gleich 20 Minuten für die Bewegungsmelder als durchaus sinnvoll erscheint, um die vorherrschende Aktivität in einem Zeitfenster bei einem zweiwöchigen Lernzeitraum unterzubringen, sollte  $t_{diff}$  bereits bei einem vierwöchigen Lernzeitraum reduziert werden. Unabhängige Aktivitäten werden dann nicht fälschlicherweise in einem Zeitfenster zusammengelegt. Je mehr Wochen der Lernzeitraum umfasst, desto kleiner ist  $t_{diff}$  zu wählen. Bei einem Lernzeitraum von mehr als zwei Monaten sollte sogar eine maximale Zeitdifferenz von weniger als fünf Minuten in Betracht gezogen werden.

(ii) *vierwöchiger Lernzeitraum*

$t_{diff}$	Sonntag	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	Samstag	$\Sigma$
5	40	42	61	38	26	26	26	259
10	13	27	30	28	21	19	8	146
15	7	16	18	21	15	14	7	98
20	5	12	12	19	12	10	5	75
25	3	10	7	14	11	10	4	59
30	3	10	5	11	9	9	3	50
45	2	3	3	5	8	7	2	30
60	2	1	1	2	6	5	2	19

**Tabelle 4.2:** Anzahl explorierter Zeitfenster  $t_{\Delta}$ : Vierwöchiger Lernzeitraum ((iii)  $t_{diff}, d$ )

Hingegen ist  $t_{diff}$  für die Magnetkontakte höher anzusetzen, denn Events von Türen oder Schubladen werden nur durch Interaktion und nicht passiv wie bei den Bewegungsmeldern durch die reine Anwesenheit der Probanden mit einem Mindestsendeabstand von 15 Sekunden ausgelöst. In einem Lernzeitraum ist es gewöhnlich, dass eine Person in der einen Woche

Montag um 7:30 Uhr das Haus verlässt und in der darauffolgenden Woche erst um 8:00 Uhr. Gleichmaßen kann ein Frühstück statt um 6:45 Uhr auch um 7:15 Uhr zubereitet werden.

Die Ergebnisse der Schwellwertüberschreitung (*(iv) abs.diff*) (Digitaler Anhang - AnomalyDetection) weisen zu viele FPs auf, als dass dem Angehörigen ein Feedback aller Falschmeldungen zuzumuten wäre. Die hohe Anzahl der FPs ist zum einem dadurch zu begründen, dass der Rahmen dieser Arbeit zur Verfügung stehende Lernzeitraum zu knapp bemessen ist und zum anderen dadurch, dass bei der in Sektion 3.1 beschriebenen Semantik des Algorithmus zur Exploration der Zeitfenster  $t_{\Delta}$  auch alleinstehende Events als eigenständiges Zeitfenster aufgefasst werden. Alleinstehende Events bedeuten in diesem Zusammenhang, dass die Probanden einen kurzzeitigen Raumwechsel vollziehen - also ein Aufenthalt im Wohnzimmer nur wenige Sekunden besteht - oder die Haustiere der Probanden eine Auslösung der Melder verursachen. Werden die Zeitfenster mit alleinstehenden Events nicht festgehalten, kann z.B. die Rate der FPs über eine Woche in einem vierwöchigen Lernzeitraum mit einer maximalen Zeitdifferenz  $t_{diff} = 10$  Minuten und einem sensiblen Schwellwert um 25% gesenkt werden. Es ist also in Zukunft durchaus sinnvoll zusätzlich zu  $t_{diff}$  auch die Häufigkeit bei der Exploration der Zeitfenster zu berücksichtigen. In der Abbildung 6.12 ist die Validierung der Wochentage von Montag bis Mittwoch aus der Woche vom 23-29.01.2016 mit  $t_{diff} = 20$  Minuten zu sehen (vierwöchiger Lernzeitraum). Eine Betrachtung zeigt die Problematik der alleinstehenden Events und die notwendige Lernzeitvergrößerung.

Die Ergebnisse der Simulation (2) zeigen initial gelernte Wissensbasen mit 4637 (zweiwöchiger Lernzeitraum) und 7620 (vierwöchiger Lernzeitraum) eindeutigen Eventsequenzen für  $t_{db}$  von 1-5 Minuten. Diese Wissensbasen werden für das Training der Prognosemodelle "CPTPlus" und "AKOM" eingesetzt. Ausgehend von den gelernten Prognosemodellen wird eine Prognose aus dem im Zeitraum von  $t_{db}$  angetroffenen Events - in ihrer Konkatenation die Sequenz - sukzessive für eine ganze Woche aufgestellt.

*CPT+ Validierung: (i) zweiwöchiger Lernzeitraum*

$t_{db}$	Prog. Wahr	Prog. Falsch	Prog. Insg.	Genauigkeit (~%)
<i>(a) 02-08.01.2016</i>				
1	266	1014	1280	20,78
2	118	674	792	14,90
3	69	535	604	11,42
<i>(a) 09-15.01.2016</i>				
1	289	926	1215	23,79
2	111	628	739	15,02
3	76	482	558	13,62

**Tabelle 4.3:** Genauigkeitstabelle für CPT+: zweiwöchiger Lernzeitraum - Validierung (a)

Wie in den Tabellen 4.3 und 4.4 dargestellt, ist bei dem vom CPT+-Algorithmus aufgestellten Prognosen nicht von einer Genauigkeit zu sprechen, sondern bei einer prozentualen Fehlprognose von 76,21 - 91,12% sind die Vorhersagen schlichtweg ungenau. Die Ergebnisse zeigen, dass die "Genauigkeit" der Prognose bei steigendem  $t_{db}$  sinkt, d.h. je länger die eingegebene Eventsequenz ist, desto schlechter die Vorhersage. Die Ergebnisse für  $t_{db} > 3$  sind aufgrund dieser Tatsache nicht bewusst weggelassen, sondern könnten mit den zur Verfügung stehenden Hardwareressourcen nicht errechnet werden, obwohl laut [FV16a] mit den verwendeten Eingabeparameter "CCF:true" und "CBS:true" (Anwendung der Frequent Subsequence Compression (FSC) und Simple Branches Compression (SBC) Strategien) der Speicherbedarf zu Lasten der Prognosegeschwindigkeit reduziert wird.

*CPT+ Evaluierung: (i) zweiwöchiger Lernzeitraum*

$t_{db}$	Prog. Wahr	Prog. Falsch	Prog. Insg.	Genauigkeit (~%)
<i>(b) 16-22.01.2016</i>				
1	281	960	1241	22,64
2	127	624	751	16,92
3	76	493	569	13,36
<i>(b) 23-29.01.2016</i>				
1	209	789	998	20,94
2	84	540	624	13,46
3	43	441	484	8,88

**Tabelle 4.4:** Genauigkeitstabelle für CPT+: zweiwöchiger Lernzeitraum - Evaluierung (b)

Eine Validierung des Prognosemodells bei vierwöchigem Lernzeitraum ( $t_{db} = 1$  Minute) zeigt keine signifikante Verbesserung der Prognosegenauigkeit (~21,84%). Ein Blick auf die gestellten Prognosen offenbart, dass in über 80% der Fälle die Prognose "(3)" bzw. "M02=0" (im Flur findet keine Bewegung mehr statt) gestellt wird. Ein Test ohne die Anwendung der Prediction with improved Noise Reduction (PNR) Strategie zeigt, dass die Anzahl der Prognosen "(3)" nur marginal gesenkt, die absolute Genauigkeit jedoch nicht gesteigert wird. Darüber hinaus stellt der CPT+-Algorithmus in seltenen Fällen gar keine Prognose!

Wie sind diese (Un-)Genauigkeiten der CPT+-Prognose zu begründen? Entweder sind die in dieser Arbeit vorliegenden Messdaten nicht für eine Prognose mit dem CPT+-Algorithmus geeignet und Genauigkeiten von 73,52% (Schriftzeichen der Bibel) sind nicht zu erreichen, oder die Implementierung in der SPMF-Bibliothek (Version 0.99d) entspricht nicht der beschriebenen Semantik [Ted+16, S.1 ff.].

In Abgrenzung zu den Ergebnissen der CPT+-Prognose sind im Folgenden die Ergebnisse der Eventprognosen mit den AKOM-Prognosemodell gelistet.

*AKOM Validierung: (i) zweiwöchiger Lernzeitraum*

$t_{db}$	k-Ordnung	Prog. Wahr	Prog. Falsch	Prog. Insg.	Genauigkeit (~%)
<i>(a) 02-08.01.2016</i>					
1	5	851	429	1280	66,48
2	5	515	277	792	65,03
3	5	396	208	604	65,65
4	5	329	179	508	64,76
5	5	277	161	438	63,24
1	10	871	409	1280	68,05
2	10	577	215	792	72,85
3	10	428	176	604	70,87
4	10	344	164	508	67,72
5	10	296	142	438	67,58
1	15	886	394	1280	69,22
2	15	570	222	792	71,97
3	15	417	187	604	69,04
4	15	341	167	508	67,13
5	15	290	148	438	66,21
<i>(a) 09-15.01.2016</i>					
1	5	762	453	1215	62,72
2	5	481	258	739	65,09
3	5	335	223	558	60,04
4	5	275	188	463	59,40
5	5	249	151	400	62,25
1	10	803	412	1215	66,10
2	10	545	194	739	73,75
3	10	393	165	558	70,43
4	10	313	150	463	67,60
5	10	265	135	400	66,25
1	15	805	410	1215	66,26
2	15	547	192	739	74,02
3	15	397	161	558	71,15
4	15	312	151	463	67,39
5	15	269	131	400	67,25

**Tabelle 4.5:** Genauigkeitstabelle für AKOM: zweiwöchiger Lernzeitraum - Validierung (a)

Die Tabelle 4.5 zeigt die validierte Prognosegenauigkeit bei einem zweiwöchigen Lernzeitraum in Abhängigkeit der Parameter  $t_{db}$  und k-Ordnung. Dabei liegen die eruierten Genauigkeiten

zwischen 59,40% und 74,02%, wobei die durchschnittliche Sequenzlänge aus 15,07 bzw. 9,46 Events besteht, d.h. werden in das Prognosemodell längere Eventsequenzen eingegeben als die  $k$ -Ordnung aufweist, scheint die Genauigkeit zu sinken. Denn auch für Markov-Ketten 15-Ordnung ( $t_{db} = 4$ ) steigt die Genauigkeit von 59,40% (5-Ordnung) auf 67,39% an.

Die weiteren Ergebnisse der AKOM-Prognose - entsprechend eine Validierung im vierwöchigen Lernzeitraum und die Evaluierungen im zwei- und vierwöchigen Lernzeitraum - sind im Anhang 6.6 in den Tabellen 6.4 - 6.6 hinterlegt. Die Evaluierung im zweiwöchigen Lernzeitraum weißt dabei Genauigkeiten von 41,65% bis 50,28% auf. Entgegen der Annahme, dass aus der Validierung im vierwöchigen Lernzeitraum höhere Genauigkeiten resultieren, variieren diese lediglich zwischen 57,82% und 73,56%. Dafür sind die die evaluierten Genauigkeiten für vier Wochen leicht erhöht (42,36% - 56%).

Diese Resultate mögen in erster Instanz als unbrauchbar für eine Verhaltensprognose erscheinen, im Kontext des Gesamtkonzept ist dabei folgende Kenntnis zu beachten: Bei dem vorliegenden Eventalphabet (s. Tabelle 3.3) liegt die Wahrscheinlichkeit für eine richtige Prognose eines einzelnen Events bei ~5,56%. Eine ADL wie z.B. "Essen und Trinken" besteht aber aus mehreren unterschiedlichen Events (Bewegung in der Küche, Küchenschublade und Kühlschranktür öffnen oder schließen), d.h. AAL-Systeme, die eine Aktivitätsprognose realisieren, setzen nicht auf eine Erkennung in Abhängigkeit eines einzelnen Events, sondern berücksichtigen mehrere hintereinander folgende Eventprognosen. Ist in den Eventfolgen die "Bewegung im Flur" als Fehlprognose verzeichnet, kann die ADL "Essen und Trinken" anhand der übrigen zutreffenden Prognosen erkannt werden. Gleichermaßen gilt dies für alle übrigen ADL wie z. B. dem "Sich waschen und kleiden", "Ausscheiden" und "Raum und Zeit gestalten".

## 4.2 Ergänzende Ansätze

Mit den hier rudimentär beschriebenen Ansätzen wird nicht die Auswahl eines weiteren Konzept der Datenanalyse oder anderen herstellerabhängigen Sensoren vorgestellt. Denn der Entwurf und die Realisierung des AAL-Systems gleicht Funktionsbausteinen, deren Kern durch andere Algorithmen bzw. Prognosemodelle ersetzt werden könnte. Zudem ist die Wahl der Sensoren herstellerunabhängig und irrelevant (o.E. der geforderten Nutzerziele und -wünsche), denn durch den Einsatz der openHAB-Software wird die vorherrschende Inkompatibilität der herstellerabhängigen Hardwarekomponenten geschlossen. Somit kann für das ausgewählte Konzept jede Sensorik eingesetzt werden, die kategorische Messdaten als diskrete Events produziert. Die folgenden Ansätze befolgen somit Anwendungsszenarien, welche durch "Context-Aware Applikationen" abgedeckt werden und mit den in Abschnitt 2.4.2 gelisteten Kategorien von Algorithmen für AAL-Systeme korrespondieren. Neben einer "Anomaly Detection" und der "Event Prediction" (nutzbar für ein "Planning") kann ein AAL-System auch im Anwendungskontext

der "Activity Recognition", "Context Modeling" und "Location and Identify Identification" genutzt werden.

#### Ergänzender Ansatz Nr.1:

Um mit dem vorgestellten AAL-System zudem eine Aktivitätserkennung umzusetzen, kann die vom Endanwender als Teil der Kontextmodellierung formulierte openHAB-Sitemap das System um Kontextinformationen bereichern, die Auskunft über Einsatzort, Sensorart und Sensorbeschreibung liefern. Diese Informationen können dann in der Echtzeitüberwachung in einem ersten, trivialen Ansatz dazu dienen, die Kontextinformationen der Events, die am häufigsten Auftreten als Bezeichner einer Aktivität zu verwenden. Weißt die konfigurierte Sitemap zudem eine feingranulare Hierarchie auf, d.h. sind einzelne Etagen und deren Räume in verschachtelten Sitemap-Frames eingruppiert, können die Kontextinformationen auch eine Identifizierung des derzeitigen Aufenthaltsort der betroffenen Person liefern, dessen Verständnis dem Angehörigen auch ohne weitere Kontextgrößen möglich ist.

Derzeitig bieten die Persistence-Bindings lediglich das Speichern der Messdaten als Zeit-Werte-Paare an, d.h. die für die Realisierung des Ansatzes notwendigen Kontextinformationen können nicht mit abgespeichert werden. Zur Umsetzung des Ansatzes muss das eingesetzte Persistence-Binding (hier: MySQL-Binding) erweitert werden, sodass die aus der Sitemap geparsten Kontextinformationen mit abgespeichert werden können. Aufbauend auf der geleisteten Implementierung wird der Aufwand zur Realisierung - mit initialer Einarbeitungszeit - auf 30 Personentage geschätzt.

#### Ergänzender Ansatz Nr.2:

Die bisher vorgestellten Ansätze gehen davon aus, dass die bestehende Infrastruktur der vorliegenden Wohnung erweitert wird, d.h. die eingesetzten Bewegungsmelder sind zwar unauffällig aber präsent und sind für das AAL-System zu erstehen. Ausgehend von einem bestehenden Internetanschluss ermöglicht der ISP den Zugang durch einen mitgelieferten WAP, wodurch gerade im städtischen Wohnbereich eine Mehrfachverfügung von WAPs gewährleistet ist. In [Mas] ist eine Bewegungserkennung mit handelsüblichen WAPs realisiert, d.h. anstatt PIR-Sensoren einzusetzen, könnte die bestehende Hardware genutzt werden. Um darauf aufbauend eine Verhaltensabweichung oder Aktivitätserkennung zu realisieren oder weitere Kontextinformationen, wie Aufenthaltsort in der Wohnung oder im Raum zu erhalten, wird weiteres Wissen benötigt, inwieweit der menschliche Körper die Signale eines spez. Frequenzbandes der WAPs beeinflusst. Dazu wird exemplarisch in [Dep+16] eine Personenabschätzung realisiert. Zusammen mit WLAN-Fingerprints aus zuvor erstellten Heatmaps der Wohnung auf Basis von analytischen Ausbreitungsmodellen wie z. B. dem "Multi-Wall"-Modell [Zva+16] und materialspezifischen Dämpfungsfaktoren, könnte, in Abgleich mit WLAN-Fingerprints der

Erkennungsphase, die Bewegung einer Person erkannt und ein Lokalisierung mindestens raumgenau erfolgen.



## 5 Zusammenfassung und Fazit

Der demografische Wandel in Deutschland zeigt, dass die Generation 65+ bis zum Jahr 2060 um mehr als acht Millionen Menschen ansteigt. Zudem steigt auch das Risiko im Alter auf Pflege angewiesen zu sein und der Wunsch von älteren Menschen trotz Erkrankungen oder körperlicher Gebrechlichkeit weiterhin in ihrer gewohnten Umgebung leben zu können, anstatt in einer Pflegeeinrichtung ihr Leben zu fristen. Die Entscheidung für ein bestehendes Leben in der privaten Wohnung obliegt aber nicht allein der betroffenen Person, denn auch nahestehende Angehörige übernehmen meist eine Bürde, die sie nicht nur vor physische Belastungen stellt, sondern auch zu emotionalen Anspannungen führt. Das AAL-System als "unsichtbarer Pfleger" kann die psychischen Strapazen der Angehörigen lindern und den Wunsch der betroffenen Personen erfüllen. Dabei existiert bislang keine Sprache die den Alltag oder die ADL formalisiert. Um den Alltag bzw. das Verhalten der Personen dennoch nachzubilden, werden unterschiedlichste Verfahren und Methodiken aus dem Bereich des "Data Mining" von den AAL-Systemen angewandt.

Bei den AAL-Systemen unterscheidet die Literatur drei Varianten, deren Ziel gleichermaßen die Bewerkstelligung einer altersgerechten Assistenzfunktion ist: "Robotics"-Systeme helfen beim Aufstehen, Ankleiden oder bei der Zubereitung von Mahlzeiten. Die Systeme mit "Mobile and Wearable Sensors" realisieren ein Gesundheitsmonitoring, Aktivitätserkennung oder Erinnerungsfunktionen mit dem Defizit, dass die Sensorik an der Person angebracht ist. Die AAL-Systeme der "Smart Home" Variante - Synonym dazu der Begriff AHCS - setzen unterschiedliche Arten von Algorithmen ein, um z. B. eine Aktivitätserkennung, eine Kontextmodellierung oder eine Anomaliedetektion zu verwirklichen. Dabei ist keine Sensorik an der Person befestigt, sondern unauffällig und unaufdringlich in der Wohnung installiert. Es existieren fünf Subkategorien von Smart Homes, wobei das Proof-of-Concept dieser Arbeit ein AAL-System der "Event-tracking" und "Learning Home" Kategorien beschreibt. Die Systeme unter dem Begriff der "Context-Aware"-Applikationen greifen unterschiedliche Kontexte auf, um z. B. Aktivitätsmodelle oder Nutzerprofile zu erstellen - wobei der Begriff eher in dem übergeordneten Forschungsgebiet der Aml anzutreffen ist.

Für die Systementwicklung greifen die AAL-Systeme oftmals auf den agentenbasierten Ansatz des maschinellen Lernens zurück. Dabei stellen Systeme, die überwacht lernen, im Gegensatz zu Systemen die ein bestärkendes oder unüberwachtes Lernen implementieren, den Angehöri-

gen vor eine große Hürde in der initialen Konfiguration und Installation vor Inbetriebnahme des Systems. Das in dieser Arbeit realisierte AAL-System führt ein Lernen nach Inbetriebnahme eigenständig durch. Während der Echtzeitauswertung, dem Abgleich von gelernten und aktuellem Verhalten, ruft das System auftretende Verhaltensabweichungen aus und stellt zudem Eventprognosen. Ein Lernen in der Erkennungsphase ist bestärkt, da der informierte Angehörige evtl. Falschmeldungen im Lernprozess beeinflussen kann.

Schlussendlich ist es nicht nur der Heimautomatisierungssoftware "openHAB" zu verdanken, welche die Kluft zwischen herstellerabhängigen Standards und Hardwarekomponenten der Heim- und Gebäudeautomatisierung schließt, sondern auch der anhaltenden Strukturverkleinerung der Halbleiter, die eine Bauweise von leistungsstarken "Mini-PC" ermöglicht, dass sich die Personen der Zielgruppe 65+ aber auch deren Angehörigen spätestens im nächsten Jahrzehnt über einen produktreifen und autonomen "unsichtbaren Pfleger" erfreuen dürfen, dessen Anschaffungskosten den derer eines Aufenthaltes in einer Pflegeeinrichtung oder den Einsatz von geschulten Pflegepersonal unterschreiten.

# **6 Anhang**

## **6.1 Tabellarischer Vergleich ausgewählter AAL-Systeme**

Projekt (2.4.3)							
Hard- und Softwarekomponenten							
Fehlerklassen	Zeitfenster	Sensoren	Middleware	Formate	AlgoDat	Feldexperiment	
(2)	parallele ADL, täglich, wöchentlich [1]-[4]	O, C, M	eigene Entw., ProM	MXML, Event-Logs	CCT, TSM(+)	PN, ✓	
(3)	k.A. zeitnah	O, C, M	SOPHIA, Vita-phone	csv, XML, PDF.	k.A.	✓	
(4)	[1] zeitnah	C, M	k.A.	audiovisuell	k.A.	✓	
(5)	k.A.	ZigBee	ODS	regelbasiert	OSGi-FW, Bundles	✗, Entwicklungsphase	
(6)	Sturzerkennung k.A.	O, C, M, KNX	CMS	Text- u. Anruf-formate, XMPP	"LOC.SENS"	✓	
(10)	k.A. k-event	O, C, M, ZigBee	P.S.-Manager	XMPP	k-Label, SVM	✓	
(11)	[1]-[4] minütlich	O, C, M	CORBA, Zero-Conf, SQL	Events	ED, MDL, ALZ, MDP	O, Messdaten, Labor	
(12)	Patientenaktivität stündlich	O, M	MatLab	XML	FSM	✗, Datenanalyse	
(13)	[1]-[4] variabel	O, C, M	k.A.	Event-Logs	SAX, growth, k-Pattern, ANN, ATR, DT	✗	

**Tabelle 6.1:** Vergleich ausgewählter Projekte: Kriterien für Hard- und Softwarekomponenten

**Abkürzungen:**

k.A. keine Angabe | ✗ nicht realisiert | ✓ realisiert | [1]-[4] Fehlerklassen ADL (2.4.4) | **O-C-M**-Typisierung (2.3)

AlgoDat - Algorithmen u. Datenstrukturen

Projekt (2.4.3)		Anforderungen der Personas (2.1)				
Nutzerprofile	Echtzeitauswertung	Alarmierung	Schutz der Privatsphäre	Bedienung	Einfache Inbetriebnahme	
(2) 1 Pers, Besucher-Ident. geplant	X	X, Ampelzustand	✓	O, Editor f. Be- treuer	X	
(3) n Pers.	✓	✓, Ampelzustand	X, ext. Leitzentrale	O, Sicherheitsarm- band	k.A.	
(4) 1 Pers.	✓	✓, autom. Notruf	X	✓	✓	
(6) n Pers.	k.A.	✓, E-Mail, SMS, Telefon, etc.	X, ext. Zugriff, Community	O, Tablet (Anrdo- id, iOS)	✓, UCD	
(10) 1 Pers.	✓	X	✓	k.A.	✓, SHIB	
(11) 1 Pers.	X	X	k.A.	k.A.	k.A.	
(12) 1 Pat.	X	X	O, kontextbezogen	X	X	
(13) n Pers.	X	X	k.A.	X	X	

**Abkürzungen:** **Tabelle 6.2:** Vergleich ausgewählter Projekte: Kriterien für Anforderungen der Personas

k.A. keine Angabe | X - unzureichend umgesetzt | ✓ ausreichend erfüllt | O neutrale Bewertung

## 6.2 Tabellarischer Vergleich ausgewählter Bibliotheken

Bibliothek (2.6)	Programmier- sprache	Lizenz	GUI	API + Doku	Query- Manager	Formate	Algorithmen verfügbar?
(1)	Java	GPL	✓, Anleitung + Übung	○, reine Aufzählung	✗	CSV, Event- Logs	✗, Heuristic- Miner
(2)	Java, Groovy	AGPL	Übungen + Videos	○, Bsp.- Integration	Con.- Manager, SQL-Query	Excel, CSV, SQL	✗, NB, DT, ANN, SVM
(3)	Java	GPL 3.0 3.7.5	> Explorer, Knowledge Flow, etc.	○, Aufzählung, veraltet	✓	CSV, ARFF, JSON, Mat- Lab, etc.	○, FP- growth, SAX, PAX
(4)	k.A.	proprietär	Persona-l & Group- Training	k.A.	k.A.	CSV, TXT, XLS, ODBC	k.A.
(5)	Python	GPL	✓, Anleitung + Beispiele	✓, Beispiele	○, SQL- Table	CSV, TSV, XLS, Basket	✗, Regr. Trees, SVM, NB, ANN
(6)	R	GPL	○, eher IDE	✓, viele Bsp.	JDBC- Support	CSV, XLS, JSON	○, SVM, DT, ANN, FP-growth
(7)	Java	GPL 3.0	einfache UI	✗, Bsp. In- und Output	✗, kein SQL	CSV, ARFF, SPMF- Format	○, FP- growth, CPT+

**Tabelle 6.3:** Vergleich ausgewählter Software-Bibliotheken: "Data-Mining"

**Abkürzungen:**

k.A. keine Angabe | ✗ nicht realisiert | ✓ realisiert | ○ neutrale Bewertung

## 6.3 Java-Quellcode für k-Pattern-Ansatz

```

1: //notwendige Import-Anweisungen
2: ...
3: public class PaperApproachKPattern {
4:     public static void main(String[] args) throws Exception {
5:         // Retrieve data through db (declared in DatabaseUtils.
           props - jdbc con)
6:         InstanceQuery query = new InstanceQuery();
7:         query.setUsername("root");
8:         query.setPassword("openhab1!");
9:         // Query: Weekday, Hour, Minute, Sensor, Value
10:        query.setQuery("..._SQL-QUERY...");
11:        Instances datadb = query.retrieveInstances();
12:        // Lists, Arrays for ARFF
13:        ArrayList<Attribute> atts;
14:        ...
15:        ArrayList<String> attListSensor;
16:        Instances data;
17:        double[] vals;
18:        // List - contains all attributes on hierarchy lvl 1
19:        atts = new ArrayList<Attribute>();
20:        // 1.1 Attribute Weekday
21:        attVals = new ArrayList<String>();
22:        for (int i = 0; i < datadb.size(); i++) {
23:            attVals.add(String.valueOf((int)datadb.get(i).value(0))
                );
24:        }
25:        attListWeekday = new ArrayList<String>(new LinkedHashSet<
            String>(attVals));
26:        for (int i = 0; i < attListWeekday.size(); i++) {
27:            atts.add(new Attribute("w"+attListWeekday.get(i), new
                ArrayList<String>(Arrays.asList("t","f"))));
28:        }
29:        // 1.2 Attribute Hour...1.3 Attribute Minute...1.4
           Attribute for each Sensor (name or id possible)...
30:        // 2. Instance Object
31:        data = new Instances("DataFromDB", atts, 0);
32:        // 3. Fill Attributes with Data (per instance)
33:        int weekday, weekdayindex;
34:        int hour, hourindex;
35:        int minute, minuteindex;

```

```
36:     int sensor, sensorindex;
37:     for (int i=0; i<datadb.size(); i++) {
38:         //Array for values - initialize for each step mandatory
39:         vals = new double[data.numAttributes()];
40:         //3.1 Fill Weekday values: weekday=true for current
            instance, others=false; Weekday: true = 0, false = 1,
            Double.NaN = ?
41:         weekday = (int) datadb.get(i).value(0);
42:         weekdayindex = data.attribute("w"+weekday).index();
43:         vals[weekdayindex] = 0;
44:         for (int j=0; j < attListWeekday.size(); j++) {
45:             if (weekdayindex != data.attribute(j).index()) {
46:                 vals[j] = 1;
47:             }
48:         }
49:         //3.2 Fill Hour values...3.3 Fill Minute values...3.4
            Fill Sensor values...
50:         //4. Add Data to Instance Object
51:         data.add(new DenseInstance(1.0, vals));
52:     }
53:     //save the processed sql-query to arff
54:     ArffSaver saver = new ArffSaver();
55:     saver.setInstances(data);
56:     saver.setFile(new File("./DataforFPgrowth.arff"));
57:     saver.writeBatch();
58:     String[] options;
59:     //FP-growth needs ARFF as input
60:     ArffLoader loader = new ArffLoader();
61:     loader.setFile(new File("./DataforFPgrowth.arff"));
62:     Instances dataarff = loader.getDataSet();
63:     FPGrowth fpg = new FPGrowth();
64:     options = weka.core.Utils.splitOptions("-P_2");
65:     fpg.setOptions(options);
66:     fpg.setNumRulesToFind(10);
67:     fpg.buildAssociations(dataarff);
68:     System.out.println(fpg.getAssociationRules().getRules());
69: }
70: }
```

Quelltext 6.1: Weka: k-Pattern-Ansatz - Vorverarbeitung und FP-growth (Ausschnitt)



## 6.4 Visualisierungen der Anforderungen FA, QA, RB

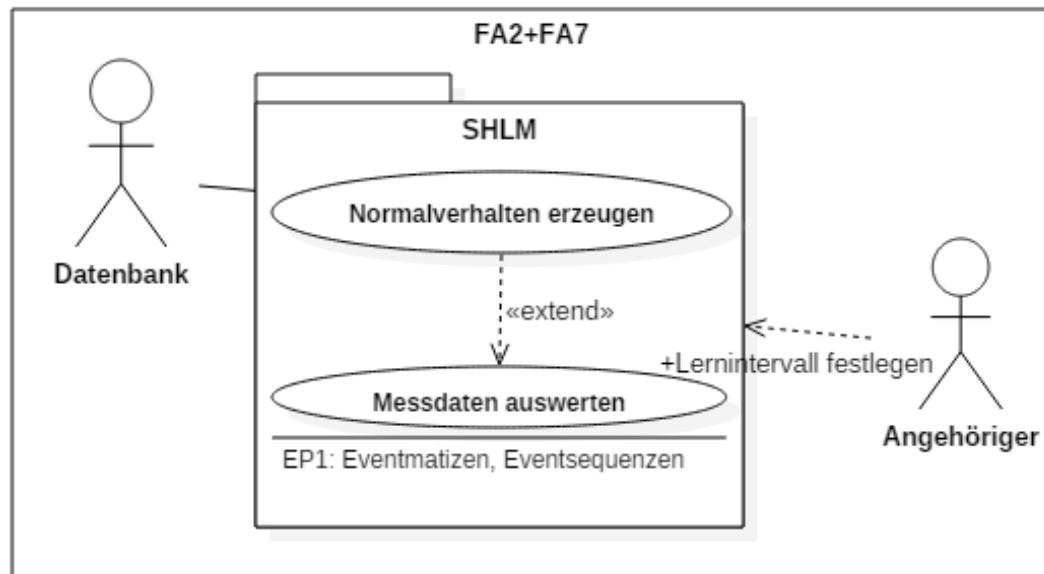


Abbildung 6.1: Use-Case-Diagramm: FA2+FA7

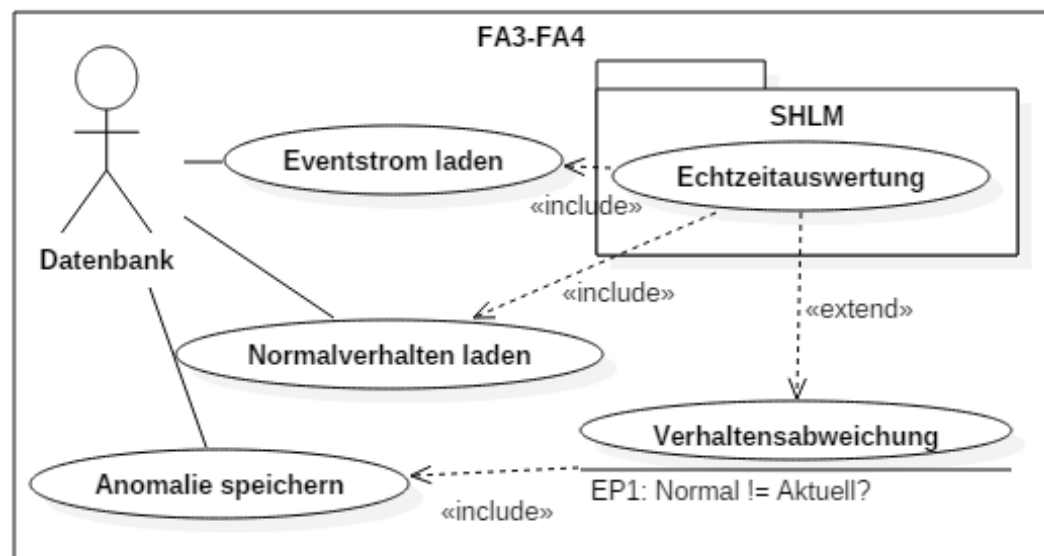


Abbildung 6.2: Use-Case-Diagramm: FA3-FA4

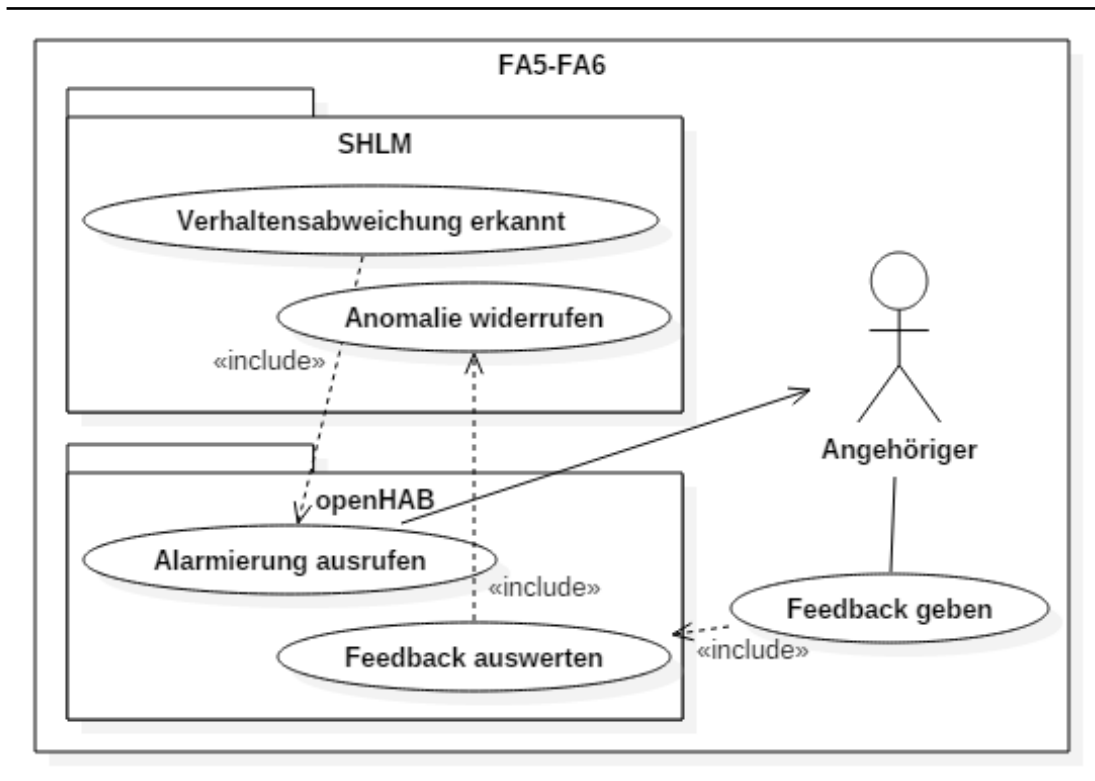


Abbildung 6.3: Use-Case-Diagramm: FA5-FA6

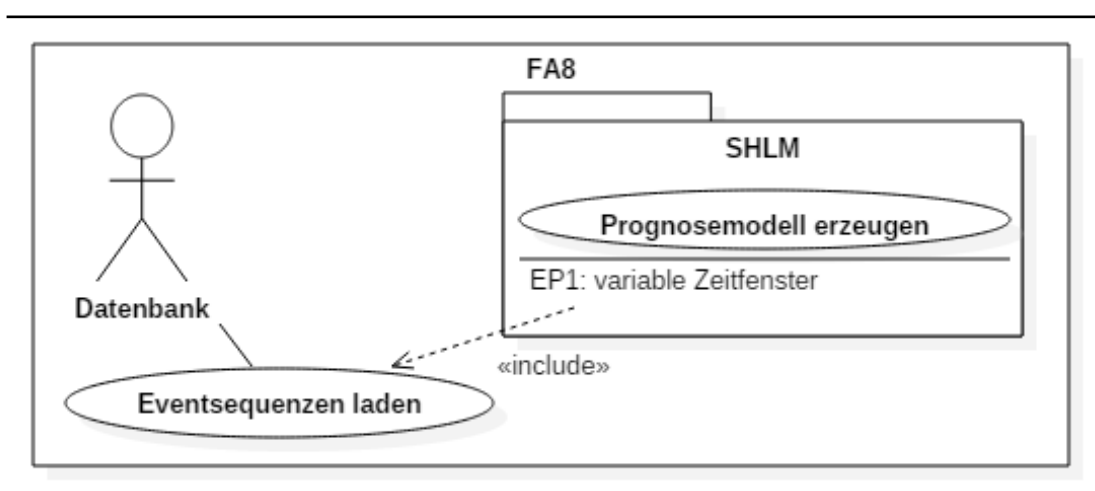


Abbildung 6.4: Use-Case-Diagramm: FA8

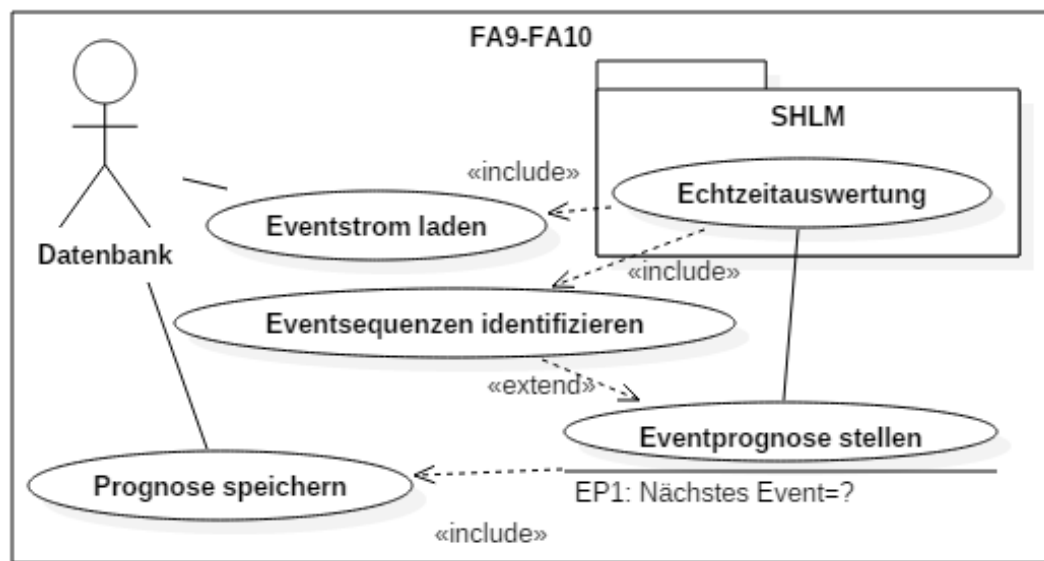


Abbildung 6.5: Use-Case-Diagramm: FA9-FA10

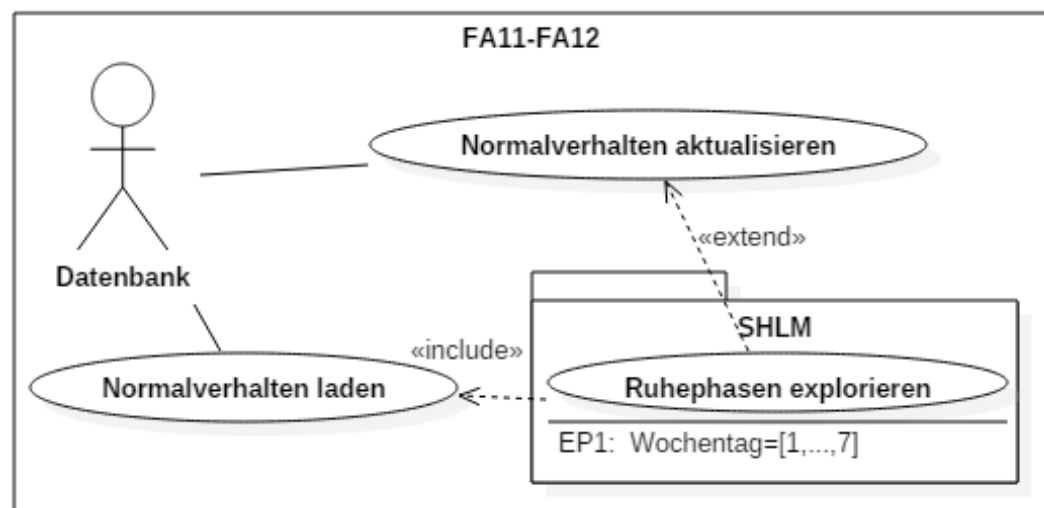


Abbildung 6.6: Use-Case-Diagramm: FA11-FA12

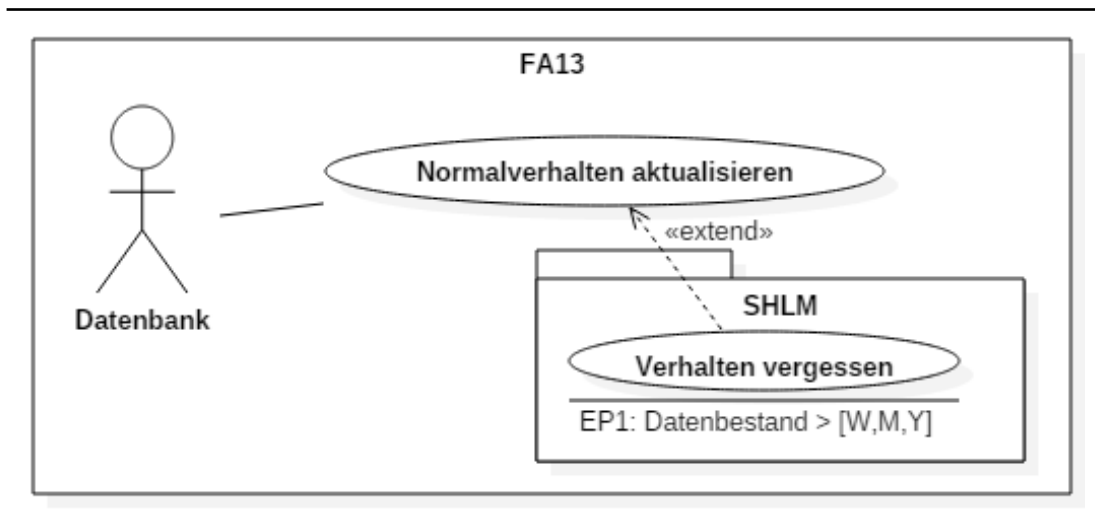


Abbildung 6.7: Use-Case-Diagramm: FA13

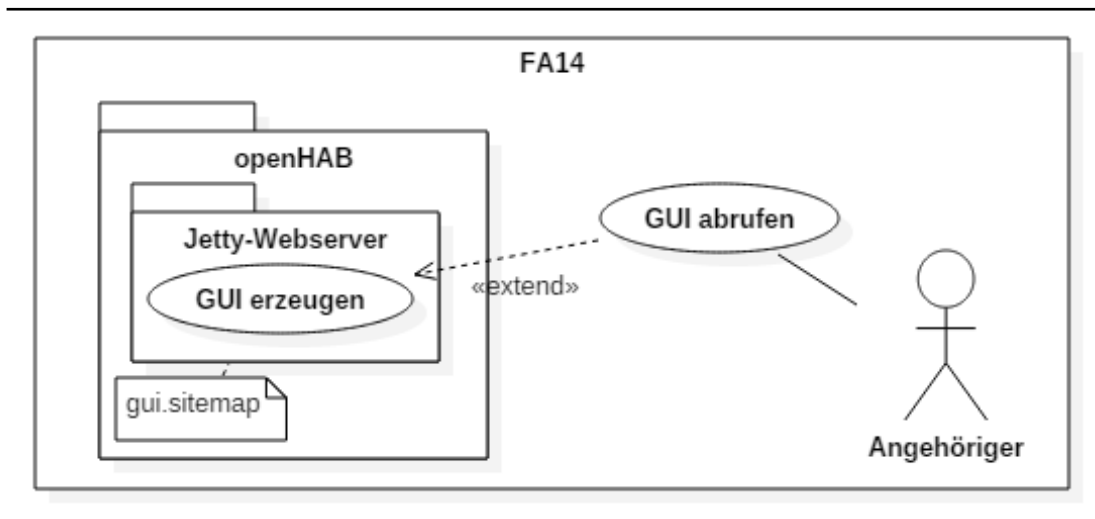


Abbildung 6.8: Use-Case-Diagramm: FA14

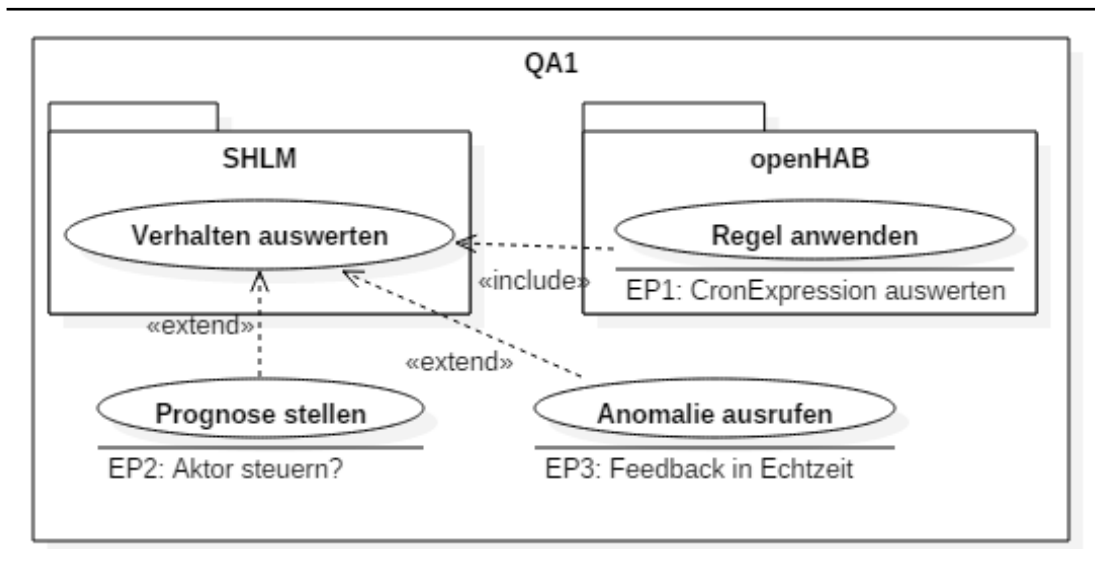


Abbildung 6.9: Use-Case-Diagramm: QA1

## 6.5 Auszüge der openHAB-Konfiguration

---

```

1: ...
2: # the database url like 'jdbc:mysql://<host>:<port>/<database>
   ' (without quotes)
3: mysql:url=jdbc:mysql://localhost:3306/openhab
4: # the database user
5: mysql:user=root
6: # the database password
7: mysql:password=openhab1!
8: # the reconnection counter
9: mysql:reconnectCnt=1
10: ...
11: # Hostname / IP address of the Homematic CCU
12: homematic:host=192.168.2.112
13: ...

```

---

**Quelltext 6.2:** Auszug der openHAB-Config: MySQL- und HomeMatic-Parameter

---

```

1: sitemap demo label="Smart_Home"
2: {
3:   Frame label="Räume" {
4:     Text label="3-Zimmer_Wohnung" icon="firstfloor" {
5:       Text item=C01
6:       Text item=C02
7:       Text item=C03
8:       Text item=C04
9:       Text item=M01
10:      Text item=M02
11:      Text item=M03
12:      Text item=M04
13:      Text item=M05
14:    }
15:  }
16:  Frame label="Smart_Home_Learning_Module_(SHLM)" {
17:    ...
18:  }
19: }

```

---

**Quelltext 6.3:** Auszug der openHAB-Sitemap

---

```

1: Group Home
2: Group Sensors (Home)

```

---

```

3: Group Actors (Home)
4: Group Motion (Sensors)
5: Group MContact (Sensors)
6: ...
7: Contact C01 "C01_Haustür_[%s]" <contact> (MContact) {homematic
    = "address=LEQ0404897,channel=1,parameter=STATE"}
8: Contact C02 "C02_Badezimmertür_[%s]" <contact> (MContact) {
    homematic="address=LEQ0404923,channel=1,parameter=STATE"}
9: ...
10: String M01 "M01_Wohnzimmer_[%s]" <switch> (Motion) {homematic=
    "address=LEQ0643729,channel=1,parameter=MOTION"}
11: String M02 "M02_Flur_[%s]" <switch> (Motion) {homematic="
    address=LEQ0643450,channel=1,parameter=MOTION"}
12: String M03 "M03_Schlafzimmer_[%s]" <switch> (Motion) {
    homematic="address=LEQ0643182,channel=1,parameter=MOTION"
    }
13: ...

```

---

#### Quelltext 6.4: Auszug der openHAB-Items

---

```

1: Strategies {
2:     everyFiveSecond : "0/5*_*_*_*_*?"
3:     default = everyChange
4: }
5:
6: Items {
7:     C01: strategy = everyChange, restoreOnStartup
8:     C02: strategy = everyChange, restoreOnStartup
9:     C03: strategy = everyChange, restoreOnStartup
10:    C04: strategy = everyChange, restoreOnStartup
11:    M01: strategy = everyChange, restoreOnStartup
12:    M02: strategy = everyChange, restoreOnStartup
13:    M03: strategy = everyChange, restoreOnStartup
14:    M04: strategy = everyChange, restoreOnStartup
15:    M05: strategy = everyChange, restoreOnStartup
16: }

```

---

#### Quelltext 6.5: Auszug der openHAB-MySQL-Persistence

---

## 6.6 Eventstreams, Validierung und Evaluierung

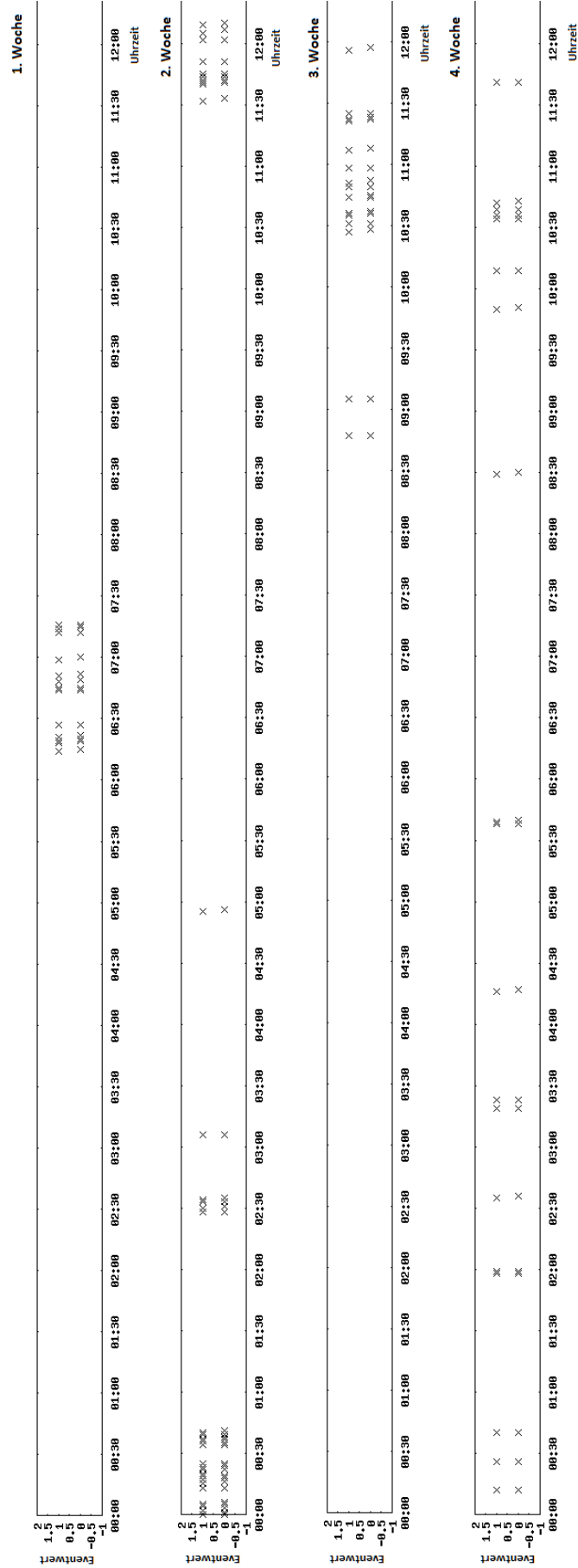


Abbildung 6.10: Eventstream: Bewegungsmelder 'M01' Wohnzimmer - Sonntags: 4 Wochen (1. Teil)



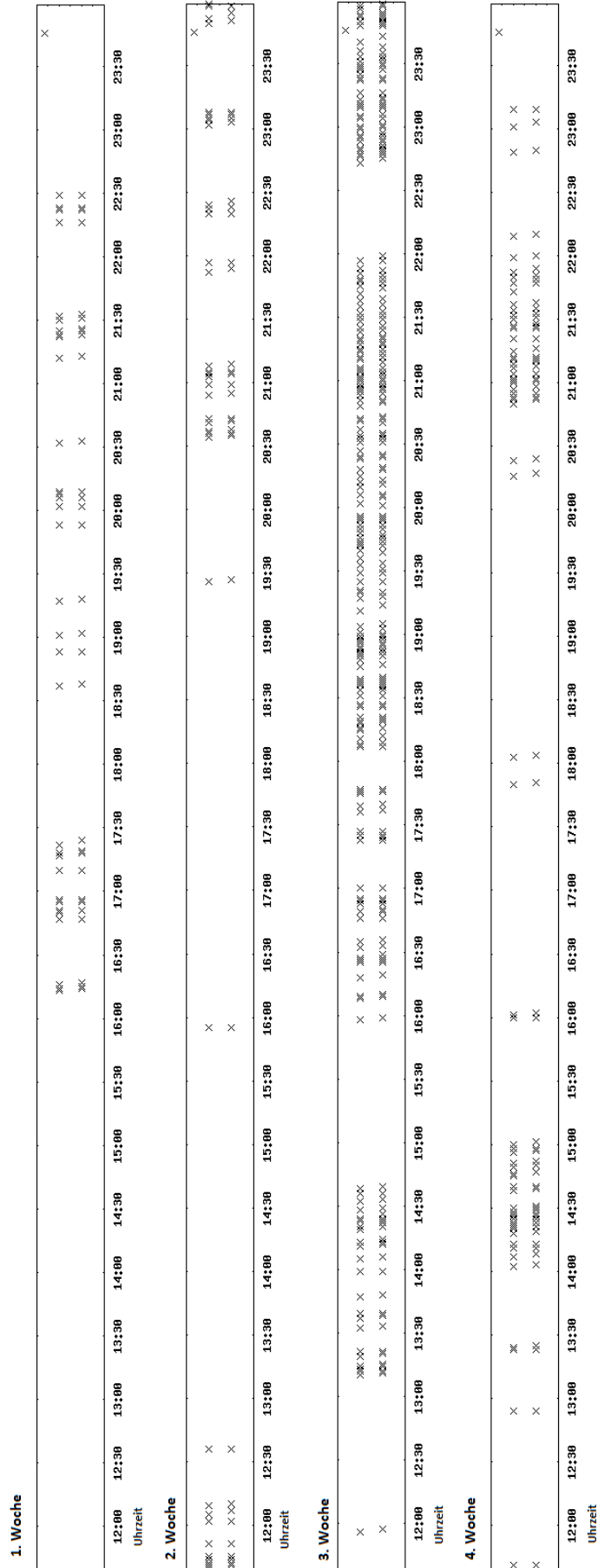
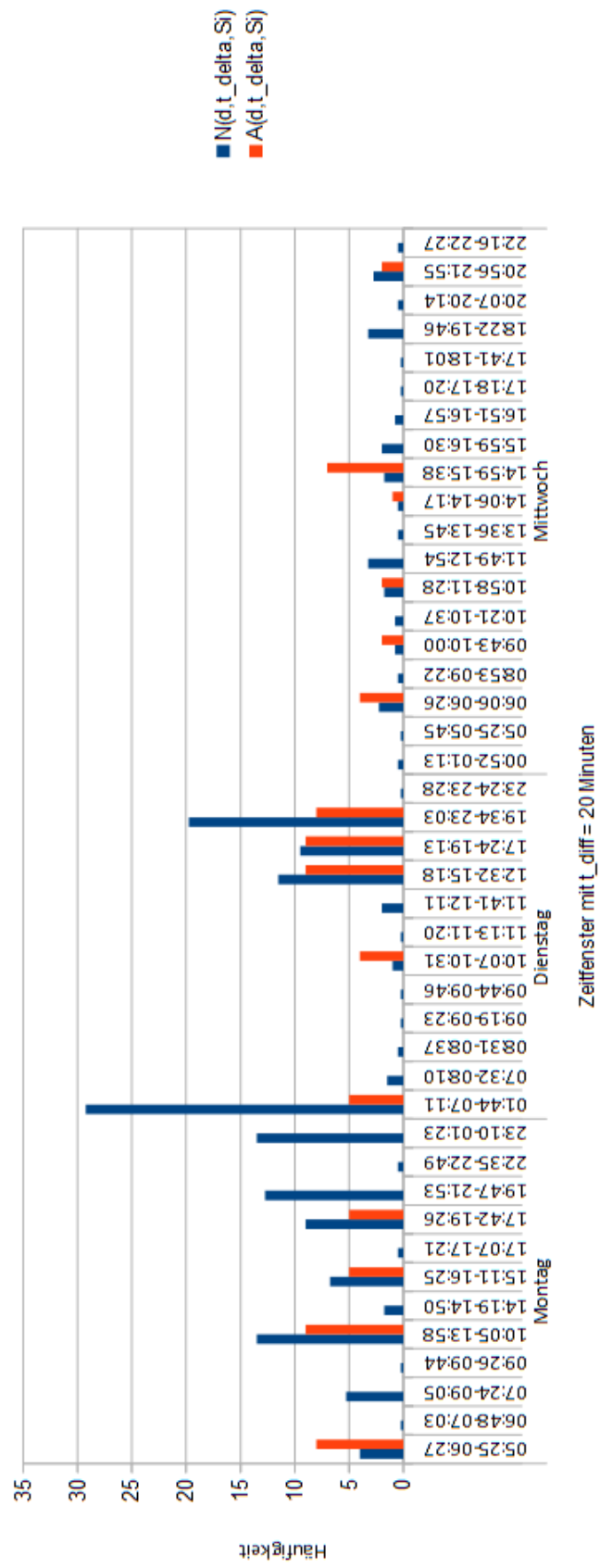


Abbildung 6.11: Eventstream: Bewegungsmelder 'M01' Wohnzimmer - Sonntags: 4 Wochen (2. Teil)

## Bewegung im Wohnzimmer: (ii) Vierwöchiger Lernzeitraum

(a) Validierung 23-29.01.2016

Abbildung 6.12: Ausschnitt Validierung der Woche "23-29.01-2016" - vierwöchiger Lernzeitraum ( $t_{diff} = 20$ )

## AKOM Evaluierung: (i) zweiwöchiger Lernzeitraum

$t_{db}$	k-Ordnung	Prog. Wahr	Prog. Falsch	Prog. Insg.	Genauigkeit (~%)
<i>(b) 16-22.01.2016</i>					
1	5	526	715	1241	42,39
2	5	334	417	751	44,47
3	5	254	315	569	44,64
4	5	231	239	470	49,15
5	5	196	204	400	49,00
1	10	532	709	1241	42,87
2	10	318	433	751	42,34
3	10	237	332	569	41,65
4	10	222	248	470	47,23
5	10	178	222	400	44,50
1	15	534	707	1241	43,03
2	15	332	419	751	44,21
3	15	246	323	569	43,23
4	15	224	246	470	47,66
5	15	186	214	400	46,50
<i>(b) 23-29.01.2016</i>					
1	5	443	555	998	44,39
2	5	287	337	624	46,00
3	5	232	252	484	47,93
4	5	198	203	401	49,38
5	5	178	176	354	50,28
1	10	440	558	998	44,09
2	10	269	355	624	43,11
3	10	213	271	484	44,01
4	10	182	219	401	45,39
5	10	164	190	354	46,33
1	15	449	549	998	44,99
2	15	275	349	624	44,07
3	15	218	266	484	45,04
4	15	186	215	401	46,38
5	15	165	189	354	46,61

Tabelle 6.4: Genauigkeitstabelle für AKOM: zweiwöchiger Lernzeitraum - Evaluierung (b)

## AKOM Validierung: (ii) vierwöchiger Lernzeitraum

$t_{db}$	k-Ordnung	Prog. Wahr	Prog. Falsch	Prog. Insg.	Genauigkeit (~%)
<i>(a) 16-22.01.2016</i>					
1	5	726	515	1241	58,50
2	5	451	300	751	60,05
3	5	329	240	569	57,82
4	5	286	184	470	60,85
5	5	243	157	400	60,75
1	10	768	473	1241	61,89
2	10	524	227	751	69,77
3	10	374	195	569	65,73
4	10	310	160	470	65,96
5	10	248	152	400	62,00
1	15	776	465	1241	62,53
2	15	524	227	751	69,77
3	15	371	198	569	65,20
4	15	310	160	470	65,96
5	15	253	147	400	63,25
<i>(a) 23-29.01.2016</i>					
1	5	613	385	998	61,43
2	5	390	234	624	62,50
3	5	298	186	484	61,57
4	5	253	148	401	63,10
5	5	209	145	354	59,04
1	10	645	353	998	64,63
2	10	459	165	624	73,56
3	10	339	145	484	70,04
4	10	263	138	401	65,59
5	10	222	132	354	62,71
1	15	649	349	998	65,03
2	15	451	173	624	72,28
3	15	341	143	484	70,45
4	15	270	131	401	67,33
5	15	220	134	354	62,15

Tabelle 6.5: Genauigkeitstabelle für AKOM: vierwöchiger Lernzeitraum - Validierung (a)

## AKOM Evaluierung: (ii) vierwöchiger Lernzeitraum

$t_{db}$	k-Ordnung	Prog. Wahr	Prog. Falsch	Prog. Insg.	Genauigkeit (~%)
<i>(b) 30.01-05.02.2016</i>					
1	5	555	621	1176	47,19
2	5	324	396	720	45,00
3	5	274	269	543	50,46
4	5	220	230	450	48,89
5	5	202	190	392	51,53
1	10	544	632	1176	46,26
2	10	305	415	720	42,36
3	10	250	293	543	46,04
4	10	209	241	450	46,45
5	10	183	209	392	46,68
1	15	533	643	1176	45,32
2	15	307	413	720	42,64
3	15	242	301	543	44,57
4	15	216	234	450	48,00
5	15	188	204	392	47,96
<i>(b) 06-12.02.2016</i>					
1	5	583	572	1155	50,48
2	5	385	347	732	52,60
3	5	304	258	562	54,10
4	5	253	209	462	54,76
5	5	229	180	409	56,00
1	10	579	576	1155	50,13
2	10	355	377	732	48,50
3	10	277	285	562	49,29
4	10	212	250	462	45,89
5	10	202	207	409	49,39
1	15	575	580	1155	49,78
2	15	365	367	732	49,86
3	15	275	287	562	48,93
4	15	211	251	462	45,67
5	15	195	214	409	47,67

Tabelle 6.6: Genauigkeitstabelle für AKOM: vierwöchiger Lernzeitraum - Evaluierung (b)

## 6.7 SHLM: Klassendiagramme

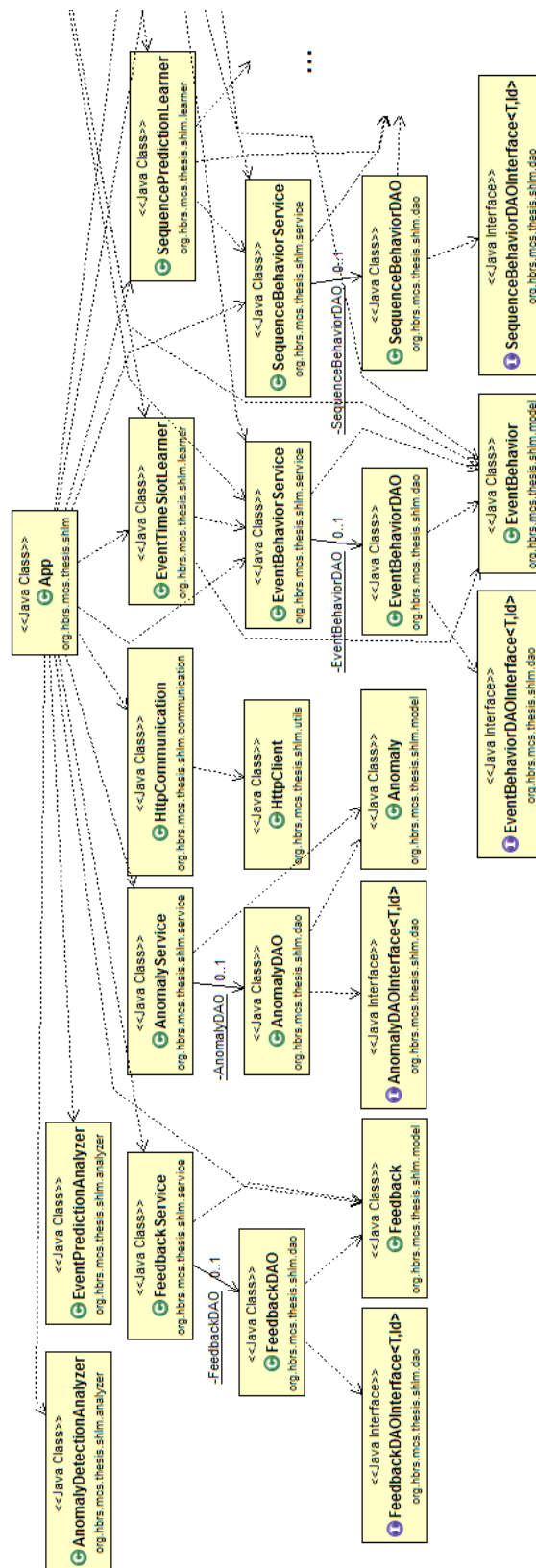


Abbildung 6.13: UML-Klassendiagramm des SHLM (alle Klassen - 1.Teil)

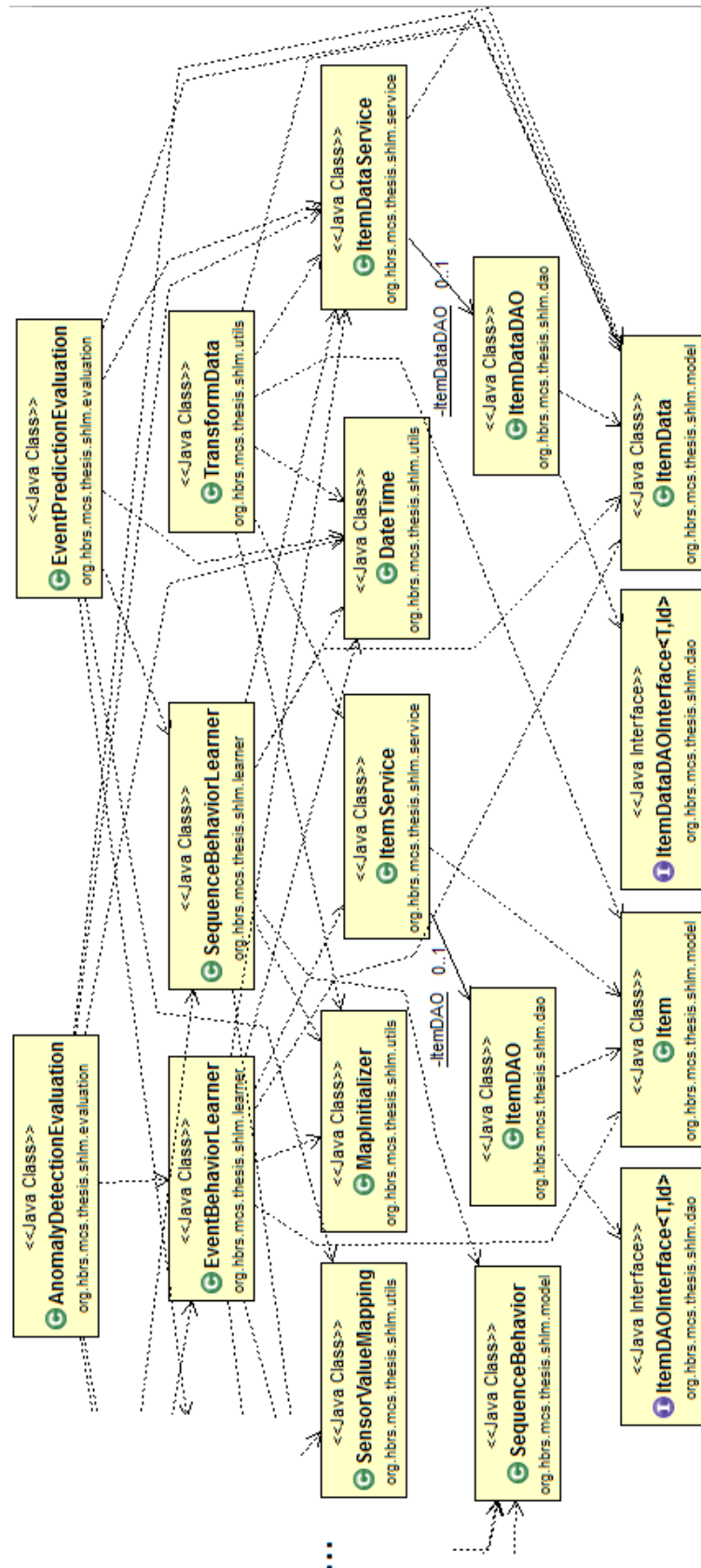


Abbildung 6.14: UML-Klassendiagramm des SHLM (alle Klassen - 2.Teil)

# Literaturverzeichnis

- [3 A16a] eQ 3 AG. *HomeMatic Zentralen - Zentrale CCU2*. 2016.  
URL: [http://files.elv.de/Assets/Produkte/10/1035/103584/Downloads/103584\\_ccu2\\_produktdatenblatt\\_data.pdf](http://files.elv.de/Assets/Produkte/10/1035/103584/Downloads/103584_ccu2_produktdatenblatt_data.pdf) (besucht am 25.03.2016) (zitiert auf Seite 71).
- [3 A16b] eQ 3 AG. *Sensor - Funk-Fensterkontakt*. 2016.  
URL: [http://files.elv.de/Assets/Produkte/13/1317/131775/Downloads/Funk-Fensterkontakt\\_131775\\_produktdatenblatt\\_V1.1.pdf](http://files.elv.de/Assets/Produkte/13/1317/131775/Downloads/Funk-Fensterkontakt_131775_produktdatenblatt_V1.1.pdf) (besucht am 25.03.2016) (zitiert auf Seite 71).
- [3 A16c] eQ 3 AG. *Sicherheit - Funk-Bewegungsmelder, innen*. 2016.  
URL: [http://files.elv.de/Assets/Produkte/13/1317/131776/Downloads/Funk-Bewegungsmelder-innen\\_131776\\_produktdatenblatt.pdf](http://files.elv.de/Assets/Produkte/13/1317/131776/Downloads/Funk-Bewegungsmelder-innen_131776_produktdatenblatt.pdf) (besucht am 25.03.2016) (zitiert auf Seite 71).
- [3 L16] eQ 3 Ltd. *Handbuch - HomeMatic WebUI - mit Erweiterungen für die HomeMatic Zentrale CCU2*. 2016.  
URL: [http://www.ehomeportal.de/downloads/homematic/103584\\_webui\\_de.pdf](http://www.ehomeportal.de/downloads/homematic/103584_webui_de.pdf) (besucht am 30.03.2016) (zitiert auf Seite 74).
- [AHGC15] pentaho A Hitachi Group Company. *Time Series Analysis and Forecasting with Weka*. 2015.  
URL: <http://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka> (besucht am 02.12.2015) (zitiert auf Seite 54).
- [AAL15a] Fraunhofer AAL. *Über Fraunhofer AAL*. 2015.  
URL: <http://www.aal.fraunhofer.de/about.html> (besucht am 28.09.2015) (zitiert auf Seite 5).
- [AAL15b] Fraunhofer AAL. *JUTTA - Just-in-Time-Assistance*. 2015.  
URL: <http://www.aal.fraunhofer.de/projects/JUTTA.html> (besucht am 09.11.2015) (zitiert auf Seite 30).
- [AAL15c] Fraunhofer AAL. *PareSense - Lautbasierte Steuerung von Gebäudetechnik, Sicherheitstechnik und Multimedia für Menschen in Pflegeumgebungen*. 2015.  
URL: <http://aal.fraunhofer.de/projects/PareSense.html> (besucht am 12.11.2015) (zitiert auf Seite 34).



- [AAL15d] Fraunhofer AAL. *sens@home - Notfallerkennung im häuslichen Umfeld*. 2015.  
URL: <http://www.aal.fraunhofer.de/projects/sensathome.html> (besucht am 10. 11. 2015) (zitiert auf Seite 31).
- [AG15] ELV Elektronik AG. *Smart Home Systeme – 8 Systeme zur Auswahl*. 2015.  
URL: <http://www.elv.de/hausautomation-smart-home.html> (besucht am 29. 10. 2015) (zitiert auf Seite 19).
- [al15] Heck R. et. al. *Preiswerte Do-It-Yourself-AAL für jedermann - Machbarkeitsbetrachtung für die Planung einer AAL-Lösung für die persönliche Wohnsituation*. 2015.  
URL: <https://www.vde-verlag.de/proceedings-en/453484022.html> (besucht am 12. 10. 2015) (zitiert auf Seiten 17, 19, 52).
- [alni] Meyer S. et. al. *Technische Assistenzsysteme für ältere Menschen - eine Zukunftsstrategie für die Bau- und Wohnungswirtschaft. Wohnen für ein langes Leben/AAL. Abschlussbericht*. Fraunhofer IRB Verlag, Juni 2015.  
(Zitiert auf Seite 27).
- [Alp10] Ethem Alpaydin. *Introduction to Machine Learning*. 2nd. The MIT Press, 2010.  
(Zitiert auf Seiten 47, 48).
- [AIG15] Wohnen im Alter Internet GmbH. *Aktivitäten des täglichen Lebens (ATL) Pflegemodell*. 2015.  
URL: <http://www.wohnen-im-alter.de/pflege-aktivitaeten-des-taeglichen-lebens-atl.html> (besucht am 22. 10. 2015) (zitiert auf Seite 10).
- [Art62] GILL Arthur. *Introduction to the theory of finite-state machines*. 1962.  
(Zitiert auf Seite 37).
- [Avg+15] Konstantinos Avgerinakis, Alexia Briassouli und Ioannis Kompatsiaris. *Activity detection and recognition of daily living events*. 2015.  
URL: [http://mklab.itl.gr/files/ADL\\_det\\_rec.pdf](http://mklab.itl.gr/files/ADL_det_rec.pdf) (besucht am 31. 10. 2015) (zitiert auf Seiten 18, 23).
- [Baj+07] Javier Bajo, Dante I. Tapia, Sara Rodríguez, Ana Luis und Juan M. Corchado. "Computational and Ambient Intelligence: 9th International Work-Conference on Artificial Neural Networks, IWANN 2007, San Sebastián, Spain, June 20-22, 2007. Proceedings". In: Hrsg. von Francisco Sandoval, Alberto Prieto, Joan Cebestany und Manuel Graña. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. Kap. Nature-Inspired Planner Agent for Health Care, S. 1090–1097.  
URL: [http://dx.doi.org/10.1007/978-3-540-73007-1\\_132](http://dx.doi.org/10.1007/978-3-540-73007-1_132) (zitiert auf Seite 39).
- [Bec+06] M. Becker, E. Werkman, M. Anastasopoulos und T. Kleinberger. "Approaching Ambient Intelligent Home Care Systems". In: *Pervasive Health Conference and Workshops, 2006*. 2006, S. 1–10.  
(Zitiert auf Seite 5).

- [BB15] Versicherungsanstalt öffentlich Bediensteter (BVA). *12 Aktivitäten des täglichen Lebens*. 2015.  
URL: <http://www.bva.at/portal27/portal/bvaportal/content/contentWindow?action=2&contentid=10007.677814> (besucht am 22. 10. 2015) (zitiert auf Seite 10).
- [Ber+16] Bruegge Bernd und H. Dutoit Allen. *Use Case Template*. 2016.  
URL: <http://faculty.washington.edu/jtenenbg/courses/360/s01/project/use-case-template.html> (besucht am 16. 03. 2016) (zitiert auf Seite 67).
- [Bou+15] Serge Thomas Mickala Bourobou und Younghwan Yoo. "User Activity Recognition in Smart Homes Using Pattern Clustering Applied to Temporal ANN Algorithm". In: *Sensors* 15.5 (2015), S. 11953.  
URL: <http://www.mdpi.com/1424-8220/15/5/11953> (zitiert auf Seiten 16, 38, 39, 55, 58).
- [Bun15a] Statistisches Bundesamt. *Bevölkerung Deutschlands bis 2060 - 13. koordinierte Bevölkerungsvorausberechnung*. 2015.  
URL: [https://www.destatis.de/DE/Publikationen/Thematisch/Bevoelkerung/VorausberechnungBevoelkerung/BevoelkerungDeutschland2060Presse5124204159004.pdf?\\_\\_blob=publicationFile](https://www.destatis.de/DE/Publikationen/Thematisch/Bevoelkerung/VorausberechnungBevoelkerung/BevoelkerungDeutschland2060Presse5124204159004.pdf?__blob=publicationFile) (besucht am 17. 08. 2015) (zitiert auf Seite 1).
- [Bun15b] Statistisches Bundesamt. *Die Generation 65+ in Deutschland*. 2015.  
URL: [https://www.destatis.de/DE/PresseService/Presse/Pressekonferenzen/2015/generation65/Pressebrochure\\_generation65.pdf?\\_\\_blob=publicationFile](https://www.destatis.de/DE/PresseService/Presse/Pressekonferenzen/2015/generation65/Pressebrochure_generation65.pdf?__blob=publicationFile) (besucht am 17. 08. 2015) (zitiert auf Seiten 1, 6).
- [Bun15c] Statistisches Bundesamt. *Pflegebedürftige nach Altersgruppen 1999-2060*. 2015.  
URL: [http://www.demografie-portal.de/SharedDocs/Downloads/DE/ZahlenFakten/pdf/Pflegebeduerftige\\_Anzahl.pdf?\\_\\_blob=publicationFile&v=2](http://www.demografie-portal.de/SharedDocs/Downloads/DE/ZahlenFakten/pdf/Pflegebeduerftige_Anzahl.pdf?__blob=publicationFile&v=2) (besucht am 17. 08. 2015) (zitiert auf Seite 1).
- [Bun15d] Bundesnetzagentur. *Allgemeinzuteilung von Frequenzen zur Nutzung durch Funkanwendungen mit geringer Reichweite für nicht näher spezifizierte Anwendungen*. 2015.  
URL: [https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen\\_Institutionen/Frequenzen/Allgemeinzuteilungen/2014\\_69\\_SRD\\_pdf.pdf?\\_\\_blob=publicationFile&v=1](https://www.bundesnetzagentur.de/SharedDocs/Downloads/DE/Sachgebiete/Telekommunikation/Unternehmen_Institutionen/Frequenzen/Allgemeinzuteilungen/2014_69_SRD_pdf.pdf?__blob=publicationFile&v=1) (besucht am 03. 11. 2015) (zitiert auf Seite 23).
- [Bun15e] Bundesnetzagentur. *Funkanwendungen auf den ISM-Bändern*. 2015.  
URL: <http://emf3.bundesnetzagentur.de/pdf/ISM-BNetzA.pdf> (besucht am 02. 11. 2015) (zitiert auf Seite 20).
- [Bun15f] Deutscher Bundestag. *I. Grundrechte*. 2015.  
URL: <https://www.bundestag.de/bundestag/aufgaben/rechtsgrun>

- dlagen/grundgesetz/gg\_01/245122 (besucht am 02. 11. 2015) (zitiert auf Seite 21).
- [Coo15] Diane J. Cook. *Health Monitoring and Assistance to Support Aging in Place*. 2015.  
URL: <http://eecs.wsu.edu/~cook/pubs/ph07.pdf> (besucht am 12. 11. 2015) (zitiert auf Seiten 35–37).
- [Coo+13] D.J. Cook, A.S. Crandall, B.L. Thomas und N.C. Krishnan. "CASAS: A Smart Home in a Box". In: *Computer* 46.7 (2013), S. 62–69.  
(Zitiert auf Seiten 34, 35).
- [Cor16a] Oracle Corp. *Download MySQL Community Server*. 2016.  
URL: <https://dev.mysql.com/downloads/mysql/> (besucht am 29. 03. 2016) (zitiert auf Seite 71).
- [Cor16b] Intel Corporation. *Mini-PC: Intel® NUC-Kit DN2820FYKH*. 2016.  
URL: <http://www.intel.de/content/www/de/de/nuc/nuc-kit-dn2820fykh.html> (besucht am 25. 03. 2016) (zitiert auf Seite 71).
- [Cor16c] Oracle Corporation. *How to download and install prebuilt OpenJDK packages*. 2016.  
URL: <http://openjdk.java.net/install/> (besucht am 30. 03. 2016) (zitiert auf Seite 74).
- [Dep+15] S. Depatla, A. Muralidharan und Y. Mostofi. "Occupancy Estimation Using Only WiFi Power Measurements". In: *Selected Areas in Communications, IEEE Journal on* 33.7 (2015), S. 1381–1393.  
(Zitiert auf Seite 19).
- [Dep+16] Saandeep Depatla, Arjun Muralidharan und Yasamin Mostofi. *Occupancy Estimation Using Only WiFi Power Measurements*. 2016.  
URL: [http://www.ece.ucsb.edu/~ymostofi/papers/JSAC15\\_DepatlaMuralidharanMostofi.pdf](http://www.ece.ucsb.edu/~ymostofi/papers/JSAC15_DepatlaMuralidharanMostofi.pdf) (besucht am 13. 04. 2016) (zitiert auf Seite 100).
- [Dey00] A. K. Dey. "Providing Architectural Support for Building Context-aware Applications". AAI9994400. Diss. Atlanta, GA, USA, 2000.  
(Zitiert auf Seite 14).
- [Ele+15] Institute of Electrical und Electronics Engineers (IEEE). *IEEE 802: OVERVIEW & ARCHITECTURE*. 2015.  
URL: <http://standards.ieee.org/about/get/802/802.html> (besucht am 29. 10. 2015) (zitiert auf Seite 19).
- [Ert09] W. Ertel. *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung*. Computational Intelligence. Vieweg+Teubner Verlag, 2009.  
URL: <https://books.google.de/books?id=G72CLusVtEkC> (zitiert auf Seiten 35, 47, 49, 50, 54).

- [FV+15] Alvaro Fides-Valero, Matteo Freddi, Francesco Furfari und Mohammad-Reza Tazari. *The PERSONA Framework for Supporting Context-Awareness in Open Distributed Systems*. 2015.  
URL: [http://www.researchgate.net/publication/225123221\\_The\\_PERSONA\\_Framework\\_for\\_Supporting\\_Context-Awareness\\_in\\_Open\\_Distributed\\_Systems](http://www.researchgate.net/publication/225123221_The_PERSONA_Framework_for_Supporting_Context-Awareness_in_Open_Distributed_Systems) (besucht am 11. 11. 2015) (zitiert auf Seite 32).
- [For15] Bundesministerium für Bildung und Forschung. *Assistenzsysteme im Dienste des älteren Menschen*. 2015.  
URL: [http://www.iat.eu/ehealth/downloads/20\\_plugin-projektportrats-aal.pdf?PHPSESSID=fb2d2398cdd5fd5af0877ea8da6179dc](http://www.iat.eu/ehealth/downloads/20_plugin-projektportrats-aal.pdf?PHPSESSID=fb2d2398cdd5fd5af0877ea8da6179dc) (besucht am 20. 08. 2015) (zitiert auf Seite 27).
- [Fou15] Free Software Foundation. *GNU PSPP*. 2015.  
URL: <http://www.gnu.org/software/pspp/> (besucht am 03. 12. 2015) (zitiert auf Seite 55).
- [Fou16a] The Apache Software Foundation. *HTTP Components*. 2016.  
URL: <https://hc.apache.org/> (besucht am 29. 03. 2016) (zitiert auf Seite 72).
- [Fou16b] The Apache Software Foundation. *Welcome to Apache Maven*. 2016.  
URL: <https://maven.apache.org/index.html> (besucht am 29. 03. 2016) (zitiert auf Seite 72).
- [Fou16c] The R Foundation. *Available CRAN Packages*. 2016.  
URL: <http://mirrors.softliste.de/cran/> (besucht am 03. 03. 2016) (zitiert auf Seite 55).
- [Fou16d] The R Foundation. *What is R?* 2016.  
URL: <https://www.r-project.org/about.html> (besucht am 03. 03. 2016) (zitiert auf Seite 54).
- [FV16a] Philippe Fournier-Viger. *Example 104 : Perform Sequence Prediction using the CPT+ Sequence Prediction Model*. 2016.  
URL: <http://www.philippe-fournier-viger.com/spmf/index.php?link=documentation.php#cptPlus> (besucht am 11. 04. 2016) (zitiert auf Seiten 92, 97).
- [FV16b] Philippe Fournier-Viger. *SPMF - An Open-Source Data Mining Library*. 2016.  
URL: <http://www.philippe-fournier-viger.com/spmf/> (besucht am 03. 03. 2016) (zitiert auf Seiten 55, 62, 85).
- [Fra16a] Java Code Geeks T. Fragkouli. *Hibernate JPA DAO Example*. 2016.  
URL: <https://examples.javacodegeeks.com/enterprise-java/hibernate/hibernate-jpa-dao-example/> (besucht am 29. 02. 2016) (zitiert auf Seite 82).

- [Fra16b] Eibe Frank. *Weka mailing list - Classification of time series data*. 2016.  
URL: <http://weka.8497.n7.nabble.com/Classification-of-time-series-data-td34996.html#a35022> (besucht am 07.03.2016) (zitiert auf Seite 57).
- [Fra16c] Eibe Frank. *Weka mailing list - procedures to implement FP - Growth*. 2016.  
URL: <http://weka.8497.n7.nabble.com/what-are-the-procedures-to-implement-Fp-Growth-using-weka-3-6-td35750.html> (besucht am 07.03.2016) (zitiert auf Seite 57).
- [GG10] VDI-Gesellschaft Bauen und Gebäudetechnik (GBG). *Assistenzfunktion zum Wohnen*. 2010.  
URL: [https://www.vdi.de/uploads/tx\\_vdirili/pdf/1566965.pdf](https://www.vdi.de/uploads/tx_vdirili/pdf/1566965.pdf) (besucht am 24.10.2015) (zitiert auf Seite 14).
- [gee16] Home Automation for geeks. *openHAB Persistence*. 2016.  
URL: [http://www.homeautomationforgeeks.com/openhab\\_persistence.shtml](http://www.homeautomationforgeeks.com/openhab_persistence.shtml) (besucht am 31.03.2016) (zitiert auf Seite 74).
- [GN12] College of Nursing Hartford Institute for Geriatric Nursing New York University. *Katz Index of Independence in Activities of Daily Living (ADL)*. 2012.  
URL: [http://consultgerirn.org/uploads/File/trythis/try\\_this\\_2.pdf](http://consultgerirn.org/uploads/File/trythis/try_this_2.pdf) (besucht am 23.10.2015) (zitiert auf Seite 11).
- [Gmb15a] Atral-Secal GmbH. *Speziell für Tierhalter - Bewegungsmelder mit Tierimmunfunktion*. 2015.  
URL: <http://www.daitem.de/pressemeldung/speziell-f%C3%BCr-tierhalter-bewegungsmelder-mit-tierimmun-funktion> (besucht am 02.11.2015) (zitiert auf Seite 19).
- [Gmb15b] CCDM GmbH. *ePEN...einfach genial, genial einfach! - Portfolio -Daily Care Journal*. 2015.  
URL: <http://www.epen.berlin/portfolio/daily-care-journal/> (besucht am 07.11.2015) (zitiert auf Seite 27).
- [Gmb15c] ELDAT GmbH. *FunkübertragungFrequenzen*. 2015.  
URL: [http://www.eldat.de/wissenswertes/infofunk/frequenzen\\_de.html](http://www.eldat.de/wissenswertes/infofunk/frequenzen_de.html) (besucht am 02.11.2015) (zitiert auf Seite 20).
- [Gmb15d] Rapid Miner GmbH. *Guide - How to Extend RapidMiner*. 2015.  
URL: <http://docs.rapidminer.com/downloads/RapidMiner-extensions.pdf> (besucht am 02.12.2015) (zitiert auf Seite 53).
- [Gmb15e] Rapid Miner GmbH. *The Open Source Platform of Choice*. 2015.  
URL: <https://rapidminer.com/> (besucht am 02.12.2015) (zitiert auf Seite 53).
- [Gmb15f] Rapid Miner GmbH. *Weka Extension*. 2015.  
URL: <https://marketplace.rapidminer.com/UpdateServer/faces/produ>

- ct\_details.xhtml?productId=rmx\_weka (besucht am 02.12.2015) (zitiert auf Seite 54).
- [Gmb15g] VDI/VDE Innovation + Technik GmbH. *Ökonomische Potenziale altersgerechter Assistenzsysteme*. 2015.  
URL: <http://www.mtidw.de/grundsatzfragen/begleitforschung/dokumente/oekonomische-potenziale-und-neuartige-geschaeftsmodelle-im-bereich-altersgerechter-assistenzsysteme-2> (besucht am 06.10.2015) (zitiert auf Seiten 6, 23–25).
- [Gmb15h] ZEIT Online GmbH. *Altenpflege - Der unsichtbare Pfleger*. 2015.  
URL: <http://www.zeit.de/2013/02/Pflege-Technologie-Ambient-Assisted-Living> (besucht am 18.08.2015) (zitiert auf Seite 1).
- [Gro+16a] Process Mining Group und Eindhoven Technical University. *Nightly Javadoc API documentation*. 2016.  
URL: <http://www.promtools.org/prom6/nightly/doc/> (besucht am 03.03.2016) (zitiert auf Seite 53).
- [Gro+16b] Process Mining Group und Eindhoven Technical University. *ProM 6 getting started*. 2016.  
URL: <http://www.promtools.org/doku.php?id=gettingstarted:start> (besucht am 03.03.2016) (zitiert auf Seite 53).
- [Gro+15] Process Mining Group und Eindhoven Technical University. *ProM Tools*. 2015.  
URL: <http://www.promtools.org/doku.php> (besucht am 01.12.2015) (zitiert auf Seiten 29, 53).
- [Hal+15] Mark Hall und Peter Reutemann. *WEKA KnowledgeFlow Tutorial for Version 3-5-8*. 2015.  
URL: <http://software.ucv.ro/~eganea/AIR/KnowledgeFlowTutorial-3-5-8.pdf> (besucht am 02.12.2015) (zitiert auf Seite 54).
- [Ham+13] Maik Hampicke, Carsten Hoherz, Harald Pötter, Robby Rochlitzer, Holger Quaas u. a. "BMBF-Projekt SELBST: Vom Konzept zum Produkt – Entwicklungen im Projekt SELBST". In: *Lebensqualität im Wandel von Demografie und Technik*. VDE-Verlag, 2013, S. 157–161.  
(Zitiert auf Seite 33).
- [Hei+03] E. O. Heierman und D. J. Cook. "Improving home automation by discovering regularly occurring device usage patterns". In: *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. 2003, S. 537–540.  
(Zitiert auf Seite 36).
- [Mas] *Herstellerübergreifende Heim-/Gebäudeautomatisierung beim Einsatz von open-HAB*. 2015.  
URL: [http://www2.inf.fh-bonn-rhein-sieg.de/~kjonas2m/lab/projects/verhaltensanalyse/2015\\_MCS\\_Praxisprojekt\\_SS15\\_Projektberic](http://www2.inf.fh-bonn-rhein-sieg.de/~kjonas2m/lab/projects/verhaltensanalyse/2015_MCS_Praxisprojekt_SS15_Projektberic)

- ht\_Peter\_Manheller.pdf (besucht am 18.08.2015) (zitiert auf Seiten 2, 19, 42, 55, 69, 100).
- [Hri+08] A. Hristova, A.M. Bernardos und J.R. Casar. "Context-aware services for ambient assisted living: A case-study". In: *Applied Sciences on Biomedical and Communication Technologies, 2008. ISABEL '08. First International Symposium on*. 2008, S. 1–5.  
(Zitiert auf Seite 5).
- [(Hr15] Olimpia.com (Hrsg.) *Systems and software engineering — Architecture description - ISO/IEC/IEEE 42010*. 2015.  
URL: <http://www.iso-architecture.org/ieee-1471/> (besucht am 04.12.2015) (zitiert auf Seite 56).
- [Hru+14] Peter Hruschka und Gernot Starke. *Knigge für Softwarearchitekten: reloaded*. Entwickler.press, 2014.  
(Zitiert auf Seite 69).
- [i15] rapid i. *Integrating RapidMiner into your application*. 2015.  
URL: [http://rapid-i.com/wiki/index.php?title=Integrating\\_RapidMiner\\_into\\_your\\_application](http://rapid-i.com/wiki/index.php?title=Integrating_RapidMiner_into_your_application) (besucht am 02.12.2015) (zitiert auf Seite 53).
- [Inc16] Google Inc. *UI: what's available to let the user fill data like a PIN code*. 2016.  
URL: <https://groups.google.com/forum/#!topic/openhab/H7SiUcrBD2I> (besucht am 23.04.2016) (zitiert auf Seite 86).
- [Inc15] SPSS Inc. *PASW Statistics*. 2015.  
URL: <http://www.spss.com.hk/statistics/> (besucht am 03.12.2015) (zitiert auf Seite 55).
- [Inf15] Deutsches Institut für Medizinische Dokumentation und Information. *Hamburger Einstufungsmanual zum Barthel-Index*. 2015.  
URL: [http://www.dimdi.de/static/de/klassi/icd-10-gm/systematik/hamburger\\_manual.pdf](http://www.dimdi.de/static/de/klassi/icd-10-gm/systematik/hamburger_manual.pdf) (besucht am 23.10.2015) (zitiert auf Seite 11).
- [IPA15] Fraunhofer IPA. *safe@home*. 2015.  
URL: [http://www.ipa.fraunhofer.de/safe\\_at\\_home.html](http://www.ipa.fraunhofer.de/safe_at_home.html) (besucht am 20.08.2015) (zitiert auf Seite 30).
- [JBo16] JBossDeveloper. *Hibernate ORM - Idiomatic persistence for Java and relational databases*. 2016.  
URL: <http://hibernate.org/orm/> (besucht am 29.03.2016) (zitiert auf Seite 71).
- [K.15] Kreuzer K. *openHAB 1.7 und 2.0 alpha 2 Release*. 2015.  
URL: <http://kaikreuzer.blogspot.de/> (besucht am 18.08.2015) (zitiert auf Seite 2).

- [KZ+16] Haigh Karen Zita und Kiff Liana M. *The Independent LifeStyle Assistant (I.L.S.A.): AI Lessons Learned*. 2016.  
URL: <http://www.cs.cmu.edu/~khaigh/papers/haigh-iaai04.pdf> (besucht am 29.02.2016) (zitiert auf Seite 40).
- [Keo16] Eamonn Keogh. *Naive Bayes Classifier*. 2016.  
URL: [http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect\\_examples.pdf](http://www.cs.ucr.edu/~eamonn/CE/Bayesian%20Classification%20withInsect_examples.pdf) (besucht am 29.02.2016) (zitiert auf Seite 35).
- [KG15] Heise Medien GmbH & Co. KG. *Intel-Chef verabschiedet sich vom bisherigen "Moore's Law"*. 2015.  
URL: <http://www.heise.de/newsticker/meldung/Intel-Chef-verabschiedet-sich-vom-bisherigen-Moore-s-Law-2751848.html> (besucht am 29.09.2015) (zitiert auf Seite 7).
- [Ke09] Medizinischer Dienst des Spitzenverbandes Bund der Krankenkassen e.V. *Richtlinien des GKV-Spitzenverbandes zur Begutachtung von Pflegebedürftigkeit nach dem XI. Buch des Sozialgesetzbuches*. 2009.  
URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.362.4175> (besucht am 23.10.2015) (zitiert auf Seite 11).
- [Laf+01] John D. Lafferty, Andrew McCallum und Fernando C. N. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Proceedings of the Eighteenth International Conference on Machine Learning*. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, S. 282–289.  
URL: <http://dl.acm.org/citation.cfm?id=645530.655813> (zitiert auf Seite 35).
- [LeB+06] G. LeBellego, N. Noury, G. Virone, M. Mousseau und J. Demongeot. "A model for the measurement of patient activity in a hospital suite". In: *Information Technology in Biomedicine, IEEE Transactions on* 10.1 (2006), S. 92–99.  
(Zitiert auf Seiten 37, 38).
- [Lim16] Object Refinery Limited. *Welcome To JFreeChart*. 2016.  
URL: <http://www.jfree.org/jfreechart/> (besucht am 11.04.2016) (zitiert auf Seite 141).
- [Lin+14] Yu-Feng Lin, Hsin-Han Shie, Yi-Ching Yang und Vincent S. Tseng. "Design of a Real-Time and Continua-Based Framework for Care Guideline Recommendations". In: *International Journal of Environmental Research and Public Health* 11.4 (2014), S. 4262.  
URL: <http://www.mdpi.com/1660-4601/11/4/4262> (zitiert auf Seite 15).
- [Lin+15] Norbert Link, Barbara Steiner, Marius Pflüger, Julia Kroll und Ronny Egeler. *safe@home - Erste Erfahrungen aus dem Praxiseinsatz zur Notfallerkennung mit optischen Sensoren*. 2015.



- URL: [http://publica.fraunhofer.de/eprints/urn\\_nbn\\_de\\_0011-n-2498958.pdf](http://publica.fraunhofer.de/eprints/urn_nbn_de_0011-n-2498958.pdf) (besucht am 10. 11. 2015) (zitiert auf Seiten 30, 31).
- [Lin15] Hendrik Linka. *Laufende Arbeiten - Entwicklung einer intelligenten, selbstversorgenden Türüberwachung mit Hilfe von modernen Single-Board Computern*. 2015. URL: <http://mc-lab.inf.h-brs.de/projects/personenerkennung/> (besucht am 23. 11. 2015) (zitiert auf Seite 44).
- [Liv15a] Fraunhofer-Allianz Ambient Assisted Living. *Sensorbasiertes adaptives Monitoringsystem für die Verhaltensanalyse von Senioren*. 2015. URL: <http://www.aal.fraunhofer.de/projects/SAMDY.html> (besucht am 20. 08. 2015) (zitiert auf Seite 28).
- [Liv15b] Fraunhofer-Allianz Ambient Assisted Living. *Unsere Projekte*. 2015. URL: <http://www.aal.fraunhofer.de/projects.html> (besucht am 20. 08. 2015) (zitiert auf Seite 27).
- [Liv15c] Fraunhofer-Allianz Ambient Assisted Living. *Willkommen bei der Fraunhofer-Allianz AAL*. 2015. URL: <http://www.aal.fraunhofer.de/index.html> (besucht am 20. 08. 2015) (zitiert auf Seite 27).
- [LLC15] Loyolan Ventures LLC. *INTEL CORE I5-3427U BENCHMARKS – DC53427HYE NUC IN LINUX*. 2015. URL: <http://www.servethehome.com/intel-core-i5-3427u-benchmarks-dc53427hye-nuc-linux/> (besucht am 03. 11. 2015) (zitiert auf Seite 22).
- [LLC16] Tangient LLC. *Use WEKA in your Java code*. 2016. URL: <https://weka.wikispaces.com/Use+WEKA+in+your+Java+code> (besucht am 07. 03. 2016) (zitiert auf Seite 57).
- [May04] Rene Mayrhofer. "An Architecture for Context Prediction". In: *In Advances in Pervasive Computing, number 3-85403-176-9. Austrian Computer Society (OCG*. 2004. (Zitiert auf Seiten 12–14, 20, 44).
- [Min15] Orange Data Mining. *Orange Data Mining Library*. 2015. URL: <http://docs.orange.biolab.si/3/data-mining-library/> (besucht am 03. 12. 2015) (zitiert auf Seite 54).
- [MK12] Maria Mischo-Kelling. *Zur Theorie des Pflegehandelns - eine explorative Studie zur Bedeutung des Selbst, Selbstkonzepts und Körperbildnis für die Transformation des Pflegemodells von Roper, Logan & Tierney in eine pragmatisch-interaktionistische Theorie des Pflegehandelns*. Freiburg i.B., 2012. (Zitiert auf Seite 10).
- [Mü15] Ludwig-Maximilians-Universität München. *ELKI: Environment for Developing KDD-Applications Supported by Index-Structures*. 2015.

- URL: <http://sci2s.ugr.es/keel/algorithms.php> (besucht am 03.12.2015) (zitiert auf Seite 55).
- [Mun15] Marco Munstermann. *Technisch unterstützte Pflege von morgen - Innovative Aktivitätserkennung und Verhaltensermittlung durch ambiente Sensorik*. 2015. Aufl. Wiesbaden: Springer Fachmedien Wiesbaden, 2015. (Zitiert auf Seiten 1, 6, 9–23, 25, 26, 28–30, 43–48, 53, 59, 60, 77).
- [Nic+16] Mora Niccola, Losardo Agostino, De Munari Ilaria und Ciampolini Paolo. *Self-tuning behavioral analysis in AAL "FOOD" project pilot environments*. 2016. URL: [https://www.researchgate.net/publication/281167822\\_Self-tuning\\_behavioral\\_analysis\\_in\\_AAL\\_FOOD\\_project\\_pilot\\_environments](https://www.researchgate.net/publication/281167822_Self-tuning_behavioral_analysis_in_AAL_FOOD_project_pilot_environments) (besucht am 29.02.2016) (zitiert auf Seite 40).
- [Nie12] Lene Nielsen. *Personas - User Focused Design*. 1. Aufl. Berlin Heidelberg: Springer Science u. Business Media, 2012. (Zitiert auf Seite 7).
- [Nor15] Donald A. Norman. *Ad-Hoc Personas u. Empathetic Focus*. 2015. URL: [http://www.jnd.org/dn.mss/personas\\_empath.html](http://www.jnd.org/dn.mss/personas_empath.html) (besucht am 20.10.2015) (zitiert auf Seite 7).
- [onl15] Pharmazeutische Zeitung online. *Wenn Helfer Hilfe brauchen*. 2015. URL: <http://www.pharmazeutische-zeitung.de/index.php?id=38926> (besucht am 18.08.2015) (zitiert auf Seite 1).
- [PH13] Helga Pelizäus-Hoffmeister. *Zur Bedeutung von Technik im Alltag Älterer - Theorie und Empirie aus soziologischer Perspektive*. 1. Aufl. Berlin Heidelberg New York: Springer-Verlag, 2013. (Zitiert auf Seite 9).
- [Pir+12] Hamed Pirsiavash und Deva Ramanan. "Detecting Activities of Daily Living in First-person Camera Views". In: *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE. 2012. (Zitiert auf Seite 18).
- [Poh+15] Klaus Pohl und Chris Rupp. *Basiswissen Requirements Engineering - Aus- und Weiterbildung nach IREB-Standard zum Certified Professional for Requirements Engineering Foundation Level*. 4. überarb. Aufl. Heidelberg: Dpunkt.Verlag GmbH, 2015. (Zitiert auf Seiten 64–67, 69).
- [Ras+13] P. Rashidi und A. Mihailidis. "A Survey on Ambient-Assisted Living Tools for Older Adults". In: *Biomedical and Health Informatics, IEEE Journal of* 17.3 (2013), S. 579–590. (Zitiert auf Seiten 1, 18–21, 24–26).

- [RFo16] RForge.net. *JRI - Java/R Interface*. 2016.  
URL: <https://rforge.net/JRI/> (besucht am 04.03.2016) (zitiert auf Seite 55).
- [Ris98] Jorma Rissanen. *Stochastic complexity in statistical inquiry*. Bd. 15. World scientific, 1998.  
(Zitiert auf Seite 36).
- [Rus+12] Stuart Russell und Peter Norvig. *Künstliche Intelligenz : Ein moderner Ansatz*. 3. Aufl. Pearson, Higher Education, 2012.  
(Zitiert auf Seiten 47–52).
- [Sch15a] Oliver Schonschek. *Recht auf Vergessen: Nicht nur Google und Bing sind betroffen*. 2015.  
URL: <http://www.searchsecurity.de/lernprogramm/Recht-auf-Vergessen-Nicht-nur-Google-und-Bing-sind-betroffen> (besucht am 03.11.2015) (zitiert auf Seite 22).
- [Sch15b] Wolfrid Schröer. *SAMDY - Sensorbasiertes adaptives Monitoringsystem für die Verhaltensanalyse von Senioren : Schlußbericht zum Teilvorhaben*. 2015.  
URL: <http://opac.tib.uni-hannover.de/DB=1/SET=1/TTL=8/MAT=/NOMAT=T/CLK?IKT=12&TRM=816118779> (besucht am 09.11.2015) (zitiert auf Seite 28).
- [Sco16] W. Ambler Scott. *UML 2 Use Case Diagrams: An Agile Introduction*. 2016.  
URL: <http://www.agilemodeling.com/artifacts/useCaseDiagram.htm> (besucht am 16.03.2016) (zitiert auf Seite 67).
- [Scu+15] David Scuse und Peter Reutemann. *WEKA Experimenter Tutorial for Version 3-5-8*. 2015.  
URL: <http://www.inf.ed.ac.uk/teaching/courses/dme/experimenter-tutorial.pdf> (besucht am 02.12.2015) (zitiert auf Seite 54).
- [SE15] SAP SE. *SAP Extends the Power of Predictive Analytics to Unlock Big Data With Acquisition of KXEN*. 2015.  
URL: <http://global.sap.com/corporate-en/news.epx?PressID=21521> (besucht am 02.12.2015) (zitiert auf Seite 54).
- [Sen13] Michael Sengpiel. "Das SMILEY-Projekt: Von der Anforderungsanalyse zu Apps für ein selbständiges Leben". In: *Lebensqualität im Wandel von Demografie und Technik*. VDE-Verlag, 2013, S. 226–227.  
(Zitiert auf Seite 32).
- [She15] GMDH Shell. *Choose Your GMDH Shell Version*. 2015.  
URL: <https://www.gmdhshell.com/choose-version> (besucht am 03.12.2015) (zitiert auf Seite 54).
- [Shi+12] Karen A. Shire, Jan Marco Leimeister, Karen A. Shire und Jan Marco Leimeister. "JUTTA – JUSt-in-Time Assistance: Betreuung und Pflege nach Bedarf".

- In: *Technologiegestützte Dienstleistungsinnovation in der Gesundheitswirtschaft*. Springer-Verlag, 2012, S. 325–349.  
(Zitiert auf Seite 30).
- [Son+11] Kai-Tai Song und Wei-Jyun Chen. “Human activity recognition using a mobile camera”. In: *Ubiquitous Robots and Ambient Intelligence (URAI), 2011 8th International Conference on*. 2011, S. 3–8.  
(Zitiert auf Seite 18).
- [Sta+15] Gernot Starke und Peter Hruschka. *arc42 - Ressourcen für Softwarearchitekten*. 2015.  
URL: <http://www.arc42.de/index.html> (besucht am 07.12.2015) (zitiert auf Seite 56).
- [Ste+12] A. Steinhage und C. Lauterbach. “Markteinführung eines AAL-Produkts aus Hersteller- und Kundensicht: Praxiserfahrungen mit der SensFloor-Matte”. In: *Technik für ein selbstbestimmtes Leben (AAL 2012)*. VDE-Verlag, 2012.  
(Zitiert auf Seite 33).
- [Ste+13a] Axel Steinhage, Janosch Rieß, Philipp Novotny und Herbert Plischke. “Das SensBed - eine Weiterentwicklung der SensFloor-Technologie zur Schlafanalyse”. In: *Lebensqualität im Wandel von Demografie und Technik*. VDE-Verlag, 2013, S. 517–520.  
(Zitiert auf Seite 33).
- [Ste+13b] Axel Steinhage und Christl Lauterbach. “Ein Sensorfußboden zur Verbesserung der Pflegequalität: Ergebnisse des SensFloor - Projekts”. In: *Lebensqualität im Wandel von Demografie und Technik*. VDE-Verlag, 2013, S. 517–520.  
(Zitiert auf Seite 33).
- [Sto+12] J.A. Stork, L. Spinello, J. Silva und K.O. Arras. “Audio-based human activity recognition using Non-Markovian Ensemble Voting”. In: *RO-MAN, 2012 IEEE*. 2012, S. 509–514.  
(Zitiert auf Seite 18).
- [Suk+15] P. Sukanya und K. S. Gayathri. *An Unsupervised Pattern Clustering Approach for Identifying Abnormal User Behaviors in Smart Homes*. 2015.  
URL: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.362.4175> (besucht am 23.10.2015) (zitiert auf Seiten 16, 24, 38, 55).
- [SI15] Fraunhofer-Institut für Software-und Systemtechnik ISST. *SMILEY Projektabschlussbericht*. 2015.  
URL: [http://www.isst.fraunhofer.de/de/e-health\\_infrastrukturen\\_und\\_dienste/it-fuer-die-plege-und-wohnassistentz/referenzprojekte/smiley.html](http://www.isst.fraunhofer.de/de/e-health_infrastrukturen_und_dienste/it-fuer-die-plege-und-wohnassistentz/referenzprojekte/smiley.html) (besucht am 11.11.2015) (zitiert auf Seiten 32, 33, 74).
- [Tec16] House\_n Research Group Massachusetts Institute of Technology. *House\_n Materials and Media*. 2016.

- URL: [http://web.mit.edu/cron/group/house\\_n/publications.html](http://web.mit.edu/cron/group/house_n/publications.html)  
(besucht am 29.02.2016) (zitiert auf Seite 40).
- [Ted16] Gueniche Ted. *IPredict, a sequence prediction Framework*. 2016.  
URL: <https://github.com/tedgueniche/IPredict/blob/master/README.md>  
(besucht am 11.03.2016) (zitiert auf Seite 62).
- [Ted+16] Gueniche Ted, Fournier-Viger Philippe, Raman Rajeev und S. Tseng Vincent. *CPT+: Decreasing the time/space complexity of the Compact Prediction Tree*. 2016.  
URL: [http://www.philippe-fournier-viger.com/PAKDD2015\\_sequence\\_prediction.pdf](http://www.philippe-fournier-viger.com/PAKDD2015_sequence_prediction.pdf) (besucht am 11.03.2016) (zitiert auf Seiten 62, 97).
- [Tor+05] B.U. Toreyin, Y. Dedeoglu und A.E. Cetin. "Wavelet based real-time smoke detection in video". In: *Signal Processing Conference, 2005 13th European*. 2005, S. 1–4.  
(Zitiert auf Seite 18).
- [ubu16a] ubuntuusers.de. *MySQL*. 2016.  
URL: <https://wiki.ubuntuusers.de/MySQL/> (besucht am 31.03.2016)  
(zitiert auf Seite 74).
- [ubu16b] ubuntuusers.de. *Ubuntu Installation*. 2016.  
URL: [https://wiki.ubuntuusers.de/Ubuntu\\_Installation/](https://wiki.ubuntuusers.de/Ubuntu_Installation/) (besucht am 30.03.2016) (zitiert auf Seite 74).
- [Uh15a] openHAB UG (haftungsbeschränkt). *Bindings*. 2015.  
URL: <https://github.com/openhab/openhab/wiki/Bindings> (besucht am 03.12.2015) (zitiert auf Seite 56).
- [Uh16] openHAB UG (haftungsbeschränkt). *Getting Started*. 2016.  
URL: <http://www.openhab.org/getting-started/> (besucht am 30.03.2016) (zitiert auf Seiten 74, 75).
- [Uh15b] openHAB UG (haftungsbeschränkt). *Persistenz - Introduction*. 2015.  
URL: <https://github.com/openhab/openhab/wiki/Persistence> (besucht am 30.11.2015) (zitiert auf Seiten 2, 55).
- [UNNG15] unn UNITED NEWS NETWORK GmbH. *Fraunhofer-Forscher zu SmartHome, Lifestyle und Industrie 4.0: Potenzial der Aktivitätserkennung wird noch lange nicht ausgeschöpft*. 2015.  
URL: <http://www.handelsblatt.com/technik/vernetzt/vernetztes-heim-schlaues-notfallssystem-macht-wohnen-sicherer/9632418.html>  
(besucht am 20.08.2015) (zitiert auf Seite 17).
- [Wai16] Machine Learning Group at the University of Waikato. *Class FPGrowth*. 2016.  
URL: <http://weka.sourceforge.net/doc.dev/weka/associations/FPGrowth.html> (besucht am 07.03.2016) (zitiert auf Seite 57).

- [Wai15a] Machine Learning Group at the University of Waikato. *Weka - Documentation*. 2015.  
URL: <http://www.cs.waikato.ac.nz/ml/weka/documentation.html> (besucht am 02. 12. 2015) (zitiert auf Seite 54).
- [Wai15b] WEKA The University of Waikato. *How do I use the package manager?* 2015.  
URL: <https://weka.wikispaces.com/How+do+I+use+the+package+manager%3F> (besucht am 02. 12. 2015) (zitiert auf Seite 54).
- [Was15] Steffen Wasmund. *Sozialgesetzbuch (SGB XII)*. 2015.  
URL: <http://www.sozialgesetzbuch-sgb.de/sgbxii/64.html> (besucht am 29. 09. 2015) (zitiert auf Seite 6).
- [Wel+13] Jan Wellmann, Anna Heindorf, Danilo Hollosi, Jens-E. Appell und Stefan Goetze. "MOBECS - Mobility by Safety: Konzept und Nutzeranforderungen". In: *Lebensqualität im Wandel von Demografie und Technik*. VDE-Verlag, 2013, S. 504–507. (Zitiert auf Seite 40).
- [Wic+14] Reiner Wichert und Helmut Klausning. *Ambient Assisted Living - 7. AAL-Kongress 2014 Berlin, Germany, January 21-22, 2014*. Berlin, Heidelberg: Springer, 2014. (Zitiert auf Seite 19).
- [Wil15] Arno Willig. *1% Regel*. 2015.  
URL: [http://fhemwiki.de/wiki/1%25\\_Regel](http://fhemwiki.de/wiki/1%25_Regel) (besucht am 02. 11. 2015) (zitiert auf Seite 20).
- [WS15] KEEL Ministerium für Bildung und Wissenschaft Spanien. *KEEL Included Algorithms*. 2015.  
URL: <http://sci2s.ugr.es/keel/algorithms.php> (besucht am 03. 12. 2015) (zitiert auf Seite 55).
- [Woj+15] Manfred Wojciechowski, Helmut Ristok, Wolfgang Brandes und Bernd Lange. *Architektur des 'Daily Care Journal' zur Unterstützung von Pflegenetzwerken*. 2015.  
URL: <https://www.vde-verlag.de/proceedings-de/453323013.html> (besucht am 07. 11. 2015) (zitiert auf Seite 27).
- [Wyb+15] Aline Wybranietz und Enrico Löhcke. *AAL in der Praxis eines sozialen Dienstleisters – Erfahrungsbericht aus dem SAMDY-Projekt*. 2015.  
URL: <https://www.vde-verlag.de/proceedings-de/453484091.html> (besucht am 09. 11. 2015) (zitiert auf Seite 28).
- [Zaa+15] S. Zaad und S. B. Allouch. *The Influence of Control on the Acceptance of Ambient Intelligence by Elderly People: An Explorative Study*. 2015.  
URL: [http://link.springer.com/chapter/10.1007%2F978-3-540-89617-3\\_5](http://link.springer.com/chapter/10.1007%2F978-3-540-89617-3_5) (besucht am 06. 10. 2015) (zitiert auf Seiten 6, 21).

- [Ziv+78] J. Ziv und A. Lempel. "Compression of individual sequences via variable-rate coding". In: *IEEE Transactions on Information Theory* 24.5 (1978), S. 530–536. (Zitiert auf Seite 37).
- [Zva+16] Stanislav Zvanovec, Pavel Pechac und Martin Klepal. *Wireless LAN Networks Design: Site Survey or Propagation Modeling?* 2016.  
URL: [http://www.radioeng.cz/fulltexts/2003/03\\_04\\_42\\_49.pdf](http://www.radioeng.cz/fulltexts/2003/03_04_42_49.pdf)  
(besucht am 13.04.2016) (zitiert auf Seite 100).

# Digitaler Anhang

Alle aufgezeichneten Daten, getätigte Auswertungen und der Quellcode der Implementierung sind in digitaler Form auf der beiliegenden CD wiederzufinden. Die nachstehende Abbildung zeigt die Ordnerstruktur der CD.





# Eidesstattliche Erklärung

Peter Manheller  
Kop Nück 8  
53902 Bad Münstereifel

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbst angefertigt habe; die aus fremden Quellen direkt oder indirekt übernommenen Gedanken sind als solche kenntlich gemacht.

Die Arbeit wurde bisher keiner Prüfungsbehörde vorgelegt und auch noch nicht veröffentlicht.

Bad Münstereifel, den 04.05.2016

---

(Ort, Datum, Unterschrift)