

**Eine offene Mobile Health Plattform zur Erfassung und
Verarbeitung von physiologischen Parametern in der
Präventivmedizin**

Dissertation

zur

Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

der Fakultät für Informatik und Elektrotechnik

der Universität Rostock



vorgelegt von

Dipl.-Inf. Weiping Zhang

geb. am 21.02.1982 in Shanghai

Rostock, April 2014

1. Gutachter: Prof. Dr. med. habil. Regina Stoll

Institut für Präventivmedizin, Universitätsklinikum Rostock

2. Gutachter: Prof. Dr.-Ing. habil. Kerstin Thurow

Institut für Automatisierungstechnik, Universität Rostock

3. Gutachter: Prof. Dr. Karen Frith

College of Nursing, University of Alabama in Huntsville

Verteidigungsdatum: 01.09.2014

Danksagung

Mein herzlicher Dank gilt Frau Prof.Dr.med.habil. Regina Stoll, die mir an ihrem Institut die Möglichkeit einer interdisziplinären Dissertation einräumte und die mir jederzeit mit Rat und Tat zur Seite gestanden hat. Ebenso möchte ich mich bei Frau Prof. Dr.-Ing. habil. Kerstin Thurow und Herrn Prof. Dr.-Ing. Norbert Stoll für die Vermittlung, die Themenvergabe und die hilfreiche Unterstützung, insbesondere in der Anfangs- und der Endphase meiner Arbeit, bedanken.

Weiterhin danke ich auch dem gesamten Team des Instituts für Präventivmedizin für die gewährte Unterstützung, wobei mein besonderer Dank an Frau Dr. rer. hum. Annika Rieger geht, die mich mit fachlichen Ratschlägen und anregenden Diskussionen tatkräftig unterstützte. Ebenfalls ein besonderer Dank gilt Herrn Dr.-Ing. Sebastian Neubert, Herrn Dipl.-Ing. Peter Passow und Herrn Dr. rer. hum. Matthias Weippert, die bei fachlichen Fragen des Öfteren ihre Erfahrungen mit mir geteilt haben und damit maßgeblich zum Gelingen dieser Arbeit beitrugen.

Ganz besonders möchte ich an dieser Stelle auch meiner Familie danken, die über die Jahre des Studiums und der Promotion so viel Vertrauen in mich gesetzt haben und mich mit großer Begeisterung unterstützten.

Mein größter Dank jedoch gilt meiner Frau Jingzhi, die mich während der gesamten Promotionszeit motiviert und unterstützt haben und mir in so mancher schwierigen Phase den Rücken freigehalten haben.

Inhaltsverzeichnis

Inhaltsverzeichnis	2
Abbildungsverzeichnis	7
Tabellenverzeichnis	9
Abkürzungsverzeichnis.....	10
1 Einleitung.....	12
1.1 Zielstellung.....	12
1.2 Technologien und Geschichte der Telemedizin	16
1.3 Mobile Health Systeme - Stand der Forschung und Technik	18
1.3.1 Tragbare Sensortechnologie.....	19
1.3.2 Drahtlose Sensornetzwerke	21
1.3.3 Kommunikationstechnologie.....	22
1.3.4 Mobiles Computing	24
1.3.5 Standardisierung	25
1.4 Standards für den Datenaustausch im Gesundheitswesen	26
1.4.1 Interoperabilität von Telemedizin-Systemen.....	27
1.5 Gliederung der Arbeit	28
2 Drahtloses Sensornetzwerk und medizinische Versorgung	31
2.1 Definition eines medizinischen drahtlosen Sensornetzwerksystems	31
2.2 Architektur des medizinischen drahtlosen Sensornetzwerksystems.....	32
2.2.1 Logische Struktur des Systems.....	32
2.2.2 Layout des Systems.....	34
2.3 Die Schlüsseltechnologien des medizinischen drahtlosen Sensornetzes	36
2.3.1 Energieverbrauch in drahtlosen Sensornetzen	37
2.3.2 Kommunikationslayout des drahtlosen Sensornetzes	39
2.3.3 Drahtlose Kommunikationstechnik für den Nahbereich.....	42
2.3.3.1 Infrarot-Technologie.....	43

2.3.3.2 Bluetooth.....	43
2.3.3.3 ZigBee	44
2.3.3.4 WiFi.....	44
2.3.3.5 ANT	44
2.3.3.6 Vergleich und Analyse	45
2.4 Hauptanwendungsbereiche von drahtlosen Sensornetzen in medizinischen Versorgungssystemen	47
3 Anforderungsanalyse	52
3.1 Business-Subsystem.....	52
3.1.1 Mobiler Erfassungsclient	52
3.1.2 Fernüberwachungszentrale.....	53
3.1.3 IPM-Auswertungsmodule	53
3.2 Integration der Subsysteme	54
3.2.1 Nahbereich-Kommunikation des mHealth-Clients	54
3.2.2 Fernmeldeverbindung.....	55
3.2.3 Datenschnittstelle.....	56
3.3 Entwurf der Schnittstellenfunktionen zwischen den Systemmodulen	56
3.3.1 Schnittstelle zwischen Patient und Monitoring Center.....	57
3.3.2 Schnittstelle zwischen mHealth Client und mHealth Server	58
3.3.3 Schnittstelle zwischen Healthcare Monitoring-Zentrum und den Auswertungsmodulen	59
4 Entwurf und Umsetzung des mobilen Erfassungsclients	61
4.1 Android-Plattform	61
4.1.1 Android-Entwicklungstechnologie.....	62
4.2 Entwurf und Umsetzung des Erfassungsclients.....	63
4.2.1 Sensorsystem.....	63
4.2.2 Wesentliche Module des Erfassungsclients.....	65
4.3 Sensor Kommunikation	66
4.3.1 Drahtlose Verbindung von IPM-MC über Bluetooth.....	66

4.3.2 Verbindung von IPM-MC über ANT	69
4.3.3 Echtzeit Datenerfassungsprozess	70
4.3.4 Echtzeit Analyse der Sensordaten	72
4.4 Datenvisualisierung und Aktualisierung	77
4.5 Netzwirkommunikationsmodul	79
4.6 Management-Modul für Untersuchungen	87
5 Einführung in SOTH: SOA-basiertes offenes Telemonitoring framework für Healthcare-Anwendungen	89
5.1 SOTH : SOA-basiertes offenes Telemonitoring framework für Healthcare.....	90
5.2 Granularität	91
5.2.1. Business Collaboration Management Model (BCM)	92
5.2.2. Business-Model (BM).....	93
5.2.3. Control Service Model (CM).....	93
5.2.4. Utility Service Models	93
5.3 Beziehungen zwischen den Modellen	94
5.3.1 Zusammenfassende Definition der Architektur.....	95
5.4 Beschreibung der medizinischen Ressourcen und Dienste.....	96
5.4.1 Beziehungen des Systemkonzepts	97
5.4.2 Beschreibung der Ressourcen.....	98
5.4.3 Beschreibung von Diensten	100
6 Konzeption und Implementierung des mHealth Zentral-Server-Systems	104
6.1 Ziele des Systementwurfs.....	104
6.2 Healthcare-Service-Bus.....	106
6.2.1 Gesamtstruktur des HSB.....	108
6.2.2 Grundlegende Konzepte	109
6.3 Entwurf und Umsetzung des mHealth Server-Moduls.....	111
6.3.1 Kommunikationsserver.....	112
6.3.2 Benutzerverwaltung	116

6.3.3 Live Visualisierungsmodul	119
6.3.4 Studien-Management.....	121
6.3.5 Datenmanagement	123
6.3.5.1 Struktur der Datenbank.....	123
6.4 Implementierung der Services.....	125
6.4.1 HSB Service-Integration.....	125
6.4.2Anwendungsszenarien.....	126
7 Entwurf der automatisierten Aktivitätserkennung.....	133
7.1 Vorstellung des Equivital Eq01 Multiparameter-Sensors.....	133
7.1.1 ADXL330 3-Axis Accelerometer	134
7.1.2Dreiachsige Beschleunigungssensoren.....	135
7.2 Entwurf des Bewegungserkennungsalgorithmus.....	137
7.2.1 Erfassung der Beschleunigungssensordaten	137
7.2.2 Glättung der Beschleunigungssensordaten.....	138
7.2.3 Extraktion der Merkmale der Beschleunigungssensordaten.....	139
7.2.4 Analyse des Klassifikationslernprozesses	140
7.2.5 Analyse des Bewegungserkennungsprozesses	144
8Evaluierung und Anwendung	146
8.1 Aufbau und Durchführung der Untersuchung	146
8.1.1 Untersuchung der Auslastung desIPM Servers.....	146
8.1.2 Ergebnisse der Untersuchung.....	148
8.1.3 Echtzeit-Datenübertragungstest	150
8.1.4 Stabilitätstest.....	150
8.1.5 Leistungsaufnahme des Smartphones.....	151
8.2 Anwendung in der Praxis.....	151
8.2.1 Untersuchung physiologischer Parameter während Langzeitfelduntersuchungen.....	152
8.2.2 Beanspruchung von Chemikern, technischen Assistenten und Biologen des Center for Life Science Automation (Celisca)	154

9 Zusammenfassung und Ausblick	156
9.1 Zusammenfassung	156
9.2 Ausblick	157
Literaturverzeichnis	160
Anhangsverzeichnis	173
Kurzreferat / Abstract	180
Thesen.....	182

Abbildungsverzeichnis

Abbildung 1-1 Architektur des IPM-mHealth Systems.....	12
Abbildung 1-2 Telemonitoring-System.....	17
Abbildung 1-3 Bild vom Radio doctor	18
Abbildung 1-4 Body Media System [20].....	20
Abbildung 1-5 Beziehung zwischen Health Care WSN und Kommunikationsgeräten	23
Abbildung 2-1 Logische Struktur des medizinischen Sensornetzwerksystems.....	33
Abbildung 2-2 Layout des medizinischen Sensornetzwerksystems	34
Abbildung 2-3 Layer-basierte Struktur des medizinischen Sensornetzwerksystems.....	35
Abbildung 2-4 Überwachung der physiologischer Daten in Echtzeit [59].....	48
Abbildung 2-5 Anwendungsszenarien der drahtlosen Sensornetze in medizinischen Versorgungssystemen[61].....	49
Abbildung 3-1 Systemarchitektur des IPM-mHealth Systems.....	52
Abbildung 3-2 Lokales Netzwerk-Gateway	55
Abbildung 3-3 Fernverbindung zwischen Patientengeräten und dem Monitoring-Zentrum	56
Abbildung 4-1 Architektur des Android-Systems [71].....	62
Abbildung 4-2 Struktur und Arbeitsprozesse der Anwendungsprogramme in Android.....	63
Abbildung 4-3 Kernkomponenten vom IPM-MC.....	66
Abbildung 4-4 Bluetooth und Ant Transceiver [84] [85]	66
Abbildung 4-5 IPM-MC Bluetooth Device Discovery-Prozess	68
Abbildung 4-6 Etablieren eines ANT-Kommunikationskanals zwischen Master- und Slave-Knoten	69
Abbildung 4-7 Ablauf des Echtzeit-Sensordatenerfassungsprozesses.....	70
Abbildung 4-8 Synchronisation von Study information zwischen IPM-MC und IPM Server	71
Abbildung 4-9 Ablauf der Sensordatenanalyse	73
Abbildung 4-10 Datenübertragungsprotokoll für den Mutisensor EQ01 (Hidalgo, Coimbridge UK) [97]	74
Abbildung 4-11 Datenübertragungsprotokoll für den Beschleunigungssensor (Wahoo Fitness) [98] ..	74
Abbildung 4-12 UML-Klasse-Diagramm der Schnittstellenorientierten Sensordatenanalyse	77
Abbildung 4-13 EKG-Diagramm vom IPM-MC	79
Abbildung 4-14 Übermittlung der physiologischen Daten vom IPM-MC an den IPM-Server.....	80
Abbildung 4-15 Arbeitsprozess des TCP-Socket Fernübertragungsthreads.....	82
Abbildung 4-16 Kontrollbenachrichtigungen des IPM-Servers	86
Abbildung 4-17 Formate der Datenpakete für Verwaltungs- und Kontrollbenachrichtigungen	86
Abbildung 4-18 Vorgang der Synchronisierung von Untersuchungsinformationen zwischen IPM-MC und IPM-ZS	87
Abbildung 4-19 IPM-MC-Fragebogen.....	88
Abbildung 5-1 Hierarchische Domainmodelle, die Brücke zwischen Anforderungen und den Diensten	89
Abbildung 5-2 SOTH-Architektur.....	91
Abbildung 5-3 Kommunikation zwischen Diensten des gleichen Layers	95
Abbildung 5-4 Konzept der SOTH-Architektur	97
Abbildung 5-5 Konzept des Ressourcenmodelles	98
Abbildung 5-6 Beziehung zwischen Konzept und Diensten	101
Abbildung 6-1 Mule- und CXF- basierte Healthcare Service Bus (HSB).....	107
Abbildung 6-2 Gesamtstruktur des HSB.....	107

Abbildung 6-3 Ablauf der Message Bearbeitung im HSB	108
Abbildung 6-4 Model.....	109
Abbildung 6-5 Service.....	109
Abbildung 6-6 Transport	110
Abbildung 6-7 Transformer	110
Abbildung 6-8 Router	110
Abbildung 6-9 Component	111
Abbildung 6-10 Kernkomponenten vom IPM-ZS.....	111
Abbildung 6-11 Bearbeitung der physiologischen Parameter durch die Kommunikationsserver	112
Abbildung 6-12 Klassediagramm der Identifikations-Service	117
Abbildung 6-13 Zustandsdiagramm der Benutzeranmeldung	118
Abbildung 6-14 Verlauf der verschiedenen Funktionen im EKG-Zeichenprozess.....	119
Abbildung 6-15 Diagramm der erfassten physiologischen Parameter.....	122
Abbildung 6-16 Hauptbedienungsfläche des Studienmanagers	122
Abbildung 6-17 Speichervorgang für die Funktion: Add_New_Patient	123
Abbildung 6-18 Tabellenstruktur in der IPM-ZS Datenbank	125
Abbildung 6-19 Messszenario im IPM-mHealth System	127
Abbildung 6-20 HSB Transformer-Modell	128
Abbildung 7-1 Hidalgo Equivital Muti-parameter Sensor [146].....	133
Abbildung 7-2 Equivital Sensor Brustgurt [148].....	134
Abbildung 7-3 ADXL330 3-Axis Accelerometer [149].....	135
Abbildung 7-4 Direction of the EQ01 3-Axis Accelerometer.....	136
Abbildung 7-5 Verlauf der Bewegungserkennung	136
Abbildung 7-6 Aufrufprozess der Beschleunigungssensordaten	137
Abbildung 7-7 Zeitreihenanalyse der Beschleunigungssensordaten	138
Abbildung 7-8 Prozess der Bewegungserkennung.....	144
Abbildung 8-1 Testumgebung und Versuchsgeräte.....	146
Abbildung 8-2 Prozessorzeiten des gesamten Serversystems bei verschiedenen Belastungen (keine fortlaufende Datenbankabfrage)	148
Abbildung 8-3 Prozessorzeiten des gesamten Serversystems bei verschiedenen Belastungen (mit fortlaufender Datenbankabfrage).....	148
Abbildung 8-4 Prozessorzeiten des Kommunikationsservers	149
Abbildung 8-5 Verhältnis zwischen Datenrate und Anzahl tranferierender Clients für maximale Auslastung des Serversystems	150
Abbildung 8-6 Tabellarische Darstellung der physiologischen Daten.....	150
Abbildung 8-7 Häufigkeitsverteilung der Antworten auf die Frage: Waren die vergangenen 24 Stunden repräsentativ für Sie?.....	153
Abbildung 8-8 Online-Visualisierung mit den Beispieldaten eines Probanden im Verlauf eines Arbeitstages mit den richtigen Belastungskategorien	154
Abbildung 8-9Einfluss unterschiedlicher Tätigkeiten (sitzend und stehend) auf die Atemfrequenz..	155

Tabellenverzeichnis

Tabelle 1-1 Typische Sensoren verwendet in WSN-basierten Health Care systems.....	21
Tabelle 1-2 Eigenschaften der Wireless Technologien.....	23
Tabelle 1-3 Standard: Healthcare WANS.....	25
Tabelle 2-1 Vergleich der verschiedenen Sensor-Netzwerk-Plattformen.....	38
Tabelle 2-2 Energieverbrauch der Sensorknoten.....	39
Tabelle 2-3 Vergleich der Nahbereich-Kommunikationstechnik	45
Tabelle 4-1 Sensoren List.....	64
Tabelle 4-2 Formates für die Datenpakete der physiologischen Parameter	84
Tabelle 5-1 Attribute der Beschreibung der Ressourcen	99
Tabelle 5-2 Beschreibung der Service-Attribute	102
Tabelle 5-3 Definition der Service-Attribute	102
Tabelle 6-1 Rollen im IPM-ZS.....	118
Tabelle 6-2 Tabelle Measurement	125
Tabelle 6-3 IPM-Server-Umgebung	127
Tabelle 7-1 Confusion Matrix der Nächster-Nachbar-Klassifikation (IB1).....	141
Tabelle 7-2 Confusion Matrix der Bayes-Klassifikation	142
Tabelle 7-3 Confusion Matrix der Entscheidungsbaum-Klassifikationsalgorithmus.....	142
Tabelle 7-4 Einfluss verschiedener Merkmale auf die Einteilung der Daten	143
Tabelle 7-5 Testergebnisse.....	145
Tabelle 8-1 Versuchsergebnisse	151
Tabelle 8-2 Durchgeführt Study List.....	151
Tabelle 8-3 Mittelwerte der normalisierten HRV-Parameter (Liegen = 1; Schlafen = 2)	153

Abkürzungsverzeichnis

IPM	Institut für Präventivmedizin
IPM-mHealth	Institut für Präventivmedizin -Mobile Health System
IPM-MC	Mobile Client des IPM-mHealth System
IPM-ZS	Server System des IPM-mHealth System
API	Application Programming Interface
BAN	Body Area Network
BSC	Base Station Controller
BSS	Base Radio Sub-System
BTS	Base Transceiver Stations
EKG / ECG	Elektrokardiogramm
EEG	Elektroenzephalografie
EMG	Elektromyografie
FIFO	First-In First-Out
GSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSM	Groupe Spécial Mobile
GUID	Globally Unique Identifier
HIMSS	Healthcare Information and Management Systems Society
HL7	HealthLevel 7
HSDPA	High Speed Downlink Packet Access
HSPA	High Speed Packet Access
HSUPA	High Speed Uplink Packet Access
HSB	Healthcare Service Bus
ICNIRP	International Commission on Non-Ionising Radiation Protection
IEEE	Institute of Electrical and Electronics Engineers
IPM	Institut für Präventivmedizin
ISM-Band	Industrial, Scientific, and Medical Band
MSC	Mobil Switch Center
PDA	Personal Digital Assistant
PIN	Persönliche Identifikationsnummer
PKG / PCG	Phonokardiogramm
PCU	Packet Control Unit
RNC	Radio Network Controller

RSNARadiological Society of North America
SARSpezifische Absorptionsrate
SDKSoftware Developers Kit
SEMSensor Electronics Module
SGSNServing GPRS Support Node
SPPSerial Port Profile
TCPTransmission Control Protocol
UDPUser Datagram Protocol
UMTSUniversal Mobile Telecommunications System
UTRANUMTS Terrestrial Radio Network
WISEWireless Intelligent Sensor
WLANWireless Local Area Network

1 Einleitung

1.1 Zielstellung

Die heutige Lebenswelt in den hochentwickelten Industrieländern ist zunehmend durch einen akuten Bewegungsmangel eines großen Teils der Bevölkerung gekennzeichnet. Insbesondere die Beurteilung der Belastbarkeit von Risikopatienten mit einer Herz-Kreislauf-Erkrankung im Alltag ist bisher kaum oder nur sehr eingeschränkt möglich. Im Bereich der Präventivmedizin basiert ein großer Teil der Forschung auf den Untersuchungen, die für individuelle Belastungs-Beanspruchungs-Untersuchungen ausschließlich im Feld durchgeführt werden müssen. Mit dem Projekt „Mobile Health“ vom Institut für Präventivmedizin Rostock wird in dieser Arbeit die technische Voraussetzung: IPM-mHealth-System (siehe Abbildung 1) geschaffen, die Belastung sowohl stationär als auch im häuslichen Umfeld dynamisch zu ermitteln und diese zeitsynchron zu den physiologischen Parametern aufzunehmen. Der Mediziner bekommt damit neue Ansätze zur Verbesserung seiner Diagnose und Empfehlungen hinsichtlich der anzustrebenden körperlichen Belastung. Auch die Sicherheit der Patienten erhöht sich, da kritische Zustände bei bestimmten Bewegungsabläufen (z.B. Treppensteigen) frühzeitig erkannt werden. Letztlich bietet dieses System die automatisierte Aktivitätsanalyse im Rahmen von Langzeituntersuchungen einer vielversprechenden Möglichkeit Patienten beim Erlernen eines optimalen Bewegungsprofils zu unterstützen [1].

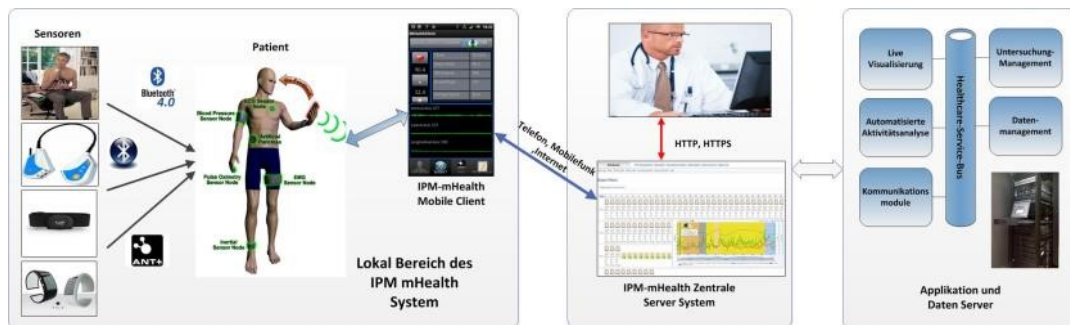


Abbildung 1-1 Architektur des IPM-mHealth Systems

Die auf tragbaren Sensoren und intelligentem mobilen Endgerät basierende medizinische Fernüberwachung ist eine moderne Art des telemedizinischen Systems. Es bedient sich der medizinischen Sensoren als Gesundheits-Datenerfassungsschnittstelle und sendet die erfassten physiologischen Daten über ein drahtloses Nahbereich-Kommunikationsnetzwerk an den lokalen Gateway (mobiles Gerät). Erst dann werden die Daten an das Datenerfassungszentrum des Telemonitoring-Systems gesendet. Im medizinischen Zentrum werden die Daten analysiert, Diagnosen erstellt und entsprechende Dienste bzw. Gegenmaßnahmen angeboten. Auf diese Weise wird das Modell des „Remote Health Care“ [2]

realisiert. Das auf leichten, tragbaren Sensoren [3] basierende System ermöglicht den Patienten eine relativ große Bewegungsfreiheit. So können die Patienten ohne zeitliche und örtliche Einschränkungen jederzeit auf die Betreuung des medizinischen Personals zurückgreifen und sich im Notfall auf notwendige Rettungsmaßnahmen verlassen. Die neue, auf Sensoren und mobiler Kommunikation basierende Lösung ist eine Verbesserung der veralteten Kabel-basierten medizinischen Erfassungssysteme und stellt für die Telemedizin eine vielversprechende, zukunftsweisende Entwicklung dar [4].

Andererseits, trotz einer zehnjährigen Entwicklung, befindet sich das medizinische Telemonitoring aber noch in vielerlei Hinsicht auf Forschungsniveau [5]. Bis heute wurde die Anwendung – wenn auch in deutlich zunehmendem Maße - nicht in großer Breite verfolgt. Durch die Analyse der aktuellen Forschungsarbeit zeigt sich eindeutig: Neben den sozialen und wirtschaftlichen Faktoren steht das Mobile-Health-System derzeit vor den folgenden drei großen technischen Herausforderungen.

1. Dynamik. Auf dem Markt gibt es eine breite Palette von medizinischen Sensoren zur Messung von Daten, aber es mangelt an einheitlichen Standards. Momentan sind die meisten Telemonitoring-Systeme für spezielle Geräte ausgelegt. Mit anderen Worten, bei der Entwicklung eines Systems muss der Code für jedes Gerät geschrieben und es müssen an die Geräte angepasste Operations-Workflows erarbeitet werden, sodass die Skalierbarkeit des Systems erheblich eingeschränkt wird. Dies ist auch einer der wichtigen Gründe dafür, warum Telemonitoring noch nicht hinreichend weit verbreitet ist [6].

2. Heterogenität. Aufgrund des Unterschieds der Struktur der Softwaresysteme von Telemonitoring-Systemen und den herkömmlichen medizinischen Systemen wie Krankenhausinformationssystem (KIS) und Laborinformationssystem (LIS), kommt es zu Schwierigkeiten beim Austausch von Daten und Bereitstellung von medizinischen Diensten zwischen dem mHealth- und anderen medizinische-Anwendungen. Es entstehen sogenannte „Informationssilos“ [7]. Dieser Aspekt schränkt die Verbreitung von Telemonitoring ebenfalls ein.

3. „Big Data“-Massendatenspeicherung. Anders als bei der Elektronischen Gesundheitsakte (EHR) müssen mHealth-Systeme in der Lage sein, das 10-fache und sogar 100-fache an Vitaldaten zu speichern. Wie z.B. bei einem 2-kanaligen EEG (256HZ) das über 24 Stunden ausgezeichnet wird, fallen für einen Patienten mehrere Millionen Daten ($256 * 2 * 60 * 60 *$

24) an, und diese Daten sind oft dynamisch und semistrukturiert. Dies beinhaltet auch eine sehr hohe Anforderung an die technische Auslegung der Monitoring-Server-Systeme.

In dieser Arbeit wurde ein Mobile Health (mHealth) System entwickelt, das konsequent auf tragbaren Sensoren basiert. Um die oben genannten Probleme zu lösen, wurde im Rahmen dieses Systems ein SOA (Service Oriented Architecture) basiertes offenes Telemonitoring-Framework (SOTH) konzipiert und eingesetzt. SOTH verwendet den SOA-Ansatz zur Erzeugung eines strukturellen Rahmens für Ingenieure und Softwareentwickler, um ihnen Hilfestellungen bei der Entwicklung von Komponenten einer Mobile-Health-Plattform zu geben. In dieser Plattform werden die medizinischen Dienstleistungen als das Ergebnis der Kombination von verschiedenen abstrakten Ebenen betrachtet, was zur losen Kopplung der medizinischen Leistungen und zur Verbesserung der Interoperabilität zwischen den Software-Dienst-Modulen führt.

In den mobile clients (MC) wurde eine Middleware konzipiert, um verschiedene Nahbereich-Kommunikationstechnologien zu unterstützen. Die Middleware erfasst die aktuelle interne und externe Rechnerumgebung des Geräts und gibt diese Informationen an den oberen Anwendungs-Layer über eine einheitliche Schnittstelle weiter. Auf diese Weise wurde auf der Hardware-Ebene das Ziel der flexiblen Anbindung mit heterogener Sensorik erreicht.

Für die Auswertungsmodule wurde in dieser Arbeit eine automatisierte Aktivitätsanalyse-methode entwickelt. Ziel dieser Komponente war die Auswertung von telemedizinisch erhobenen Vital-Daten in Verbindung mit einer Bewegungsklassifizierung. Diese bietet beispielhaft qualitativ neue Möglichkeiten für Langzeituntersuchungen verschiedener Vitalfunktionen. Aktivitätsprofile können einerseits eigenständig als Grundlage diagnostischer Aussagen dienen, andererseits in Kombination mit anderen Messgrößen deren Aussagekraft maßgeblich erweitern.

Die wichtigsten Innovationen und Forschungsergebnisse dieser Arbeit werden aus den drei relevanten Aspekten, nämlich den drei Teilen des mHealth-Systems im Folgenden zusammenfasst: (1) lokale Seite des mHealth Systems: Sensorsystem und Android-basierte Mobile Clients (*IPM-MC*), (2) Server-Seite: Zentrales Server System (*IPM-ZS*) und (3) Auswertungsmodule: Automatisierte Aktivitätsanalyse (*IPM-AM*) – beispielhaft -

1. Diese Arbeit untersucht nun als erstes den Aufbau und die Struktur von drahtlosen Sensor-Netzwerken [8], die der medizinischen Versorgung bzw. dem Monitoring dienen. Anhand der Merkmale der medizinischen Versorgung wird eine geeignete Definition des

medizinischen Sensor-Netzwerks herausgearbeitet. Die Bereiche des Sensor-Netzwerks für die medizinische Versorgung, die mit Problemen behaftet sind, werden näher untersucht. So gibt es Defizite in den Bereichen Energieverbrauch, Routing Topologie Kontrolle und Positionierung, die die Entwicklung der drahtlosen Sensor-Netzwerke hemmen. Die vorhandenen Sensor-Netzwerke, die sich überwiegend auf den Nahbereich der Telekommunikation beschränken, werden verglichen. Eine neue Nahbereichs-Kommunikationstechnologie ANT wird eingeführt. Das in dieser Arbeit vorgestellte mHealth-System verfügt sowohl über die ANT- als auch über die Bluetooth-Technologien und kann gleichzeitig auf diese zurückgreifen und flexibel zwischen ihnen wechseln.

2. Um die einfache Integration mit verschiedenen Arten von Sensoren zu realisieren, wurde in dieser Arbeit bei der Implementierung des Android basierten mobilen Clienten (IPM-MC) eine Middleware umgesetzt, deren echtzeitfähige Kommunikationsmodule verschiedene drahtlose Kommunikationstechnologien unterstützen, wie Bluetooth, ANT, ANT+ und Bluetooth Low Energie. Bei der Sensor-Daten-Analyse wird ein schnittstellen orientierter Ansatz verwendet. Durch die Zusammenfassung und ausführliche Analyse der verschiedenen Sensor-Kommunikationsprotokolle wurde eine Universal-Schnittstelle definiert, die den allgemeinen Vorgang des Analyseablaufs enthält. Im Wesentlichen wird der obere Layer (Display Layer) des IPM-MC und der untere Layer (Datenverarbeitung Layer) über diese Schnittstellen lose gekoppelt. Hierdurch wird ein flexibler Austausch der Sensorgeräte ermöglicht und das Ziel der Erweiterbarkeit vom mHealth System erreicht.
3. In der Implementierungsphase des mHealth-Zentral-Systems (IPM-ZS) wurde der Healthcare-Service-Bus (HSB) verwendet. HSB bringt durch die Verknüpfung verschiedener Dienste und die Zusammenlegung von Systemen, Daten, Anwendungen und Prozessen ein flexibleres Design mit sich. Darüber hinaus nutzt diese Arbeit eine Kombination aus Softwaretechnik und Untersuchungsumgebung vom Institut für Präventivmedizin der Universität Rostock als Grundlage, um eine telemedizinische Untersuchungsplattform mit hohem Verallgemeinerungsgrad in der arbeits- und präventivmedizinischen Forschung zu entwerfen.
4. Bei der Datenauswertung wird beispielhaft ein automatisierter Aktivitätsanalyse-Algorithmus auf der Grundlage von Beschleunigungssensoren entwickelt. Aktivitätsprofile können einerseits eigenständig als Grundlage diagnostischer Aussagen dienen, andererseits können sie in Kombination mit anderen Messgrößen deren

Aussagekraft maßgeblich erweitern. Bewegungserkennung kann als Klassifizierungsproblem des maschinellen Lernens aufgefasst werden. Die „Nearest-Neighbor-Classification“ wird verwendet, um die Messdaten unter den verschiedenen Zuständen zu klassifizieren und die Bewegungserkennung des Patienten zu vervollständigen und effektiv zu interpretieren.

1.2 Technologien und Geschichte der Telemedizin

In diesem Abschnitt werden, basierend auf den vielen nützlichen Arbeiten von Wissenschaftlern und Experten im Bereich medizinisches Telemonitoring in den letzten Jahren, die mittels der mit Mobile Health Systemen verbundenen wichtigen Techniken sowie tragbare Technologie, drahtlose Sensornetzwerke, Kommunikationstechnologien, mobiles Computing, Netzwerk-Storage-Technologie (NST), Standardisierung untersucht. Das Ziel ist, alle technischen Komponenten des mobilen Monitoringsystems umfassend zu analysieren.

Mit dem technischen Fortschritt im Industrialisierungsprozess steht die menschliche Gesundheit vor immer mehr Herausforderungen. Auf der einen Seite nehmen mit der Erweiterung der menschlichen Kommunikationsmöglichkeiten die Einflüsse und das Ausmaß von verschiedenen Krankheiten zu. Beispielsweise haben die neu entstandenen Erkrankungen wie Vogelgrippe, SARS und das Influenza-A-Virus H1N1 einen schockierenden Einfluss auf die Menschheit ausgeübt. Um gegen sie besser vorgehen zu können, müssen bessere Monitoringeinrichtungen gegen den Ausbruch von Krankheiten aufgebaut werden. Auf der anderen Seite ist es bei einer alternden Bevölkerung notwendig, das Gesundheitsmonitoring und Pflegesystem für das Alter, besonders für alte Menschen mit chronischen Krankheiten, zu implementieren und zu vervollkommen. Nicht nur in Europa, sondern auch in Asien oder in Nordamerika können bei der gegebenen Anzahl professioneller Pfleger und des Versorgungsgebiets die Herausforderungen, die aus der Alterung der Menschen entstehen, mit herkömmlichen Betreuungsmethoden nicht einfach überwunden werden [9]. Es muss darüber nachgedacht werden, ob mittels bestimmter Techniken ein effektives, verlässliches und preiswertes Gesundheitsmonitoring- und Pflegesystem zu schaffen ist, um ein Echtzeit-Monitoring der Patienten zu erreichen. Seit vielen Jahren sind zahlreiche Experten damit beschäftigt [10].

Bereits in den 1970er Jahren wurde das „RemoteHealthcare System“, das aus einer Kombination von Kommunikationstechnologien und Medizintechnik bestand, als eine der

Zukunftstechnologien betrachtet [11]. Diese Technik wurde Telemedizin genannt. Es wird allgemein angenommen, dass Telemedizin bedeutet, dass mittels der modernen Informationstechnologie bzw. bidirektionaler Audio/Video-Kommunikation, Computer und „Remote-Presence-Systemen“ die Fernpflege für Patienten, die Plattform für die Übertragung von Ferndaten und der Austausch von Informationen zwischen Patienten, niedergelassenen Ärzten und medizinischen Experten möglich sind. Abbildung 1-2 stellt anschaulich das Telemonitoring System dar.

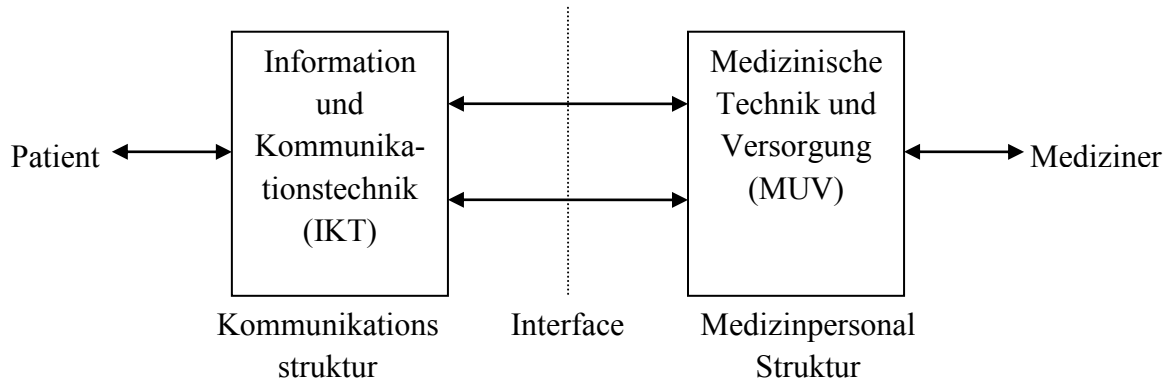


Abbildung 1-2 Telemonitoring-System

In der Tat sind die Vorstellungen der Anwendungen der Informations- und Kommunikationstechnologie auf das medizinische Versorgungssystem bis in das Jahr 1924 zurückzuverfolgen. In Abbildung 1-3 ist die Vorderseite des <<Radio News>>Magazins dargestellt. Zu dieser Zeit hatten die Wirtschaftler die Idee, mittels der Verwendung des Fernsehens und Radios Telemedizin zu erreichen. Diese Technik wurde „Radio Doctor“ genannt. In den 1970er Jahren haben die US National Aeronautics and Space Administration NASA eine Vielzahl von Experimenten durchgeführt, um die „Tele-Diagnose“ per Satellit zu realisieren. Seit dem Jahr 1995 findet mit der Verbreitung des Internets und der Entwicklung von drahtlosen Kommunikationstechnologien sowie der Ausweitung des Telekommunikationsnetzes die TV-netzwerkbasierte Telemedizin immer mehr Anwendungen in der Praxis [12].

Trotz der langjährigen Entwicklung konnte aufgrund des Engpasses der Entwicklung der drahtlosen Übertragungstechnik und Sensortechnik das Telemonitoringsein großes Potenzial noch nicht vollständig entfalten. Im 21. Jahrhundert werden mit dem Fortschritt der medizinischen Sensortechnik, der Entstehung des Smartphones, der Entwicklung der drahtlosen Kommunikationstechnologie, der Vergrößerung der Bandbreite und der Entwicklung von Cloud-Computing und verteilten Systemen die neuen Techniken in den Bereichen mobiles Computing oder Cloud-Speicherung im Gesundheitswesen eingesetzt. In

der medizinischen Technik entwickelt sich ein Forschungsfokus im Bereich Telemonitoring [13].

1.3 Mobile Health Systeme - Stand der Forschung und Technik

Die Entwicklung des Telemonitorings ist derzeit noch unausgewogen. Das Entwicklungsniveau in den USA und Europa ist viel höher als in anderen Ländern. Schon sehr früh haben sich europäische Länder mit diesem Bereich beschäftigt und viel in die Entwicklung dieser Technik investiert. Daher haben diese Länder bereits eine ausgereifte unterstützende Technologie im Telemedizin-System erreicht. Die Hauptanwendungsbereiche umfassen derzeit noch Telemonitoring, Telekonsultation, Tele-Therapie und Notdienst im Krieg. Mehrere Subsysteme der Telemedizin weisen ein hohes Entwicklungsniveau auf [14]. Beispielsweise sind das Krankenhausinformationssystem (KIS), die elektronische Patientenakte und das Picture Archiving and Communication System (PACS) sehr ausgereift.



Abbildung 1-3 Bild vom Radio doctor

In Luxemburg wird das Satellitennetzwerk für das medizinische e-learning verwendet [15]. Durch Satellitenkommunikation, die ein effektives Kommunikationsmittel ist, kann ein großes Gebiet abgedeckt werden. Durch diese Technik werden viele entlegene Gebiete erreicht, was mit konventionellen Kommunikationsmitteln nicht einfach möglich ist. Der Vorteil ist ein schneller Datentransfer, der es ermöglicht, Telemedizin z.B. im fliegenden Flugzeug oder auch dem fahrenden Schiff zu verwirklichen. Der größte Nachteil sind die hohen Kosten, daher ist die Verbreitung der Technik sehr begrenzt. Aus geographischen Gründen verwendet man in Japan oft Satellitenkommunikation, um Experimente der Telemedizin durchzuführen [16]. In

manchen Gebieten wurden hier gute Effekte erreicht. Die USA haben keine vergleichbaren Vorteile bei der Satellitentechnologie. Satellitenkommunikation begünstigt die große Region Alaska sehr, die am weitesten vom Festland der Vereinigten Staaten entfernt liegt [17]. Diese Region ist sehr dünn besiedelt und hat einen großen Mangel an medizinischen Ressourcen. Jedoch wurde mittels Telemedizin das medizinische Niveau der Region stark verbessert. Durch die Entwicklung eines GSM-basierten Telemonitoring-Systems können z.B. physiologische Parameter erfasst werden, wie u.a. EKG, Blutdruck, Sauerstoffsättigung, Herzfrequenz und Körpertemperatur. Die Erfassungsfrequenz des EKGs beträgt 200Hz. Zudem können statische Bilder übertragen und das Transfer-Protokoll durch TCP/IP realisiert werden. Durch den JPEG-Algorithmus werden die Bilder komprimiert. Die meisten europäischen Länder nutzen die mobile Kommunikationstechnik als Plattform für die Telemedizin. Der Grund liegt in ihren gut entwickelten mobilen Kommunikationsnetzwerken, die an der Spitze liegen, damit Telemedizin realisiert werden kann.

Die EU hat drei im Bereich Telemonitoring leistungsfähige Labore für biomedizinische Technik, zehn große Unternehmen und 20 Pathologielabore und andere Endkunden motiviert [18], an den umfassenden Experimenten zur Einführung eines Monitoring-Systems teilzunehmen. Im Hinblick auf Telemonitoring hat Europa eine hohe Leistung erreicht. Die EU hat eine große Menge von Projekten finanziert und organisiert. Eines davon ist das Ambulance and Emergency-112.

Das US-Georgia Education Medical System [19](CSAMS) ist das weltweit größte und umfangreichste Fernstudium- und Telemedizin-Netzwerk. Es tätigt drahtgebundene, drahtlose und Satellitenkommunikationsaktivitäten. Tele-Therapie ist nur ein Teil davon. Das US-Militär hat ein für Kriegszeiten verwendetes persönliches Physiological-State-Monitor System (PSM) entwickelt, um Atmung, Körpertemperatur, Herzfrequenz und andere physiologische Parameter zu messen.

1.3.1 Tragbare Sensortechnologie

Tragbare Sensortechnologie ist eine in den letzten Jahren neu entstandene Technologielinie. Im Medizinwesen wird sie in den Bereichen Fernüberwachung, klinische Versorgung, Familiengesundheit, Schlafanalyse, Notdienst, Luftfahrt, Pflege für besondere Bevölkerungsgruppen, psychologische Untersuchung, körperliches Training, militärische Aktivitäten usw. verwendet.

Wie der Name schon vermuten lässt, bezieht sich die Anwendung von „Wearable Sensor Technologie“ im medizinischen Bereich darauf, dass ein Mensch das physiologische Datenerfassungssystem trägt. Durch diese Technik wird die physiologische Datenerfassungstechnologie ins tägliche Leben der Menschen integriert. Die Forscher gestalten verschiedene Formen wie Gürtel, Ring oder Uhr als physiologische Datenerfassungsclients, damit der Träger sie als angenehm empfindet und sich wohl fühlt. Zur gleichen Zeit können EKG, Blutdruck, Sauerstoffsättigung, Atmung, Körpertemperatur und Aktivitätsdaten des Patienten erkannt und gemessen werden.



Abbildung 1-4 Body Media System [20]

Das Wearable Monitoring System besteht aus grundlegenden Funktionsmodulen wie Wearable Träger, physiologische Signaldetektion, Signalverarbeitung, Diagnose und drahtlose Übertragungskomponente. Durch diese Module sind nichtinvasive Überwachung, Diagnose und Alarm zu realisieren. In der Regel ist das System bequem zu tragen und einfach zu bedienen. Zudem kann es für lange Zeit arbeiten und intelligente Diagnose, Alarm, drahtlose Übertragung und Fernpositionsbestimmung erreichen. Die Forschungsschwerpunkte umfassen elektronische Fabrikation, Design der Sensoren, Biokompatibilität, Multi-Sensor-Datenfusion, Body-Area-Network, Systemoptimierung, Verlängerung der Akkulaufzeit, Echtzeit-Datenübertragung sowie die Erhöhung der Systemsicherheit und -zuverlässigkeit. Je nach Form kann das Wearable Monitoring System in tragbare (intelligente Ringe, Handschuhe, Armbänder, Abzeichen) und „wearable“ (intelligente Kleidung, Brustgurt) unterteilt werden. Die Vorteile des Erstgenannten liegen in geringer Größe, hoher Tragbarkeit und einfacher Messung. Body Media (siehe Abbildung 1-4) ist ein „wearable“ physiologisches Überwachungsgerät in Form eines Armbandes [20]. Es kann die Körpertemperatur, den Kalorienverbrauch sowie die Körperaktivität messen und die Informationen an einen PC zur Analyse und Verarbeitung senden. In Tabelle 1-1 sind die derzeit wichtigsten Sensoren dargestellt.

Tabelle 1-1 Typische Sensoren verwendet in WSN-basierten Health Care systems

Sensors	Messe Parameters	Geeignet Krankheit	Kommentar	Reference
Pulse oximete	Blood oxygen saturation,Pulse rate	Cardiac,respiratory	Generi	MONIN
Plethysmog-raphig ring sensor	BP, HR, Oxygen saturation, heart rate variability	Cardiac	integrated, wearable sensor	MIT
AMON sensors	SpO2,BP,ECG,temper ature	Cardiac	integrated, wearable sensor	AMON
ECG	ECG, blood oxygen saturation, BP	Cardiac,respiraton	integrated, wearable sensor	UbiMon
HRV/BP sensors	HR,blood oxygen saturation, BP	Seniors,chronic,cardi ac	integrated,wea rable sensor	Mobi Health

1.3.2 Drahtlose Sensornetzwerke

Mit der Entwicklung der Mikroelektronik- und Computertechnologie, der drahtlosen Kommunikationstechnik und des mikroelektromechanischen Systems steht die drahtlose Sensornetzwerktechnologie im Fokus der Aufmerksamkeit der Wissenschaftler und Experten. Es wird deutlich, dass das drahtlose Sensornetzwerk die Informationswelt mit der natürlichen Welt verbindet und die Interaktionsweise zwischen Mensch und Natur verändert.

Das drahtlose Sensornetzwerk besteht aus Sensorknoten, Aggregationsknoten und einem Management-Modul. Sensorknoten sind üblicherweise in ein Mikrosystem eingebettet und sammeln und verarbeiten lokale Informationen. Außerdem speichern, verarbeiten und leiten sie die von anderen Knoten übertragenen Daten weiter. Jedoch ist ihre Fähigkeit zur Datenverarbeitung, -speicherung und -kommunikation oft (noch) nicht ausreichend entwickelt. Der Aggregationsknoten ist in der Regel ein Knoten mit erweiterten Funktionen. Er wird verwendet, um den Gateway des externen Netzwerks und das drahtlose Sensornetzwerk zu verbinden. Ihre Verarbeitungs-, Speicherungs- und Kommunikationsfähigkeit ist stärker. Diese Funktionen übernehmen gegenwärtig oft Mobile Geräte wie z.B. PDA (Personal Digital Assistants) oder Smartphones. Für das Management-Modul ist der Server (oder PC) verantwortlich. Üblicherweise werden Datenbanken und benutzerfreundliche User Interfaces ausgestattet, um den Datenzugriff und das Netzwerkmanagement zu vereinfachen.

Im Vergleich zu dem traditionellen drahtlosen Netzwerk hat das drahtlose Sensornetzwerk folgende Eigenschaften:

1. **Anwendungsorientiertheit:** Die verschiedenen Anwendungsbereiche stellen unterschiedliche Anforderungen an WSN. Dies führt dazu, dass die Designs der Hardware/Software-Layouts und Kommunikationsprotokolle auf jeden Fall unterschiedlich sind. So ist es sehr schwierig, ein universelles Kommunikationsprotokoll zu entwickeln. Die aktuelle und die neue Entwicklung der Protokolle müssen an die bestimmten Anwendungsbereiche angepasst werden.
2. **Begrenzte Knotenkapazität:** Da Energie, Kommunikationsfähigkeit, Rechenleistung und Speicherkapazität der Knoten begrenzt sind, ist das Hauptziel des drahtlosen Sensornetzwerks, durch die Erledigung der zuverlässigen und wirksamen Überwachungsaufgaben Ressourcen aller Art effizient zu nutzen und zur gleichen Zeit die Lebensdauer des Netzwerks zu verlängern.
3. **Selbstorganisationsfähigkeit:** Weil das drahtlose Sensornetzwerk in großem Umfang eingesetzt wird, braucht das System eine große Menge von Sensorknoten im Überwachungsbereich, und in der Regel besteht keine Network-Infrastructure im Überwachungsbereich. Daher sollte der Sensorknoten eine Selbstorganisationsfähigkeit haben, durch die das skalierbare Protokoll und die verteilten Algorithmen selbst das Multihop-Netzwerk bilden.
4. **Datenorientiertheit:** In vielen Fällen gibt es keine zwangsläufige Verbindung zwischen Identifikation und Lokationskennung der Sensorknoten. Wenn der Benutzer das drahtlose Sensornetzwerk nutzt, ist das Auftreten des Ereignisses wichtig, aber nicht der Sensorknoten, welche das Ereignis wahrnimmt. Diese Idee, die Daten als Eventstreiber zu verwenden, nähert sich dem Verhalten der menschlichen Kommunikation und soll auch im Designprozess des Sensor Netzwerk Systems durchgeführt werden.

1.3.3 Kommunikationstechnologie

In den letzten Jahren stand die drahtlose Kommunikationstechnik im Zentrum der Forschung. Derzeit werden viele Untersuchungen in relevanten Bereichen durchgeführt, z. B. in Bezug auf das Mobile Health System, das verschiedene Kommunikationstechniken wie BAN (Body Area Network), PAN (Personal Area Network), WLAN (Wireless Local Network), WWAN (Wireless Wide Area Network) und das Internet umfasst. Abbildung 1-5 beschreibt die Verbindungen zwischen dem mobilen Monitoring System und dafür eingesetzter Kommunikationstechnik, wobei die gestrichelte Linie die Kommunikation innerhalb des

homogenen Netzwerkdarstellt und die gepickte Linie die Kommunikation zwischen den verschiedenen Netzwerken beschreibt [21].

Aus der Abbildung ist deutlich zu erkennen, dass jede Kommunikationstechnik das Ziel hat, die Anforderungen der medizinischen Überwachung und Pflege zu erfüllen. Nur wenn die Knoten inBANund PAN effektiv und in großem Umfang mit dem öffentlichen Netz verbunden sind, kann das System seine Vorteile entfalten.

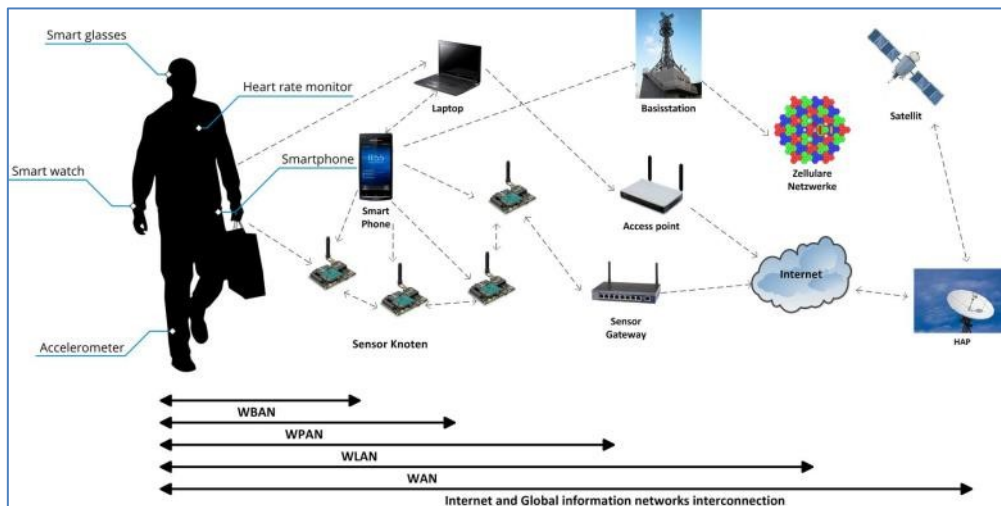


Abbildung 1-5 Beziehung zwischen Health Care WSN und Kommunikationsgeräten

In den Bereichen BAN und PAN sind Bluetooth, Zigbee [22], Radiofrequenz, Ultra-Wideband [23] und ANT [24] am weitesten verbreitet. Im Bereich WLAN spielt das Protokoll (wie 802.11a/b/g, e, p, s, n) eine beherrschende Rolle. Im Bereich WWAN wird WIMAX immer beliebter. Internet, Satellitenkommunikation und alle Mobilfunknetze (wie GSM, CDMA, 3G, LTE) werden weltweit breit zur Netzwerkbildung eingesetzt. Gemäß der Beschreibung in der Literatur [25], [26] werden hier die Haupteigenschaften der das medizinische Sensornetzwerk betreffenden drahtlosen Kommunikationstechniken aufgeführt (siehe Tabelle 1-2).

Tabelle 1-2 Eigenschaften der Wireless Technologien

Name	Effective distance	Frequeney band	Bitrate	Physical Layer	Related Technology
ETSI: EN301, EN302	BAN	9-135kHz, 315-600kHz, 12.5-20MHz, 30-37.5MHz, 402-405MHz	-	-	ULP-AMI
Medical Implant Communicatio	BAN	402-405MHz	-	-	-

n Service(MICS)					
RFID	BAN/PAN	125-134.2kHz,13.56MHz,868-954MHz	106kbits/s	OOK,ASK	
DECT	PAN(<50m)	1.88 — 1.9GHz,24GHz	32.64,384kbits/s	TDD/TDMA,FDMA	
Near Field Communication(NFC)	BAN/PAN	13.56MHz	106,212,424kbits/s	-	-
IEEE802.15.1	PAN	2.4GHz	721kbits/s750kbits/s22Mbits	FHSS	Bluetooth
IEEE802.15.3a	High rate PAN	2.4GHz	>100Mbits	OQPSK,OFDM	UWB
IEEE802.15.4	PAN	868MHz,905MHz,2.4GHz	20kbits/s,40kbits/s,250kbits/s	BPSK	Zigbee, Wireless
IEEE802.15.6	BAN/PAN	1THz	> 1Gbits/s	-	
IEEE802.15.11a	WLAN<100m	5Ghz	6-54Mbits/s	OFDM	-
IEEE802.15.11b	WLAN<100m	2.4GHz	1,2,5.5,11 Mbits/s	DSSS,CCK	Wifi
IEEE802.15.11g	WLAN	2.4GHz	<22Mbits/s	OFDM	-
IEEE802.16(x)	WLAN<50km	2-11GHz	0.5-30Mbits/s	OFDM	WiMax, WiBro
IEEE802.20	WLAN<15km	1.25,5GHz	08,3.2,4,16 Mbits/s	-	For mobile users speed<250km/h
Hiper-LAN	WLAN<150m	5GHz	11.54 Mbits/s	OFDM	-
Hiper-Acess	WLAN<5km	11-43.5GHz	25-100Mbits/s	LOS PMP,back bone access for GSM	-

1.3.4 Mobiles Computing

Mit der Entwicklung des Mobile Computing werden Smartphones als Middleware eingesetzt, um Sensoren mit fernmedizinischen Zentren im Mobile Health System zu verbinden. Mobile Computing ist ein verteilter Rechenmodus, der auf Mobilfunk und einem Wireless-Netzwerk aufbaut. Technologien und Applikationsanforderungen sind die Antriebskraft für seine Weiterentwicklung. Das Ziel des Mobile Computing ist, einen Informationsaustausch aller Art zu erreichen, um intelligente Kommunikationslösungen und -dienstleistungen anzubieten.

1.3.5 Standardisierung

Standardisierung steht immer im Fokus eines Gesundheitssystems. Jedoch steckt die Standardisierung des mHealth aufgrund der unterschiedlichen Gesundheitssysteme der verschiedenen Nationen noch immer in den Kinderschuhen [27]. Derzeit wird die Standardisierung der Gesundheitssysteme wesentlich von der International Organization for Standardization (ISO)[28] und den Arbeitsteams 11073 und 802.15.6 des International Institute of Electrical and Electronics Engineers (IEEE), der International Telecommunication Union (ITU)[29], dem Arbeitsteam Telemedicine-Alliance-Project (ATLP) [30] aus der Weltgesundheitsorganisation (WHO) und der Europe Space Agency (ESA) [31] sowie der Telemedizin Standardization Coordination Group (TSCG) [32] vorangetrieben. Die Literatur[33] fasst den aktuellen Stand des Einsatzes des Informations- und Kommunikationstechnik-standards im Bereich medizinische Versorgung zusammen, in [34] wird ein weitergehender Überblick über den derzeitigen Standard im Bereich der medizinischen Versorgung weltweit dargestellt.

In diesem Abschnitt werden die Standards im Bereich der medizinischen Versorgung nach Kommunikation, System und Devices, Interconnection, medizinische Information und Datenmanagement geordnet dargestellt (siehe Tabelle 1-3).

Tabelle 1-3 Standard: Healthcare WANS

Standard	Beschreibung	Klassifikation
Fcc47, CFR for Radio Transmitter Requirements Part 2	Frequency allocations, general rules and regulations	<i>Kommunikation</i>
ETSI EN300674; FCC47 CFR Part 15, 18 and 95	using radio frequency equipment and systems	<i>Kommunikation</i>
ETS300339, ETS30068	EMC for radio and or specific frequency radio Communication equipment	<i>Kommunikation</i>
IEEE802.11x; IEEE802.15x; ISO/IEEE 11073X.73-ITU	Using radio frequency equipment and system	<i>Kommunikation</i>
ETSI EN 302 195-1; ISO14708	general safety requirements of active implantable medical devices	<i>System und Devices</i>
ANSI C63.19	Compatibility between wireless devices and hearing aids	<i>System und Devices</i>
IEEE 1073.X; ISO/TR 16056-1; ISO/TR 16056-2; ISO/TR; 18307; ISO/IEEE 11073-30300; ISO 12052; HL7	Communication at the point-of-care and interoperability of TH systems and networks for Health informatics, DICOM	<i>Interconnection</i>
US Universal Healthcare Identifier (UHID)	Classification and codes of basic	<i>Medizinische</i>

	personal information, including health	<i>Information und Daten Management</i>
ISO/TS 18308	Supports using, sharing, and exchanging Electronic health records across different health sectors, countries and models of healthcare delivery	<i>Medizinische Information und Daten Management</i>

1.4 Standards für den Datenaustausch im Gesundheitswesen

Seit den 80er Jahren des 20. Jahrhunderts versuchen viele Forschungseinrichtungen und Unternehmen für medizinische Systeme, unter Berücksichtigung von Sicherheit und Vertraulichkeit der Daten von Patienten elektronische Patientenakten mit unterschiedlichen Funktionen an unterschiedlichen Orten leicht zugänglich zu machen. Bis jetzt ist bereits viel erreicht worden und es gibt viele erfolgreiche Konzepte. Im Folgenden werden sie ausgehend von zwei Aspekten, nämlich Nachrichten- und Geschäftsanwendungen, betrachtet und vorgestellt.

HL7 (Health Level 7) aus der Perspektive der Nachricht betrachtet, kann die Entwicklung der Interoperabilität auch in zwei Ebenen unterteilt werden: Messageebene und Dateiebene. Die Verwendung einer international standardisierten Nachrichten-Architektur verbessert die Lesbarkeit von ausgetauschten Nachrichten. HL7 ist derzeit der am weitesten verbreitete Standard für den Datenaustausch im medizinischen Bereich [35].

DICOM standardisiert den Austausch von medizinischem Imaging und den dazugehörigen Daten, sodass DICOM oft als Ergänzung zu HL7-Anwendungen dient. Mehr und mehr medizinische Einrichtungen verwenden das einheitliche HL7 sowie das DICOM-Format und die Beschreibung, um die übermittelten medizinischen Daten zu standardisieren. Gefördert durch die RSNA (Radiological Society of North America) und HIMSS (Healthcare Information and Management Systems Society) werden im Projekt IHE verschiedene medizinische Daten-Standards (wie HL7 und DICOM) kombiniert, sodass die verschiedenen Protokolle zusammenarbeiten können, um für die beste Arbeit im medizinischen Bereich (vor allem klinische Arbeit) bessere Dienste anbieten zu können [36].

Der **EHR** (Electronic Health Record) von einem Patienten beschreibt digitale Aufzeichnungen, bei denen die Gesundheit, Pflege und Behandlung eines Individuums im Zentrum stehen. Er ist eine menschenorientierte digitale Patientenakte, in der Informationen über Gesundheit, Pflege und Behandlungen eines Individuums im Laufe des Lebens von der Geburt bis zum Tod

verzeichnet sind. EHR ermöglicht den Austausch und das Teilen von medizinischen Informationen von unterschiedlichen Informationsbereitstellern und -nutzern zwischen den verschiedenen Institutionen und Systemen. Derzeit unterstützen die standardisierten Dateien CEN und HL7 die Interoperabilität von elektronischen Patientenakten. Allerdings geht aus dem Bericht der Medical Group Management Association von 2007 hervor, dass nur 30% der Organisationen im Gesundheitswesen und 19% der Praxen betriebsfähig elektronische Patientenakten nutzen [37].

HSSP(Healthcare Services Specification Program): Betrachten wir die Interoperabilität aus der Perspektive der Entwicklung von Business-Anwendungen, so muss in einem Projekt oder bei mehreren Projekten, wie oben erwähnt, eine Vielzahl von medizinischen Data-Messaging-Standards und Nachrichten-Standards einfach und vollständig integriert werden. Ähnliche Projekte wie HSSP bieten eine Spezifikation, wie die dazugehörigen Funktionen und Schnittstellen von medizinischen Diensten zu beschreiben sind. In den Vereinigten Staaten wurde bundesweit NHIN eingeführt, damit EHR-Informationen besser ausgetauscht werden können. Analysieren wir aus technischer Sicht den Business Layer und Interoperabilität und analysieren wir die Projekte aus dem fünften und sechsten Rahmenprogramm der EU (FP5, FP6), so ist SOA eine Methode, welche immer mehr verwendet wird [38].

1.4.1 Interoperabilität von Telemedizin-Systemen

Die Digitalisierung von medizinischen Daten und Behandlung bedeutet nicht nur, dass die Ärzte bei der Diagnose und Behandlung mit mehr Möglichkeiten arbeiten können, sondern auch, dass der Austausch, die Speicherung, der Zugriff und die Weiterleitung von medizinischen Daten für die ganze Gesellschaft vereinfacht werden. Allerdings hat ein Krankenhaus eine sehr komplexe Struktur und die verschiedenen Fachabteilungen haben unterschiedliche Bedürfnisse, sodass es für ein einzelnes Produkt von einem Unternehmen schwierig ist, alle Fachgebiete abzudecken. Daher gibt es in einem Krankenhaus oft eine große Anzahl von heterogenen, autonomen, verteilten medizinischen Bildgebungsgeräten, Datenbanken, Informationssystemen und wissensbasierten Anwendungen etc. von verschiedenen Unternehmen und dadurch wird die Interoperabilität von heterogenen medizinischen Systemen immer wichtiger.

Mit der Interoperabilität im medizinischen Bereich sind die genaue, effiziente und konsistente Kommunikation und der Datenaustausch zwischen unterschiedlichen Informationssystemen und Anwendungssoftwares gemeint. Dabei können die ausgetauschten Informationen richtig

eingesetzt werden. Die Verbesserung der Interoperabilität des Informationssystems führt zur Minimierung der Gesundheitskosten. Darüber hinaus erhalten die Patienten eine zeitnahe und wirksame medizinische Dienstleistung.

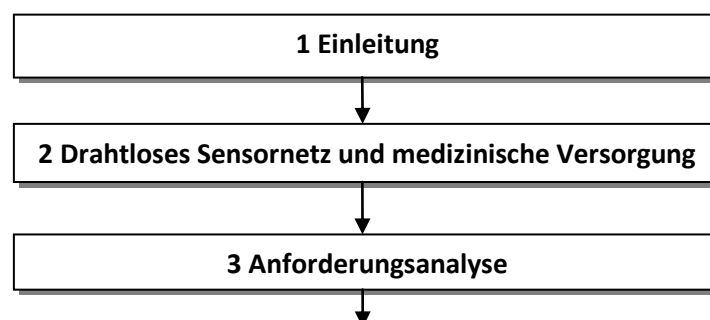
Die oben genannten Aspekte führen dazu, dass das Feld der mHealth-Dienstleistungen spezielle Anforderungen hat und vom Umfang her extrem groß ist. Dies erfordert einen ganzheitlichen Plan, der die Informationstechnik der Systeme fördert. In der Implementierungsphase müssen die Business-Anforderungen und die funktionalen Anforderungen noch genauer umgesetzt werden. Zudem muss im Entwurfsprozess des Systems stets ganzheitlich vorgegangen werden, um eine bessere Interoperabilität zwischen den Systemen zu erreichen.

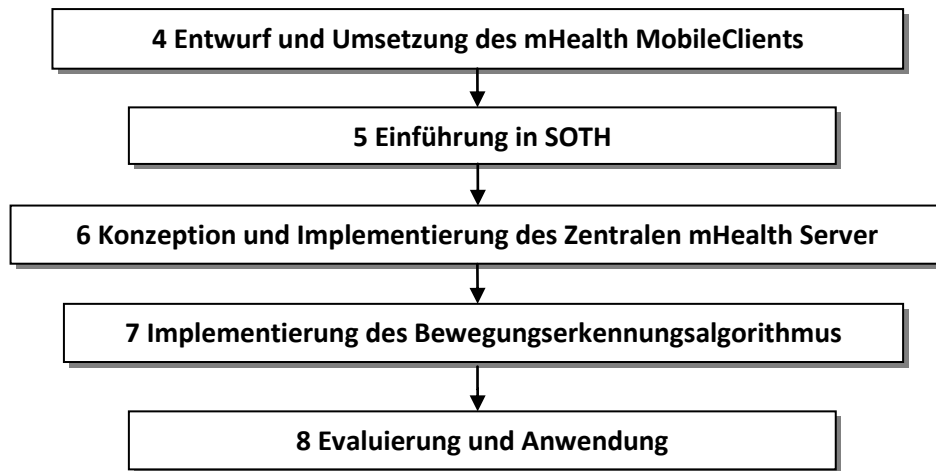
1.5 Gliederung der Arbeit

Auf Basis der Eigenschaften des Telemonitorings, besonders des mHealth Systems, zeigt diese Arbeit die Schwierigkeiten beim System-Design auf und beschreibt die Struktur des serviceorientierten mHealth-Systems. Diese Arbeit untersucht zunächst einige Schlüsseltechnologien des mHealth-System, wie drahtlose Kommunikationstechnologien, drahtlose Sensornetze, Mobile Computing, Datenverwaltung und -analyse sowie Datenqualität und -sicherheit. Zudem werden die Methoden zur Optimierung der Struktur, die logische Organisation, die Kommunikationsschnittstelle sowie die Datenverarbeitung im Bereich des medizinischen Monitorings vorgestellt.

Der Aufbau dieser Arbeit stimmt mit der Struktur eines Telemonitoring-Systems überein und die Arbeit folgt dieser Struktur: **Lokale Seite des Telemonitoringsystems** → **Fernüberwachungszentrale** → **Datenverarbeitung**.

Abbildung 1-5 zeigt den Aufbau der Arbeit





Kapitel 1 erklärt den aktuellen Forschungsstand bezüglich mobile Health Systemen und stellt wichtige Technologien des Systems wie Wireless Sensornetz, Kommunikationstechnologien, Mobile-Computing und Standards für den medizinischen Datenaustausch usw. vor.

Basierend auf den Eigenschaften des Gesundheitssystems, besonders des Telemonitorings, wird in **Kapitel 2** eine neue Definition von **Medizinisches Sensornetzwerk** gegeben und die Struktur des modular basierten medizinischen Sensornetzes beschrieben. Zudem werden die wesentlichen Eigenschaften der Sensornetze untersucht.

In **Kapitel 3** „Anforderungsanalyse“ wird im Wesentlichen die grundlegende Architektur eines mHealth-Systems vorgestellt. Das Szenario und die gesamten Anforderungen des Systems werden ebenfalls geschildert. Diese basieren auf den Business-Anforderungen des oberen Applikation Layer, technischen Anforderungen des unteren Layers sowie den Anforderungen der Schnittstellen.

Kapitel 4 lautet „Entwurf und Umsetzung des mHealth Mobile Client“. Entsprechend der Anforderungen an das System aus den vorherigen Kapiteln umfassen die mobilen Erfassungsclients des IPM-mHealth Systems drei Kernkomponenten: Ein *Sensorkommunikationsmodul*, welches mit den medizinischen Sensoren kommuniziert; ein intelligentes *Echtzeit-Überwachungsmodul* und ein *Datenübertragungsmodul* für die Weiterleitung erfasster physischer Parameter auf den Server. In diesem Kapitel werden die Grundsätze des Entwurfs und die Implementierung der Komponenten detailliert beschrieben.

In **Kapitel 5** „Einführung in SOTH“ wird die vorgeschlagene SOA-basierte „Open-Telemonitoring-Framework for Healthcare“ detailliert beschrieben. Nach der „Layered-Architecture“ wird eine Reihe von Methoden vorgestellt, die zur Beschreibung von medizinischen Software Ressourcen und Diensten dienen.

In Kapitel 6, „Konzeption und Implementierung des Zentralen mHealth Server Systems“ werden die Ziele und Grundsätze des Entwurfs des IPM-mHealth Server Systems (IPM-ZS) anhand der Anforderungen an das System aus den vorherigen Kapiteln erläutert. Zunächst werden die Ziele des Entwurfs des Systems analysiert. Nach der im vorherigen Kapitel ausgeführten SOTH-Struktur wird hier ein Healthcare-Service-Bus (HSB) entworfen. Der HSB erlaubt es Entwicklern, schnell und einfach mehrere medizinische Anwendungen als HSB-Service zu verbinden, damit der gesamte medizinische Service durch die Kombination verschiedene Module abbilden kann. Im Teil der Systemarchitektur wird die Funktion jedes Moduls beschrieben.

In Kapitel 7, „Implementierung des Aktivitätserkennungsalgorithmus“ werden hauptsächlich das Design und die Umsetzung einer Bewegungserkennung auf der Grundlage von Beschleunigungssensoren dargestellt. Zunächst wird der für diese Arbeit verwendete Sensor beschrieben. Im Anschluss werden neue Abläufe und Methoden dargelegt, wie mittels Beschleunigungsdaten Patientenverhalten erkannt werden kann.

Am Institut für Präventivmedizin wurden bereits mehrere medizinische Untersuchungen in Verbindung mit dem entwickelten System durchgeführt. In **Kapitel 8** werden dazu exemplarisch einige Untersuchungen näher erläutert. Die Belastung und Stabilität des Systems werden ebenfalls dargestellt.

Schließlich werden die Vor- und Nachteile der entwickelten Systems und neue Lösungen zusammengefasst und Potenziale für die zukünftige Weiterentwicklungen genannt.

2 Drahtloses Sensornetzwerk und medizinische Versorgung

Dieses Kapitel befasst sich mit der systematischen Struktur eines medizinischen Wireless-Sensornetzwerks. Entsprechend der Eigenschaften der medizinischen Pflege und Überwachung im Speziellen wird die Definition eines Sensornetzwerks für medizinische Versorgung erläutert, außerdem wird die Netzwerkstruktur eines modularen medizinischen Sensornetzwerks beschrieben.

2.1 Definition eines medizinischen drahtlosen Sensornetzwerksystems

Heute entwickelt sich die Forschung im Bereich der medizinischen Sensornetze sehr schnell. In Kapitel 1 wurden die relevanten Forschungsthemen vorgestellt. Daraus ist deutlich zu ersehen, dass die Forschungsinhalte aufgrund der Differenzen der technischen Niveaus und der Forschungszwecke voneinander abweichen. Oft wird ein medizinisches drahtloses Sensornetzwerk wie folgt definiert: Das Wireless-Sensornetzwerk dient als Plattform für die Umsetzung der fernmedizinischen Pflege. Diese Definition ist allerdings sehr unscharf und kann daher nicht als eine einheitliche Definition verwendet werden. Auf Basis der Beschreibung von Telemedizin und der drahtlosen Sensornetzwerk-Beschreibung ist die Definition des medizinischen Sensornetzwerks wie folgt gegeben:

„Dasmedizinische Sensornetzwerk ist eine Kommunikationsinfrastruktur, die eine Vielzahl von Sensorknoten verbindet. Es kann die physiologischen Parameter und Umgebungsparameter von einem oder mehreren Objekten der Pflege aufzeichnen, und über die selbst organisierte Form eines homogenen oder heterogenen Netzwerksunter verschiedenen Bedingungen einen bidirektionalen nahtlosen Informationsaustausch zwischen Objekten der Pflege und dem professionellem Management zu erreichen.“[39]

Laut der obigen Definition besitzt das medizinische Sensornetzwerk folgende Merkmale:

- (1) Die von Sensorknoten des medizinischen Sensornetzwerks aufgezeichneten Inhalte umfassen nicht nur wichtige physiologische Parameter wie *Herzschlagfrequenz* oder *Puls*, sondern auch Umgebungsparameter wie *Umgebungstemperatur* oder *Luftfeuchtigkeit*. So sollte das medizinische Sensornetzwerk den Transfer sowohl normaler Daten als auch von Echtzeitdaten unterstützen.
- (2) Aus Sicht der Struktur betrachtet, umfasst das medizinische Sensornetzwerk Personal Area Netzwerke, Indoor-Sensornetzwerke und öffentliche Sensor-Netzwerke. Solche

Netzwerke können homogen oder heterogen sein. So ist dabei eine offene Netzwerkstruktur für die Vernetzung notwendig.

- (3) Aus Sicht der Datenverwaltung kann die Datensammlung des medizinischen Sensornetzwerks zentralisiert, dezentralisiert oder hybrid erfolgen. Beim Protokolldesign ist es notwendig, die Kompatibilität von verschiedenen Datensammlungsmethoden zu berücksichtigen [40].
- (4) Aus Sicht der Netzwerk-Bereitstellungsszenarien sollte ein medizinisches Sensornetzwerk den Anschluss von verschiedenen Hauptnetzwerken unterstützen, um die nahtlose real-time Überwachung zu ermöglichen. Es ist somit erforderlich, das Problem der Mobilität unterschiedlicher Netzwerke zu lösen.
- (5) Aus Sicht der Überwachung, Dienstleistung und Netzwerk-Bereitstellungsszenarien kann das medizinische Sensornetzwerk nicht nur für bestimmte Nutzer eingesetzt werden, sondern auch zur Überwachung bestimmter Nutzergruppen. Die Zahl von Dienstobjekten des Netzwerks kann bei einem oder mehreren liegen. Deshalb müssen der Anschluss und die Kommunikation von verschiedenen Subnetzwerken ebenfalls berücksichtigt werden.

2.2 Architektur des medizinischen drahtlosen Sensornetzwerkssystems

Gemäß der in Abschnitt 2.1 angegebenen Definition hat das in dieser Arbeit beschriebene medizinische Sensornetzwerk das Ziel, das Alltagsleben der Objekte der Pflege nicht zu stören und durch eine geeignete systematische Netzwerkstruktur nahtlos und hocheffizient die Objekte der Pflege in Echtzeit zu überwachen und dynamischen Alarm zu ermöglichen. In diesem Abschnitt wird durch die Analyse der **logischen Struktur** und **Vernetzungsstruktur** die modular-basierte Struktur des medizinischen Sensornetzwerkssystems vorgestellt.

2.2.1 Logische Struktur des Systems

Wie in Kapitel 1 gezeigt, gibt es für das medizinische Sensornetzwerkssystem drei wichtige Anwendungsszenarien: *Familien- und Krankenhauspflege, öffentliche Überwachung und Pflege in Not*. Durch den Anschluss an verschiedene drahtlose Netzwerke oder Kabelnetzwerke und das Wireless-Sensornetzwerk als grundlegendes Mittel werden medizinische Informationen an Dienstleister übermittelt, welche medizinische Überwachung und Dienstleistungen anbieten. Weil es keine einheitliche Definition der systematischen logischen Struktur gibt, steht die systematische logische Struktur in dieser Arbeit für eine bestimmte Struktur, welche die Zusammenschaltung aller Netzwerkgeräte beschreibt [41].

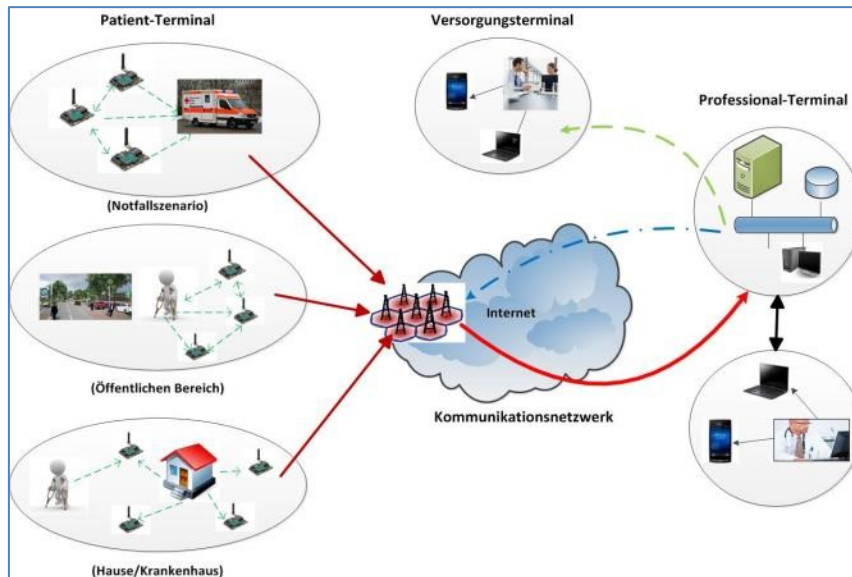


Abbildung 2-1 Logische Struktur des medizinischen Sensornetzwerksystems

Die in der Arbeit behandelte logische Struktur des medizinischen Sensornetzwerksystems besteht aus folgenden vier Teilen: *Patient-Terminal*, *Kommunikationsnetzwerk*, *Professional Terminal* und *Versorgungsterminal*. Am Patient-Terminal haben Personal Area Netzwerke und öffentliche Sensornetzwerke die Aufgabe, Daten zu sammeln, und es deckt viele Szenarien wie familiäre Pflege, Krankenhäuser, öffentliche Plätze usw. ab. Die grundlegenden Merkmale lauten wie folgt: eine große Menge von Daten fließt innerhalb des Netzwerks (wie die grüne gepunktete Linie der Abbildung 2-1 zeigt). Ein Teil der Daten wird durch den Gateway aus diesem Port herausfließen (wie die rote gepunktete Linie der Abbildung 2-1 zeigt). Nur ganz wenige Daten fließen herein (wie die blaue gepunktete Linie der Abbildung zeigt). Das Professional Terminal dient als der Hauptort der Datenspeicherung und hat die Aufgabe, die Daten zu bewerten. Üblicherweise besteht das System aus Hochleistungs-Computern. Es empfängt die herein fließenden Daten (wie die dicke rote Linie in der Abbildung 2-1 zeigt). Es leitet die Kontrollinformationen oder Befehlsinformation an das Patient-Terminal (wie die dicke blaue Linie in der Abbildung 2-1 zeigt) und Warnungsinformationen an die Familie des Patienten weiter (wie die grüne gepunktete Linie in der Abbildung 2-1 zeigt). Am Versorgungsterminal erhält das medizinische Personal über verschiedene Plattformen wie Handy, PDA und PC usw. die Informationen über spezifische physiologische Parameter der überwachten Patienten. Am Kommunikationsnetzwerk werden verschiedene Zugangsnetze und Kommunikationstechnologien jedes Terminals verbunden, um einen nahtlosen Anschluss zu erreichen.

2.2.2 Layout des Systems

Derzeit wird die systematische Struktur des medizinischen Sensornetzwerks nur auf logischer Ebene analysiert. Aber die Analyse der hierarchischen Vernetzungs-Struktur des Systems ist daneben unerlässlich. Wenn die logische Struktur des Systems die Anwendungsszenarien im System bestimmt, determiniert die Vernetzungsstruktur des Systems die grundlegenden Arbeitsleistungen des Systems [42]. Dies liegt daran, dass Netzwerkdesigner die gestalteten und verbesserten Inhalte dem Netzwerklayer zuordnen müssen. Für die speziellen Anwendungen im Bereich der medizinischen Versorgung werden in dieser Arbeit bei der Systemarchitektur folgende Methoden verwendet: Auf Basis der Zusammenfassung der grundlegenden Anwendungsszenarien der medizinischen Versorgung werden die hauptsächlichen Anforderungen des Strukturdesigns festgelegt. Basierend darauf werden sie in mehrere ausführlichere Anforderungen unterteilt und diese Anforderungen werden der passenden Netzwerkschicht zugeordnet. Bei dem Design jeder Schicht müssen die Anforderungen gestaltet und optimiert werden, um hierarchische Rahmenlösungen zu bilden und schließlich einen ganzheitlichen Rahmen zu erhalten. Abbildung 2-2 stellt den Aufbau des ganzheitlichen Rahmens des medizinischen Sensornetzwerks dar.

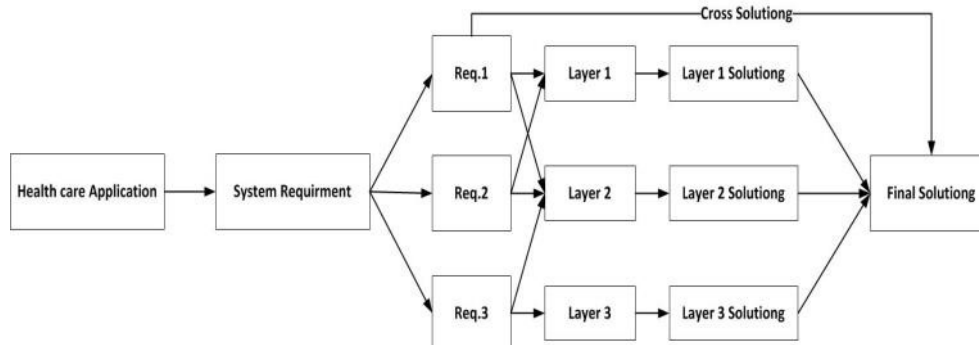


Abbildung 2-2Layout des medizinischen Sensornetzwerksystems

In Bezug auf den systematischen Rahmen des Gesundheitssystems bestehen bereits mehrere internationale standardisierte Definitionen. Sie umfassen zwei einflussreiche Standards, wie IEEE/ISO11073 und HL7. IEEE12073, welche von der International Organization for Standardization, IEEE und dem European Standardization Committee (ECS) zusammen festgelegt wurden. Der Zweck ist es, die Bestandteile des Healthcare-Systems zu definieren und die Probleme wie die Verbindung zwischen medizinischen Geräten, Datenaustausch, Authentifizierung und Fernbedienung zu lösen. HL7 definiert auf der Datenebene ein flexibles medizinisches Informationsformat, Codes, Speicherung und Austauschstandards, um sicherzustellen, dass alle medizinischen Systeme einen Informationsaustausch und eine

vereinheitlichte Informationsverarbeitung realisieren können [43]. Es lässt sich sagen, dass HL7 mehr Wert auf die Definition der medizinischen Informationsstruktur legt, während IEEE11073 auf der Transportschicht die Interaktion zwischen medizinischen Geräten definiert. Zurzeit entwickeln Philips, Toshiba und Panasonic diese medizinischen Geräte unter Einhaltung solcher Standards [44].

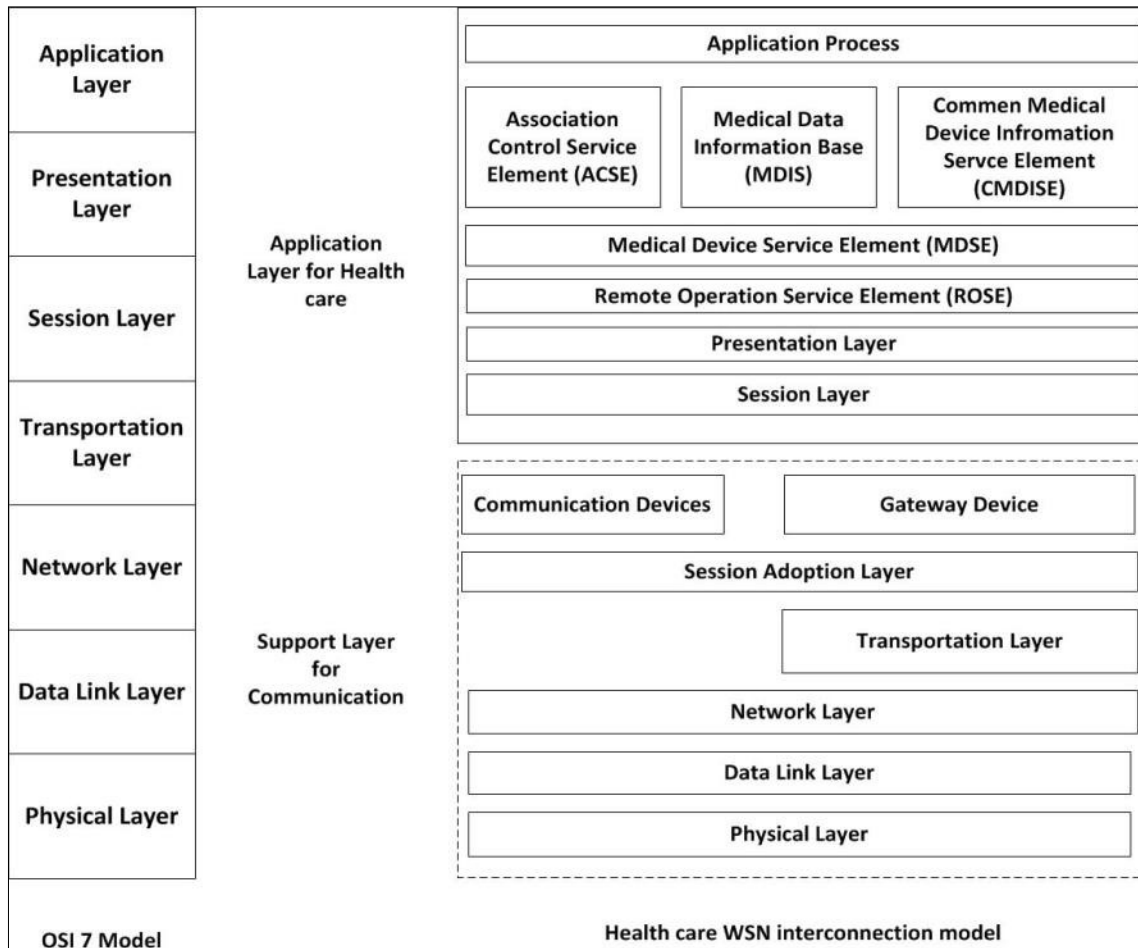


Abbildung 2-3 Layer-basierte Struktur des medizinischen Sensornetzwerksystems

Basierend auf den vorgenannten Informationen stellt diese Arbeit eine hierarchische Vernetzungsstruktur eines drahtlosen Sensornetzwerksystems nach den Standards IEEE11073 auf. Abbildung 2-3 zeigt die entsprechenden Beziehungen zwischen diesem Modell und dem OSI-Modell [45]. Insgesamt wird dieses Modell in die Netzwerk-Anwendungsschicht und Kommunikations-Trägerschicht unterteilt.

Die Netzwerk-Anwendungsschicht umfasst die Anwendungsschicht, die Präsentationsschicht und Session-Schicht des OSI Modells, und entspricht dem IEEE110746 Standard. Bei der Kommunikations-Trägerschicht werden die Besonderheiten der medizinischen Pflege berücksichtigt und ein Session Adaption Layer im Personal Area Netzwerk und Gateway besonders für spezielle Dienste der Gesundheitsversorgung hinzugefügt, um die

Kompatibilität zwischen dem Session-Protokoll für drahtlose medizinische Sensor-Netzwerk-Systeme und dem aktuellen Internet-Session-Protokoll zu erreichen und die nahtlose Abdeckung der medizinischen Versorgung zu realisieren.

Geringe Unterschiede bestehen zwischen der Kommunikations-Trägerschicht und dem OSI-Modell. Der Schwerpunkt liegt darauf, Probleme wie physikalische Schicht, Sicherungsschicht und Netzwerkschicht zu lösen. Es muss darauf hingewiesen werden, dass die in dieser Arbeit behandelten wichtigen Kommunikationstechniken sich auf die Kommunikations-Trägerschicht konzentrieren, welche das Design des Routing-Protokolls und Session-Protokolls umfassen.

2.3 Die Schlüsseltechnologien des medizinischen drahtlosen Sensornetzes

Im Fernüberwachungssystem sind die Verwaltung der medizinischen Versorgung, das Durchführen von Krankheitsdiagnosen und die Arbeitsvorbereitung medizinischer Organisationen die wichtigsten Forschungsrichtungen. Krankheitsdiagnosen schließen die automatisierte Diagnose sowie die Erkennung und Klassifizierung von unterschiedlichen Krankheiten ein. Die Verwaltung der medizinischen Versorgung bedeutet die Speicherung, das Management und die Umwandlung der medizinischen Daten [46]. Die Anwendung der drahtlosen Technologie in der medizinischen Versorgung kann nicht nur die Effizienz und Qualität [47] der Versorgung erhöhen, sondern auch den Zeitaufwand und die Kosten der Patienten erheblich reduzieren, womit die medizinischen Kosten der ganzen Gesellschaft reduziert werden.

In diesem Abschnitt werden die wichtigsten Techniken im Bereich der drahtlosen Sensornetze detailliert diskutiert. Bei der Forschung im Bereich der drahtlosen Sensornetze gibt es insgesamt drei wesentliche Richtungen: (1) *Die Erforschung von Sensorknoten*, (2) *die Erforschung von Software-Protokollen* und (3) *die Erforschung der Anwendungsbereiche des drahtlosen Sensornetzes*. Bei der Erforschung von Sensorknoten sind die zwei wichtigen Themen der Energieverbrauch des Knotens und die Stromversorgung. Sensorknoten sind in der Regel batteriebetrieben. Wegen des unbeaufsichtigten Betriebs ist eine dauerhafte Aufladung unmöglich. So ist es wesentlich, die Lebensdauer des Knotens zu verlängern, um so eine Verlängerung des Lebenszyklus des Netzwerks zu erreichen. Dies ist ein wichtiges zu lösendes Problem für ein drahtloses Sensornetzwerk [48]. Bei der Erforschung von Software-Protokollen gilt, wenn die Verteilung der Sensorknoten zufällig

und dicht ist, so ist die Überlappung zwischen den Reichweiten vieler Knoten möglich. Dies kann dazu führen, dass Redundanzknoten entstehen [49]. Die Abdeckung der Redundanzknoten kann bedeuten, dass sogar beim Ruhezustand die vom Netzwerk abgedeckten Knoten nicht beeinflusst werden. Es ist ein sehr wichtiges Forschungsziel, durch Software-Protokolle solche Abdeckung der Redundanzknoten beim Ruhezustand zu verwalten, um Energieeinsparungen zu erreichen. Bei der Anwendung von drahtlosen Sensornetzwerken werden je nach Anwendungsbereich unterschiedliche Leistungsanforderungen gestellt. Im Allgemeinen sind die Sicherheit des drahtlosen Sensornetzwerks, die Dienstqualität und die Interaktion zwischen verschiedenen Plattformen zu berücksichtigen.

2.3.1 Energieverbrauch in drahtlosen Sensornetzen

Das Problem des Energieverbrauchs des drahtlosen Sensornetzwerks sollte zuerst gelöst werden. Eine energieeffiziente Architektur und Kommunikationsprotokolle des drahtlosen Sensornetzwerks unterliegen den Einschränkungen des Energieverbrauchs des Netzwerks. Deshalb sind bei der Erforschung des Energieverbrauchs des drahtlosen Sensornetzwerks hocheffiziente Kommunikation und Daten-Management-Protokolle sehr wichtig für die Anwendung des drahtlosen Netzwerks, aber die Fortschritte in der Akku-Technologie können dies noch nicht leisten [50]. Es gibt neben der Stromversorgung mittels Batterie noch vier Methoden zur Energieversorgung: (1) Mechanische Energie, die in elektrische Energie umgewandelt wird, um die Stromversorgung des Knotens zu gewährleisten. Meninger [51] nutzt einen variablen Kondensator, um mechanische Energie in elektrische Energie zu verwandeln, und somit eine Strommenge von 5mw zu erzeugen. (2) Sonnenenergie: Sonnenenergiebatterien eignen sich gut für im Freien gelegene Netzwerke. Aber das Wetter hat einen großen Einfluss auf das Gewinnen von Sonnenenergie. Bei sonnigem Wetter beispielsweise kann ein Quadratmeter Sonnenkollektor eine Energiemenge von 15mw sammeln, aber bei bewölktem Wetter kann nur eine Energiemenge von 0.05mw gesammelt werden. (3) Thermische Energie: In einigen Fällen ist die Stromerzeugung der Sensorknoten durch Geothermie möglich. Weil manche Sensoren nur wenig Energie verbrauchen, ist es möglich, durch thermische Energie Strom zu generieren, um Sensorknoten mit Strom zu versorgen. (4) Radiofrequenz: Der Pittsburgher Zoo verwendet Radiofrequenz-Technologie, um die Energie für das drahtlose Sensornetzwerk bereitzustellen. Durch eine kontinuierlich arbeitende Stromversorgungsanlage wird ein dauerhaftes, niederfrequentes Signal abgegeben, um die Energieversorgung für das drahtlose Sensornetzwerk bereitzustellen [52]. Jeder Sensorknoten ist mit einer HF-Antenne als Empfänger ausgerüstet. Durch diese Anlage

werden die Batterien aufgeladen. Diese Methode löst effektiv das Problem der Stromversorgung für Sensorknoten. Aber die Effizienz der Stromversorgung ist dabei ziemlich niedrig.

Tabelle 2-1 Vergleich der verschiedenen Sensor-Netzwerk-Plattformen

Sensorknoten Plattform	Mikrocontroller			Sensor	
	CPU	Taktfrequenz	Hauptspeicher(KB)	Type	Max. Übertragungsrate (Kbs)
Nesc	Atmel AT90s8535	4	8	RFM TR1000	10
Sene mote	Atmel AT90s8535	4	8	RFM TR1000	10
Rene2 mote	Atmel Tmega 163	20	16	RFM TR1000	10
MIT MAMPS	Intel StrongARM SA-1100	4	128	National Semiconductor LMX3162	1024
Crossbow MICA	Atmel ATmega 128L	4	128	RFM TR1000	40
Crossbow MIC2DOT	Atmel ATmega 128L	4	128	Chipon CC 1000	40
Crossbow MICA2	Atmel ATmega 128L	4	128	Chipon CC 1000	38.4
Crossbow MICAZ	Atmel ATmega 128L	4	128	Chipon CC 1000	250
EYES(Nedap)	TI MP430F 149	2	60	RFM TR1001	57.6

Tabelle 2-1 listet verschiedene Sensor-Netzwerk-Plattformen auf. Nutzer können nach ihren Bedürfnissen verschiedene Sensor-Netzwerk-Hardware gestalten. Verschiedene Sensoren haben eine stark unterschiedliche Lebensdauer. Bei Volumen und Kosten bestehen ebenfalls große Differenzen. Tabelle 2.2 zeigt, dass der Energieverbrauch einen signifikanten Unterschied zwischen den verschiedenen Arten von Sensorknoten aufweist. Bei einem drahtlosen Sensornetzwerk verbrauchen die Datensendung und Übertragung die meiste Energie. Gewöhnlich erhöht jeder Bestandteil des Sensorknotens durch die Reduktion der Datensendung den Lebenszyklus. Normalerweise hat ein Sensorknoten nur einen Datenkanal. Die hauptsächliche Arbeitsfrequenz liegt auf den ISM-Bändern [53] zwischen 868MHz, 433MHz und 2.4MHz. Die benötigte Energie zur Übertragung von 1bit Daten entspricht der Energie, die 1000 CPU Zyklusberechnungen verbrauchen. Deshalb ist es sinnvoll, durch die Reduktion der Datenübertragung die Lebensdauer des Netzwerks zu verlängern.

Tabelle 2-2 Energieverbrauch der Sensorknoten

Sensortype	Strom (mA)	Energieverbrauch (mJ)
Solarstrahlung	0.350	0.525
Atmosphärischem Druck	0.025	0.003
Feuchtigkeit	0.500	0.500
Oberflächentemperatur	5.6	0.0056
Umgebungstemperatur	5.6	0.0056
Acceleration	0.6	0.0048
Magnetometer	5	0.2595

2.3.2 Kommunikationslayout des drahtlosen Sensornetzes

Bezüglich des Kommunikationsprotokolls des drahtlosen Sensornetzwerks gibt es eine große Menge von Forschungsthemen. Aber es ist möglich, eine Klassifikation auf Knotenebene und Netzwerkebene vorzunehmen. Aus der Sicht des Sensorknotens sind die Forschungsinhalte wie folgt:

1. Low-Power-Medium Access Control (MAC)-Protokoll

Das MAC-Protokoll wird eingesetzt, um die zugrunde liegende Infrastruktur zu kontrollieren und den Kommunikationsprozess der Sensorknoten sowie die Betriebsart zu steuern. Wegen der begrenzten Energie der Knoten und des großen Netzwerks muss das MAC-Protokoll des drahtlosen Sensornetzwerks zunächst die Energieeinsparung und Skalierbarkeit berücksichtigen. Dann sollten Effizienz und Rechtzeitigkeit usw. herangezogen werden [54]. Um Energie einzusparen, sollte das MAC-Protokoll die Arbeit der Knoten im Ruhestand unterstützen. Falls erforderlich, sollten Kommunikationsunit und Teile der Funktionsunit von der CPU ausgeschaltet werden, um den Ruhestand zu erreichen.

2. Energieeffizientes Routing-Protokoll

Das Routing-Protokoll ist verantwortlich für das Bilden und Wählen des Routings und gilt als Kern des Netzwerks, das aus Sensorknoten besteht [55]. Die Eigenschaften des drahtlosen Sensornetzwerks erfordern Routing-Protokolle mit einfachen Berechnungen und niedrigen Speicherkosten. Energieeinsparung ist ein wichtiger Indikator für das Routing-Protokoll des drahtlosen Sensornetzwerks. Im drahtlosen Sensornetzwerk bestimmt das Routing-Protokoll nicht nur den Energieverbrauch eines einzelnen Knotens, sondern auch den gleichmäßigen Energieverbrauch des ganzen Netzwerks, um einen progressiven Energieverbrauch zu verhindern. Darüber hinaus sollte nach den Verteilungsmerkmalen des

Flusses des Netzwerks und den Anforderungen der Anwendungsaufgaben die Routing-Technik optimiert werden, um die Lebensdauer des Netzwerks zu verlängern.

3. Netzwerkabdeckung und -planung

Ein sehr dicht ausgestattetes drahtloses Sensornetzwerk hat viele Redundanzknoten. Es ist ein wichtiges Forschungsgebiet, wie mittels der Redundancy-Eigenschaft der Knoten die Energienutzungseffizienz erhöht und die Lebensdauer des drahtlosen Sensornetzwerks verlängert werden kann. Durch die Kontrolle von Netzwerkabdeckung und -planung wie Node Scheduling ist eine Kontrolle der Dichte erreichbar, ohne die Netzwerkabdeckung zu beeinflussen und die Anzahl der aktiven Knoten im Netzwerk zu reduzieren. Dabei kann die abwechselnde Arbeit von Redundanzknoten realisiert werden, um das Ziel der Verlängerung der Lebensdauer des Netzwerks zu erreichen. Die Netzwerkplanung bezieht sich darauf, wie das drahtlose Sensornetzwerk ausgestattet werden sollte, um die Netzwerkabdeckungsqualität zu optimieren und zu möglichst niedrigen Kosten eine nahtlose und drahtlose Abdeckung zu realisieren.

4. Topologie-Kontrolltechnik

Das Hauptproblem der topologischen Kontrolle ist, wie sich durch die Leistungssteuerung und die Wahl der Hauptkette-Knoten die unnötigen Kommunikationsverbindungen lösen lassen, damit die Netzwerktopologie mehrere Eigenschaften wie Konnektivität aufweisen kann. Das Ziel besteht darin, Signalkonflikte zu reduzieren, den Energieverbrauch der drahtlosen Übertragung zu senken und die Lebensdauer des Netzwerks zu erhöhen. Die topologische Kontrolle kann die Effizienz des Routingprotokolls und MAC-Protokolls steigern und einen Grundstein für Datenfusionen, Uhrensynchronisation und Ziellokalisierung legen. Es ist ein Kernproblem der Forschung im Bereich der drahtlosen Sensornetze.

5. Lokalisierungstechnik

Die Bestimmung des Vorfalles und Sammlung von Datenknoten gehört zu den grundlegenden Funktionen eines drahtlosen Sensornetzwerks. Um effektive Standort-Informationen zu liefern, müssen Sensorknoten selbst ihre Positionen bestimmen können. Da Knoten über Merkmale wie begrenzte Ressourcen, Anfälligkeit für Störungen und die zufällige Verteilung verfügen, muss die Positionierungsmethode die folgenden Bedingungen erfüllen: Selbstorganisation, Robustheit, hohe Energieeffizienz und Distributed Computing [56]. Das aktuelle Hauptproblem ist, wie mit Verkehrszeichen oder ohne Verkehrszeichen

mittels Abstandsbestimmungstechnik die Verteilung von Netzwerksensoren synchron positioniert werden kann, um die absolute oder relative Position zu lokalisieren.

6. Uhrzeitsynchronisationstechnik

Uhrzeitsynchronisation ist ein wichtiger Mechanismus für das drahtlose Sensornetzwerk. NodeScheduling und Zielverfolgung benötigen die Unterstützung durch die Uhrzeitsynchronisation. Kosten, Volumen und Energieverbrauch sind wesentliche Einschränkungsfaktoren für die Uhrzeitsynchronisation des Sensornetzwerks. Aufgrund der hohen Kosten sowie des hohen Volumens und Energieverbrauchs kann die traditionelle GPS-Technologie nicht effektiv für drahtlose Sensornetzwerke verwendet werden. Beim Design der Uhrzeitsynchronisation des drahtlosen Sensornetzwerks müssen Energieeffizienz, Skalierbarkeit, Robustheit und Stabilität berücksichtigt werden.

7. Datenfusionstechnik

Redundante Knoten führen zu redundanten Daten. Durch die Reduzierung von redundanten Daten bei der Übertragung im Netzwerk kann effektiv Energie eingespart werden. Bei der Datensammlung von Sensorknoten ist es möglich, durch ihre eigene Kalkulations- und Speicherfähigkeit eine Datenfusion zu realisieren, um redundante Information zu entfernen. Obwohl die Datenfusionstechnik den Energieverbrauch reduziert, entsteht jedoch eine Übertragungsverzögerung. Entsprechend der Bedürfnisse bestimmter Anwendungen ist die Erforschung von Cache und die Integration von Daten sowie von Methoden zur Datenintegration eine wichtige Forschungsrichtung im Bereich drahtloser Sensornetzwerke.

8. Datenverwaltungstechnik

Aus der Perspektive der Datenspeicherung kann das Sensornetz als verteilte Datenbank betrachtet werden. Das Datenmanagement im drahtlosen Sensornetzwerk ist durch Datenbanken zu realisieren, um die logische Sicht der im Netzwerk gespeicherten Daten und die logische Sicht im Netzwerk zu trennen. In dem Fall brauchen die Nutzer nicht mehr für die Details der Implementierung, sondern nur für die Datenabfrage und logische Struktur zu sorgen. Die Datenverwaltung des drahtlosen Sensornetzwerks unterscheidet sich stark von der traditionellen verteilten Datenbank. Es ist notwendig, gemäß der Merkmale des Sensornetzwerks ein energieeffizientes Design und eine robuste Datenverwaltungstechnik zu schaffen.

9. Netzwerksicherheitstechnik

Das drahtlose Sensornetzwerk als ein aufgabenorientiertes Netzwerk verantwortet nicht nur die Datenübertragung, sondern auch die Datenerfassung und –integration sowie kooperative Steuerung der Aufgaben. Es ist ein zu lösendes Problem für das drahtlose Sensornetzwerk, wie die Vertraulichkeit, Aufgaben umzusetzen, die Zuverlässigkeit der Daten und die Vertraulichkeit der Datenübertragung gewährleistet werden. Um die obigen Ziele zu realisieren, muss das drahtlose Sensornetzwerk mehrere grundlegende Sicherheitsmechanismen erreichen, wie Vertraulichkeit, Message-Authentifizierung, Integrität der Nachrichten-Authentifizierung, Zertifizierungsbroadcast, Sicherheitsmanagement usw.. Obwohl auf dem Gebiet der Sicherheitsforschung nicht so viele Informationen über drahtlose Sensornetze verfügbar sind, resultieren aus den Merkmalen von drahtlosen Sensornetzen große Unterschiede der Forschungsmethoden und der Mittel für die Umsetzung zwischen einem drahtlosen Sensornetzwerk und einem traditionellen Sicherheitsnetzwerk.

10. Applikations-Layer Technik

Der Applikations-Layer des drahtlosen Sensornetzes besteht aus Softwaresystemen. Die Forschung im Bereich Applikations-Layer bezieht sich vorwiegend auf die Koordination zwischen der Entwicklung verschiedener Anwendungssysteme und Multi-Tasking. Zudem behandelt die Forschung zur Anwendung von drahtlosen Sensornetzen Programmiersprachen für drahtlose Sensornetze, die Methodik der Programmierung, Software Entwicklungsumgebungen und Tools, Software-Testing-Tools, ein anwendungsorientiertes System, Netzwerktests und Konfigurationsmanagementtools.

2.3.3 Drahtlose Kommunikationstechnik für den Nahbereich

Es gibt verhältnismäßig viele drahtlose Kommunikationstechnologien für den Nahbereich, welche für drahtlose Sensornetze verwendet werden können. Die entscheidende Frage bleibt die Auswahl einer Technologie, welche sich für das Telemonitoring eignet. Telemonitoring-Sensornetze stellen verhältnismäßig strikte Ansprüche bezüglich Kosten, Größe und Stromverbrauch. Unter gewöhnlichen Bedingungen entscheiden die Kosten in einem hohen Maße über die Verbreitung eines Systems. In diesem Abschnitt werden entsprechend der Besonderheiten des Telemonitorings häufig anzutreffende Technologien der drahtlosen Kommunikation im Nahbereich bezüglich Abdeckungsbereich, Anzahl der möglichen Knoten, Komplexität der Technik, Sicherheit und Kosten etc. verglichen.

2.3.3.1 Infrarot-Technologie

Infrarot-Technologie verwendet Infrarotlicht, um Daten und Informationen zu übermitteln. Diese Technologie kommt ohne spezielle Lizenz für eine spezifische Frequenz aus und besitzt die notwendigen Vorzüge für die mobile Datenübertragung, wie geringe Größe und einen niedrigen Stromverbrauch. Sie ist nicht von Funkstörungen betroffen und garantiert so eine sichere und effiziente Datenübertragung. Die Geschwindigkeit der Datenübertragung ist relativ hoch und erreicht bis zu 16Mbps. Trotz der genannten Vorteile ist diese Technologie für den Einsatz in einem intelligenten Wohnraum ungeeignet. Zunächst einmal kann sie nur eine Punkt-zu-Punkt-Datenübertragung leisten und ist nicht in der Lage, an mehrere Punkte zu übertragen. Darüber hinaus ist die Infrarot-Technologie für den Einsatz auf Sichtweite vorgesehen, d.h. es dürfen keine Hindernisse die Datenübertragung behindern, und der effektive Sendebereich liegt zwischen null und etwa einem Meter. Diese beiden Punkte schränken den Einsatz der Technologie für Heimnetzwerke stark ein.

2.3.3.2 Bluetooth

Bluetooth ist eine Radiofrequenztechnologie, welche auf dem lizenzfreien ISM-Band mit 2,4 GHz sendet. Bluetooth eignet sich für die drahtlose Verbindung im Nahbereich von verschiedensten fixen und mobilen Geräten. Im Wesentlichen soll Bluetooth als Kabelersatz fungieren. Bluetooth ermöglicht mittels des Multiplexverfahrens eine vollständige Duplex-Kommunikation und die Übertragung von Ton und Daten in alle Richtungen mit einer maximalen Übertragungsgeschwindigkeit von 1 Mbps. Das Frequenzsprungverfahren ist eine Schlüsseltechnologie von Bluetooth. Durch die hohe Rate, mit welcher die Frequenzen gewechselt werden, besitzt Bluetooth eine höhere Resistenz gegenüber Störungen als andere Kommunikationstechnologien. Es können maximal 8 Geräte in einem Bluetooth-Netzwerk verbunden werden, die Anzahl der Knoten ist relativ gering, weswegen sich diese Technologie nur schlecht für einen intelligenten Wohnraum eignet. Der ideale Bereich für eine Verbindung mittels Bluetooth liegt bei 10 cm bis 10 m. Der Sendebereich ist sehr gering und die Leistung durch Wände hindurch ist sehr schwach, besonders im Fall von Stahlbeton ist der Sendebereich ungenügend. Darüber hinaus ist der Preis eines Bluetooth-Chips hoch, woraus sich hohe Systemkosten ergeben, was eine eingeschränkte Verbreitung zur Folge hat. Das Auftauchen der Bluetooth-Technologie hat drahtlose Kommunikation im Nahbereich ermöglicht, aber Eigenschaften wie das verhältnismäßig komplizierte Protokoll und der relativ hohe Stromverbrauch machen sie ungeeignet für Haushaltsnetzwerke, welche geringe Kosten und einen geringen Stromverbrauch benötigen.

2.3.3.3 ZigBee

ZigBee ist eine Drahtlosnetzwerk-Technologie für den Nahbereich mit geringer Komplexität, niedrigem Stromverbrauch und geringen Kosten. ZigBee basiert auf dem Standard IEEE802.15.4[60] und operiert hauptsächlich auf dem lizenzfreien 2,4 GHz-Band. Die Übertragungsgeschwindigkeit ist niedrig und beträgt auf dem 2,4 GHz-Band 250 Kbps, die Übertragungreichweite beträgt 10 – 75 m, die klassische Entfernung beträgt 30 m. Das ZigBee-Netzwerk ist sehr flexibel und kann bis zu 254 Slave oder Master-Geräte aufnehmen. Technische Eigenschaften wie geringe Zeitverzögerung, geringer Stromverbrauch, niedriger Preis, hohe Leistungsfähigkeit, Zuverlässigkeit und Sicherheit eröffnen zahlreiche Einsatzmöglichkeiten für die ZigBee-Technologie. Obwohl ZigBee in der Konzeption von intelligentem Wohnraum oft genannt wird, sind in der Praxis selten mehr als 10 Knoten in einem Drahtlosnetzwerk zu finden, weswegen sich ZigBee bis heute noch nicht durchgesetzt hat und nur bei einigen Firmen getestet wird.

2.3.3.4 WiFi

WiFi ist die Abkürzung von Wireless Fidelity. Die Technologie verwendet Frequenzen nahe am 2,4 GHz-Band. Diese Frequenzen sind derzeit noch lizenzfrei. Es gibt zur Zeit zwei Standards unter dem IEEE802.11 Standard, IEEE802.11a und IEEE802.11b. WiFi ist im Grunde genommen zur hauptsächlichsten Norm von WLAN geworden. Die Qualität der drahtlosen Kommunikation der WiFi-Technologie ist nicht besonders gut und die Datensicherheit weist Mängel auf, allerdings ist die Übertragungsgeschwindigkeit sehr hoch und erreicht bis zu 11 Mbps. Der effektive Sendebereich liegt bei 100 m.

Aus dem Vergleich dieser häufig vorkommenden Technologien für drahtlose Kommunikation wird ersichtlich, dass bei jeder Technologie einige Gesichtspunkte für die Eignung für einen intelligenten Wohnraum sprechen, allerdings gleichzeitig andere Gesichtspunkte dagegen.

2.3.3.5 ANT

ANT— ist ein auf 2.4 GHz arbeitendes, bidirektionales Kommunikationsprotokoll für Wireless Sensor Networks. ANT passt bezüglich Geschwindigkeit, Zeitverzögerung, Stromverbrauch, Kosten und Kapazität im Vergleich besser zur Anwendung im Bereich des Telemonitoring. Tatsächlich ist ANT im individuellen Fitnessbereich bereits der drahtlose Standard [57].

Das ANT-Netzwerk ist von Nordic Dynastream angeregt und vorangetrieben worden. Es handelt sich um ein Drahtlosnetzwerk mit sehr geringem Stromverbrauch, welches auf die bestmögliche Ressourceneffizienz bei niedrigst möglichem Energieverbrauch abzielt. Die Technologie wird für drahtlose individuelle LAN (IVN) verwendet. ANT verwendet eine Duplex-Kommunikation über das 2,4 GHz-Band, adaptive synchronisierte Kanäle und PTP und PTMP als Übertragungsmethode. Es besitzt eine Air Data Rate von 1 Mbps und eine Netzwerk-Datenübertragungsgeschwindigkeit von 20 Kbps. Der Sendebereich beträgt 10 bis ungefähr 30m. Es werden viele Frequenzen und umfangreiche Netzwerke unterstützt. Typische Anwendungsszenarien umfassen beispielsweise Sensornetzwerke, Fernsteuersysteme und intelligente Wohnräume.

2.3.3.6 Vergleich und Analyse

Daneben gibt es noch Kommunikationstechnologien wie RuBee, Zarlink und Sensium, welche gegenwärtig verwendet werden, aber nicht sehr verbreitet sind [58]. RuBee verwendet dabei eine Duplex-Kommunikation mit langwelligen Magnetsignalen, 128 Byte und dynamischem Wireless-Protokoll. Es eignet sich für Bestandsanwendungen in Echtzeit und ungünstige Umgebungen und funktioniert sogar in Umgebungen in unmittelbarer Nähe zu Wasser, Metall oder elektromagnetischen Störungen. Sensium bietet eine Plattform, welche auf verschiedene Anwendungen zum menschlichen Körper mit niedriger Übertragungsgeschwindigkeit und extrem niedrigem Stromverbrauch abzielt (3n1A, 1.2v). Darüber hinaus zielt Zarlink auf Kommunikationstechnik für medizinische Implantate und verwendet eine Reed-Solomon-Kodierung und CRC-Prüfung, um einen sicheren Telelink zu garantieren. Die Funktionalität wird über einen speziellen Sender-Chip mit niedrigem Energieverbrauch erreicht. Das Wireless-Modul der ersten als Kapsel oral einnehmbaren Kamera basiert ebenfalls auf dem Sender-Chip-Design von Zarlink. Die ANT-Technologie verfügt über die folgenden drei großen Vorteile:

Tabelle 2-3 Vergleich der Nahbereich-Kommunikationstechnik

Standard	WIFI	Infrarot	Bluetooth	Zigbee	ANT
Betriebsfrequenz	2.4GHz	38kHz	2.4GHz	2.4GHz	2.4GHz
Max. Geschwindigkeit	11 Mbps	16Mbps	1Mbps	250Kbps	1Mbps
Effektive Reichweite	100m	1m	5-10m	10m	10-30m
Kosten (\$)	hoch	<5	20-25	3	<5
Durchschnittlicher	100-300mW	20mW	30mW	9mW	115uW

Stromverbrauch					
Anzahl der Knoten	30	2	8	2¹⁶	2¹⁶
Netzwerk-Latenz	3s	20-40ms	3-10s	15-30ms	<15-30us
Protokoll Stapel Bytes	>200k	1k	250k	100k	2k
Übertragung	1to1,1toN	1to1	1to1,1toN	1to1,1toN	1to1,1toN

Äußerst niedriger Stromverbrauch: Ein drahtloses System einer ANT-Lösung verfügt normalerweise über einen Standbybetrieb und einen Ruhezustand. Den größten Teil der Zeit befindet sich das System im Ruhezustand mit dem niedrigsten Stromverbrauch, dieser liegt dann nur im Bereich von Mikroampere. Gleichzeitig ist die Zeitverzögerung bei ANT nur kurz und es besitzt eine schnelle Reaktionsgeschwindigkeit. Das System kann bei Bedarf schnell den Ruhezustand verlassen und in kürzester Zeit Übertragungen durchführen. Danach kehrt es schnell wieder in den Ruhezustand zurück, wodurch es durchschnittlich nur eine Leistung von einigen Mikroampere benötigt. Der mittlere Energieverbrauch von ANT liegt bei 115 uW, wogegen der durchschnittliche Energieverbrauch von ZigBee bei 9 mW und damit deutlich höher liegt. Im Vergleich zu den konkurrierenden Konzepten verfügt ANT also über einen äußerst niedrigen Energieverbrauch. Vergleichbare Technologien halten bei ähnlicher Anwendungsintensität nur einige Monate oder sogar kürzer. Während ANT also beispielsweise mit einer Knopfatterie 5 Jahre läuft, verbraucht sich diese bei Bluetooth innerhalb von 2 bis 5 Monaten. ZigBee funktioniert mit zwei AA-Batterien für etwa 6-24 Monate.

Niedrige Kosten: Zunächst einmal operiert ANT auf dem lizenzfreien 2,4 GHz-Band, was zusätzliche Kosten vermeidet. Darüber hinaus benötigt das ANT-Protokoll nur geringe Ausgaben. Das Bluetooth-Protokoll benötigt 250.000 Byte, ZigBee 100.000 Byte, ANT dagegen lediglich 2000 Byte. Außerdem sind die Kosten der bereits der Öffentlichkeit präsentierten ANT-Produkte mit einem ANT-Wireless-Chip und die direkt darauf zurück greifenden Module bisher allesamt sehr preisgünstig, was die Einrichtungs- und Nutzungskosten für die Anwender reduziert und gleichzeitig auch die Kosten der Anwender für die Entwicklung und Produktion von Produkten senkt.

Einfache Bedienung: ANT kann durch die einfache Konfiguration der Knoten von Arbeitskräften ohne besondere Qualifikation eingerichtet werden, wo zuvor ein Programmierer mit Konfiguration, Test und Debugging beauftragt werden musste. Dies

reduziert die Hürden für den Einsatz der Technologie und verkürzt die benötigte Zeit für die Einrichtung.

ANT ist bei drahtlosen Übertragungen im Nahbereich ZigBee deutlich überlegen: (1). Die Qualität der Module von ANT, der Firmware-Support und das Aufrufen von Dateien erreichen in der Praxis bessere Resultate; (2). ANT-Module benötigen nur 60 % der Größe der ZigBee-Module; (3). ANT-Module benötigen eine geringe Leistung, um eine sehr hohe Datenübertragungsgeschwindigkeit zu realisieren.

2.4 Hauptanwendungsbereiche von drahtlosen Sensornetzen in medizinischen Versorgungssystemen

In der medizinischen Betreuung gibt es verschiedene Instrumente zur Informationserfassung, unter gewöhnlichen Umständen lassen sich die Instrumente bezüglich ihrer Handhabung in der medizinischen Betreuung in zwei Kategorien einteilen: (1) die erste Kategorie sind Geräte, welche nur von Ärzten oder Fachpersonal in einem Krankenhaus bedient werden können; (2) die andere Kategorie kann von Durchschnittspersonen zuhause bedient werden oder unter Verwendung von medizinischen Fernbetreuungssystemen betreut werden, welche physiologische Informationen sammeln und an den Arzt senden. Sensoren werden am Patienten angebracht und übermitteln Informationen an die Überwachungsgeräte. Komplizierte Geräte, viele Verbindungen können Patienten nervös und angespannt machen, wodurch möglicherweise der Gesundheitszustand des Patienten beeinflusst werden kann. Die in der Diagnose erhaltenen Informationen können daher vom tatsächlichen Zustand abweichen, was dem Patienten und dem medizinischen Personal Probleme bereiten und die Richtigkeit der Diagnose beeinflussen kann. Gleichzeitig kann bei einer notwendigen häufigen Erhebung von physiologischen Daten (z.B. bei chronisch Kranken oder alten Patienten) die Erhebung zuhause stattfinden. Die Messungen können so in einem vertrauten und bequemen Umfeld stattfinden, wobei der Einsatz kabelloser Geräte die Vernetzung zwischen medizinischen Sensoren und Überwachungsgeräten fördert und bei der Überwachung verhältnismäßig große Freiheiten einräumt, wodurch einerseits präzisere Messungen möglich sind und andererseits dem Patienten auch ein unnötiger Gang zum Krankenhaus erspart wird.

Gleichzeitig können auf einer Station eines Krankenhauses zahlreiche Analysen über das kabellose Überwachungsnetzwerk mit den Krankbetten ausgetauscht werden. Hierdurch können die Behandlung des Patienten enorm vereinfacht sowie das Informationsmanagement

modernisiert und die Arbeitseffizienz des Krankenhauses gesteigert werden. Darüber hinaus kann das Telemonitoring-System auf das Personal abgelegener Krankenhäuser und anderer medizinischer Einrichtungen ausgeweitet und so jederzeit die notwendige medizinische Behandlung geboten werden. Des Weiteren ist es möglich, wenn erforderlich, Fernkonsultationen vorzunehmen.



Abbildung 2-4 Überwachung der physiologischer Daten in Echtzeit [59]

Die Anwendung von drahtlosen Sensornetzen im Gesundheitsbereich beinhaltet im Wesentlichen: Überwachung von Kranken, Diagnose, Verwaltung der Medikamente des Krankenhauses, Beobachtung von Schädlingen und kranken Tieren sowie Fernüberwachung von physiologischen Daten. Mittels der Verbindung drahtloser Kommunikationsmedien werden physiologische Daten an eine medizinische Datenbank übermittelt und den Kranken Freiheiten verschafft, gleichzeitig ist eine Überwachung der Daten in Echtzeit möglich (Abbildung 2-4). Bei der medizinischen Behandlung und Überwachung ist die Vollständigkeit und Sicherheit individueller medizinischer Daten ein wesentlicher zu berücksichtigender Faktor. Diese Arbeit erörtert einen Sicherheitsrahmen für ein medizinisches Informationssystem und betrachtet eventuelle Behinderungen von Signalen oder einen möglichen „Denial of Service“ einer solchen medizinischen Behandlungsumgebung genauer.

Die wesentlichen Anwendungsbereiche von drahtlosen Sensornetzwerken in medizinischen Überwachungssystemen sind folgende:

(1) **Entwicklung neuer Medikamente.** Am Körper des Patienten werden Sensoren zur Überwachung von Körpertemperatur, Atmung, Blutdruck etc. angebracht. Der Arzt kann sich über Telemonitoring über den Zustand des Patienten informieren. Die vom Sensornetz über einen längeren Zeitraum gesammelten physiologischen Daten werden ausgewertet und

möglicherweise noch eine DNA-Analyse vorgenommen. Diese Daten sind für die Entwicklung neuer Medikamente von großer Wichtigkeit.

(2) **Überwachung älterer Patienten.** Interne Sensoren können die Tätigkeiten von älteren Menschen überwachen und so beispielsweise auf Stürze aufmerksam machen. Diese winzigen Sensoren erlauben den beobachteten Zielen einen höheren Grad an Freiheit und ermöglichen dem Arzt, anfängliche Symptome zu bestimmen. Gegenüber einem medizinischen Zentrum bleibt so der beobachteten Person ein höherer Grad an Lebensqualität erhalten.

An der Harvard University wurde das Projekt Codeblue [60] gestartet, bei welchem ein Sensornetzwerk mit Hardware und Software aufgebaut wird, welches Patienten medizinische Betreuung in Echtzeit bietet. Codeblue basiert auf medizinischen Sensoren und publish/subscribe Routing und ist in der Lage, medizinische Überwachung und Betreuung in Echtzeit zu leisten. Die medizinischen Sensoren bei Codeblue verwenden hauptsächlich Micaz und Telosmote Software-Designs und beinhalten im Wesentlichen EKG-Geräte, Blutdruckgeräte und Beschleunigungssensoren. (Abbildung 2-5)

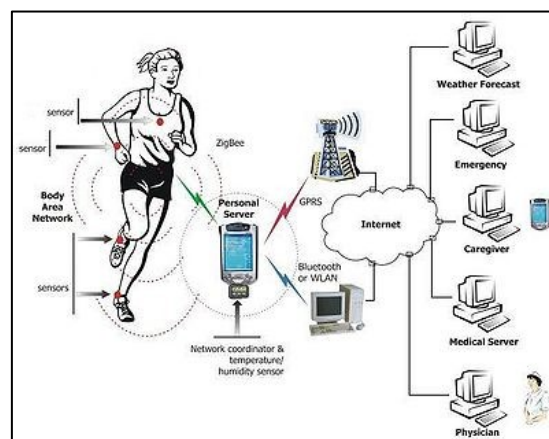


Abbildung 2-5 Anwendungsszenarien der drahtlosen Sensornetze in medizinischen Versorgungssystemen [61]

(3) **Notfallmedizin.** Bei plötzlich eintretenden gesundheitlichen Notfällen ist die Rettungszeit der wichtigste Faktor. Bei der medizinischen Notfallhilfe ist insbesondere die Zuverlässigkeit von Software und Hardware ein wichtiger zu berücksichtigender Faktor. Das System muss zuverlässig arbeiten und bequem vom Benutzer verwendbar sein. Über ein drahtloses Netzwerk sind medizinische Sensoren und Software verbunden und können dem Patienten medizinische Notfallhilfe bieten. Die Johns Hopkins University hat drei Technologien vereint, um ein System für Gesundheitsschutz und Katastrophenhilfe zu entwickeln, welches im Notfall Patienten eine rasche medizinische Behandlung bietet [62]. Dieses System eignet sich für die Behandlung und Rettung von schwerkranken Patienten und für schnelle Hilfe bei großen Opferzahlen im Katastrophenfall.

(4) **Heimpflege.** Die Heimpflege ist in den USA ein diversifiziertes Dienstleistungsgewerbe, welches verschiedenste Formen der Heimpflege als Dienstleistung anbietet. Die Pflegebedürftigen können dabei normale Kranke, Behinderte oder unheilbar erkrankte Patienten sein. Intel bietet derzeit ein entsprechendes System an. Dieses System ist ein Projekt und soll Technologien bieten, welche als Reaktion auf die Alterung der Gesellschaft verstanden werden. Es stattet Schuhe, Möbel und Heimelektrik mit Sensoren aus. Dieses System kann Kranken, älteren Menschen und Behinderten ein eigenständiges Leben ermöglichen, wobei im Bedarfsfall Hilfestellung durch medizinisches Personal oder Individualfürsorger erfolgt. Die Forscher entwickelten ein drahtloses Sensornetz auf der Grundlage von mehreren Beschleunigungssensoren, um körperliche Bewegungsabläufe, wie Sitzen, Stehen, Liegen, Gehen, Fallen oder Kriechen zu überwachen. Das System verwendet verschiedene Sensorknoten, welche an verschiedenen spezifischen Stellen am Körper angebracht werden. Das System erstellt in Echtzeit dreidimensionale Bewegungsinformationen. Es fehlt nicht mehr viel, dann kann dieses System zu einer Hilfe für ältere Menschen und Patienten werden, welche sich nicht frei bewegen können. Gleichzeitig kann das System auch in Rehabilitationszentren eingesetzt werden, wo der Fortschritt der Erholung der Patienten präzise erfasst werden kann, wodurch sich wertvolle Anhaltspunkte für den Erholungsplan der Patienten ergeben.

Das Gesundheitssystem kann durch Telemonitoring die Kommunikation von Informationen und die medizinischen Dienstleistungen verbessern. Durch die Kombination mit einer drahtlosen Sensornetz-Technologie kann Telemonitoring die Arbeit des medizinischen Personals erleichtern, korrigieren und die neuesten medizinischen Daten bieten. Unterschiede im Niveau der Dienstleistung können behoben werden. Domenico Zito [63] verwendet eintragbares System zur Überwachung der Herz-Lungen-Funktion von Patienten. Dieses System besteht aus Sensoren zur Überwachung von Herzschlag und Atmung. Es ist ausgestattet mit einem ZigBee Interface mit niedrigem Stromverbrauch nach IEEE802.15.4 Standard. Über das ZigBee Interface wird die Datenerfassung per Fernzugriff ermöglicht. Dagtas et al. schlagen vor, ein drahtloses Sensornetz mit ZigBee Protokoll zu verwenden, um eine sichere Grundstruktur für die drahtlose Gesundheitsüberwachung in Echtzeit zu schaffen. Diese Grundstruktur verwendet ein BAN um Signale zu messen, Daten zu übertragen und Kanäle zuzuweisen. Obermiller verwendet allgemein passende Technologien, um ein Schlafmessungssystem aufzubauen, welches erlaubt, durch die Messung der Werte während des Schlafs den Gesundheitszustand des Beobachteten zu diagnostizieren. Rall et al. schlagen ein adaptives Heim-System vor, welches wirksam die Aktivitäten älterer Menschen

überwachen und aus den gesammelten Sensordaten eine Entwicklung erstellen kann. Youcef Rahal [64] et al. empfehlen ein intelligentes Heimsystem auf der Grundlage von Positionsbestimmungen, welches den bayesschen Filter verwendet sowie drahtlose Sensornetztechnologie, um allein wohnende ältere Menschen zu überwachen. Anciaux hat zu gewöhnlicher Datensicherheit und Fragen der Privatsphäre bei Telemonitoring geforscht, mittels mobiler Geräte wurden medizinische Daten des Patienten zurückgegeben [65].

3 Anforderungsanalyse

In diesem Kapitel wird im Wesentlichen die grundlegende Architektur eines mHealth-Systems vorgestellt. Sämtliche Anforderungen in einem Modell darzulegen, könnte die Komplexität des Entwurfsprozesses erhöhen und bringt Konflikte und Widersprüche mit sich. Entsprechend der modellbasierten Entwicklung wird in dieser Arbeit der Systementwurf in verschiedene Subsysteme unterteilt. In diesem Kapitel werden ebenfalls von diesem Standpunkt aus die gesamten Anforderungen des Systems geschildert. Diese basieren auf den Business-Anforderungen des oberen Applikation Layer, technischen Anforderungen des unteren Layers sowie den Anforderungen der Kommunikationsschnittstellen zwischen den Subsystemen.

3.1 Business-Subsystem

Das in dieser Arbeit entworfene mHealth-System (siehe Abbildung 3-1) besteht aus drei Subsystemen: dem Mobile-Erfassungsclient (IPM-MC), dem Monitoring-Zentral-Server System (IPM-ZS) und dem Auswertungsmodul (IPM-AM).

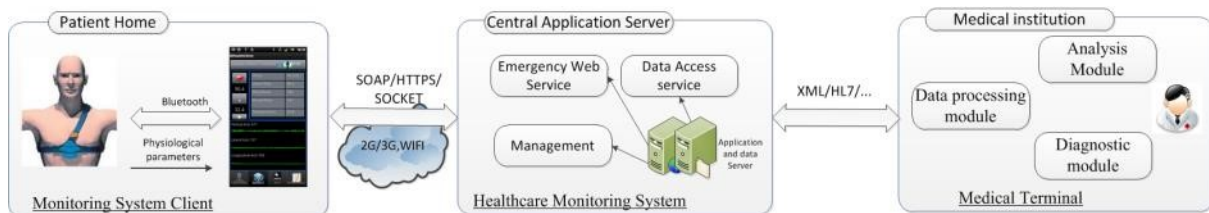


Abbildung 3-1 Systemarchitektur des IPM-mHealth Systems

3.1.1 Mobiler Erfassungsclient

Das Projekt IPM-mHealth umfasst ein Langzeit-Telemonitoring, welches es dem medizinischen Personal gestattet, den Gesundheitszustand von Patienten aus der Entfernung zu kontrollieren. Der mobile Erfassungsclient wird beim Patienten eingerichtet, er beinhaltet das in Kapitel 2 vorgestellte medizinische drahtlose Sensor-Netzwerk und ein lokales Gateway (Controller Unit, meist dient das Smartphone als CU). Diese beiden sind wechselseitig miteinander verbunden. Die CU hat die Aufgabe, medizinische Daten von Sensoren zu empfangen, zu analysieren und über GPRS, 2G oder 3G-Mobile-Netzwerk die Daten auf den Zentralserver weiterzuleiten [66]. Um das korrekte Einrichten und die korrekte Funktionsweise des Monitoring-Systems zu garantieren, muss das IT-Personal am Anfang die grundlegende Einstellung für die CU entsprechend der Sicherheitsstrategie und der

Bedürfnisse von Arzt und Patient einstellen. Danach kann diese Konfiguration auch durch Synchronisieren mit dem Zentral-Server-System wieder aktualisiert werden.

3.1.2 Fernüberwachungszentrale

Die Aufgabe eines mHealth-Servers umfasst, welche physiologischen Parameter im System verwaltet werden, wie und wo diese Daten abgespeichert werden, wie die Sicherheit der Daten gewährleistet wird (wie ein Zugriff für berechtigte Nutzer auf Daten möglich ist, bei gleichzeitigem Schutz der Daten) und schließlich, wie das System flexibel und schnell an die Anforderungen der Gesundheitsversorgung angepasst werden kann. Die Antwort auf diese Fragen geht einher mit der Organisation des Gesundheitsversorgungssystems, dem Unterschied der Erfahrung des IT-Personals und den Anforderungen der Nutzer. Gezielt auf diese Fragen hin wurden bereits viele verschiedene Lösungskonzepte in Betrieb genommen, allerdings zwingt die Entwicklungstendenz in diesem Bereich die Entwickler dazu, ein offeneres Konzept mit besserer Interoperabilität und Kompatibilität zu bieten. Der Lösungsansatz in dieser Arbeit berücksichtigt diesen Punkt und bietet ein offenes und erweiterbares Lösungskonzept, welches in verschiedenen Monitoring -Zentren eingesetzt werden kann.

Um einen medizinischen Dienst anzubieten oder zu verwenden, müssen die Subsysteme in der mHealth-Plattform mit einem Monitoring-Zentrum verbunden werden. Dieser Dienst ist mit Schnittstellen festgelegt, allerdings verwenden verschiedene Anbieter von Diensten bei der konkreten Umsetzung verschiedene Technologien. Um die Informationen auszutauschen, wird eine verschlüsselte XML(SOAP, HL7Message)-Schnittstelle für die Interaktion verwendet [67]. Der Grund für die Auswahl von XML liegt ebenfalls darin, dass XML heute breite Anwendung findet und die Struktur flexibel ist. Andere Anwendungen können direkt und über die Anfrage von Diensten mit den Informationen der Datenbank verbunden werden. Die anonyme Übermittlung von Nachrichten garantiert ebenfalls die Sicherheit der Daten.

3.1.3 IPM-Auswertungsmodule

Am anderen Ende des mHealth-Systems stehen verschiedene medizinische Diagnosemodule zur Verfügung. Diese Module verbinden sich mit dem mHealth Zentral-Server über ein spezielles Protokoll, die Daten-Schnittstelle, und die Kommunikationstechnik. Diese werden als mHealth- Auswertungsmodule (IPM-AM) bezeichnet.

Das IPM-AM ist eine Dienst-Plattform, welche entsprechend der Anforderungen des medizinischen Personals die Dienste verwaltet und organisiert. Es ist für folgende Dienste verantwortlich: Erzeugen formatierter Informationen (wie Formulare, Icons, Warnungen, Berichte, etc.), Datenanalyse, Auswertung oder Diagnose etc. An dieser Stelle interessieren vornehmlich der Datenaustausch mit dem Monitoring-Zentral-Server und die Schnittstelle, welche das Monitoring-Zentrum den Auswertungsmodulen bietet bzw. von diesem diesbezüglich anfragt wird.

3.2 Integration der Subsysteme

Die oben vorgestellten drei Subsysteme bilden zusammen ein vollständiges mHealth-Telemonitoring-System. Bei der Verbindung der einzelnen Teile gibt es unterschiedliche Anforderungen. Beispielsweise wird die Kommunikation von Messegeräten und Mobile Client über das drahtlose Nahbereich-Protokoll des BAN-Netzwerks erreicht. Die Verbindung von IPM-ZS und IPM-AM dagegen fällt völlig anders aus. Hier ist ein deutlich komplexeres Szenario nötig: Die Identität des Benutzers muss überprüft werden, die internen und externen Kanäle für die Übertragung müssen geschützt werden etc. Die Verbindung zwischen den Dienstmodulen, die genauere Definition der allgemeinen Merkmale und Schnittstellen zur Umgebung müssen alle gleichermaßen in der Architektur klar erläutert werden.

Aus der Analyse des oberen Abschnittes lässt sich erkennen, dass drei Kommunikationspfade in den oben genannten drei Subsystemen bestehen: Die Nahbereich-Kommunikation des mobilen Erfassungsclients, die Fernverbindung zwischen *IPM-MC* und *IPM-ZS* und die medizinische Dienstleistungskommunikation zwischen *IPM-ZS* und *IPM-AM*.

3.2.1 Nahbereich-Kommunikation des mHealth-Clients

In der physischen Umgebung des Patienten werden zwei Anlagen zum Aufnehmen der Datensets benötigt: Das Body Area Network (BAN) und das lokale Gateway (Mobile Client).

Body Area Netzwerk

Wie im zweiten Kapitel bereits detailliert vorgestellt, handelt es sich beim BAN um ein drahtloses Sensornetzwerk. Im BAN sorgt ein Smartphone-Knoten im Innen- und Außenbereich für einen festen Monitoring-Bereich. Das BAN hat die Aufgabe, Sensordaten zu sammeln. Über drahtlose Sensoren auf dem Körper des Patienten werden physiologische Informationen (wie z.B. Herzfrequenz, Atemfrequenz, Temperatur etc.) gesammelt und danach an das Smartphone weitergeleitet. Das Smartphone dient hier als ein lokaler Gateway.

Es leitet die physiologischen Informationen dann eine Ebene höher weiter. Die im BAN gesammelten Daten werden letztlich zum Server gesendet.

Lokales Gateway

Das lokale Netzwerk-Gateway (IPM-MC) dient hauptsächlich der Kommunikation des lokalen Netzwerks mit dem Telemonitoring-Zentrum. Nachdem das lokale Netzwerk Daten mittels eines in Kapitel 2 beschriebenen drahtlosen Nahbereich-Kommunikationsprotokolls wie Bluetooth oder ANT empfangen hat, erfolgt eine Voranalyse der physiologischen Daten und eine Protokoll-Konvertierung (weiterführende Details werden in Kapitel 4 erläutert). Die Auswahl des Übertragungsprotokolls ist dabei sehr wichtig. Es kann möglich sein, dass Patienten für dieselben physiologischen Parameter unterschiedliche Sensoren verwenden oder mit verschiedenen Arten von Sensoren (ANT sensor, Bluetooth sensor) mit einem spezifischen IPM-MC verbunden werden. Danach wird der gegenwärtige Zeitpunkt hinzugefügt, alles im Datenformat des Applikation Layer verpackt und an das IPM-ZS weitergeleitet. Das lokale Netzwerk-Gateway empfängt auch die Warnhinweise und Anweisungen des mHealth-Zentrums und kann ggf. die Anweisungen des mHealth-Zentrums über Nahbereich-Kommunikationstechnologien an die Messelemente weiterleiten. Mit diesen Anweisungen können beispielsweise Informationsarten oder die Frequenz der Informationserfassung geändert werden. Der lokale Netzwerk-Gateway ist ein zentraler Punkt der Informationsübermittlung und erfüllt eine wichtige Brückenfunktion, er bietet auch die Möglichkeit der Umwandlung von Protokollen und Anweisungen, sowie Datenverarbeitungsmodule (siehe Abbildung 3-2).

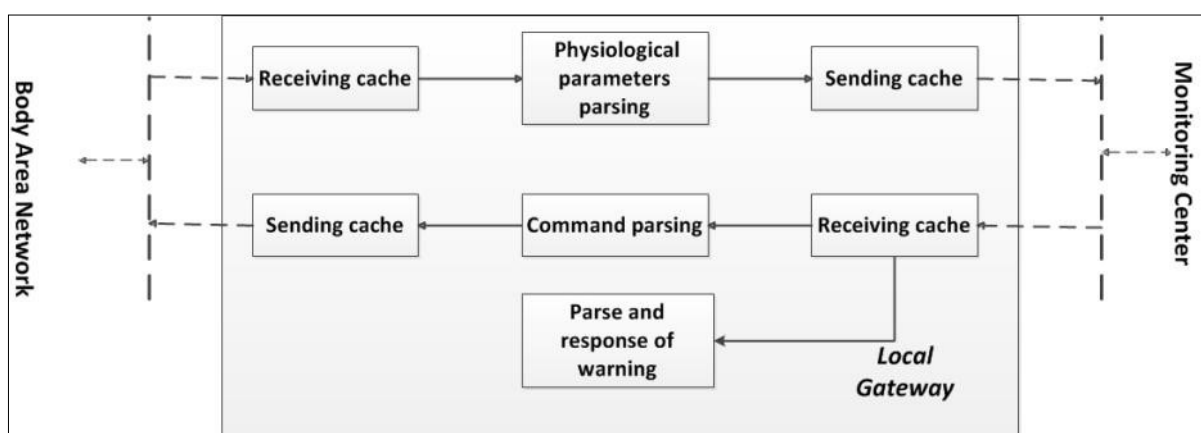


Abbildung 3-2 Lokales Netzwerk-Gateway

3.2.2 Fernmeldeverbindung

Der zweite Kommunikationspfad wird hauptsächlich zur Sammlung und zum Austauschen der Informationen verwendet, er ist zuständig für die Übertragung oder den Empfang der

Informationen zwischen Patient und IPM-ZS (siehe Abbildung 3-3). Wie bereits zuvor erwähnt, werden die wichtigen Daten des Patienten lokal gesammelt und in der IPM-MC abgespeichert. Die Datenbank des IPM-ZS empfängt diese Daten und versorgt das medizinische Personal damit. Das TEC („Telecommunication“) wird zwischen IPM-MC und IPM-ZS errichtet. Es handelt sich um einen sicheren Kommunikationskanal, welcher die sensiblen Daten schützt. Einige Technologien der Umsetzung sind bereits enthalten, mittels des FTP des VPN, HTTP/HTTPS oder TCP-SOCKET etc. können sie für die Übertragung der Daten verwendet werden [68]. Jede Methode hat ihre eigenen Vor- und Nachteile. Die konkreten Umsetzungsmethoden werden in Kapitel 6 detailliert analysiert.

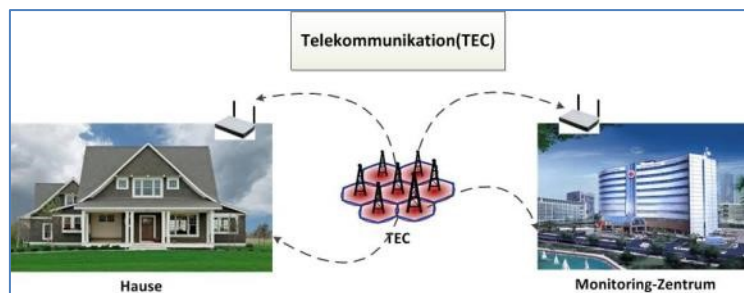


Abbildung 3-3 Fernverbindung zwischen Patientengeräten und dem Monitoring-Zentrum

3.2.3 Datenschnittstelle

Bei der Krankenbetreuung zuhause müssen medizinische Dienstleistungen stets mit der Datenanalyse und dem Gesundheitszustand des Patienten koordiniert werden. Spezifisch autorisierte medizinische Fachkräfte können auf die Daten des Patienten in der medizinischen Datenbank zugreifen und den Gesundheitszustand des Patienten einsehen. Die medizinischen Analysemodule wandeln diese Daten in verständlichere Informationen um und bieten eine notwendige Basis für die Diagnose. Dabei muss gesichert werden, dass diese Anwendungen stets ungehindert und reibungslos mit dem Server verbunden sind und die vom Überwachungszentrum gebotenen verschiedenen Dienste (z.B. Datenabfrage, Sicherheit etc.) anfragen können. Über das Protokoll müssen entsprechend der festgelegten offenen Standards auch strikte Sicherheits-Prozesse implementiert werden.

3.3 Entwurf der Schnittstellenfunktionen zwischen den Systemmodulen

Ein Telemonitoring-Prozess (Aktion) besteht aus einer Sequenz von separaten, kleinen logischen Einheiten (oder Aktionen). Diese kleinen Einheiten müssen für das Erreichen des Gesamtziels notwendigerweise zusammenarbeiten. Die Definition der Schnittstellen zwischen diesen kleinen Einheiten erfolgt nach den jeweiligen Eigenschaften der Anwendungsarchitektur. In dieser Architektur bildet eine ausgewählte logische Einheit

(bezeichnet als Task) einen medizinischen „Dienst“. In diesem Abschnitt werden die verschiedenen Spezifikationen bezüglich der drei Plattformen (IPM-MC, IPM-ZS und IPM-AM) erörtert, welche Interaktionen zwischen den Subsystemen notwendigerweise umgesetzt werden müssen.

3.3.1 Schnittstelle zwischen Patient und Monitoring Center

Die Schnittstelle zwischen Patient und Monitoring-Zentrum ist verhältnismäßig einfach zu gestalten. Auf der Grundlage des Systementwurfs wird versucht, die Interaktion herzustellen und das Risiko der falschen Bedienung von Geräten zu reduzieren. Die hauptsächlichen Dienste sind wie folgt:

Identifikationsverwaltungsdienst: Dieser legt fest, dass ein Gerät in einem definierten Zeitraum nur einem Patienten Dienste anbietet. Die Identifikation des Patienten ist der erste Schritt des Tasks der Datenmessung. Der Patient kann über die Übermittlung von Benutzername und Passwort oder eine unabhängige IP-Adresse authentifiziert werden.

Messungsverwaltungsdienst: Der Zweck dieses Dienstes liegt darin, den im System definierten Messungstask zu überwachen. Bei Fehlern soll der Patient gewarnt und die Patientendaten sollen an das medizinische Zentrum übermittelt werden. Zur Realisierung dieses Dienstes müssen demnach zwei weitere Dienste realisiert werden: (1) ein Dienst zum Lesen der Informationen von Sensoren und (2) ein Alarm-Dienst. Der Dienst liest die Datenaus, welche die Messelemente gesammelt haben, die lokale Datenbank empfängt und speichert dann die Daten. Um die Informationen über den eigenen Gesundheitszustand zu aktualisieren, kann der Patient Kommentare hinzufügen (in ein vom medizinischen Personal bereitgestelltes Formular/Fragebögen eingetragenen). Der Prozess der Messung kann vom Patienten jederzeit abgebrochen werden. Die Daten werden aber keinesfalls gelöscht, sondern zwischen folgenden Zeichen abgespeichert: „measure interrupte=on“ und „measurement period= xxx seconds“. Diese Informationen werden zudem in einer XML-Datei abgespeichert.

Im **Alarm-Dienst** werden drei Arten von Warnungen konzipiert: (1) Falls die zuvor festgelegte Messkonfiguration nicht erfüllt wurde, erhält der Patient oder das medizinische Zentrum eine Warnmeldung. (2) Falls das medizinische Personal eine Message an den Patienten schickt, erhält der Patient eine Warnmeldung. (3) Falls die Verbindung des medizinischen Zentrums scheitert, erhält der Patient einen Warnhinweis (das medizinische Zentrum benötigt einen eigenen Alarm-Dienst).

Der **Message-Verwaltungsdienst** informiert den Patienten (wenn der Patient online ist), wenn der Patient eine Message erhält. Dieser Dienst muss daher notwendigerweise ebenfalls mit dem Alarm-Dienst kooperieren. Im gegenwärtigen Entwurf besitzt der Patient nicht die Berechtigung, die IPM-MC-Konfiguration aufzurufen, allerdings müssen beim Design der IPM-MC mögliche Entwicklungen im Heimpflege-Bereich berücksichtigt werden. Zwischen Patient und Geräten gibt es noch kompliziertere Zusammenhänge, bei der zu realisierenden Umgebung muss es daher möglich sein, einfach neue Charakteristika hinzuzufügen. Ein Beispiel für neue Anforderungen wäre, dass möglicherweise in Zukunft dem Patienten der Zugriff auf das Konfigurationsdokument der eigenen Geräte gestattet wird, was im gegenwärtigen Entwurf noch nicht der Fall ist.

3.3.2 Schnittstelle zwischen mHealth Client und mHealth Server

Das Austauschen der Message zwischen IPM-MC und IPM-ZS beschränkt sich auf die Übermittlung von XML Messages. Der IPM-MC fordert bei der IPM-ZS-Konfiguration und beim Message-Dokument an, der IPM-ZS antwortet mit der Übermittlung der Informationen. Der Empfang des Dokuments erfordert die Bestätigung des IPM-ZS, der IPM-ZS übermittelt dem IPM-MC das XML-Dokument des Patienten und fordert gleichzeitig eine Bestätigung an. Die Umsetzung dieser Schnittstelle benötigt die Unterstützung von vier Diensten, darunter zwei Dienste in den Geräten im Haushalt des Patienten: Der **Messungsverwaltungsdienst** ist zuständig für das Versenden der XML-Dokumente der gespeicherten Patienteninformationen. Der **Aktualisierungsverwaltungs-Dienst** verantwortet den Empfang der Konfiguration und Message-Dokumente vom IPM-MC, den Empfang von Bestätigungen, die Aktualisierung der Gerätekonfiguration und das Versenden von Message-Dokumenten an den Message-Verwaltungsdienst.

Die anderen beiden Dienste befinden sich auf dem zentralen Server IPM-ZS: Der **Konfigurations- und Message-Dienst**. Dieser erzeugt Informationen der Gerätekonfiguration und verschiedene Messages. Diese werden an den IPM-ZS versendet und es wird auf eine Bestätigung gewartet.

Der **Elektronische Gesundheitsakte (EG)-Aktualisierungsdienst** übernimmt den Empfang der Patientendaten und speichert diese in der Datenbank des medizinischen Zentrums. Auch um die Umwandlung in das entsprechende Datenformat kümmert sich dieser Dienst.

Sämtlicher Austausch basiert auf „Messages“, daher muss die Architektur der „Message“ standardisiert werden. Aus diesem Grund müssen alle Dienste das gleiche Format und Übertragungsprotokoll verwenden. Darüber hinaus wird in der SOA-Architektur eine Anwendung mit Message als Zentrum konzipiert, um die zunehmenden Business-Daten zu unterstützen. Das Verschicken der „Message“ durch die Dienste ist bereits zum grundlegendsten Element der SOA-Funktion geworden. Eine Weiterentwicklungsfähigkeit des Message-Frameworks erfordert daher eine extrem hohe Flexibilität und Erweiterbarkeit [69].

3.3.3 Schnittstelle zwischen Healthcare Monitoring-Zentrum und den Auswertungsmodulen

Das Auswertungsmodul verwendet die vom IPM-ZS angebotenen Dienste. Egal wie eine medizinische Anwendung umgesetzt wird, die Kommunikation mit dem IPM-ZS folgt dem WebService-Standard und verwendet SOAP Message zum Austausch von Informationen. Das IPM-ZS bietet dem IPM-AM folgende Dienste:

Identifikationsverwaltungsdienst. Patienten und medizinisches Personal benötigen jeweils einen eigenen Benutzernamen und ein Passwort für die Authentifikation.

Der Verwaltungsdienst für die Daten des medizinischen Personals. Dieser Dienst verwaltet sämtliche Daten des medizinischen Personals: Autorisierung zur Anzeige und Änderung bestimmter Parameter, Einführung von individuellen Informationen und Anzeige von speziellen Befugnissen etc.

Der Message-Verwaltungsdienst bietet einen sicheren Informationsaustausch zwischen den Nutzern (Erstellung von Patienten-Diagnoseergebnissen und Patienteninformationen, Prüfen individueller Informationen).

Der Datenzugriffsdienst. Medizinische Anwendungen benötigen bessere Methoden, um Patientendaten zu erhalten. Andere Dienste, welche vom medizinischen Personal angeboten werden, müssen Patientendaten lesen oder ändern (Einfügen von Diagnoseergebnissen, Kommentare etc.). Nachdem sie autorisiert wurden, können sie Anfragen abschicken. Der Datenzugriffsdienst befriedigt diese Anforderungen.

Dienst zur Integration externer Module. Die Informationen der definierten Dienste, welche auf eine Schnittstelle zugreifen, entsprechend der Konfiguration von medizinischem Personal

und IT-Personal, und können vom IPM-AM abgerufen werden. Diese Art von Diensten sind gewöhnlich externe Datenverarbeitungsmodule, welche derzeit meist als Webservice angeboten werden, wie beispielsweise das Stressmessmodul des eHealth-Projekts des Instituts für Präventivmedizin der Universität Rostock [70].

Der Gerätekonfigurationsdienst ermöglicht dem Patienten sowie dem medizinischen und IT-Personal die Kontrolle der notwendigen Informationen der Geräte in der Heimpflege. Die Behandlungsabläufe des Patienten (wie Messzeitpunkt und -dauer), die Verbindungszeit und die verwendeten Messtypen werden von spezialisiertem, medizinischen Personal festgelegt. Wenn diese Daten verschickt werden, fügt das System automatisch die ID des medizinischen Personals hinzu.

Der Alarmdienst definiert die Warnereignisse und Alarm Messages. Wenn das medizinische Personal eine Message erhält oder Patientendaten an das IPM-ZS übermittelt werden, führt der Server eine Hinweisprozedur aus (es kann eine Message, E-Mail oder kurze Message an das IPM-AM verschickt werden).

4 Entwurf und Umsetzung des mobilen Erfassungsclients

Im vorangegangenen Kapitel wurden die verschiedenen Komponenten des mHealth-Systems dargelegt und die Methode der Interaktion zwischen den verschiedenen Subsystemen erörtert. Entsprechend der Schilderung in Abschnitt 3.3 umfasst der mobile Erfassungsclient des mHealth-Systems die folgenden drei Kern-Komponenten: (1) ein Sensorkommunikationsmodul, welches mit den medizinischen Sensoren kommuniziert; (2) ein intelligentes Echtzeit-Überwachungsmodul und (3) ein Datenübertragungsmodul für die Weiterleitung der physiologischen Parameter auf den IPM Server. In diesem Kapitel soll zunächst kurz die Android-basierte Entwicklungstechnologie für die mobile Anwendung vorgestellt werden, danach werden die in dieser Arbeit eingebundenen Datenerfassungsgeräte (Sensoren) aufgelistet. In der Konzeption des mobilen Clients (IPM-MC) wird in der vorliegenden Arbeit eine neue Middleware-Komponente entworfen, um verschiedenste Sensorgeräte flexibel zu integrieren und gleichzeitig die Kommunikation zwischen Sensoren verschiedener Arten (darunter Sensoren, welche Bluetooth und ANT verwenden) zu erreichen. Zuletzt werden das Design und die Umsetzung der Echtzeit-Überwachung der physiologischen Parameter und die Übertragung der Daten auf den IPM-ZS detailliert dargelegt.

4.1 Android-Plattform

Die Android-Plattform [71] bietet die notwendigen Schnittstellen und Werkzeuge für die Entwicklung von Anwendungen auf den mobilen Geräten. Diese enthält vier grundlegende Eigenschaften:

1. Auf dem Applikation Layer der Plattform werden austauschbare und wiederverwendbare Module verwendet;
2. Grafiken werden bei Android gut unterstützt;
3. Multimediaetechnik, Mobilkommunikation und Wireless-LAN Technologien der Plattform sind abhängig von der verwendeten Hardwareplattform [72];
4. Es wird ein Rahmenkonstrukt mit verschiedenen Layern verwendet, auf Basis dessen von verschiedenen Entwicklern in unterschiedliche Richtungen entwickelt werden kann.

Die Android-Plattform verwendet eine Betriebssystem-Architektur mit verschiedenen Layern. Wie Abbildung 4-1 zeigt, unterteilt sich die Plattform in fünf wesentliche Bestandteile. Dies sind derLinux Kernel, die Laufzeitumgebung des Systems, die für Entwicklung notwendige Systembibliothek, das Applikation Framework und die Applikation. Da eine relativ deutlich getrennte Layerstruktur verwendet wird, ergibt sich daraus auch eine unterschiedliche

Arbeitsteilung für Programmierer. Hinsichtlich der Layer von Systembibliothek und Applikation Framework wird im Wesentlichen mit C++ entwickelt, entsprechend der von den unteren Layern bereitgestellten API. Für die Applikationen des darüber liegenden Layers wird eine Funktionsaufruf-Schnittstelle geschaffen. Bei der Applikation wird hauptsächlich Java für die Entwicklung verwendet. Gewöhnlich dient ECLIPSE als Entwicklungswerkzeug.

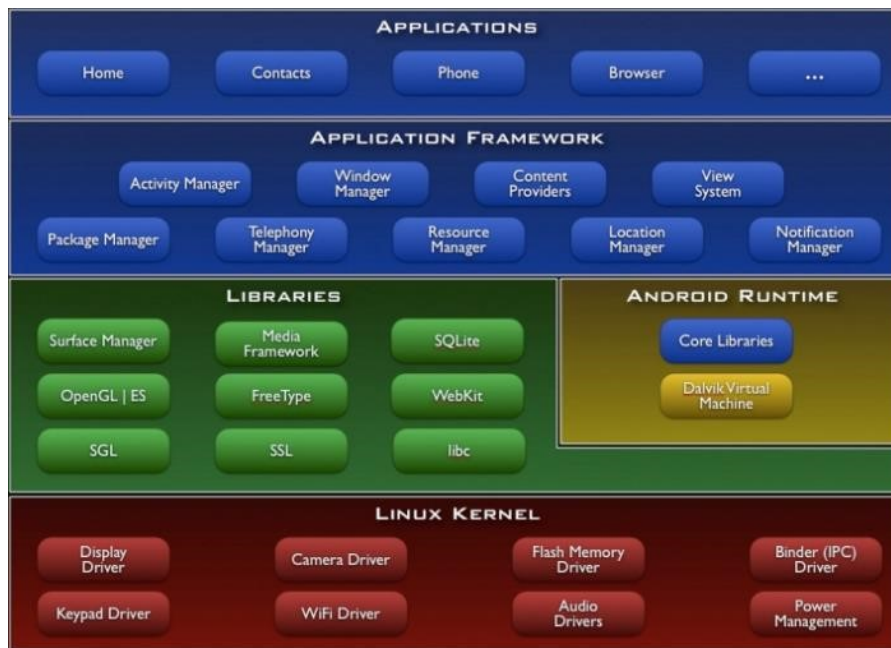


Abbildung 4-1 Architektur des Android-Systems [71]

4.1.1 Android-Entwicklungstechnologie

Anders als Software-Anwendungen auf Windows, Linux, IOS oder anderen Betriebssystemen besitzen die Anwendungen der Android-Plattform keinen gemeinsamen, von allen Nutzern verwendeten Einsprungspunkt. Jede Software-Anwendung ist vollständig unabhängig und läuft in ihrem eigenen Anwendungsprozess, die Anwendungen stören sich nicht gegenseitig. Nach den unterschiedlichen Funktionen unterteilt Android die Applikation in vier Kern Klassen: *Content Provider*, *Broadcast Receiver*, *Service* und *Activity*. Diese werden gewöhnlich als die „Big-Four“ der Komponenten bezeichnet, allerdings fallen längst nicht alle Anwendungen in diese vier Klassen. Der Austausch von Informationen zwischen gleichen und unterschiedlichen Komponenten wird über Intent erreicht. In Android wird auch die View Klasse definiert, welche der Visualisierung der Oberfläche dient. Wenn die oben genannten Komponenten verwendet werden sollen, muss in der Konfigurationsdatei der Applikation (*AndroidManifest.xml*) eine entsprechende Einstellung erfolgen. Die Struktur und der Ablauf der Anwendungsprogramme in Android sind in Abbildung 4-2 zu sehen.

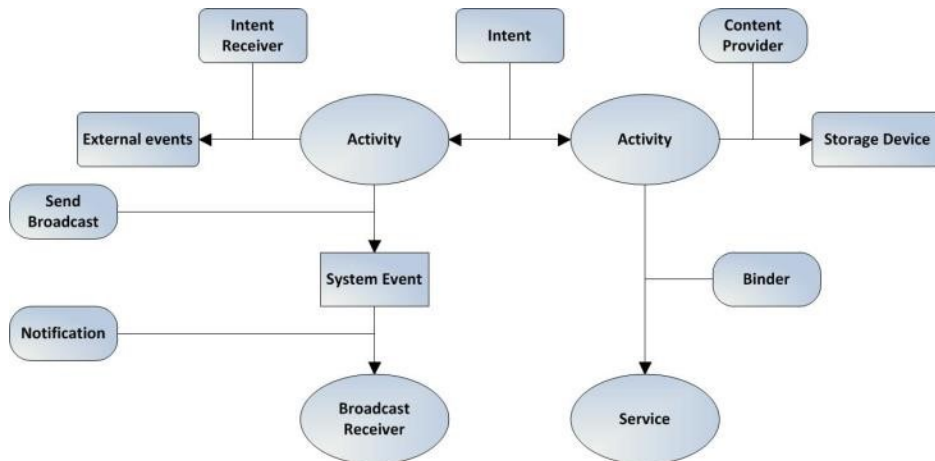


Abbildung 4-2 Struktur und Arbeitsprozesse der Anwendungsprogramme in Android

Die Entwicklungsumgebung von Android kann auf Windows, Linux und Mac eingerichtet werden. Um aber das Projekt bequem entwickeln zu können, wird in dieser Arbeit auf die Plattform des Windows XP SP2 zurückgegriffen und als Software-Entwicklungsumgebung werden die derzeit verhältnismäßig gängigen JDK1.7+ Eclipse3.6.2+ AndroidSDK-r15+ADT-15 verwendet. Aufgrund des Umfangs wird an dieser Stelle auf die Konfiguration nicht im Detail eingegangen.

4.2 Entwurf und Umsetzung des Erfassungsclients

4.2.1 Sensorsystem

Bevor die Beschreibung der Implementierung von IPM-MC erfolgt, werden hier zunächst die in diese Arbeit integrierten medizinischen Sensoren vorgestellt. Die Fernüberwachung ist das klassische Szenario einer drahtlosen Sensoranwendung [73]. Die unteren Layer des mHealth-Systems setzen sich aus vielen unterschiedlichen tragbaren Sensoren zusammen. Über die Wahrnehmungsfunktion der Sensoren werden die physiologischen Parameter von Menschen gesammelt, womit das Ziel der Langzeit-Überwachung des Gesundheitszustandes von Menschen in Echtzeit erreicht wird. Gegenwärtig gibt es in vergleichbaren Anwendungen zwei Konzepte für die Auswahl der Sensoren: (1) Die Auswahl von Sensoren auf dem Markt. (2) Der Einsatz selbst entwickelter Sensoren. Beide Konzepte haben ihre Vor- und Nachteile. Beim ersten Konzept können auf dem Markt eingekaufte Sensoren oftmals nicht alle Anforderungen der Anwendung in der Praxis erfüllen. Zudem sind Sensoren für mehrere Parameter verhältnismäßig kostspielig. Beim zweiten Konzept reduziert die Entwicklung eigener Sensoren zwar die Kosten, ist aber oftmals zeitaufwändig und die entwickelte Hardware erfüllt auch nicht unbedingt die Anforderung zum Sammeln jeglicher Art von physiologischen Parametern.

Daher werden in dieser Arbeit sowohl einige auf dem Markt befindliche Sensoren, gleichzeitig aber auch eigenständig entwickelte Sensoren von [74] verwendet. Dieses System erfasst hauptsächlich mittels dieser Geräte individuelle Gesundheitsdaten und führt danach die Übermittlung, Analyse, Darstellung sowie die Auswertung der Daten durch. Ein Ziel des Entwurfs des Datenerfassungsclients (IPM-MC) ist die flexible Unterstützung neuer Sensorgeräte, ohne dass dabei die gesamte Systemstruktur oder der Code überarbeitet werden müsste. Auf diese Weise soll auch die Erweiterbarkeit des Systems verbessert werden. Die konkrete Umsetzung hiervon wird im nachfolgenden Abschnitt erläutert, zuvor soll aber die bereits im System eingebundene Sensorhardware aufgelistet werden, welche hauptsächlich drahtlose Thermometer, EKG-Geräte, Pedometer, Beschleunigungs-Sensoren etc. umfasst, wie Tabelle 4-1 zu entnehmen ist.

Tabelle 4-1 Sensoren List

Sensor	Parameter	Protokoll	Bild
Hidalgo Equivital [75]	Muti-parameter(Vital, Acc, ECG)	Bluetooth	
Zephyr HXM BT [76]	Heartrate, RRinterval	Bluetooth	
Zephyr BioHansess3 [77]	Muti-parameter(Vital, Acc, ECG)	Bluetooth	
Mio Alpha [78]	Pulsuhr	Bluetooth Low Energy	
Polar H7 [79]	Heartrate, RRinterval	Bluetooth Low Energy	
Wahoo Speed [80]	Speed, Step	ANT+	
Wahoo Heartrate [81]	Heartrate	ANT+	
Garmin Tempe [82]	Umgebungstemperatur	ANT+	
Eigene Entwicklung [74]	Mutiparameter	ANT	

Nach Funktion unterteilt, kommen zwei Arten von Sensoren häufig vor:

1. Sensoren zum Sammeln von physiologischen Parametern: wie EKG-Geräte, HerzfrequenzMonitor, Thermometer, Atmungsmesser, etc. Darunter auch *Hidalgo Equivital*, *Zephyr BioHarness3* in der Liste.
2. Aktivitätssensoren, wie Positionssensoren, Schwerkräftensensoren, Geschwindigkeits-sensoren, Gassensoren, etc. Diese beinhalten *Wahoo Speed*, *Hidalgo EquivitalEQ01* und *Zephyr BioHarness3* in der Liste.

Unterteilt nach der verwendeten Kommunikationsschnittstelle gibt es drei häufig auftretende Arten von Sensoren:

1. Sensoren, welche Bluetooth für die drahtlose Kommunikation im Nahbereich verwenden. Darunter fallen in der Liste *Hidalgo Equivital*, *Zephyr BioHarness2* und *Zephyr BioHarness3*.
2. Sensoren, welche ANT für die drahtlose Kommunikation im Nahbereich verwenden. Darunter fallen in der Liste *Wahoo Speed*, *Wahoo Heartrate*, *Garmin Tempe* und die *eigenständig entwickelten Sensoren*.
3. Sensoren, welche Bluetooth Low Energy für die drahtlose Kommunikation im Nahbereich verwenden. Darunter fallen in der Liste *Mio Alpha* und *Polar H7*.

Da gegenwärtig die Unterstützung von Zigbee durch Smartphones noch unausgereift ist, werden in dieser Arbeit keine Zigbee-Sensoren verwendet.

4.2.2 Wesentliche Module des Erfassungsclients

Die Entwicklung des IPM-MC auf der Grundlage einer mobilen Plattform muss auf vielfältiges Wissen über Multithreading und die Kommunikation von Smartphones zurückgreifen [83]. Konkret beinhaltet dies: (1) Kenntnisse der Programmierung von Multithreading auf der Android-Plattform, (2) der Netzwerkprogrammierung von Bluetooth/ANT, TCP-SOCKET, (3) der HTTP Netzwerk-Programmierung und (4) Kenntnisse über Content Provider, etc. Entsprechend des Gesamtentwurfs aus Abschnitt 3.3 wird beim Entwurf des IPM-MC das Konzept der komponentenbasierten Entwicklung verwendet. Das gesamte System wird untergliedert in die vier Module: **Sensorinteraktionsmodul**, **Visualisierungsmodul**, **Netzwerkkommunikationsmodul** und **Verwaltungsmodul**.

Die Struktur der verschiedenen Funktionsmodule wurde in Abbildung 4-3 dargestellt. Jedes Modul erfüllt seine vorgesehene Aufgabe eigenständig und ist unabhängig von den jeweils anderen. Über die Dateninteraktion erfüllen die Module gemeinsam die Aufgabe des mHealth Systems.

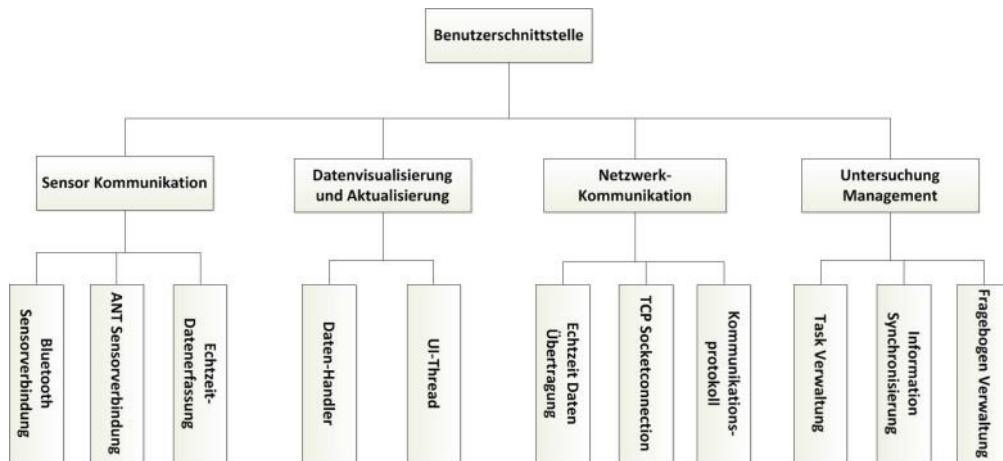


Abbildung 4-3Kernkomponenten vom IPM-MC

4.3 Sensor Kommunikation

Die Kernfunktion des IPM-MC ist die Erfassung und Übermittlung der Informationen über die physiologischen Parameter des Patienten. Diese Funktion verwendet die Nahbereich-Kommunikation via Transceiver vom Smartphone als Hardwaremodul zur Kommunikation mit den medizinischen Sensoren. In dieser Arbeit werden die in Sony Xperia (Android 2.3.4) integrierten Bluetooth- und ANT-Schnittstellen für die drahtlose Kommunikation verwendet, wie in Abbildung 4-4 zu sehen ist. Im Folgenden wird differenziert der Ablauf des Aufbaus einer drahtlosen Verbindung über Bluetooth/ANT dargelegt.



Abbildung 4-4Bluetooth und Ant Transceiver[84] [85]

4.3.1 Drahtlose Verbindung von IPM-MC über Bluetooth

Android 2.0 und höhere Versionen bieten eine öffentliche Bluetooth-API, welche zur Kommunikation und Verbindung über Bluetooth dient. Der Prozess der Operation mittels Bluetooth vom IPM-MC umfasst vier Schritte:

1. das Öffnen der lokalen Bluetooth-Schnittstelle,
2. Suche nach dem Sensorgerät,
3. die Etablierung der Verbindung des virtuellen Socket
4. und die Datenübertragung.

1. Steuerung des lokalen Bluetooth

Zunächst, um die lokale Bluetooth-Schnittstelle zu betätigen, muss in der *manifest.xml* der IPM-MC eine Berechtigung hinzugefügt werden:

```
<uses-permissionandroid:name="android.permission.BLUETOOTH_ADMIN"/>
<uses-permissionandroid:name="android.permission.BLUETOOTH" />
```

Die Klasse `BluetoothAdapter` beschreibt den lokalen Bluetooth-Adapter. Dieser muss über `getDefaultAdapter()` gesteuert werden und diese Methode ermittelt die lokale `BluetoothAdapter`-Instanz. Wenn der Rückgabewert null ist, so zeigt dies, dass das Gerät nicht über ein Bluetooth-Modul verfügt.

Die `BluetoothAdapter`-Klasse bietet zahlreiche Methoden, um die lokale Bluetooth-Schnittstelle zu steuern. Die Methoden `enable()` und `disable()` können den Bluetooth-Adapter öffnen oder schließen. Die Methoden `getName()` und `setName()` können den Namen des lokalen Gerätes ändern, `getAddress()` kann dazu verwendet werden, um die Adresse des lokalen Gerätes zu bestimmen.

2. Identifikation und Match bei Sensormodulen

Der Vorgang des Entdeckens von Devices wird Discovery genannt. Um andere Geräte zu detektieren, kann die `startDiscovery()` Methode des `BluetoothAdapter`s verwendet werden. Der Vorgang der Discovery dauert ungefähr 12 Sekunden. IPM-MC registriert für die ACTION: „FOUND“ einen Broadcast Receiver, um die vom Gerät gesammelten Rückmeldungen einzuholen. Für jedes Gerät wird das System eine ACTION: „FOUND“ zuordnen. Der Identifikationsprozess ist in Abbildung 4-5 zu sehen.

Die Identifikation ist ein stark Ressourcen beanspruchender Prozess [86]. Nachdem das benötigte Gerät gefunden wurde, kann dann die `cancelDiscovery()` Methode abgerufen werden, um den Suchprozess anzuhalten. Wenn bereits eine Verbindung zu einem Gerät hergestellt wurde, reduziert der Scanvorgang die Bandbreite der Kommunikation. Mittels „Pairing“-Methode kann zwischen zwei Bluetooth-Geräten über die Authentifizierung durch eine vierstellige PIN eine Verbindung hergestellt werden. Die Authentifizierung stellt sicher, dass die Bluetooth-Verbindung nicht gekapert wird. Unter IPM-MC muss zunächst mit dem Sensor ein „Paar“ durchgeführt werden, erst danach ist eine Kommunikation über die Verbindung wie RFCOMM möglich.

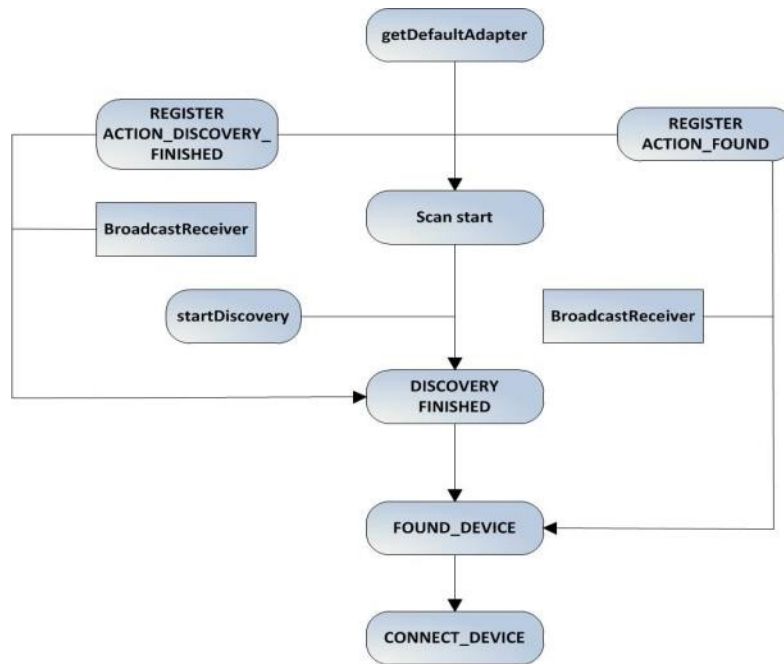


Abbildung 4-5IPM-MC Bluetooth Device Discovery-Prozess

3. Verwendung von Bluetooth für die Kommunikation

Das Ziel eines „Pairing“ Methodes mit einem Sensormodul ist eben der Aufbau von Kommunikation mit diesem. Die Bluetooth-Datenübertragung erfolgt über das RFCOMM-Protokoll. Dieses Protokoll unterstützt die Logical Link Control der unteren Layer und das virtuell serielle Kommunikationsprotokoll RS232 über dem Layer des Anpassungsprotokolls (L2CAP) [87]. Im Grunde genommen bieten diese Standards einen Mechanismus und dieser Mechanismus kann zwischen zwei Bluetooth-Geräten mit „Pairing“ ein Kommunikations-Socket öffnen. Wenn zwei Bluetooth-Geräte miteinander verbunden werden sollen, muss eines der Bluetooth-Geräte als Server fungieren, während das andere als Client fungiert. Der Server ist verantwortlich für das Empfangen der Verbindungsanfrage. Nach Empfang der Verbindungsanfrage errichtet er die Bluetooth-Verbindung über Bluetooth-Socket und antwortet auf die Anfrage des Clients. Im vorliegenden System fungieren die Sensormodule als Server, der IPM-MC dagegen fungiert als Client und sendet eine Verbindungsanfrage.

4. Datenübermittlung:

Nachdem die Geräte mit IPM-MC verbunden sind, ermittelt IPM-MC eine BluetoothSocket Verbindung. Ab diesem Zeitpunkt ist eine gemeinsame Datennutzung zwischen den Geräten möglich. Zunächst werden über die Verwendung der Methoden *getInputStream()* und *getOutputStream()*, Input- und Output-Daten ermittelt. Danach werden über die Methoden *read(byte[])* und *write(byte[])* Daten entweder gelesen oder geschrieben. Hier muss aber beachtet werden, dass der Datenfluss sowohl beim Lesen als auch beim Schreiben

blockieren kann [88]. Es muss deshalb ein unabhängiger Arbeitsthread zur Verwaltung eingerichtet werden. Der Inhalt des Echtzeit-Datenverarbeitungsthreads wird im nächsten Abschnitt erläutert.

4.3.2 Verbindung von IPM-MC über ANT

Die Verbindung von IPM-MC über ANT benötigt Unterstützung durch die ANT Radio Service Library [89]. In dieser API finden sich Bedienungsverfahren für ANT-Verbindungen. Der Aufbau einer drahtlosen Verbindung über ANT ist im Wesentlichen die Konfiguration eines ANT-Kanals mit IPM-MC.

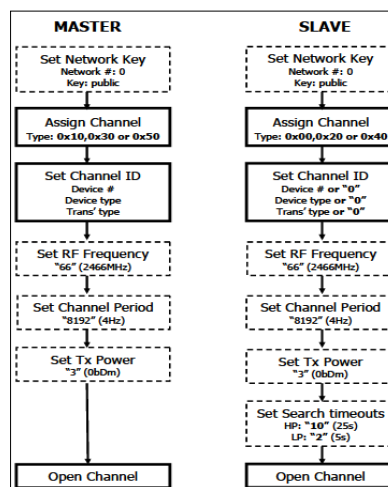


Abbildung 4-6 Etablieren eines ANT-Kommunikationskanals zwischen Master- und Slave-Knoten

Konfiguration eines ANTKanals

Das ANT-Protokoll stellt viele Befehle zur Verfügung. Zur Etablierung eines ANT-Kanals ist es notwendig, folgende Befehle zu transferieren:

- Assign Kanal
- Set Kanal ID
- Set Network Key
- Set Kanal Search Timeout
- Set Kanal RF Frequency
- Set Kanal Period

Essentiell für die erfolgreiche Konfiguration einer Verbindung ist jedoch nur der Assign-Kanal-Befehl, welcher die Übertragungsrichtung des Kanals definiert (siehe Abbildung 4-6). Die anderen Befehle werden als optional angesehen. Sie sind anzuwenden, um von den jeweiligen Default-Werten abweichende Parameter-Einstellungen vorzunehmen. Nachdem ein Kanal konfiguriert worden ist, ist dieser durch den *öffnen-Kanal-Befehl* zu öffnen/aktivieren und damit für Datentransfer zu nutzen. Zur Beendigung eines Kanals und zur Versetzung

eines der Sensoren in den Ruhezustand wird der *schließen-Kanal-Befehl* angewendet. Er deaktiviert aber den Kanal und gibt ihn nicht frei. Bevor ein neuer ANT-Kanal mittels Assign-Kanaldefiniert werden kann, ist es notwendig, den Befehl *Unassign-Kanal* anzuwenden. Zur Freigabe sämtlicher Kanäle und zum Neustart des ganzen ANT-Systems ist der *Reset-ANT-Befehl* einzusetzen. Dieser Befehl sollte beim Systemstart zuerst durchgeführt werden, um das Zurücksetzen des ANT-Systems in einen definierten Zustand zu verwirklichen. (Eine genaue Aufstellung aller Befehle wird in Dyn[90] angegeben. Anhang A stellt einen kurzen Überblick über die wesentlichen Befehle einschließlich derer Implementierung dar.)

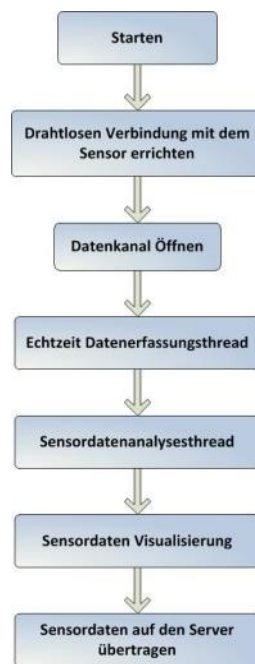


Abbildung 4-7 Ablauf des Echtzeit-Sensordatenerfassungsprozesses

4.3.3 Echtzeit Datenerfassungsprozess

Entsprechend der Aufgabe des mHealth-Systems und der funktionalen Teilung, muss das Echtzeit-Taskmodul drei Funktionen erfüllen: (1) Der erste Schritt ist die Errichtung einer drahtlosen Verbindung mit dem Sensor. (2) Danach erfolgt das Einholen in Echtzeit von Rohdaten, welche die Sensoren erfassen. (3) Zuletzt erfolgt eine Analyse der Rohdaten in Echtzeit, der bytencode wird in die entsprechende Information konvertiert. Da der Aufbau der drahtlosen Verbindung und die Datenerfassung nacheinander erfolgen, müssen diese beiden Schritte im selben Thread zustande gebracht werden. Daher werden die drei Schritte des Echtzeit-Taskmoduls in zwei Threads aufgeteilt. Ein Thread ist der Echtzeit-Datenerfassung gewidmet, der andere der Echtzeit-Analyse. Nach dem Aufbau der drahtlosen Verbindung zum Sensor liest der IPM-MC beständig die gesammelten Signaldaten. Diese Daten müssen

notwendigerweise gleichzeitig und parallel vom anderen Prozess bearbeitet werden, da sie ansonsten von den späteren Daten überschrieben würden, was zu Datenverlusten führen würde. Daher muss die Thread-Technologie [91] verwendet werden, um einen Echtzeit-Prozess zu öffnen, welcher speziell diese Daten analysiert. Die Arbeitsprozesse der verschiedenen Threads des IPM-MC sind in Abbildung 4-7 zu sehen.

Sobald der Medizinische Untersuchungsprozess [92] begonnen wird, muss der Patient zunächst eine Untersuchungsstudie im Health-System auswählen (Synchronisierung von IPM-ZS), siehe Abbildung 4-8. Die Überwachungsaufgabe wird vom Untersuchungsverwaltungsmodul verarbeitet. Dieses Modul wird später genauer vorgestellt. In der Untersuchungsstudie sind die Typen der zu sammelnden physiologischen Parameter eingestellt, wie beispielsweise Herzfrequenz, EKG, Körpertemperatur, Position etc. Nachdem die notwendigerweise zu sammelnden Daten festgelegt wurden, ist der erste Schritt der Suchprozess, während dessen nach verfügbaren Bluetooth- oder ANT-Geräten gesucht wird. Nachdem entsprechende Geräte detektiert wurden, wird der Prozess der Initialisierung des Datenübertragungskanals gestartet. Da Bluetooth- und ANT-Protokoll verschiedene Verbindungsabläufe haben, wird das System entsprechend des gefundenen Gerätetyps automatisch die entsprechende Art der Verbindung auswählen.

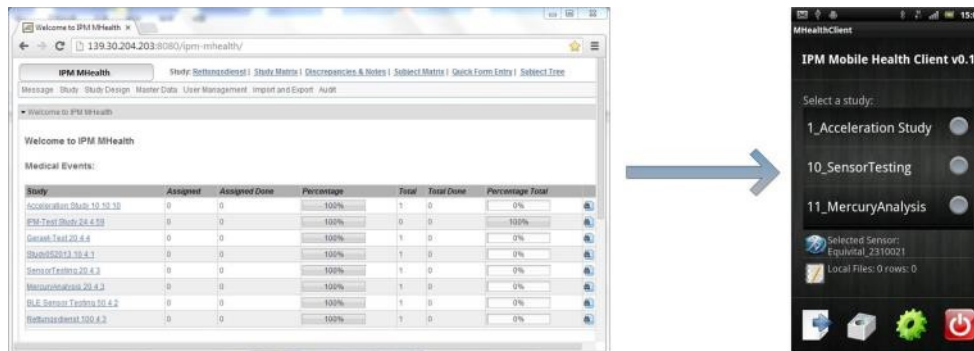


Abbildung 4-8 Synchronisation von Study information zwischen IPM-MC und IPM Server

Nach der Initialisierung der Verbindung konfiguriert das System für jedes Gerät eine eindeutige Device-ID und Kanal-ID. Danach wird der Prozess des Sammelns von Daten begonnen. Nachdem der Thread die abgefragten Daten gelesen hat, wird der folgende Vorgang gestartet:

1. Gelesene binäre Byteströme werden in einem ByteArray *rawdata(byte[])* des Puffers abgelegt.
2. Dann wird geprüft, ob 25 Datenpakete empfangen wurden. Falls 25 Datenpakete angesammelt wurden, wird der Datenanalyseprozess benachrichtigt, mit der Datenanalyse zu beginnen.

3. Der Datenanalyseprozess ruft den Datenanalysemanager auf, um die gesammelten Daten zu analysieren (entsprechend der Parser-Regelung).
4. Der Parser legt die aus dem Analyse-Prozess erhaltenen Daten im neuen ByteArray Sensordaten (byte[]) desPuffers ab.
5. Falls 25 Datenpakete empfangen wurden und das Netzwerk als „verbunden“ markiert ist, wird der Echtzeit-Übertragungsthread informiert, die Daten zur Server-Seite zu übermitteln.
6. IPM-MC prüft dann, ob 100 Datenpakete angehäuft wurden. Falls 100 Datenpakete angehäuft wurden, wird der Display-Thread gestartet, um die Physiologischer Parameter zu visualisieren.
7. Wenn der Patient die Überwachungsaufgabe beendet, sendet das System eine Nachricht um die Verbindung und Datenerfassungzuschließen.

4.3.4 Echtzeit Analyse der Sensordaten

Sobald sich der Datenempfangsthread im Arbeitszustand befindet, werden stetig Daten der Sensoren über die drahtlose Verbindung ans System übertragen. Der vorherige Sammelfhread verantwortet nur den drahtlosen Empfang der Daten und ihre Speicherung im Puffer. Ohne weitere Verarbeitung sind die Daten nur eine lange Abfolge von Bytes und unbenutzbar. Daher müssen die Informationen jedes Feldes aus dem Bytestrom im Pufferspeicher notwendigerweise extrahiert und in praktische Informationen umgewandelt werden, damit sie als Daten wertvolle Informationen zur Entscheidungsfindung beisteuern können. Allerdings gibt es momentan keinen einheitlichen Standard für physiologische Datenformate und jede Art von Sensor verwendet ein eigenes Übertragungsprotokoll [93]. Wie nun diese Protokolle dynamisch analysiert werden und die verschiedenen Arten von Sensorgeräten flexibel vom Mobile Client eingebunden werden können, ist längst zu einer wichtigen Frage in diesem Bereich geworden.

In dieser Arbeit wird ein auf die Schnittstellen orientierter Entwurfsansatz verwendet [94]. Im Wesentlichen werden der obere Layer (Display Layer) des IPM-MC und der untere Layer (Datenverarbeitung Layer) über die Schnittstellen lose gekoppelt. Hierdurch wird ein flexibler Austausch der Sensorgeräte ermöglicht und genauso das Ziel der Erweiterbarkeit vom IPM-MC erreicht. Der erste Schritt des Entwurfs ist die Analyse eines normalen Ablaufs der Sensordatenanalyse, des Weiteren wird der allgemeine Charakter darauszextrahiert und die allgemeine Schnittstelle für die Sensor-Datenanalyse definiert. Die konkreten Schritte der Umsetzung sind nachstehend aus Abbildung 4-9 zu ersehen.

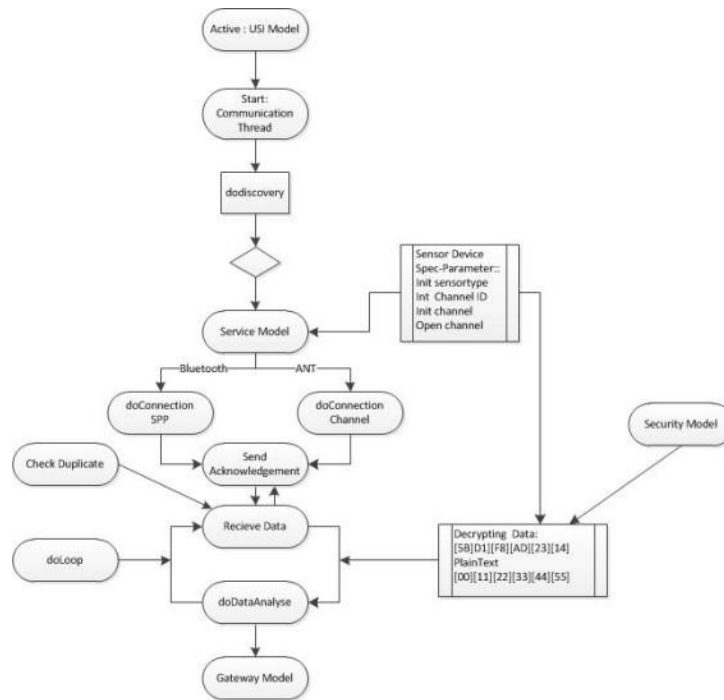


Abbildung 4-9 Ablauf der Sensordatenanalyse

1. Definition die Schnittstelle der Datenanalyse

Wie vorher beschrieben ist ein entscheidender Schritt, um eine dynamische Analyse des Sensordatenstroms zu erreichen, die Definition eines universellen Datenparsers als Analyseschnittstelle. In der Schnittstelle wird nicht die konkrete Analysemethode implementiert, sondern in ihr wird der allgemeine Vorgang des Analyseablaufs definiert. Nachfolgend werden zwei unterschiedliche Übertragungsprotokolle von kommerziellen Sensoren als Beispiel separat betrachtet. Es handelt sich um Multi-Parameter-Sensoren, „Hidalgo-Equivital“ [95], welche auf Bluetooth und „Wahoo-Speed“ [96] auf ANT basieren. Die beiden Sensoren können die Daten zu EKG, Körpertemperatur, Geschwindigkeit, Bewegung und Entfernung sammeln (siehe Abbildung 4-10, 4-11).

Daraus lässt sich erkennen, dass in normalen Übertragungsprotokollen stets ein solches Format definiert ist $\langle\langle \text{spezifisches Zeichen} + \text{Daten} \rangle\rangle$. Das spezifische Zeichen drückt den konkreten Befehl oder Datentyp aus. Beim *Equivital* Übertragungsprotokoll beispielsweise definiert r (ASCII: 114) den Herzfrequenz-Datentyp. Nachdem im Bytestrom ein „r“ entdeckt wurde, sind die folgenden zwei Bytes als Herzfrequenz-Wert zu interpretieren.

Data Item	Message Type	Data Prior to Coding
Raw Disclosure Output	-	
Primary ECG1	e	10 bit raw A/D reading + sequence no. (see 3.4 below)
Secondary ECG2	c	10 bit raw A/D reading + sequence no. (see 3.4 below)
Belt Sensor Respiration	f	10 bit raw A/D reading
Accelerometer Vertical Axis	v	10 bit raw A/D reading (see below for conversion to milli-g)
Accelerometer Lateral Axis	l	10 bit raw A/D reading
Accelerometer Longitudinal Axis	w	10 bit raw A/D reading
Skin Temperature	t	calibrated value in 0.1 °C, ie 0 to +204.7 °C
Impedance Respiration	i	10 bit raw A/D reading
R-R Interval	r	9 bit value in 1/256 s units i.e. 0.000 to 1.996 sec + sequence number (see 3.4 below)
Summary Disclosure Outputs:	-	
Belt Respiration Rate	u	12 bit value in 0.1 bpm ie 0.0 to 409.5 bpm
Heart Rate	h	12 bit value in 0.1 bpm ie 0.0 to 409.5 bpm
Impedance Respiration Rate	j	12 bit value in 0.1 bpm ie 0.0 to 409.5 bpm
EDR Respiration Rate	d	12 bit value in 0.1 bpm

Abbildung 4-10 Datenübertragungsprotokoll für den Mutisensor EQ01 (Hidalgo,Coambridge UK) [97]

Byte	Description	Length	Value	Units	Rollover
0	Data Page Number	1 Byte	Data Page Number	N/A	N/A
1	Reserved	1 Byte	Value = 0xFF	N/A	N/A
2	Reserved	1 Byte	Value = 0xFF	N/A	N/A
3	Cadence - Integer	1 Byte	Measure of strides accumulated in time. Set to 0x00 when unused.	Strides per Minute	256
4	Cadence - Fractional	Upper 4 bits	Fractional cadence. Set to 0x00 when unused.	1/16 Strides per Minute	N/A
4	Instantaneous Speed - Integer	Lower 4 bits	Instantaneous speed is intended to be appropriately filtered by the SDM, such that the receiving unit can directly display this value to the user. Set to 0x00 when unused.	m/s	N/A
5	Instantaneous Speed - Fractional	1 Byte	Fractional instantaneous speed. Set to 0x00 when unused.	1/256 m/s	N/A
6	Reserved	1 Byte	Value = 0xFF	N/A	N/A
7	Status	1 Byte	SDM status flags - see Table 5-5.	Binary	

Abbildung 4-11 Datenübertragungsprotokoll für den Beschleunigungssensor (Wahoo Fitness)[98]

Basierend auf der obigen Analyse wird in der Datenparser-Schnittstelle die erste Methode benötigt, mittels welcher eine korrekte Analyse dieser Bytes möglich ist und wodurch der Sinn der Daten entschlüsselt werden kann.

Message Type	Data1	Data2
114	00011001	11001101

a. *SensorData parserFrame(byte[] data)*

Diese Methode akzeptiert eine bestimmte Menge von Byteströmen als Parameter, zurückgegeben wird hier eine im IPM-MC definierte Klasse. In dieser Klasse werden verschiedene Typen von physiologischen Daten, wie Herzfrequenz, Körpertemperatur, etc. als Attribute definiert.

In der obigen Methode *parserFrame()* wurde definiert, wie die Daten analysiert werden. Im Anschluss muss noch bekannt sein, wie ein korrekter Data Frame von Sensoren erkannt werden kann, um die Parameter an die Methode *parserFrame(byte[]data)* zu übergeben.

Beispielsweise kann hier *Equivital* betrachtet werden, der Start eines gültigen Datenframes beginnt mit a bis n als erstem Byte. Zusammen mit den zwei Ziffern dahinter sind dies 3 Byte, welche einen korrekten Frame darstellen. In *Wahoo* sind 8 Byte als ein Standard-Frame definiert, der Wert des ersten Bytes ist 0x64. Daraus lässt sich für den Datenparser ableiten, dass die notwendigerweise zu definierenden Methoden sind:

b. `byte[] validateFrame(byte[] data)`

Diese Methode wird verwendet, um je nach Standard-Frame des Protokolls die Byteströme in einzelne Abschnitte(Standard-Frame) zu unterteilen.

c. `int findNextAlignment(byte[] data)`

Diese Methode wird verwendet, um die ursprüngliche Position des nächsten Frames im Bytestrom anzuzeigen.

d. `int getBufferSize()`

Diese Methode wird verwendet, um den Datenanalysethread zu informieren, jeweils nach einer bestimmten Menge von Bytes die Analyse zu beginnen. Nachfolgend ist der gesamte Quellcode der Datenparser-Schnittstelle dargestellt:

```
publicinterface SensorDataParser {
public SensorData parseFrame(byte[] readBuff);
publicbooleanvalidateFrame (byte[] buffer);
    publicint findNextAlignment(byte[] buffer);
    publicintgetBufferSize();
publicint updateUI(Context context);
publicvoid saveDataList();
}
```

2. Erarbeiten konkreter Implementierungsklassen

Zuvor wurde bereits ausgeführt, dass in der Schnittstelle nur abstrakte Methoden definiert sind und nicht die konkrete Implementierung. Daher muss nach der Definition einer Schnittstelle noch für jede Art von Sensor eine konkrete Analysemethode implementiert werden. Die nachfolgenden beiden Beispielcodes setzen die Analyse für Rohdaten des Sensors *Equivitalum*.

```
publicclass EquivitalDataPaser implements SensorDataParser{
private SensorData sd = new SensorData();
...
@Override
publicint findNextAlignment(byte[] buffer){
    for (int i = 0; i < buffer.length - 3; i++){
        if (packetValid(buffer, i)) {
            return i;
        }
    }
}
```

```

        }
    }
    return -1;
}
@Override
public boolean validateFrame(byte[] buffer, int i){
    int message = buffer[i] & 0xFF;
    int data1 = buffer[i + 1] & 0xFF ;
    int data2 = buffer[i + 2] & 0xFF ;
    boolean messageType = message > 95 && message < 127;
    boolean checkData1 = (data1 > 31) && (data1 < 96);
    boolean checkData2 = ( data2 > 31) && (data2 < 96);
    if (messageType && checkData1 && checkData2)
        return true;
    else
        return false;
}
@Override
public synchronized SensorData parseFrame(byte[] readBuff){
    int messageType = readBuff[0] & 0xFF;
    if (messageType == 114) {
        if (value != null) {
            String hertratevalue = hrd.toString();
            sd.setHeartRate(hertratevalue);
        }
        ...
    }
}

```

3. Schnittstelleorientiert-Entwurf

Nachdem die konkrete Analyse-Klasse vorhanden ist, kann im nächsten Schritt das auf die Schnittstellen orientierte Design realisiert werden. Wie in Abbildung 4-12 zu erkennen, zählen sowohl der Datenparser für den Wahoo Sensor als auch der Datenparser für Equivital zur Subklasse der Datenparser-Schnittstelle. In der Klasse SensorManager sind die notwendigen Datenparser registriert. Nachdem der Datenanalyseprozess die Rohdaten erhalten hat, unterscheidet er zunächst anhand der im SensorManager eingetragenen DeviceID die Art des Sensors. Danach analysiert er anhand der im Datenparser eingetragenen konkreten Methode die Daten. Der Applikation-Layer muss später nicht mehr nachvollziehen können, auf welche Weise die Rohdaten von Sensoren konvertiert wurden. Dies bedeutet auch, dass bei einem neuen Sensor nur eine Analysedatei dafür geschrieben und die dazugehörige DeviceID beim Sensor Manager eingetragen werden muss. Auf diese Weise kann der Sensor in die Applikation-Layer der IPM-MC problemlos integriert werden.

4. Datenpuffer

Zuletzt speichert der Datenanalyseprozess die analysierten Daten im neuen Puffer (Byte *sensordata*[]) ab. Diese gehen damit in den Datendisplay- und Fernübermittlungsprozess über.

4.4 Datenvisualisierung und Aktualisierung

Nach Abschluss der Datenanalyse benachrichtigt das System den Displayprozess, die Rohdaten vom Sensor graphisch darzustellen. Auf dem Mobile Client ist eine schnelle Reaktionszeit eines der wichtigsten Design-Ziele. Jedes Android-Anwendungsprogramm läuft unter eigener Dalvik Virtual Machine [99]. Wenn der Prozess beginnt (die Anwendung startet), wird ein Hauptthread (*MainThread*) gestartet. Dieser Hauptthread ist verantwortlich für die Behandlung der mit dem UI in Zusammenhang stehenden Ereignisse. Daher wird der Hauptthread für gewöhnlich auch als UI-Thread bezeichnet. Android garantiert diese schnelle Reaktionszeit der Anwendungsprogramme durch den Einsatz von Singleton UI-Threads. Sämtliche Aktionen der UI-Threads müssen im Hauptthread durchgeführt werden. Ein Nicht-UI-Thread kann die UI nicht direkt beeinflussen. Zeitraubende Netzwerkübertragungen, komplexe Berechnungen oder Datenbanksystemoperationen etc. müssen im Arbeitsthread oder in Services behandelt werden, da sie ansonsten einen „Application Not Responding“ (ANR) Fehler hervorrufen.

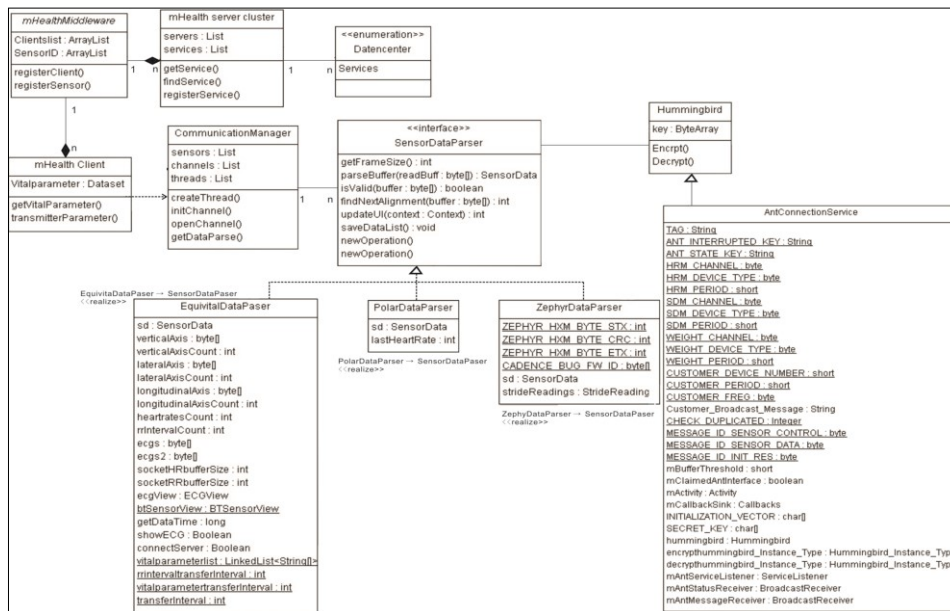


Abbildung 4-12 UML-Klasse-Diagramm der Schnittstellenorientierten Sensordatenanalyse

Im IPM-MC verantwortet der UI-Thread die Überwachung und Kontrolle der physiologischen Parameter in Echtzeit. Dies beinhaltet die Darstellung der physiologischen Daten. Da der Empfang und die Verwaltung der Daten nicht im UI-Thread erfolgen kann, öffnet IPM-MC (nachdem ein Sensorgerät eine Verbindung mit dem IPM-MC aufgebaut hat) einen

Arbeitsthread (WorkerThread) zum entsprechenden Bluetooth-Gerät, welcher für den Empfang und die Verwaltung der Daten zuständig ist. Die Daten-Interaktion zwischen dem UI-Thread des IPM-MC und dem Arbeitsthread basiert im Wesentlichen auf dem Handler-Message-Mechanismus vom Android. Bei IPM-MC wurde der Message-Handler als Variante der Rückruffunktion verwendet, um den Aufrufenden zu benachrichtigen.

Der Handler-Message-Mechanismus des IPM-MC beinhaltet im Wesentlichen die vier Teile: *Message*, *MessageQueue*, *Looper* und *Handler*.

- 1) *Message*: ist eine Klasse, in welcher Informationen der Nachrichten aufgezeichnet werden. Diese Klasse besitzt einige wichtige Attribute.
- 2) *MessageQueue*: In *MessageQueue* werden Warteschlangen von *Message*-Objekten abgelegt, die Ablege-Methode beruht auf der FIFO-Regel. Das *MessageQueue* Objekt wird durch das *Looper* Objekt verwaltet und gesichert, welches vom System automatisch generiert wird. Ein Thread kann maximal über eine *MessageQueue* verfügen. *MessageQueue*-Objekte können über die Methode *myQueue()* von *Looper* erhalten werden.
- 3) *Looper*: wird zum Verwalten und Sichern der *MessageQueue*-Objekte verwendet. In einem Thread korrespondieren die *Looper*-Objekte und *MessageQueue*-Objekte. Ein UI-Thread generiert standardmäßig ein *Looper*-Objekt. Über die *getMainLooper()* Methode des *Looper*-Objekts kann die laufende *Looper*-Instanz des UI-Threads erhalten werden.
- 4) *Handler*: verantwortlich für die Verwaltung der Nachrichten. Über *Handler*-Objekte werden *Message*-Objekte abgerufen und danach über die Methode *sendMessage(msg)* das *Message*-Objekt der *MessageQueue* hinzugefügt. Die *handleMessage()* Methode des *Handler*-Objekts verwaltet nach der FIFO-Regel die *Message*-Objekte[100].

Der EKG-Zeichner im IPM-MC verwendet die *drawline()* Methode von Canvas in der Android 2D Bibliothek. Diese Methode wird von der Methode *handleMessage()* des Handlers aufgerufen. Nachdem die Daten in einen 256 Byte-Puffer geschrieben wurde, gibt *drawline()* diese in einer Zeichenoberfläche mit der Größe 1000*256 aus. Durch das Überschreiben der alten Datenblöcke mit neu gesammelten Datenblöcken wird die Wirkung einer sich bewegenden EKG-Zeichnung erzielt. Siehe Abbildung 4-13.



Abbildung 4-13 EKG-Diagramm vom IPM-MC

4.5 Netzwerkkommunikationsmodul

Nach der Darstellung der Daten ist der nächste Schritt die Übermittlung der physiologischen Parameter vom IPM-MC an das Telemonitoring-Zentrum zur Abspeicherung. Das Netzwerkkommunikationsmodul ist ein besonders wichtiger Teil im Entwurf des IPM-MC, denn das Netzwerkkommunikationsmodul macht eine Echtzeit-Übermittlung der physiologischen Parameter des Patienten an das zentrale Monitoring-Zentrum möglich und das gesamte mHealth Client-Server-System zu einer organischen Einheit. Die Hauptaufgabe des Netzwerkkommunikationsmoduls ist die drahtlose Kommunikation mittels GPRS/UMTS/WIFI [101](GPRS verfügt theoretisch über eine maximale Download-Geschwindigkeit von 85.6 kbps und eine maximale Upload-Geschwindigkeit von 42.8 kbps) und die Übermittlung der Daten an das IPM-Server. Hierdurch ist es dem Monitoring-Zentrum möglich, parallel den Gesundheitszustand mehrerer Patienten zu überwachen und auf etwaige im Netzwerk auftretende Ereignisse schnell zu reagieren. Das Netzwerkkommunikationsmodul garantiert die Zuverlässigkeit der drahtlosen Kommunikation.

Wenn das IPM-MC eine Datenübertragung durchführt, wird eine TCP-Socket-Verbindung zwischen IPM-MC und mHealth Server erstellt. Um mit dem zentralen Monitoring-Zentrum eine Netzwerkverbindung aufzubauen, ist zunächst eine Authentifizierung nötig, danach wird eine Anfrage gesendet und auf Antwort des Monitoring-Zentrums gewartet. Sofern eine Zugriffserlaubnis, „SOK“, empfangen wird, wird mit dem Upload der Patientendaten begonnen. Wird eine Zugriffsverweigerung, „SNO“, empfangen, wird der Prozess abgebrochen und in den nächsten „Informationsloop“ übergegangen. Wenn sich der Patient

aus dem Netzwerk ausloggen muss, wird dem Monitoring-Zentrum eine Logout-Information „CQT“ gesendet und danach die Socket-Schnittstelle geschlossen. Es wird ein Informationsloop gestartet und auf den nächsten Start des Netzwerks gewartet. Das Funktionsverlaufdiagramm ist Abbildung 4-14 zu entnehmen.

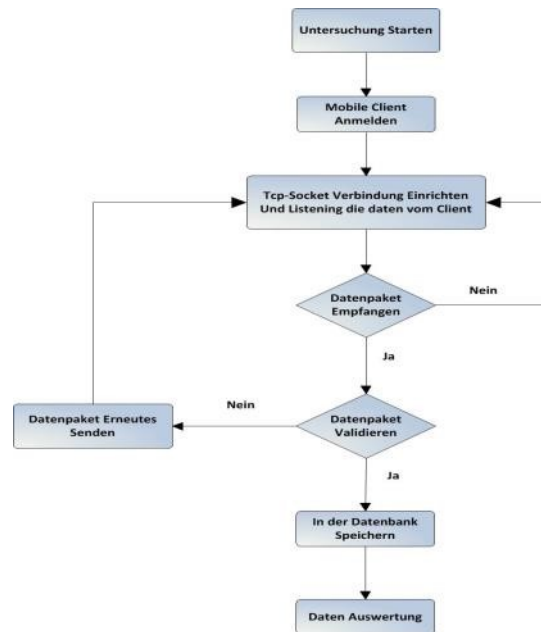


Abbildung 4-14 Übermittlung der physiologischen Daten vom IPM-MC an den IPM-Server

1. Authentifizierung des Sensors

Um Datenlecks und unrechtmäßige Verbindungen durch andere Geräte im drahtlosen Netzwerk zu verhindern, wird ein geheimer Schlüssel für den Zugriff auf den Erfassungsclient für physiologische Daten verwendet. Entsprechend der Erkenntnisse der Analyse von drahtlosen Sensornetzwerken aus Abschnitt 2.2 werden gewöhnlich die Daten eines Erfassungssensors nur an ein IPM-MC übermittelt. Es handelt sich um eine eindeutige Entsprechung [102].

2. Server-Login, durchführen der Identifikation

Um eine vernünftige Kontrolle der Anwendungsbefugnis beim IPM-MC und Login ins Monitoring-Zentrum zu erreichen, wird eine Authentifizierung mit Nutzernamen und Passwort beim Aufbau einer Verbindung zwischen IPM-MC und IPM-ZS benötigt. Dies soll verhindern, dass unberechtigte Clients Zugriff auf das Monitoring-Zentrum erhalten und unberechtigterweise an die Daten gelangen oder diese beschädigen.

3. Einrichten des Netzdienstthreads Service

Nach der Authentifizierung startet das IPM-MC den Socket Service Thread im Hintergrund und behält die Überwachung und Übermittlung der Daten bei. Service ist eines der vier von Android angebotenen Hauptmodule. Es handelt sich um eine Art von Serviceprozedur, welche normalerweise diskret im Hintergrund läuft. Sie besitzt daher oftmals kein UI und es kommt zu keiner Interaktion mit dem Nutzer. Dies beeinflusst in keiner Weise den Betrieb anderer Prozeduren.

4. Einrichten einer TCP-Socketverbindung

Dieser Thread ruft die API-Funktion von Android Socket auf, um die Kommunikation zu bewerkstelligen. Der Android Sockets Standard definiert, wie dieses Internetprotokoll verwendet wird [103]. Um die Effizienz der Fernübertragung zu erhöhen, wurde das Netzwerkmodul so geschrieben, dass der IPM-MC 50 Datenpakete zu einem Dateninhalt der Fernübertragung macht und ein Datenpaket von 8 Byte zum Server sendet. Im Verlauf der Übertragung können die Datenpakete auch in mehreren Teilen übertragen werden, um sicherzustellen, dass beim Sendevorgang keine Datenpakete verloren gehen, wodurch Korrektheit und Zuverlässigkeit garantiert wird. An dieser Stelle verwendet diese Arbeit die Funktion *SendPacket()*. Die Funktion sendet vor dem Versenden der eigentlichen Datenpakete die Information über die Datengröße an den Server. Dadurch kann das Gegenüber entsprechend der Größe der Datenpakete erkennen, ob beim Empfangsvorgang Datenpakete verloren gegangen sind, was einen partiellen Verlust von Datenpaketen verhindert. Die Funktion *Sendpacket()* wird wie folgt definiert:

```
SendPacket(intsocket, intbuffer intdwBuffer)
```

Der Parameter *Socket* bestimmt denTCP-Socket des Verbindungsaufbaus, über eben diesen Socket werden die Daten gesendet. Der Parameter *Buffer* bestimmt den Pufferbereich der gesendeten Daten. Der Parameter *dwBuffer* bestimmt die Länge der zu versendenden Daten. Ist der Wert der Antwort „True“, dann war die Datenübertragung erfolgreich; ist er „False“, dann war die Übertragung nicht erfolgreich. Der Arbeitsprozess des Fernübertragungsthreads ist Abbildung 4-15 zu entnehmen.

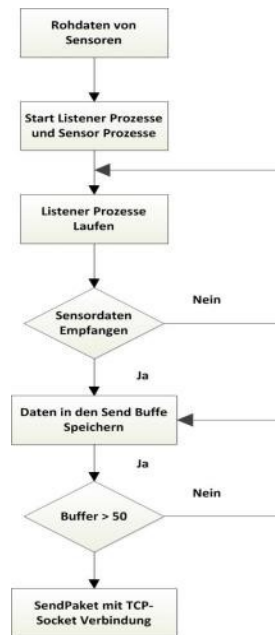


Abbildung 4-15 Arbeitsprozess des TCP-Socket Fernübertragungsthreads

5. Temporärer lokaler Speicher.

Wenn keine Netzwerkverbindung besteht, werden die Daten temporär im lokalen Datenbanksystem des IPM-MC gespeichert. Nachdem die Netzwerkverbindung wiederhergestellt ist, lädt der Client die lokal gespeicherten Daten hoch und stellt die Synchronisierung der Daten wieder her. In dieser Arbeit wurde das in Android integrierte Datenbanksystem Sqlite verwendet, um die Informationen abzuspeichern [104]. Zunächst wird eine Hilfsklasse für das Datenbanksystem eingerichtet. Diese Klasse erbt von der System-API. Die Version des Datenbanksystems ist 1.0, die Bezeichnung der Tabelle ist physiologicalMessage. Nachdem die relevanten Variablen definiert wurden, wird die Datenbank eingerichtet. Mit der Methode createDB() wird eine SQL-Anfrage durchgeführt, mit der Methode onUpgrade() wird die Tabelle aktualisiert. Beispielsweise kann die Tabelle über die Methode createDB() wie folgt gehandhabt werden:

```

private SQLiteDatabase openDatabase()
{
    try
    {
        if (!path.exists()) {
            path.mkdirs();
        }
        if (!f.exists()) {
            try {
                f.createNewFile();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
    if (database != null)
        return database;
}

```

```

        database = SQLiteDatabase.openOrCreateDatabase(f, null);
        returndatabase;
    }
    catch (Exception e)
    {
        System.out.println(e.getMessage());
    }
    returnnull;
}

publicstaticvoid createDB(){
    try {
        System.out.println("create a database");
        if(!isTableExist("daten")){
            database.execSQL("create table daten(id INTEGER PRIMARY
KEY AUTOINCREMENT, untersuchung_id varchar(20), time
varchar(50),accX varchar(50),accY varchar(50),accZ
varchar(50),belastung varchar(50),subject varchar(50))");
        }
        if(!isTableExist("rrintervaldaten")){
            database.execSQL("create table rrintervaldaten(id
INTEGER PRIMARY KEY AUTOINCREMENT, untersuchung_id
varchar(20), time varchar(50),rrinterval
varchar(50),belastung varchar(50),subject varchar(50))");
        }
        if(!isTableExist("subject")){
            database.execSQL("create table subject(id INTEGER PRIMARY
KEY AUTOINCREMENT, subject_id varchar(20), study_id
varchar(20),study_unique_id varchar(50))");
        }
        if(!isTableExist("belastung")){
            database.execSQL("create table belastung(id INTEGER PRIMARY
KEY AUTOINCREMENT, name varchar(20), study_id
varchar(20),beschreibung varchar(50))");
        }
        if(!isTableExist("fragebogeninfo")){
            database.execSQL("create table fragebogeninfo(id INTEGER
PRIMARY KEY AUTOINCREMENT, fragebogen_id INTEGER, name
varchar(20), filename varchar(20), study_id varchar(20))");
        }
        ...
    } catch (SQLException e) {e.printStackTrace();
}
}

```

Entwurf des Kommunikationsprotokolls

Im oberen Abschnitt wurde der auf TCP-Socket basierende Kommunikationsmechanismus zwischen IPM-MC und TPM-ZS vorgestellt. Um die Daten austauschen zu können, muss auf beiden Seiten noch die Bedeutung der Daten verstanden werden. Daher muss für die übermittelten Daten auch ein Informationsformat definiert werden. Die Informationen im System unterscheiden sich im Wesentlichen in zwei Arten: Lokalinformationen, welche vom IPM-MC kommen und Kontrollinformationen, welche vom Monitoring-Zentrum generiert werden. In diesem Abschnitt wird für beide Arten von Informationen ein Datenformat entworfen.

Entwurf eines Formats für die Informationen der physiologischen Parameter

Nachfolgend findet sich ein Teilbild des Entwurfs des Formates für die Datenpakete der physiologischen Parameter (siehe Tabelle 4-1). Es enthält die *Synchronisierung Byte*, *Länge der Datenpakete*, *Message-ID*, *Client-ID*, *Zeitstempel*, *Daten*, und *Frame-Ende*. Der konkrete Inhalt lautet wie nachfolgend beschrieben.

Tabelle 4-2 Formates für die Datenpakete der physiologischen Parameter

Synchronisierung Byte	Frame-Länge	Message-ID	Client-ID	Zeitstempel	Daten	Frame-Ende
2	2	4	2	4	n	2

Byte-Reihenfolge	Name	Byte-Länge	Wert	Beschreibung
0-1	Synchronisierung Byte	2	0000	Integer Type für die Framesynchronisation
2-3	Frame-Länge	2	16+n	Tatsächliche Datenlängen
4-7	Message-ID	4	l	Abfrage des Gerätezustands (Server -> Client)
			h	Messungswert : Herzfrequenz
			t	Messungswert : Temperatur
			b	Messungswert : Atemfrequenz
			r	Messungswert : R-RIntervall
			e	Messungswert : ECG
			f	Messungswert : Beschleunigung
		
8-9	Client-ID	2		Mobile Client-ID + SensorID
10-13	Zeitstempel	4		Millisekunden seit 1970
14-(14+n)	Daten	n	[Wert]	Tatsächliche Übertragener wert entsprechende Message-ID
(14+n)-(16+n)	Frame-Ende	2	1111	Frame-Ende-Marker

Entsprechend des in dieser Arbeit festgelegten Kommunikationsprotokolls zwischen IPM-MC und IPM-ZS ergibt sich bei der Körpertemperatur als Beispiel folgende Datenstruktur:

```
#define TEMPRETURE 3 //Temperatur in Grad
#define BLOODPRESS 4 // Blutdruckdaten
```

```

#define HEARTRATE 5           //Herzfrequenzdaten
#define TEXTMESSAGE 8       //Versionsdaten
typedef struct _MSGTempData
{ int Head;                 //Synchronisierung Anfang
  int Nlength;              //Anzeige der Größe des Datenbereichs
  int Type = TEMPRETURE;    //Datenart
  char Deviceid[10];        //Nummer des überwachten Endgeräts
  CTime tm;                 //Zeit der Datenerfassung
  float temp;               //Temperaturdaten
  BYTE End;                 //Synchronisierung Ende
} MSGTempData, *LPMSG MSGTempData;

```

Die Erfassungsclient-ID wird zur Identifikation jedes einzelnen IPM-MC für physiologische Parameter verwendet. Der Erfassungsclient und der Patient sind eindeutig einander zuzuordnen. Verschiedene Patienten verfügen über unterschiedliche Erfassungsclient-IDs, daher kann anhand der Erfassungsclient-ID der jeweilige Patient identifiziert werden. Der Zeitstempel besteht aus den folgenden zwei Teilen: *Datum* und *Uhrzeit*. Dieser dient der zeitlichen Eindeutigkeit der Informationen. Er bietet dem Datenserver eine zeitliche Orientierung für die Ablage der physiologischen Parameter und stellt dem medizinischen Personal eine zeitliche Basis für die genauere Diagnose des Patienten bereit. *Message-ID* ist die Art der gesammelten Parameter, *Daten* ist der konkrete Zahlenwert. Die geographische Positionsbestimmung z.B. erfolgt über das im IPM-MC integrierte GPS und unterteilt sich in die zwei Teile: GPS-Informationen und konkrete Positionsinformationen. Dies erlaubt dem medizinischen Personal anhand der geographischen Informationen dem Patienten rechtzeitig helfen zu können.

Entwurf des Formats der Kontroll- und Verwaltungsbenachrichtigungen

Nicht nur das IPM-MC übermittelt Daten an den Server, unter bestimmten Umständen kann auch das Telemonitoring-Zentrum über Fernübertragung Befehle zum Starten oder Beenden eines Erfassungsgeräts an IPM-MC senden. Beispielsweise kann in einigen Anwendungen zur Überwachung der Umgebungstemperatur das Telemonitoring-Zentrum den Befehl zum Neustart des Temperatur-Sensors ausgeben. Kontrollbenachrichtigungen unterteilen sich hauptsächlich in die folgenden zwei Arten: *Zustandsabfrage* und *Steuerung*, wie in Abbildung 4-16 zu sehen. Die Hauptfunktionen sind folgende:

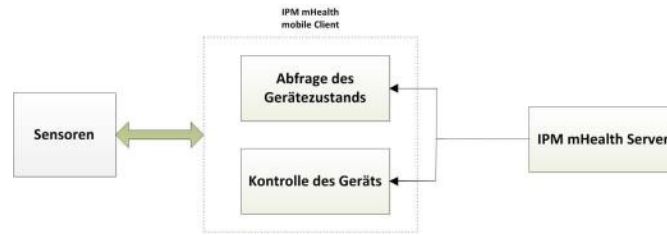


Abbildung 4-16 Kontrollbenachrichtigungen des IPM-Servers

1) Funktion der Abfrage des Gerätezustands.

Um die Geräte zusteuern, müssen zunächst die Informationen über den Zustand des Geräts abgefragt werden. Es wird überprüft, ob das Gerät aktiv ist, ob es normal funktioniert und ob die Frequenz der übermittelten physiologischen Parameterinformationen den Anforderungen entspricht.

2) Funktion der Verwaltung und steuern des Geräts.

Neben der Steuerung von Start und Beenden des Erfassungsclients ist es ebenfalls noch nötig, dass entsprechend der Ergebnisse der Diagnose des Patienten durch das medizinische Personal, die Frequenz der Übermittlung durch die Erfassungsclients für physiologische Parameter, mit welchen der Patient ausgerüstet ist, angepasst wird und je nach Ergebnis eine Rückmeldung erfolgt.

Die oberen beiden Teile sind die Hauptaufgaben der Kontrollbenachrichtigungen. Der Entwurf des Formats der Datenpakete für Verwaltungs- und Kontrollbenachrichtigungen, welcher in dieser Arbeit verwendet wird, ist Abbildung 4-17 zu entnehmen. Dabei setzen sich Datenpakete aus *Geräteinformationen*, *Kontrollmethoden* und *Zustandscode* zusammen. Der konkrete Inhalt im Detail:

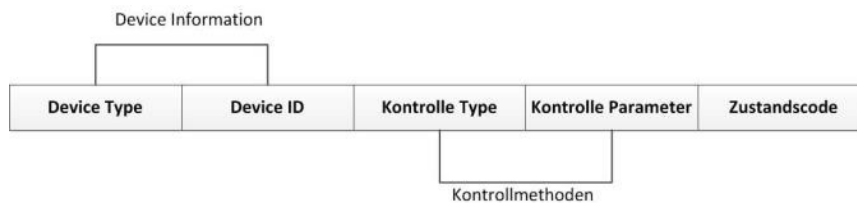


Abbildung 4-17 Formate der Datenpakete für Verwaltungs- und Kontrollbenachrichtigungen

1) Die Geräteinformationen setzen sich aus den zwei Bestandteilen Gerätetyp und Geräte-ID zusammen. An aktiven Geräten gibt es 2G/3G Modems, Netzwerkbrücken und WiFi-Sensoren mit geringem Stromverbrauch. Die Geräte-ID wird dazu verwendet, um ein aktives Gerät eindeutig zu identifizieren.

2) Die Kontrollmethode setzt sich aus den zwei Teilen *Operationstyp* und *Operationsparameter* zusammen. An Operationstypen gibt es die vier Arten: Gerät

starten, Gerät beenden, Gerät neu starten und die Regelung der Frequenz der Übermittlung. Die ersten drei Operationen setzen einen Standardwert. Der korrespondierende Operationsparameter der vierten Art von Operation ist die neue Frequenz der Übertragung.

- 3) Der Zustandscode ist das Ergebnis der oben erwähnten Verwaltung und Kontrolle, ob z.B. eine Operation erfolgreich war oder fehlgeschlagen ist, durch diese Information kann der IPM-ZS die Anfrage wiederholen.

4.6 Management-Modul für Untersuchungen

Das Untersuchungsmanagement-Modul besteht hauptsächlich aus den folgenden zwei Teilen: der Verwaltung der Monitoring-Aufgabe und der Verwaltung der Fragebögen. Der Inhalt dieser zwei Teile wird vollständig über das im obigen Abschnitt vorgestellte Netzwerk kommuniziert. Der IPM-MC erhält vom Server Detailinformationen. Die Verwaltung der Untersuchung definiert hauptsächlich die Hauptmerkmale der Monitoring-Aufgabe. Dies beinhaltet den Typ der zu sammelnden physiologischen Parameter, die Dauer des Monitorings, die Datenverarbeitungsmethode des Servers, etc. Diese Informationen werden durch Ärzte und Forscher definiert und über das Monitoring-Zentrum an den Client gesendet. IPM-MC synchronisiert bei jedem Start zunächst diese Informationen mit dem Server. Die Fragebögen dienen im Wesentlichen dem Sammeln von Informationen über den emotionalen Zustand des Patienten während des Verlaufs des Monitorings. Diese Art von Formular wird ebenfalls von Forschern im Monitoring-Zentrum entworfen und auf den IPM-ZS hochgeladen. Sobald eine Verbindung zwischen IPM-MC und IPM Server besteht, werden diese Informationen synchronisiert. Dabei ist der Fragebogen im System als XML-Format definiert. Nachdem der IPM-MC die Beschreibungsdatei des XML-Formulars erhalten hat, wird diese in eine entsprechende Android-UI umgewandelt. Der konkrete Vorgang wird in folgender Abbildung 4-18 gezeigt:



Abbildung 4-18 Vorgang der Synchronisierung von Untersuchungsinformationen zwischen IPM-MC und IPM-ZS

1. Zuerst erarbeiten Forscher einen Fragebogen auf der Basis von XML
2. Dieser wird ins System des Monitoring-Zentrums hochgeladen.

3. Der IPM-MC synchronisiert die Informationen mit dem Monitoring-Zentrum und lädt die zugehörige XML-Formulardatei herunter.
4. Dann liest der IPM-MC die Felder in der XML-Datei aus. Entsprechend der Eigenschaften der Felder wird eine entsprechende Android-Benutzeroberfläche eingerichtet, z.B. textfield, radiobutton, etc. (siehe nachfolgende Abbildung 4-19).
5. Nachdem das Ausfüllen abgeschlossen ist, werden die Werte nach Maßgabe des im oberen Abschnitt vorgestellten Formats über das MET-Kommunikationsmodul an das Monitoring-Zentrum übermittelt.



Abbildung 4-19 IPM-MC-Fragebogen

5 Einführung in SOTH: SOA-basiertes offenes Telemonitoring framework für Healthcare-Anwendungen

Im vorherigen Kapitel wurden die lokale Seite des mHealth-Systems, die Struktur des drahtlosen Sensor-Netzwerks und das IPM-MC erörtert. In diesem Kapitel wird das SOA [105] (Service-Orientiertes-Architektur)-Konzept verwendet, um ein Framework auf offenen Strukturen zu definieren, welches Ingenieure und medizinisches Personal anleitet, auf einer elektronischen Serverplattform den mHealth-Dienst zu entwerfen. In dieser Plattform werden die medizinischen Dienstleistungen als Ergebnis der abstrakten Anordnung von verschiedenen Layern betrachtet; dabei wurden verschiedene medizinische Dienste lose gekoppelt, um die Interoperabilität zwischen den Diensten zu erhöhen.

Heute strengen weltweit bereits viele Staaten Forschungen im Bereich des Telemonitorings an. Auch Deutschland verfügt diesbezüglich über zahlreiche Projekte, welche bereits verwendet werden oder sich in der Entwicklung befinden [106]. In dieser Arbeit wird ein Ablauf von zu integrierendem e-Service im Bereich des Telemonitoring beschrieben. Diese Beschreibung berücksichtigt die normalen Anforderungen in diesem Bereich, wodurch dem System große Flexibilität verliehen wird, um sich an die Veränderungen im medizinischen Dienst anzupassen. Gleichzeitig muss auch der derzeitige Stand im IT-Bereich berücksichtigt werden. Es wird die SOA-basierte Architektur für mHealth-Anwendungen SOTH (SOA based Open Telemonitoring framework for Healthcare) vorgeschlagen. SOTH kann für die Konzeption eines gesamten medizinischen Business-Prozesses verwendet werden, genauso ist es auch möglich, zur bestehenden Architektur nachträgliche Module hinzuzufügen, welche dem System neue Eigenschaften bieten.

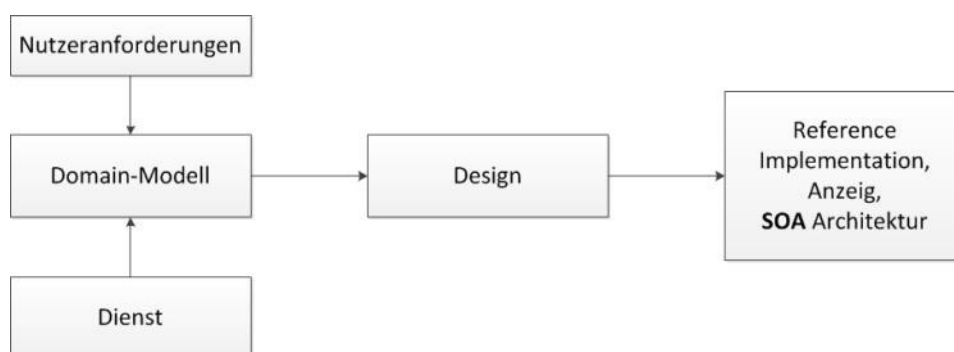


Abbildung 5-1 Hierarchische Domainmodelle, die Brücke zwischen Anforderungen und den Diensten

Auf Grundlage der Anforderungen der Benutzer werden in dieser Arbeit die Software-Dienste definiert, welche im Bereich der medizinischen Pflege verwendet werden können. Ein

hierarchisches Domain-Modell, welches dieser Abschnitt vorschlägt, stellt die Brücke zwischen den medizinischen Anforderungen und den Diensten dar (Abbildung 5-1). Die praktische Umsetzung dieser Dienste hängt von den Regeln im Domain-Modell und den Anforderungen der Nutzer ab, wonach sie letztendlich entworfen und umgesetzt werden können.

5.1 SOTH : SOA-basiertes offenes Telemonitoring framework für Healthcare

Infolge der in jüngster Zeit auftretenden plattformunabhängigen und plattform-übergreifenden Datenmodelle wurde dem wesentlichen Gedanken im SOA-Konzept immer mehr Beachtung geschenkt [107]. SOTH bietet eine Methode, um entsprechend einer praktischen Aufgabe die Ressourcen zu organisieren. Es verpackt diese Aufgabe in komplexere Module. Diese Module bilden die medizinische Dienste [108] ab, definieren die Schnittstellen zwischen Dienstleistungsanbietern/Nutzern und errichten ein Protokoll zur Darstellung der Dienste (vergleichbar mit der Organisation der menschlichen Gesellschaft). In dieser Struktur kann der Service als eine *„adressierbare und klar definierte, leicht anwendbare, Schnittstellen standardisierte Netzwerk-Entity“* betrachtet werden. Vom technischen Standpunkt aus betrachtet ist SOA eine Ansammlung von Services, welche durch das Netzwerk den Datenaustausch und Interoperationen durchführt. Eine Anwendung wie SOA scheint am wahrscheinlichsten die Anforderungen eines komplexen und aus heterogenen Elementen zusammengesetzten Informationssystems, wie es im telemedizinischen Pflegebereich der Fall ist, erfüllen zu können [109].

Durch die Verbindung verschiedener Dienste integriert SOTH lediglich Systeme, Daten, Anwendungen und Prozesse und resultiert in einem flexibleren Entwurf. Bei der Umsetzung können ergänzende Plattformen realisiert oder eine gesamte SOA-Plattform errichtet werden, welche die ursprüngliche ersetzt, um die interne Interaktion der Business-Dienste zu unterstützen.

In der SOTH-Architektur wird die Beschreibung der medizinischen Dienste in zwei Layer unterschieden: Ein Layer ist näher an der **Technologie** und der andere Layer näher am **Business** angesiedelt. Daher werden für jeden Layer Attribute aufgeführt, gleichzeitig werden zwei separate Registrations-Module etabliert (intern und extern). Dies wird im folgenden Abschnitt genauer erörtert.

5.2 Granularität

Das in diese Arbeit verwendete SOA-Konzept betrachtet die mHealth Einheiten als das Ergebnis der abstrakten Anordnung verschiedener Layer, wodurch dem System eine höhere Flexibilität verliehen wird. Nach dieser Methode kann ein abschließender Dienst (*direkt ausgerichtet auf den Endverbraucher*) mehrere (*möglicherweise parallel laufende*) Lebenszyklen besitzen (*jeder Zyklus geht von einem unterschiedlichen abstrakten Layer aus*). Nachfolgend wird die Rolle jedes abstrakten Layers dargelegt und wie diese zusammenwirken, um einen Service umzusetzen [110].

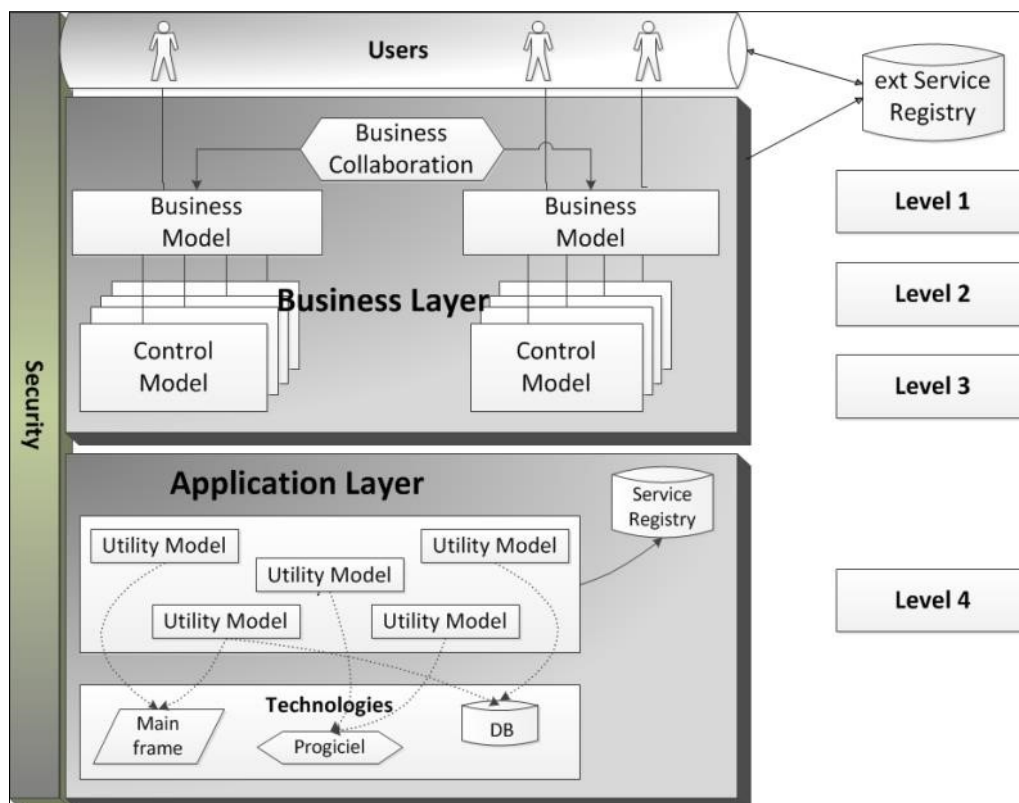


Abbildung 5-2SOTH-Architektur

Die Gesamtstruktur entspricht Abbildung 5-2. Die Elemente der Struktur werden in zwei Layer unterteilt: *Business Layer* und *Applikation Layer*. Jeder Layer befindet sich auf dem Modell eines anderen Levels. An dieser Stelle wird, um die Struktur besser erläutern zu können, das Konzept von „Layer“ und „Level“ unterschieden. Jeder Layer enthält eines oder mehrere Level. Die Elemente der Struktur werden in insgesamt fünf abstrakte Level unterteilt: Der oberste erste Level beinhaltet die Koordination der Business-Strategie, der unterste fünfte Level beinhaltet sämtliche verwendeten Technologien. Der Austausch von Messages (*Anfragen, Antworten*) zwischen den Elementen folgt dem Fluss von oben nach unten, anders gesagt kann das Element eines bestimmten Layers nur Dienste und Funktionen des Elements

des nächsten unteren Levels anfragen. Dieses Prinzip vermeidet komplexe Abhängigkeiten und Kommunikationen zwischen den Elementen.

Nachfolgend werden die Level 1 bis 5 der Architektur erörtert. Die ersten vier Level beinhalten: *Business Collaboration Level*, *Business-Level*, *Control Service* und *Application Service Level*, wie in Abbildung 5-2 zu sehen. Der Inhalt von Level 5 ist in den Ressourcen enthalten.

5.2.1. Business Collaboration Management Model (BCM)

Im ersten Level (dem obersten Level) wird der Koordinator der Kooperation zwischen verschiedene Business Modell festgelegt. Ein Business-Modell kann nicht direkt ein anderes Business-Modell anfragen (da sich dieses auf demselben Level befindet). Zwei oder mehr als zwei Business-Modelle müssen koordiniert werden, um einen Dienst-Zustand zu bewirken. Daher muss definiert werden, wie die jeweiligen Elemente bei der Ausführung des Dienstes mitwirken, z.B. „welches Element startet den Dienst und welches Element beendet den Dienst“. Das „*Business Collaboration Management*“ ist verantwortlich für die Beschreibung und Verwaltung dieser Koordination. Gewöhnlich müssen zwei Business-Modelle mit einem oder mehreren anderen Business-Modellen interagieren, ein ausführlicher Koordinationsplan wird verwendet, um ihre Anforderungen zu befriedigen.

Der Entwickler der Implementierung der Dienste muss der Beschreibung der Koordination zustimmen. Da jeder sein Gegenüber verstehen muss, ist die Verwendung einer normierten Beschreibungssprache sehr wichtig. In SOTH kann die verbreitetste Choreographie-Sprache WS-CDL [111] verwendet werden. WS-CDL kann eine präzise Koordination zwischen beliebigen Teilnehmern beschreiben, ungeachtet der unterstützten Plattform oder des Implementierungsmodells. Das Choreographie-Modell beschreibt die Koordination eines Dienstes zur Verwirklichung eines gemeinsamen Zieles. Über das Abfragen der Schnittstelle des Kooperationspartners wird die notwendige zu befolgende Interaktionssequenz (eingeschränkt durch die Beschränkungen) bestimmt. Diese Sprache muss die Abhängigkeiten zwischen den folgenden Interaktionen beschreiben können: *Abhängigkeiten der Ablaufsteuerung*, *Abhängigkeiten des Datenflusses*, *die gegenseitigen Beziehungen der Messages*, *zeitliche Einschränkungen*, *Abhängigkeiten von Transaktionen*, etc. Es ist darauf zu achten, dass der Choreographie-Plan mitnichten eine Beschreibung der internen Aktionen enthält, sondern im Gegenteil die Interaktion in der Gesamtheit im Blick hat. Anders

gesagt beinhaltet die Choreographie die gesamte am Dienst beteiligte Interaktion, und diese Dienste folgen dem Ziel der Choreographie.

5.2.2. Business-Model (BM)

Das Business-Model wird auf dem zweiten Level definiert und verbindet *Analysephase* und *Designphase*. Jede zugehörige Schnittstelle ist dabei auf den Nutzer ausgerichtet. Das Business-Model hat das Business als Zentrum und zeigt die vom Stakeholder definierten wesentlichen Funktionsgrenzen des Business und den internen Kontext jedes Business: wer sind die Endnutzer, eine Business-Analyse sowie eine Dienstarchitektur. Das Business-Model zeigt die formelle Formatierungsform des Inhaltes des Anforderungsdokuments. Dieser Inhalt ist mit der fünf-Layer-Architectureinfach verständlich. Das Modell dieses Levels wird dazu verwendet, die Business-Schnittstelle und Bedingungen (wie Umweltrestriktionen, Strategie, Sicherheit etc.) zu definieren, welche das Business ausführen können, enthält aber keinesfalls irgendwelche Einzelheiten zur Umsetzung.

5.2.3. Control Service Model (CM)

Das Control-Model beschreibt die Verbindung zwischen Business Level und Applikation Level, nämlich die Verbindung zwischen dem Business-Model und seiner Umsetzung. Bei der SOTH-Architektur handelte es sich um eine mittels Technologie strikt festgelegte Business-Logik. Jeder Business-Prozess wird von notwendigerweise auszuführenden Tasks gebildet, bis das erwartete Business-Ergebnis erreicht ist. Einige Prozesse sind sehr einfach und beinhalten nur wenige zugehörige Tasks. Andere wiederum können sehr komplex ausfallen und umfassen hunderte verschiedene Tasks. Eine optimale Methode zur Errichtung eines effektiven Business-Models kann auch für eine zukünftige Erweiterung verwendet werden. An dieser Stelle können normierte Marker und Beschreibungssprachen gewählt werden, um die Interaktivität zwischen den Modellen zu garantieren. In SOTH wird der Einsatz von BPMN [112] und BPEL [113] erwogen. Ersteres verwendet Tabellenmarkierungen, um die verstandenen internen Business-Prozesse darzulegen und eine Wahrscheinlichkeit für die gemeinsame Nutzung dieser Modelle zu bieten. Letzteres lässt Business (Anfragen) ihre Business-Prozesse beschreiben und wandelt diese danach semi-automatisch in ausführbare Informationssysteme um.

5.2.4. Utility Service Models

Dieser Level beinhaltet alle Modelle, welche zur Beschreibung der Dienste verwendet werden. Dienste sind auf diesem Level dargestellt als ausführbare Tasks. Um die Funktionen der

Tasks zu demonstrieren und die Umsetzung der Dienste anzuleiten, müssen diese Tasks ausführlich dargelegt werden. Die konkrete Beschreibungsdefinition ist in der Service Library abgespeichert. Der „Utility-Service“ organisiert und beschreibt entsprechend der Merkmale der beschriebenen Tasks (wie Architektur, Funktion, Aktion, Umgebung, Anwendung etc.) die Bildung eines Services durch die Tasks.

Das „Utility-Model“ ist näheran der Technik der Umsetzung und der Anwendungsumgebung. Es ist die Abstraktion der Ressourcenfunktion und bildet das Fundament des oberen Business-Layers. Das Utility-Model bezieht sich auf die interne Service Library. Der Prozessdesigner verwendet die interne Service Library, um das Control Modul (CM) konkret zu definieren.

Nachfolgend werden die verschiedenen konkreten Beziehungen zwischen den verschiedenen Modellen in der Vier-Layer-Struktur des SOTH erläutert.

5.3 Beziehungen zwischen den Modellen

In den vier abstrakten Level existieren zwei Arten von Service: Eine Art beinhaltet die Business-Logik (im Business-Layer) und die andere Art erläutert detailliert die Elemente der wesentlichen Dienste (im Applikation-Layer). Daraus resultiert möglicherweise die Notwendigkeit zweier unterschiedlicher Protokolle und Register. Die abstrakte Klassifizierung verleiht dem System eine größere Flexibilität, um technische und organisatorische Veränderungen zu behandeln. Auf diese Weise können unterschiedliche Gruppen von unterschiedlichen Standpunkten (Richtungen) aus Dienste und Anwendungen entwickeln. Beispielsweise können Anbieter und Anwender über das Protokoll ihre Anforderungen darlegen. Der Business-Designer kann nun wiederum über die gleichen Informationen das entworfene Modell organisieren. Diese Organisation der Aufgaben muss nicht Teil derselben Institution sein. In diesen beiden Fällen muss, um eine Koordination zwischen den verschiedenen Arbeitsgruppen zu erreichen, die Rolle jedes Teilnehmers und die Beziehungen zwischen ihnen genau definiert werden. Eines der Ziele von SOA ist es, durch das Bilden von Modulen und Beziehungen den Gefügebau des Konzeptes darzulegen.

Abbildung 5-3 zeigt, dass Dienste des gleichen Layers nicht miteinander kommunizieren können. In einem praktischen System ist dies allerdings oftmals nicht so: Die Wiederverwendbarkeit der Module bewirkt, dass der Konstrukteur nur sehr schwer den Austausch von Messages kontrollieren kann. Sämtliche Module können selbst Messages der

verschiedenen Layer senden und empfangen. Unter diesen Umständen ist es egal, in welchem Layer ein Dienst bewegt oder verschoben wird; es verändert das gesamte Modell. Auf der Grundlage, dass die Kommunikationsregeln nicht verletzt werden, wird durch die Einführung des Business Collaboration Management Modell (BCM) das wesentliche Problem, welches durch die Wiederverwendung von Diensten entsteht, gelöst. Wenn eine zuvor definierte Kooperation abgerufen wird, überwacht die BCM-Komponente das vom Control-Model erzeugte Ereignis. Während die Ausführung koordiniert wird, kontrolliert der BCM Task den Arbeitsablauf des Tasks. Wird allerdings diese Kooperation beendet, so geht die Kontrollbefugnis umgehend auf den Abrufenden des Dienstes über. Diese Methode wird verwendet, um den Grad der Klarheit, Schnelligkeit und Anonymität der Architektur zu erhöhen.

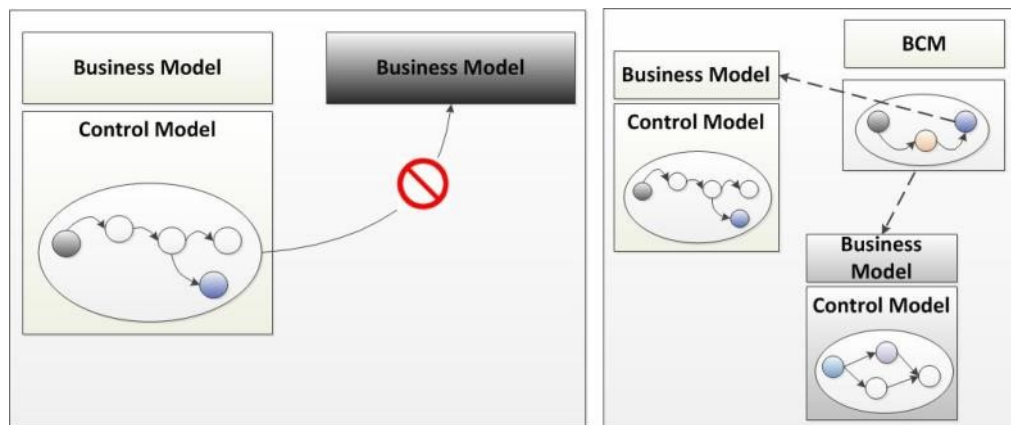


Abbildung 5-3 Kommunikation zwischen Diensten des gleichen Layers

5.3.1 Zusammenfassende Definition der Architektur

Die oben angeführten Punkte zusammenfassend, kann die Architektur wie folgt definiert werden: Im *Business Collaboration Management Model* wird die Aufgabe der benötigten Koordination der Business-Dienste definiert. Das Modell dieses Layers gibt die Aktionen aller am Koordinationsprozess beteiligten Geschäfte vor; der Austausch/die gemeinsame Nutzung von Daten verschiedener Business-Dienste ist ausführlich beschrieben; die notwendig zu kontrollierenden Ereignisse werden definiert; die Normen für die Kooperation mit anderen werden gebildet.

Das *Business-Model* ist die Darlegung der Business-Logik und definiert die speziellen Bedingungen, welche von den angebotenen Diensten verwendet werden. Sie geben dem Nutzer der medizinischen Dienste eine unmittelbare Darstellung (das Business-Model ist in der Registry eingetragen, der Nutzer greift über den Besuch der Registry auf das Modell zu). Das Modell dieses Layers definiert die Aktionen der wechselseitigen Beziehungen, um einen

Prozess zu unterstützen. Dieser Prozess wird aus der vordefinierten Technik, den Schnittstellen und offenen Normen gebildet. Es werden die eine Business-Logik berührenden Aktionen beschrieben (dabei können die Funktionen einer anderen Business-Logik nicht wiederholt werden); die Schnittstellen der Dienste werden definiert; es erfolgt eine Umsetzung über eines oder mehrere Control-Models; die Beschreibung dieses Levels kann als Referenz für den Dienstleitungsbereich dienen.

Das *Control-Model* ist die detaillierte Regelung des Business Models. Es definiert eine Reihenfolge von Tasks, welche die Business-Logik des übergeordneten Layers befriedigen. Die Umsetzung des Modells dieses Layers resultiert aus den Anforderungen des Nutzers, Prozessen, Zielen der Daten, Arbeitsabläufen und Normen. Es ist eine Beschreibung, welche eine Reihe von Prozessen und Arbeitsabläufen auf einen Dienst ausgerichtet zusammenfügt. Mindestens ein zentraler Dienst unterstützt diese Prozesse und Arbeitsabläufe. Es handelt sich um eine detaillierte Erläuterung der Regelungen des Business-Models. Es werden möglicherweise Dienste von anderen Business-Models angefordert.

Das Utility Service Model zeigt einem oder mehreren übergeordneten zentralen Services technische oder nicht-technische Details an. Diese Modelle beschreiben, wie jeder zentrale Dienst umgesetzt und verwendet wird: Das Modell des Layers ist ein auf Technik, Business-Prozesse, Wiederverwendbarkeit und Service ausgerichtetes Design. Es kann für einen oder mehrere Control-Models verwendet werden, es können einer oder mehrere Standards verwendet werden und es können die Funktionen anderer zentraler Service-Models wiederverwendet werden.

Auf dem letzten Level (Level 5) sind die praktischen Umsetzungen und Codes sämtlicher Anwendungen angeordnet. Enthalten sind (nicht ausschließlich) der Code des Utility-Models zum Erzeugen der Operationsfähigkeit, der Operationscode und die praktischen Umsetzungen der Konfiguration für die Umsetzung der Dienste.

5.4 Beschreibung der medizinischen Ressourcen und Dienste

Um in SOA die Dienste besser zu finden, zu verbinden und zu verwalten, müssen die verschiedenen Merkmale der Dienste gründlich analysiert und unterteilt werden. Dieses Kapitel zielt auf das zuvor vorgestellte SOTH-Framework und definiert eine Norm für die Darstellung der Merkmale der Dienste in den verschiedenen Modellen bezogen auf

die Architektur. Diese Darstellung wird in der Datenbank abgespeichert (Service Registry) und verwendet, um bereits existierende Dienste zu publizieren.

5.4.1 Beziehungen des Systemkonzepts

Es wird definiert, dass das gesamte System aus vier Bestandteilen besteht: *Resource*, *Service*, *Message* und *Policy* (vgl. Abbildung 5-4).

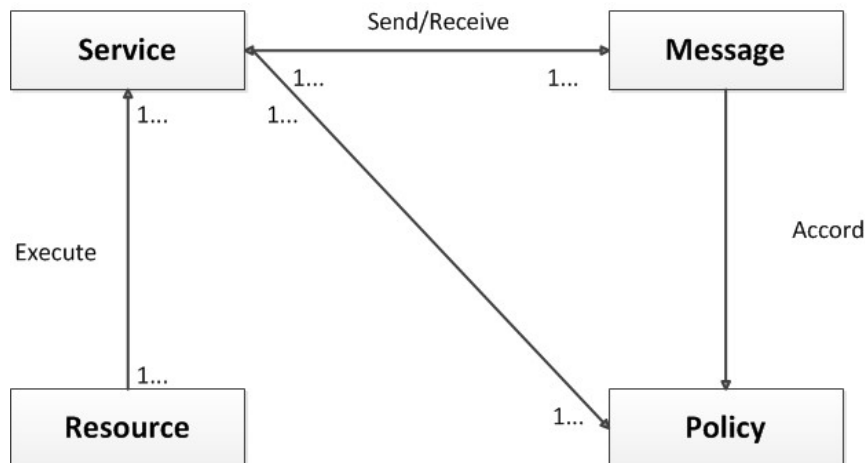


Abbildung 5-4 Konzept der SOTH-Architektur

Die Ressourcen in der Abbildung werden als physische Einheit zur Durchführung einer Reihe von logischen Aktionen betrachtet. Die jeweiligen Proxies fragen diese Aktionen an, um spezifische Tasks zu implementieren. Daher werden diese Ressourcen durch abstrakte Tasks klassifiziert. Wenn wir nun nach den zuvor definierten Zielen die Tasks unterscheiden, so werden diese Tasks als Webservice [114] bezeichnet. Ein Service kann ebenfalls durch andere Services gebildet werden. Sie können in Composed Service und Single Service differenziert werden. Wenn ein Dienst eine Message-Anfrage einer Anfrage-Utility erhält, so wird dieser Dienst aktiviert. Zu diesem Zeitpunkt entsteht zwischen Anbieter und Anfrager des Dienstes ein Kommunikationskanal durch die Übermittlung von Informationen über Message. Die Ausführung dieses Dienstes und der Austausch über Message befolgen die in den Policies definierten Regeln (Sicherheit, Preconditions und Postconditions).

Im Folgenden wird jedes Modul detailliert dargestellt. Der Schwerpunkt liegt an dieser Stelle jedoch auf den Service-Modulen. Dieses Kapitel folgt in der Beschreibung der Richtung von oben nach unten. Zunächst einmal werden das Ressourcen-Modul und seine Merkmale kurz vorgestellt. Danach werden die Details der Service-Module erläutert. In der Folge wird dann der Schwerpunkt auf die verschiedenen Layer und Level in der Architektur gelegt. Zuletzt wird dargelegt, wie der Dienst spezifische Darstellungen definiert.

5.4.2 Beschreibung der Ressourcen

Das Ressourcen-Modul ist die Grundlage der SOTH-Architektur wie auch die Grundlage der Dienste. Es wird verwendet, um die vom Nutzer des Systems angefragten Tasks aufzurufen (mittels der Dienste).

Abbildung 5-5 zeigt die vier Merkmale jeder Ressource: *URI* [115], *Autor*, *Ansicht* und *Beschreibung*.

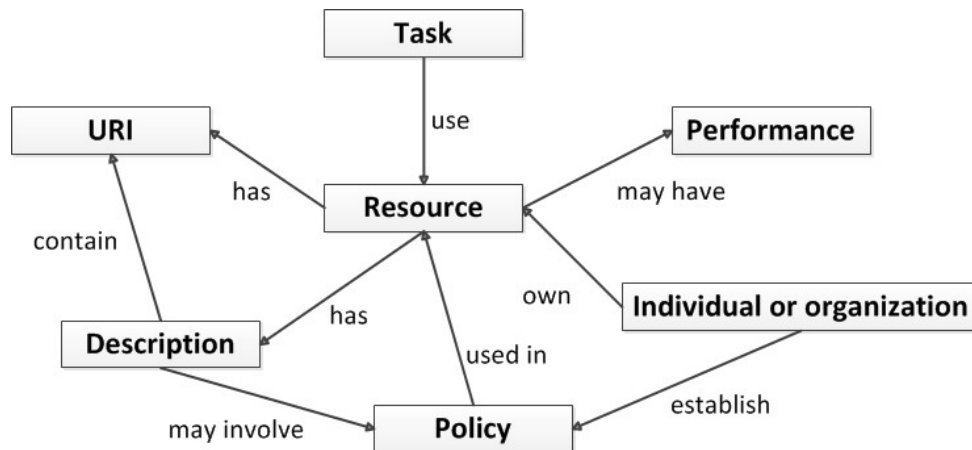


Abbildung 5-5 Konzept des Ressourcenmodelles

Der **URI** wird verwendet, um die Ressourcen zu identifizieren. Jede Ressource muss in der Architektur einen einzigartigen Namen besitzen. Der Autor einer Ressource kann sowohl ein Individuum als auch eine Organisation sein und verantwortet das Angebot und die Abfrage von Tasks.

Die **Performance** spiegelt eine Datenansicht des Zustands der Ressourcen wider.

Das wesentliche Ziel der **Description** der Ressourcen ist die Förderung der Auffindbarkeit und des Zugriffs auf die Ressourcen. Um dieses Ziel zu erreichen, kann die Beschreibung der Ressourcen die *Position*, *Zugriffsmöglichkeiten*, *sämtliche von den Ressourcen angebotene Funktionen*, *den Zweck der Anwendung*, *Postconditions* etc. enthalten. Diese Inhalte können mit der **Policy**, der Beschreibung der verwendeten Regeln, in Wechselbeziehung stehen.

Dieser Text unterteilt die **Ressourcen** in solche mit funktionalen und nicht-funktionalen Eigenschaften. Dies entspricht der beschriebenen Methode von [116] und [117]. Auch in [118] wird diese Methode verwendet, allerdings wird sie bezüglich der Ressourcen nur als Webserviceberücksichtigt, ob darauf von einem anderen Nutzer oder Dienst zugegriffen werden kann oder nicht. Auf der Grundlage der beiden vorher genannten Methoden

unternimmt diese Arbeit teilweise Änderungen. Nach den Änderungen kann die Beschreibung der Ressourcen den Dienstbeschreibungen aller Level entsprechen.

Die Beschreibung der Ressourcen umfasst insgesamt sechs Attribute: *inherent*, *functional*, *nonfunctional*, *cooperative*, *auxiliary* und *deployment*.

Das Attribut *inherent* beinhaltet Informationen der Ressourcen-Registry, wie Besitzer, Version und effektiver Zeitraum, etc. Das Ressourcen-Attribut *functional* beinhaltet die Schnittstellen begleitender notwendiger Pre- und Postconditions [119] und alles, was mit der Realisierung des Ressourcen-Modells zu tun hat. Ebenfalls enthalten sind zugehörigen Details, wie sämtliche Algorithmen, entworfene Modelle etc., Tabelle 5-1 zeigt eine Auswahl von für die Beschreibung der Ressourcen verwendeten Attributen. Für jedes Attribut werden drei Kerneigenschaften festgelegt: *ID(TagID)*, *Data Type* und *Description*. TagID beschreibt ausführlich den Namen des Attributs. Insgesamt gibt es vier Datentypen der Ressourcen: boolean, integer, float und string. Die dritte Spalte bietet eine Kurzbeschreibung des Attributs.

Tabelle 5-1 Attribute der Beschreibung der Ressourcen

Tag Id	Data Type	Description
Name	String	Resource Name
DomainName	String	Description of resource range
SystemName	String	Indicate resources belong to which system
Proprietary	String	Indicate resources belong to which Individual or organization
Description	String	Description language of resource
Computation Attributes		
Inherent Attributes		
Id	String	Unique resource identifier
Version	String	Version of the described resource
Author	String	Author of the described resource (Designer/Developer)
Date	String	Resource deployment time
Validity	Integer	Providing the life cycle of the described resource
Atomicity	Boolean	Indicate if the resource is original or not
Registration (mutiple)	String	Liste all registered of Components
Model	String	Indicate the xml description file address of system model
State	String	Indicate resources state according to the model
Functional(Mutiple)		
Function Name	String	Function Name, may be used by Other attributes
Function Description	String	Informal language of Functional description
Algorithm(complex)	Boolean	If this function use a special algorithm , will be noted here
*:Name	String	The name of Algorithm

In der Tabelle werden noch zusätzliche Symbole verwendet. Das „multiple“ in den Klammern zeigt an, dass es mehrere von diesen Attributen geben kann, „complex“ drückt aus, dass die Beschreibung eines Attributes andere Attribute benötigt. Unter diesen Umständen werden Sub-Attribute identifiziert.

5.4.3 Beschreibung von Diensten

Im ersten Abschnitt werden Ressourcen (und die zugehörigen Attribute) als virtuelle Module betrachtet. Zudem wird eine Architektur der Beschreibung der Ressourcen angeboten. Diese Architektur kann als Output des fünften Levels betrachtet werden. Die Tasks der Software Service Agents und Anfragen der Nutzer implementieren die Koordination und sind gleichzeitig zuständig für die Verwaltung der Ressourcen. Anders gesagt, um das vorher ausgearbeitete Ziel zu erreichen, regelt und verwaltet ein Agent die Durchführung der Ressourcen. Jedes Ziel ist mit einem Task verbunden. Ein Dienst wird durch eine Gruppe von Tasks definiert, insofern können diese Ressourcen als die grundlegendste Komponente eines Dienstes betrachtet werden (Ressourcen unterstützen zunächst einmal Tasks).

Die SOTH-Architektur erlaubt es einem Designer, einen Dienst mittels einer Reihe von Tasks zu definieren (Utility Model), anstatt einen Dienst fest mit speziellen Ressourcen zu verbinden. Sämtliche Ressourcen, welche diese Tasks bieten, können von diesem Dienst angefragt werden. Daher können, um einen komplexen Dienst auszuführen, verschiedene von Ressourcen unterstützte Tasks angefordert und organisiert werden. Dieser Prozess wird auf dem vierten Level abgeschlossen und als Dienst-Konfiguration bezeichnet. Der Unterschied zwischen einer komplexen Dienst-Konfiguration und einem Business ist nicht völlig klar. Dies hängt vom Hintergrund und den Vorstellungen des jeweiligen Projektdesigners ab. Der wichtigste Unterschied liegt allerdings in der verzeichneten Position und der Anwendung von Diensten/Business. Ein Dienst ist in der internen Registry publiziert und wird von Designern speziell verwendet, um Business-Module zu bilden.

In dieser Arbeit wird angenommen, dass zwischen Anbietern von Diensten und Anbietern von Tasks keine Übereinkunft über ein Protokoll besteht. Ein Business ist in der externen Registry publiziert. Durch den Designer oder tatsächliche Aufrufe der Nutzer kann es zu einem Teil der Koordination werden oder zu einem unabhängigen Modul in der Business Policy. Unter diesen Umständen kann das Protokoll zwischen Dienst-Anbieter und Business-Anbieter bestimmt werden. Zusammenfassend betrachtet wird zunächst die Methode der Beschreibung der Dienste angeboten; danach kann diese Methode erweitert werden auf andere abstrakte

Level: **Control Model**, **Business Model** und **Collaboration Model**. Der Begriff in der folgenden Abbildung lautet „concept“, und er wird verwendet, um die einzelnen Dinge und Merkmale in der praktischen Umsetzung der Architektur zu identifizieren [120]. Die Verbindungen zwischen den Blöcken zeigen die Beziehungen (relationships) zwischen den verschiedenen concepts an. Die anderen Komponenten in der SOTH-Architektur verwenden bilineare Ausdrücke. Konkret werden folgende Merkmale berücksichtigt:

1. Beschreibung, Semantik, Policy, Protokoll, Registrierungsname und Beziehungen: vorhanden (*has*)
2. Tasks und Beziehungen: ausführen (*perform*)
3. Anbieter und Beziehungen: realisieren (*realizes*)
4. Protokoll und Beziehungen: abstrahieren (*abstracts*)

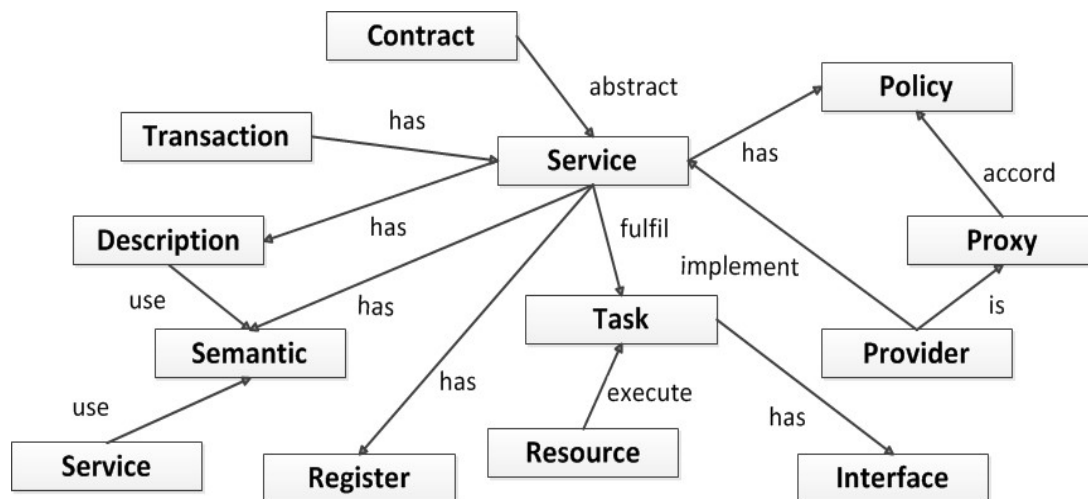


Abbildung 5-6 Beziehung zwischen Konzept und Diensten

Eine Reihe von inhärenten Informationen werden verwendet, um den Nutzer zu identifizieren: *Name*, *Besitzer*, *Version* und *Beziehungsfeld*. Ein Task verfügt ebenfalls über einige inhärente Informationen (*Name*, *Preconditions*, *Konstanten* und *Postconditions*), eine Beziehung zur Ausführung (*execute*) der Ressourcen, eine *has*-Schnittstelle und eine Anbieter-Beziehung. Ein komplexer Dienst kann verschiedene Anbieter besitzen. Der Dienst-Anbieter ist ein Agent, welcher „Message“ für die Kommunikation verwendet. Das Format von Message wird in der Schnittstelle beschrieben.

Die Attribute der Beschreibung der Dienste sind in Tabelle 5-2 zu sehen. Tabelle 5-3 zeigt eine detaillierte Erläuterung. Diese Independent-Gruppen von Informationssprachen werden in drei Typen eingeteilt: *obligatorische Informationen* (*obligatorisch*), *empfohlene Informationen* und *optionale Informationen*. Zuvor wurden die obligatorischen Informationen

bei der Beschreibung von Diensten erörtert. Im Anschluss werden Informationen ausgewählt, welche verdeutlichen, „was dieser Dienst ist“. Zuletzt werden Informationen hinzugefügt, welche dem Nutzer (Implementierer) nützliche Details zu dem Dienst anbieten.

Gleiche Informationen können ebenfalls im Business Layer verwendet werden. Unter diesen Umständen bezieht sich die Information auf die Beschreibung des Business des Dienstes und nicht auf die Tasks.

Tabelle 5-2 Beschreibung der Service-Attribute

Feature/Model	Select
Authentication	Required
: Name	
: Author	
: Version	
: Domain	
Design ideas and Transaction	Optional
Task	Required
: Name	
: Resource	
: Provider	
: Interface	
: Pre-condition	
: Invariant	
: Post-condition	
Registration	Required
Description	Recommended
: Explanation	
: Classification (life cycle)	
: Diagram	
: Contract	
Implement guide and strategies	Recommended
: Priority	
: Capacity	

Tabelle 5-3 Definition der Service-Attribute

Identifier Tag	Data Type	Description
Inherent information		
Authentication		
Name	String	Unique, simple, meaningful, concise, descriptive labels.
Author	String	Person or organization that implements the service

Version	String	Version of the service
Domain	String	Provider range of services
Provider	String	Owner of service, To request the service needs contact the owner
Design ideas and Transaction		
Decision Design	String	In the design stage used design decision
Extrinsic information		
Task		
Name	String	Task Name
Resource	String	The resources needed to execut this task
Provider	String	Noted the position of task provider
Interface	String	Task interface (WSDL)
Precondition	String	Complete description ofPrecondition
Postcondition	String	Complete description ofPostcondition
Invariant	String	Generally does not change and necessary dependencies
Registration		
Registration	String	Lists registry of published Service
Description		
Explanation	String	Natural language description of service targets
Classification (life cycle)	String	Indicates that the information belongs to which cycle
Diagram	String	Noted position of the diagram (UML) of services
Contract	String	Noted the location of contracts that associated with the service
Implement guide and strategies		

6 Konzeption und Implementierung des mHealth Zentral-Server-Systems

In diesem Kapitel wird als Schwerpunkt die Umsetzung des mHealth Zentral-Server-Systems (IPM-ZS) erörtert. Zunächst wird die Anforderung an den Entwurf des Systems analysiert. Nach der im vorherigen Kapitel ausgeführten SOTH-Struktur wird hier ein Healthcare-Service-Bus (HSB) entworfen. Der HSB erlaubt es Entwicklern, schnell und einfach mehrere medizinische Anwendungen als HSB-Service zu verbinden, damit ein gesamter medizinischer Service durch die Kombination verschiedener mHealth Module abgebildet werden kann. Im Teil der Systemarchitektur wird die Funktion jedes Moduls beschrieben. Im Studie Verwaltungsmodul wurde in dieser Arbeit entsprechend den Anforderungen und der Datenstruktur der Untersuchung in der Präventivmedizin ein eigenes CDA-Schema [121] definiert. Basierend auf der strukturierten XML-Beschreibungsdatei kann die Fernuntersuchung im IPM-ZS systematisch verwaltet und erweitert werden, dies erlaubt zugleich auch einen einfachen Datenaustausch mit externen medizinischen Anwendungen. Zuletzt wird ein Anwendungsszenario als Beispiel erörtert, anhand der Beschreibung der SOTH-Struktur werden die zuvor geschilderten Module und Prozeduren mit dem HSB verbunden. Dieses Beispiel zeigt, wie der HSB für Integrierung von Business-Prozessen eingesetzt werden kann.

6.1 Ziele des Systementwurfs

Das mHealth-System verfügt über zahlreiche Monitoring-Geräte, gleichzeitig müssen flexible Monitoring-Geräte hinzugefügt und deinstalliert werden. Das System besitzt eine sehr komplexe Heterogenität [122]. Entsprechend der verschiedenen medizinischen Signale, welche von den Sensoren gesammelt werden, muss das IPM-ZS auch in der Lage sein, neue Module zur Analyse und Auswertung der Information hinzuzufügen. Andererseits muss der Monitoring Server in Echtzeit physiologische Parameter von den verschiedenen Clients empfangen und den Schutz der darin enthaltenen sensiblen medizinischen Daten gewährleisten. Daher werden in diesem Kapitel zunächst die Eigenschaften des mHealth Zentral-Server-Systems (IPM-ZS) von den Entwurfszielen her analysiert.

1. Echtzeit

Die „Echtzeit“ bezeichnet hier das Sammeln, die Übertragung, die Verwaltung und das Feedback in Echtzeit [123]. Die physiologischen Parameter des Patienten können in Echtzeit

gesammelt und in Echtzeit an den Zielpunkt weitergeleitet werden. Im IPM-ZS können die physiologischen Parameter in Echtzeit evaluiert und ein Mess-Ergebnis erzielt werden.

2. Menschlichkeit

Die Menschlichkeit (Humanität) des Systems stellen die beiden Seiten Patient und Arzt dar. Der Patient besitzt eine räumliche Flexibilität. In der Untersuchung verfügt der Patient über einen vergleichsweise großen Aktionsradius [124]. In diesem Bereich kann er sich verhältnismäßig frei bewegen, ohne dass tägliche Aktivitäten beeinflusst würden. In einigen Fällen kann der Patient selbst die am Körper getragenen Überwachungsgeräte anlegen oder entfernen. Der Arzt kann aus der Ferne vom IPM-ZS aus den physiologischen Zustand des Patienten erfassen. Die Zeit für die Diagnose kann flexibel gestaltet werden. Auch die Betreuung des Patienten kann aus der Ferne gesteuert werden.

3. Skalierbarkeit

Das IPM-ZS muss in der Lage sein, entsprechend der von den verschiedenen Überwachungsgeräten gesammelten medizinischen Signale, die Softwaremodule zur Analyse der medizinischen Signale flexibel auszubauen [125]. Durch die lose gekoppelte Datenerfassung mit dem Applikation Layer kann effektiv die Erweiterbarkeit der Systemfunktionen erreicht werden.

4. Unterstützung einer heterogenen Umgebung

Die vom IPM-ZS verwendete Softwarestruktur muss in der Lage sein, hervorragend mit einem komplexen, aus unterschiedlichen Technologien zusammengesetzten System zurecht zu kommen. Es muss dabei möglich sein, andere Systeme einzubinden [126]. Darüber hinaus muss das System, unabhängig von spezifischen Betriebssystemen oder Hardware, nach einer einmaligen Entwicklung den Einsatz in einer beliebigen Plattform ermöglichen.

5. Sicherheit

Die Sicherheit bezieht sich hauptsächlich auf zwei Aspekte, die Verschlüsselung und Anonymisierung der Daten. Dies steht im Bezug zur Gesundheit des Patienten und zur individuellen Privatsphäre [127]. Wenn illegale Informationen im System gespeichert würden, kann die Überwachung und Diagnose des Systems beeinträchtigt werden, was ernsthafte Auswirkungen haben kann. Die Informationen, welche Patient und Arzt erhalten, sind privater Natur. Externes Personal und Personal ohne ausreichende Berechtigung darf nicht auf die

physiologischen Informationen des Patienten zugreifen. Bei der Informationssicherheit müssen drei Punkte erreicht werden:

- 1) Kein Eindringen illegaler Daten in das System;
- 2) Patienteninformationen dürfen nicht für unberechtigte Personen einsehbar sein, Daten dürfen das System nicht verlassen;
- 3) bei der Übermittlung müssen die Daten sicher verschlüsselt werden.

6.2 Healthcare-Service-Bus

Basierend auf der obigen Analyse wird das mHealth-Server-System in dieser Arbeit auf der Basis eines **Healthcare Service Bus** (HSB) umgesetzt.

Enterprise Service Bus (nachfolgend als ESB bezeichnet) [128] ist eine Integrationsmethode von Standards zwischen lose gekoppelten Diensten und Anwendungen. Es wird das „Bus“-Modell verwendet, um die Integration von Anwendungen zu verwalten und zu vereinfachen. Mit offenen Standards für breite Empfangsmöglichkeiten als Grundlage wird der Informationsaustausch zwischen den Anwendungen unterstützt. Auf dem Level von Ereignis und Dienst erfolgen dynamische gegenseitige Interaktionen. ESB ist der Agent, welcher in der SOA-Architektur die intelligente Integration und Verwaltung zwischen den Diensten übernimmt. Es handelt sich um ein abstraktes Konzept. Diese Architektur wird über den Informationsaustausch zwischen verschiedenen Systemen verwirklicht und nicht über Programmierung. Im Bereich der Gesundheitspflege wird ein auf ESB basierendes Business als Healthcare Service Bus (HSB) [129] bezeichnet. Der Realisierungsprozess der mHealth Plattform in dieser Arbeit verwendet die zwei auf ESB basierenden Open Source Werkzeuge Mule 2.0[130] und CXF [131].

Mule ist ein leichtgewichtiges und in höchstem Maße erweiterbares ESB. Es erlaubt Nutzern, einfach und schnell aus verschiedenen Elementen zusammengesetzte Systeme und Dienste zu verbinden und den Datenaustausch bei diesen zu etablieren. Mule ist mit einem unaufhörlich in Betrieb stehenden Omnibus vergleichbar, welcher an jeder Haltestelle (System/Dienst) unterschiedliche Passagiere (Informationen) aufnimmt und abliefern. Mule wird über Java realisiert, allerdings wird die Verbindung größtenteils über ein XML-Konfigurationsdokument erreicht. CXF ist eine neue Java Web Service Engine. Ihr Hauptmerkmal ist die Unterstützung von mit besonders einfachen Methoden veröffentlichten Web Services. Verglichen mit anderen Web-Service Engins ist die Konfiguration von CXF sehr einfach und es ist besonders leicht, sie mit Spring zu integrieren. Diese Methode reicht

nicht nur für die Entwicklung der Funktionen des CXF Moduls aus und vereinfacht die Umwandlung von Java-Anwendungen in Web Services, sondern reduziert auch unmittelbar den Schwierigkeitsgrad der Umsetzung von SOA und bietet eine einfach umsetzbare Methode für die Ausrichtung des Business auf SOA.

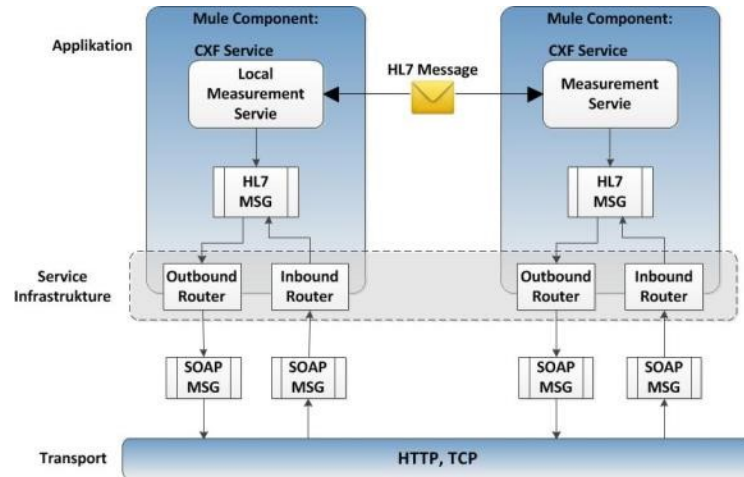


Abbildung 6-1 Mule- und CXF- basierte Healthcare Service Bus (HSB)

Zusammenfassend lässt sich sagen, dass die Umsetzung der Umgebungsstruktur den zwei Teilen in Abbildung 6-1 folgt: das Verpacken des Utility Service als CXF Service Release, und die Funktionalität von Mule wird über das Packen jedes Web Service [132] in Mule Component erreicht. Die von CXF Services erzeugten HL7 Informationen werden über den Inboundrouter und Outboundrouter von HSB Component empfangen und versendet. Die Einführung von WS-Security[133] wird ebenfalls an dieser Stelle implementiert. Danach können die von Mule erzeugten SOAP Informationen über verschiedene Netzwerkprotokolle und andere Mule Components ausgetauscht werden.

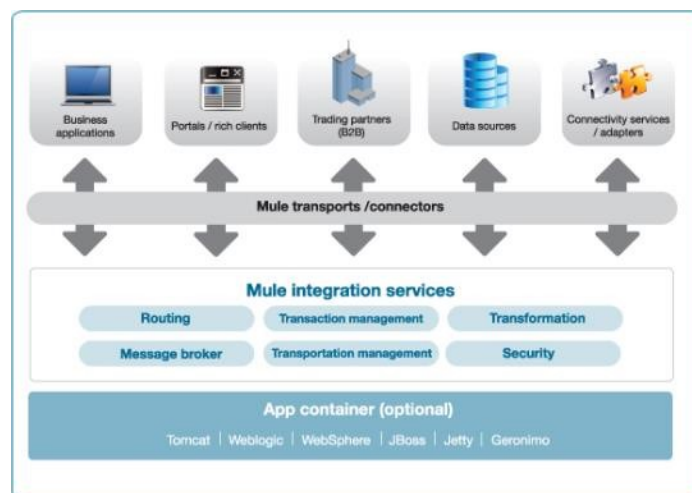


Abbildung 6-2 Gesamtstruktur des HSB

6.2.1 Gesamtstruktur des HSB

Die Gesamtstruktur des HSB ist in der Abbildung 6-2 zu sehen. Über „Transports/Connectors“ wird eine Verbindung mit heterogenen Peripheriesystemen hergestellt und die Kernmodule angeboten, wie *Routing*, *Transaction Management*, *Transformation*, *Message Broker*, *Transportation Management*, *Security*, etc.

Peripheriesysteme senden eine Anfrage über *Transport* des HSB, das HSB führt mittels *Transformer* die Konvertierung des Datenformats durch, danach erfolgt mittels des *Inbound Router* eine Filterung der Informationen (intern über den Konfigurationsfilter realisiert). Der nächste Schritt erfolgt in *Component* des HSB, eine Bearbeitung der Business Logik, für das Ergebnis wird über den *Outbound Router* die Empfangsseite festgelegt. Im Anschluss wird das Datenformat über den *Transformer* konvertiert und über *Transport* an die Empfangsseite weitergeleitet.

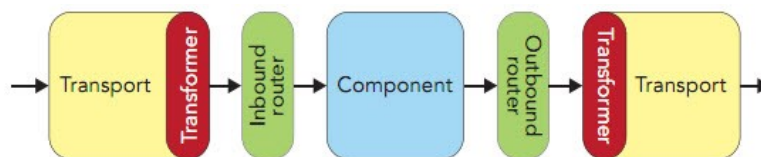


Abbildung 6-3 Ablauf der Message Bearbeitung im HSB

In dieser Abbildung 6-3 wird der klassische Ablauf der Bearbeitung im HSB aufgezeigt, welcher die verschiedenen Kernmodule im HSB beinhaltet. Dabei sind einige Bearbeitungsschritte nicht zwingend notwendig, wie beispielsweise *Inbound Router* und *Transformer*. Nachfolgend können einige andere Bearbeitungsszenarien betrachtet werden. Wenn eine Anwendung oder ein Modul als Service im HSB bearbeitet werden soll, sind folgende Schritte notwendig:

1. Definieren einer Service-Vermittlung
 - a) Definition von Business Logik und Message-Übermittlung
 - b) Definition von Informationsformat und -protokoll des Services
 - c) Bieten des Abrufs beliebiger Positionen des Services
 - d) Angebot der Überbrückung von Protokollen
2. Konvertieren der Daten
 - a) Beim Austausch von Informationen in verschiedenen Formaten zwischen Anwendungen handelt es sich im Fall von medizinischen Anwendungen hauptsächlich um die Konvertierung des HL7-Formats.
 - b) Verschlüsselung, Komprimierung und Kodierung der Information
 - c) Formatierung der Messages aus heterogenem Übertragungsprotokoll

3. Route der Message
 - a) Routing der Messages basierend auf Inhalt und Komplexität der Messages
 - b) Filterung, Zusammensetzung und Sortierung der Messages
4. Erstellen und Hosten von Services
 - a) Publizieren Endpunkte, EJB, Spring Bean und POJO als Services

Im HSB gibt es einige grundlegende Konzepte. Erst wenn diese grundlegenden Konzepte verstanden wurden, kann man die internen Mechanismen des HSB verstehen. Daraus können auch wesentliche Konzepte eines Bus' abgeleitet werden.

6.2.2 Grundlegende Konzepte

1. Model

Das Model drückt die Umgebung beim Hosten verschiedener Services aus.

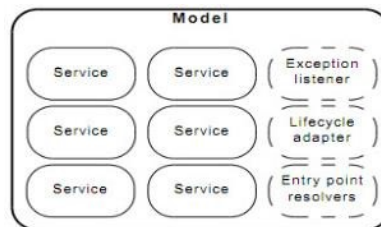


Abbildung 6-4 Model

2. Service

Service wird dazu verwendet, die Grundeinheiten der Serviceanfragen zu bearbeiten, es ruft verschiedene Module auf, um die Serviceanfragen zu bearbeiten.

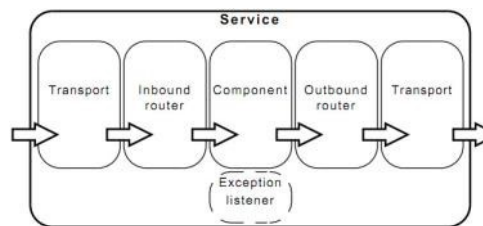


Abbildung 6-5 Service

3. Transport

Transport verarbeitet den Empfang und das Senden von Nachrichten. Der Prozess der Datenkonvertierung wird ebenfalls in Transport über den Abruf des Transformers durchgeführt.

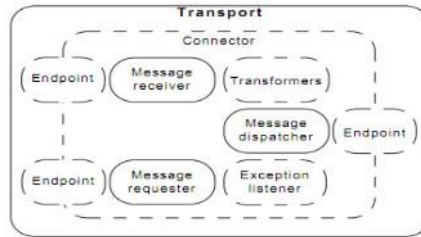


Abbildung 6-6Transport

4. Connector

Der Connector wird für die Kontrolle spezifischer Protokolle verwendet, wie beispielsweise http Connector, JMS Connector, etc.

5. End-Point

End-Point wird dazu gebraucht, um spezifische verwendete Formate einer Art von Protokoll zu beschreiben, wie beispielsweise listening/polling, auslesen/schreiben an eine bestimmte Adresse, etc. Es werden die Kanäle zum Senden und Empfangen von Messages definiert. End-Point kontrolliert, wie Einheiten(Instanzen) auf niedrigem Layer im Connector verwendet werden können. End-Point ist im *Inbound* und *Outbound Router* definiert.

6. Transformer

Transformer wird verwendet, um den Inhalt von Messages umzuwandeln.



Abbildung 6-7Transformer

7. Router

Der Router verwendet den Filter, um die Nachrichten auf der Grundlage ihrer enthaltenen Attribute in Informationen zu unterteilen. Durch die Position in Service wird die Art des Routers (*inbound*, *outbound* und *response*) und seine Rolle (*pass-through*, *aggregator*, etc.) definiert.

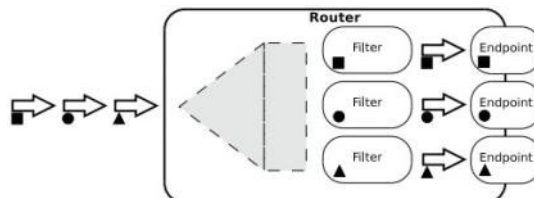


Abbildung 6-8Router

8. Component

Component ist das Kernstück des Service und die Umsetzung der Business Logic des Service.

Component kann eine Java Class (POJO, Spring Bean), ein Web Service oder ein Script, etc. sein. Component definiert den Lebenszyklus desmHealth Service: *initialise*, *start*, *stop*, *dispose*. Allerdings muss eine LifeCycle-Schnittstelle von HSB realisiert werden.

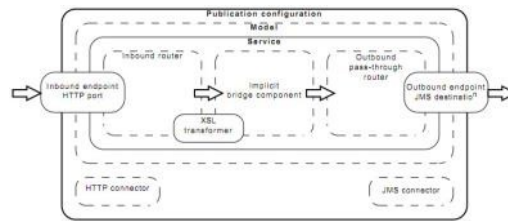


Abbildung 6-9 Component

6.3 Entwurf und Umsetzung des mHealth Server-Moduls

In diesem Abschnitt wird das Kernmodul des IPM-ZS weiter erörtert. Das IPM-ZS bietet Ärzten und Wissenschaftler eine Plattform für die medizinische Fernüberwachung und Datenerfassung. Das in dieser Arbeit entwickelte System wird auch als Forschungsplattform dienen und unterstützt die Durchführung von medizinischen Untersuchungen bei Instituten für präventive Medizin der Universität Rostock. Seine Hauptaufgabe beinhaltet folgende Module: *Kommunikationsserver (KS)*, *Benutzer Verwaltung (BV)*, *Study Management (SM)*, *Datenmanagement (DM)* und *Visualisierungsmodul (VM)*. Wie in Abbildung 6-10 zu sehen wird in jedem Teil ein Adaptionlayer zur Verbindung mit dem HSB eingerichtet. Nach den in der SOTH Struktur definierten Beziehungen zwischen *Service*, *Ressource*, *Message* und *Policies* wird die Interaktion zwischen verschiedenen Komponenten durchgeführt. Bei allen externen Schnittstellen muss eine Konvertierung des HL7-Formats durchgeführt werden. Nachfolgend werden die Hauptfunktionen und Implementierung der verschiedenen Module vorgestellt:

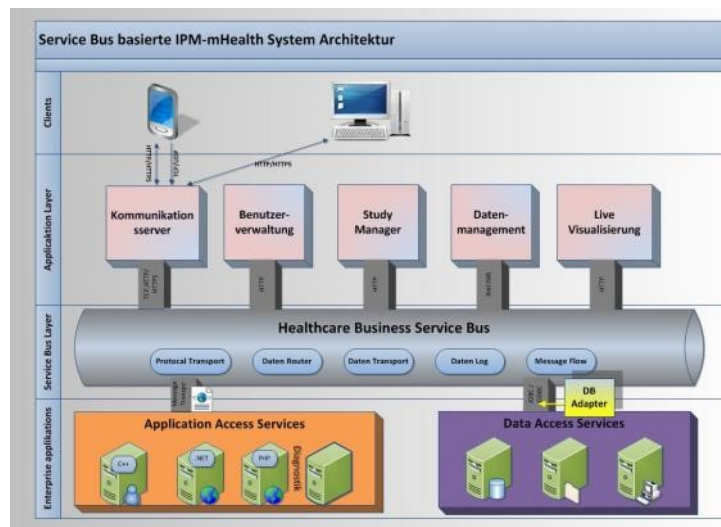


Abbildung 6-10 Kernkomponenten vom IPM-ZS

1. Kommunikationsserver (KS): Sorgt hauptsächlich für die bidirektionale Kommunikation, Synchronisierung und Echtzeit-Datenübermittlung zwischen dem IPM-MC und IPM-ZS.
2. Benutzerverwaltung (BV): Beinhaltet die Funktionen Benutzeranmeldung, Benutzerregistrierung und Änderung oder Löschung von Benutzern.
3. Study Management (SM): Beinhaltet Verwaltung, Erstellung und Anordnung von Untersuchungen
4. Datenmanagement (DM): Beinhaltet Analyse, Speicherung historischer Daten und Übermittlung der Diagnoseergebnisse.
5. Visualisierungsmanagement (VM): Hauptsächlich verantwortlich für die Visualisierung der vom IPM-MC empfangenen Vital Daten.

6.3.1 Kommunikationsserver

1. Modulfunktion

Der KS ist das Kernmodul des IPM Server. Es ist verantwortlich für den Empfang und das Beantworten der Messages des Clients. Nachdem das IPM-MC eine Verbindung mit dem IPM-ZS aufgebaut hat, werden zwei Proxy-Prozesse gestartet. Einer bearbeitet die Anfragen und der andere bearbeitet die Antworten. Der *Anfrage*-Proxy dient der Überwachung der vom Client empfangenen Daten. Er führt für alle empfangenen Daten eine Verifizierung und Evaluation der Parameter durch. Der *Antwort*-Proxy kümmert sich um die Antworten und Steuerungsaktionen des IPM-ZS an den Client. Die KS erhält einige Antwort-Task-Queues aufrecht, welche der Interaktion von Antwort-Proxy und Anfrage-Proxy dienen. Entsprechend der Priorität wird die Task-Queue in verschiedene Listen eingetragen. Wenn die KS bei der Evaluierung der Parameter abnormale physiologische Parameter feststellt, dann wird ein Error-Ergebnis später als Antwort übermittelt. Die Bearbeitung der physiologischen Parameter durch die Kommunikationsserver ist in Abbildung 6-11 dargestellt.

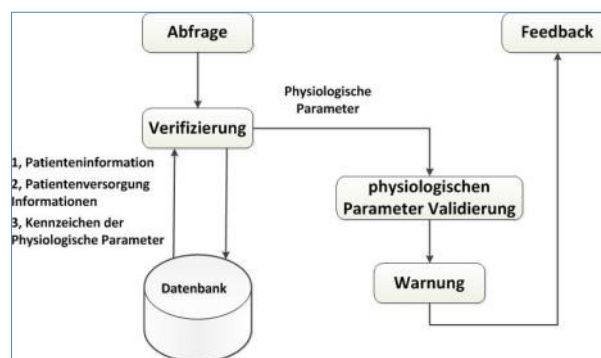


Abbildung 6-11 Bearbeitung der physiologischen Parameter durch die Kommunikationsserver

2. Implementierung

1) Anfrageproxy

Im vorausgegangenen Kapitel 4.5 wurde bereits das gegenseitige Datenformat von IPM-MC und IPM-ZS vorgestellt. Wenn derKS Parameter vom IPM-MC empfängt, so wird eine Analyse anhand des erhaltenen Message-Formats durchgeführt, damit das System dies einfacher bearbeiten kann. Es wird eine entsprechende *Message-Klasse* generiert und danach entsprechend des Typs der physiologischen Parameter eine Vorüberprüfung durchgeführt. Diese Vorüberprüfung dient der Bestimmung der Validität der Daten. Wenn die Vorüberprüfung ergibt, dass die Message nicht valide ist, wird der derzeitige Bearbeitungsprozess abgebrochen und mit der Bearbeitung der nächsten Daten fortgefahren. Falls die Vorüberprüfung erfolgreich ist, werden die Parameter in der Datenbank abgespeichert. Außerdem werden diese Daten auf Ungewöhnliches überprüft. Es wird überprüft, ob die physiologischen Parameter der Norm entsprechen. Wenn sie abnormal ausfallen, wird das Warnungsmodul zur weiteren Bearbeitung aufgerufen. Das Ergebnis der Bearbeitung des Warnungsmoduls wird in Form von Befehlen an den Client zurückgesendet.

2) Validierung

Unter den Bedingungen des Telemonitorings ist die Bearbeitung der empfangenen Daten von besonderer Wichtigkeit. Wenn fehlerhafte Daten mit abweichender Semantik oder Datenmüll in das System gelangen, so kann dieses zu nutzlosen oder fehlerhaften Bedienungen führen [134]. In den Arbeitsschritten des Systems muss sichergestellt werden, dass die Daten im Datenbanksystem korrekt, gültig und kompatibel sind, also dass das Datenbanksystem intakt ist. Wenn die Integrität des Datenbanksystems beschädigt wird, so kann dies Auswirkungen auf die Korrektheit des Systems haben und in manchen Fällen sogar zum Zusammenbruch des Systems führen. Der Schutz der Integrität des Datenbanksystems wird gewöhnlich mit einer Integritätsregel bewerkstelligt [135]. Die Fähigkeit der Integritätsregel zur Fehlererkennung in der Datenbank hat allerdings auch Grenzen und kann nicht vollständig ausschließen, dass fehlerhafte Daten in das Datenbanksystem gelangen.

Für eine Message vom IPM-MC, falls die Quell-ID im Datenbanksystem nicht existiert, dann wird die ID nicht für das Monitoring verwendet. Daraus folgt, dass die Message fehlerhaft und damit ungültig ist.

Wenn Daten fehlerhaft vom IPM-MC zum IPM-ZS übertragen werden, kann dies dazu führen, dass das IPM-ZS ungültige Messages erhält, weswegen eine Vorüberprüfung der Messages notwendig ist, um die Integrität des Datenbanksystems sicherzustellen. Die Vorüberprüfung

wird hauptsächlich zur Bestimmung der Gültigkeit der physiologischen Parameterumgesetzt. Die Überprüfung der Gültigkeit beinhaltet die Überprüfung des Typs der physiologischen Parameter sowie die Überprüfung der Gültigkeit der Quell-ID der physiologischen Parameter und ob die vorliegende ID derzeit im Monitoring eingesetzt wird. Im System ist eine Funktion zur rudimentären Überprüfung der Gültigkeit definiert, der Code dazu ist im Wesentlichen folgender:

```
public bool check_availability(int type, string DeviceID)
{.....
if (result){sqlComm.CommandText = "select from dictionary where type=1 and value=" + type + ";";
thisreader = sqlComm.ExecuteReader();
if (!thisreader.HasRows())
{result = false;}}
.....
if (result){sqlComm.CommandText = "select from Monite where
DeviceID=" + DeviceID + ";";
thisreader = sqlComm.ExecuteReader();
if (!thisreader.HasRows()){result = false;}}
thisreader.Close();
}return result;}
```

3) Konvertierung des Datenformats

Der IPM-MC ist die Middleware [136] zwischen Body Area Network und Telemonitoring-Zentrum. Er beinhaltet zwei Aspekte von Kommunikation. (1)Die Kommunikation von Client und Body Area Network. (2)Die Kommunikation von Client und IPM Server. Auf dem Applikation-Layer gibt es einige Unterschiede zwischen den Protokollen der beiden entsprechenden Schnittstellen. Um diese Protokolle zu konvertieren, werden Datenstrukturen definiert, welche die Konvertierung der Protokolle unterstützen. Im IPM-ZS werden drei Arten von Messages definiert: *Dateninformationen*, welche vom Client eingehen, *Befehlsinformationen* zur Steuerung des Body Area Network und *Warninformationen*, welche an das IPM-MC gesendet werden. Um die Analyse dieser Informationen zu unterstützen, werden drei Datenstrukturen im System definiert: *DataMessage*, *CommandMessage* und *AlarmMessage*. *DataMessage* beschreibt das Format der vom IPM-ZS empfangenen Daten-Messages. Bei der Analyse der vom IPM-ZS empfangenen Frames wird anhand von Struct *DataMessage* verfahren. Die konkrete Definition lautet wie folgt:

```
{MessageType Type;
int NodeID;
string password;
```

```

byte ValueType;
float Value;
byte Interval;
DateTime Datetime;}

```

ZentralCommandMessage(ZCM) wird verwendet, um das Format der vom IPM-ZS an das IPM-MC gesendeten Befehle zu definieren. Wenn ein Befehl die Frequenz des Samplings festlegt, beschreibt die NodeID dann die Sensor ID, Value beschreibt die neue Sampling-Frequenz. Wenn der Befehl den Neuaufbau des Netzwerks meint, so sind NodeID und Value bedeutungslos und die beiden Felder können vernachlässigt werden. Die konkrete Definition lautet wie folgt:

```

struct CommandMessage
{MessageType Type;
byte CommandNum; int NodeID;
byte value;
DateTime Datetime;}

```

AlarmMessage sind Warninformationen, welche vom IPM-ZS an das IPM-MC gesendet werden. Da die Länge der Warninformationen nicht festgelegt werden kann, müssen diese in Message length verwendet werden, um die Länge anzuzeigen. AlarmMessage beschreibt die Hinweisinformation, welche an den Client gesendet wird. Die konkrete Definition lautet wie folgt:

```

struct AlarmMessage
{
MessageType Type;
byte Level;
long length;
Array AlarmMess;
DateTime Datetime;
}

```

In den drei Datenstrukturen ist das Feld *MessageType* enthalten. *MessageType* wird wie folgt definiert:

```

enum MessageType: byte
{
DATA,
COMMAND,
ALARM
}

```

Dies sind die Faktoren, welche das Monitoring-Zentrum bei der Konvertierung zunächst bestimmen muss. Verschiedene MessageTypes haben unterschiedliche Bearbeitungsabläufe. Wenn der IPM-MC Befehlsinformationen des IPM-ZS empfängt, müssen die Befehle weitergeleitet werden. Wenn Warninformationen erhalten werden, muss im nächsten Schritt der Level der Warnung bestimmt werden und dann eine entsprechende medizinische Behandlung aufgerufen werden.

3. Service-Beschreibung für HSB Module

Service: CommunicationService

Resource: Inbound-Endpoint: Tcp-Socket, Outbound-Endpoint: SOAP,

Message: DataMessage, CommandMessage, AlarmMessage, String to Java Object

Interface:

```
public interface CommunicationService {
    public void sessionOpened(ioSession session);
    public ServerSessionHandler();
    public void messageReceived(ioSession session, Object message);
    public void messageSent(ioSession session, Object message) throws
Exception;
    public void sessionClosed(ioSession session) throws Exception;
    public void sessionIdle(ioSession session, IdleStatus status);
    public void exceptionCaught(ioSession session, Throwable cause);
}
```

6.3.2 Benutzerverwaltung

1. Modulfunktion

Dieses Modul bietet zwei grundlegende Funktionen: Es gestattet anderen Diensten die Überprüfung, ob der Nutzer im System eingeloggt ist (Überprüfungsfunktion) und es bietet die Verwaltungsfunktion des Hinzufügens, Löschens, Editierens und Wiederherstellens von Nutzern. Jeder Nutzer erzeugt einen kryptographischen Schlüssel, welcher als allgemeiner Schlüssel auf das IPM-ZS hochgeladen wird. Andere autorisierte Dienste können den allgemeinen Schlüssel eines Nutzers erhalten und einen sicheren Kommunikationskanal aufbauen. Das Klassendiagramm beinhaltet zwei Klassen: die Klasse *Identifikations-Service* und die Klasse *Nutzer*. Dies ist in der Abbildung 6-12 zu sehen.

1. Ablauf des Prozesses

Medizinische Daten stellen besondere Anforderungen an die Sicherheit. An dieser Stelle wird SOAP für die sichere Übermittlung der Informationen verwendet. Jeder Dienst und jeder Nutzer besitzt zwei Keys (private Key und public Key) [137]. Jeder Nutzer benötigt den öffentlichen Schlüssel des Servers, wenn er mit diesem kommunizieren will. Nutzer übermitteln über die Anfrage den geschützten private Key und den eigenen public Key.

Nachdem die Informationen serverendig empfangen wurden, wird der eigene public Key für die Freigabe verwendet und der in der Anfrage enthaltene Inhalt wird gelesen. Wenn eine Antwort an den Client rausgeht, wird der public Key des Clients zur Verschlüsselung verwendet. Der Client verwendet den eigenen private Key, um die Antwort zu entschlüsseln. Der konkrete Arbeitsprozess der Sicherheitsüberprüfung lautet wie folgt:

Voraussetzungen: Der Nutzernamen und das Passwort des Nutzers müssen notwendigerweise im IPM-ZS registriert und die Gesundheitsdaten des Nutzers müssen abgespeichert sein.

- 1) Der Nutzer verwendet Keytool, um den eigenen public und private Key zu erzeugen.
- 2) Der Nutzer übermittelt den public Key an den Identification Service (stellt Informationen zur Überprüfung zu Verfügung: Nutzernamen und Passwort).
- 3) Der Nutzer meldet sich beim Identification Service an und übermittelt den eigenen Nutzernamen und sein Passwort. Das korrekte Passwort ist über HASH Algorithmus [138] im Identification Service gespeichert. Entsprechen sich beide, so wird eine Session-Nummer für die Kommunikation zurück übermittelt.
- 4) Der Nutzer fragt beim Messdienst die Daten des Patienten an. Der Dienst überprüft die gebotene Kommunikation, ruft den Speicher Module auf und übermittelt die angefragten Daten.

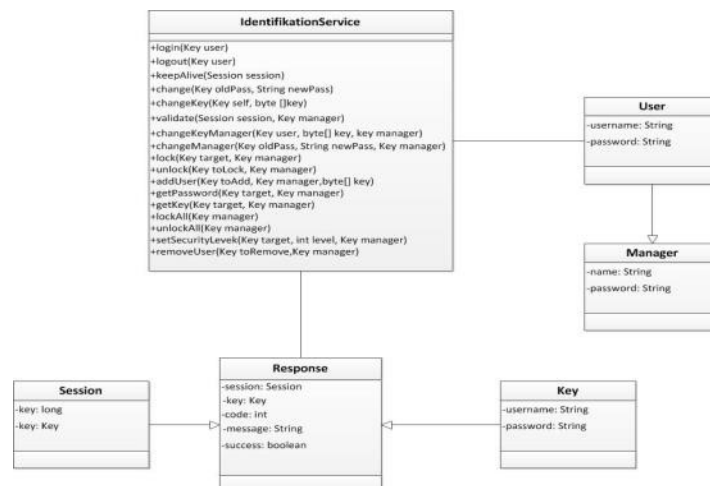


Abbildung 6-12Klassendiagrammder Identifikations-Service

2. Implementierung

Die grundlegende Funktion des Identification Service beinhaltet, dass es anderen Diensten ermöglicht wird festzustellen, ob es sich um einen registrierten Nutzer und eine reguläre Anmeldung handelt, welche nicht gesperrt (locked) ist. Das bedeutet, ob eine Berechtigung für die Benutzung des Accounts besteht. Beispielsweise könnte ein Nutzer einen Dienst einige Zeit nicht mehr benötigen oder es könnten administrative Probleme auftreten (z.B.

Abrechnungsprobleme). Das BV ist zuständig für das Abspeichern des Nutzernamens und Passworts des Nutzers und bietet eine sichere Methode, wie berechtigte Dienste die entsprechenden Informationen abrufen können. Der Dienst ist auch in der Lage, Nutzernamen und Passwörter zu verwenden, um die Anfrage des Nutzers zu verschlüsseln. Die Autorisierung umfasst folgende Schritte:

Der Client loggt sich ein und übermittelt seinen Key (Nutzername und Passwort). Sobald Nutzernamen und Passwörter verifiziert wurden und der Nutzer nicht gesperrt wurde, übermittelt das IPM-ZS eine Session, welche einen Key und eine in ein Responseobject verpackte Zufallszahl beinhaltet. Jedes Mal, wenn der Nutzer sich einloggt, wird diese Nummer erneuert. Der Nutzer verwendet diese Session, um andere Dienste von IPM-ZS aufzurufen. Diese Dienste rufen den Identifikationsservice auf, um festzustellen, ob dieser Nutzer tatsächlich registriert ist. Falls sich diese Session im idle-Zustand befindet, wird der Nutzer automatisch abgemeldet. Wenn innerhalb von 15 Minuten keine anderen Dienste diese Session überprüfen, wird diese Session in den idle-Zustand versetzt, um den Fall, dass der Nutzer vergessen hat, sich ordnungsgemäß abzumelden, abzusichern. Wenn darüber hinaus technische Probleme beim Client auftreten, wird der Zustand des Nutzers in den Zustand vor dem Fehler zurückversetzt. Nachdem der Nutzer mit der Verwendung des Systems fertig ist, sollte er sich ausloggen. Zu diesem Zeitpunkt wird der Zustand der Autorisierung der Session beendet. Die Zustände des Systems sind in den Abbildungen 6-13 zu sehen.



Abbildung 6-13 Zustandsdiagramm der Benutzeranmeldung

Entsprechend der praktischen Nutzung des Systems wurden im IPM-ZS 3 Rollen definiert, diese sind: *Administrator*, *Ärzte/Researcher* und *Patient*. Jeder Rolle wurden entsprechende Rechte zugeordnet. Details werden in der folgenden Tabelle beschrieben:

Tabelle 6-1 Rollen im IPM-ZS

User	Role	Permissions code	Permissions
Administrator	System Admin	2	Manage all users
Ärzte/Researcher	Responsible Person	1	Read all patient information and Monitoring Status
Patient	Normal user	0	Read record and Diagnostic results

2. Service-Beschreibung für HSB Module

Service: IdentificationService

Resource: Inbound-Endpoint: SOAP, Outbound-Endpoint: String

Message: SOAP to String

Interface:

```
publicinterface IdentificationService {
    publicSession login(String user, String password);
    publicSession logout(String user, String password);
    publicvoid keepAlive(ToSession session);
    publicvoid changePassword(String oldPass, String newPass);
    publicvoid exceptionCaught(ToSession session, Throwable cause);
}
```

6.3.3 Live Visualisierungsmodul

1. Modulfunktion

Das Visualisierungsmodul ist dafür verantwortlich, die vom IPM-MC erhaltenen Daten dynamisch darzustellen. Um im Browser die Funktion dieser live Vital-Daten (inkl. EKG) darzustellen, wird in dieser Arbeit die Canvas Component von HTML5 verwendet. Der konkrete Zeichenprozess wird in drawCanvas() implementiert, die Implementierungsmethode entspricht der Literatur [139], [140].

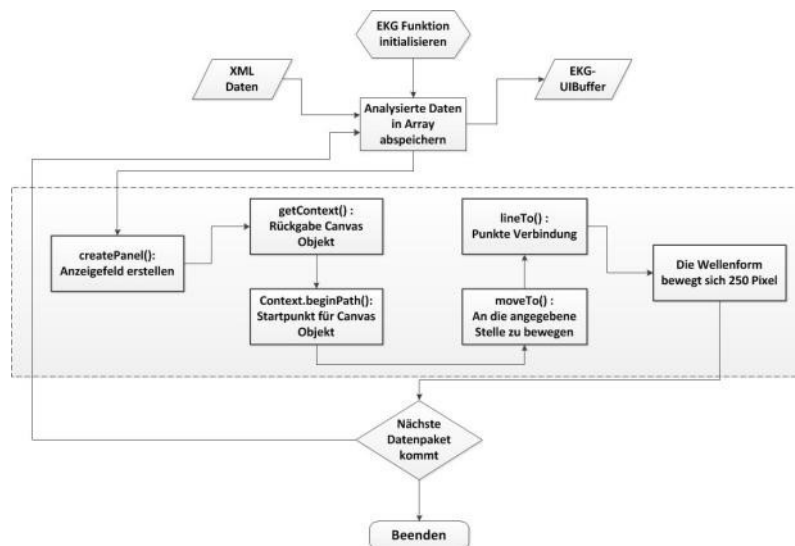


Abbildung 6-14 Verlauf der verschiedenen Funktionen im EKG-Zeichenprozess

2. Implementierung

Nachdem der Browser die vom Datenmanagementmodul gesendeten Daten empfangen hat, wird zunächst eine Analyse des XML-Formats durchgeführt und im Vitaldaten Buffer abgespeichert, welcher als Anzeigepuffer fungiert. Danach wird mit JavaScript das Canvas-

Objekt betätigt und ein Bild gezeichnet. Konkret ist `drawCanvas()` für das dynamische Zeichnen verantwortlich, es bedient die Canvas Tag von HTML5 [141] und verwendet die im Anzeigepuffer vorhandenen Daten und zeichnet auf dem von `CreatePanel()` eingerichteten Feld das Vitaldaten-Diagramm. Der konkrete Verlauf der gegenseitigen Koordination zwischen den verschiedenen Funktionen im EKG-Zeichenprozess ist wie in Abbildung 6-14. Zunächst wird über die Methode `getContext()` des *Canvas-Objektes* der Kontext der Canvas-Operation erlangt. Danach wird über die Methode `BeginPath()` dem *Context-Objekt* mitgeteilt, damit zu beginnen, einen neuen Pfad zu zeichnen. Ansonsten überlagert sich der als nächstes gezeichnete Pfad mit dem zuvor gezeichneten Pfad, womit beim Auffüllen oder Begrenzen Probleme auftreten können. Mittels `strokeStyle = 'green'` wird die Farbe des Pinsels eingestellt. Als dritter Schritt wird mittels des Loops der Daten im Vital Buffer der Pfad der Punkte entsprechend der Reihenfolge nacheinander verbunden. In Canvas wird der Pfad gezeichnet, wobei normalerweise kein Startpunkt festgelegt werden muss, da der Default-Ausgangspunkt eben der Endpunkt der letzten Zeichnung ist. Wenn daher ein Ausgangspunkt festgelegt werden muss, so muss die Funktion `moveTo()` verwendet werden, um die Position zu bestimmen. Danach wird die Methode `lineTo()` verwendet, um eine direkte Linie zur bestimmten Position zu zeichnen. `lineTo` ist der Pfad. Nachdem die Methode `lineTo` abgeschlossen wurde, bewegt sich der Startpunkt der Zeichnung in Context zum Endpunkt der gezeichneten Geraden. Darauf wird gleichzeitig mittels `ctx.fillStyle = 'gray'; ctx.fillRect(x, 0, width, height)` eine graue Vertikale zusammen mit der Verbindung der Punkte gezeichnet, um die Ansicht eines medizinischen Live-Diagramms zu schaffen. Der Pseudocode der Zeichenfunktion ist wie folgt:

```
for (i = 0; i < this.vitalUIBuffer.length; i++) {
  y = 150 - this.vitalUIBuffer[i];
  ctx.strokeStyle = 'white';
  if (i > 0) {
    ctx.moveTo(i - 1, preY);
    ctx.lineTo(i, y);
    preY = y;
  }
}
```

Nachdem das erste Event-Datenpaket analysiert und visualisiert wurde, folgt das nächste Datenpaket, woraufhin der oben genannte Zeichenprozess wiederholt wird. Die Funktion wird nach vorne erweitert. Sobald das Ende des Live-Panel erreicht ist, wird wieder von ganz links begonnen.

Was das Sensorsystem angeht, so muss gewährleistet sein, dass pro Sekunde mindestens 200 Datenwerte gesammelt werden. Entsprechend der eigenen Frequenz des Geräts mit 500 Sample/s wird in dieser Arbeit ein Datensammelintervall von 0,5 s verwendet. Dies entspricht einer Datenmenge von etwa 250 Werten pro Intervall. Daher bewegt sich die Funktion der Datenzeichnung alle 0,5 s um einen Abschnitt und die den 0,5 s entsprechenden 250 Werte werden in diesen Abschnitt eingetragen.

Entsprechend des Umfangs der vom Server jedes Mal gesammelten Datenpakete werden die Geräte des VitalBuffers auf 500 festgelegt. Auf diese Weise stellt der Browser jedes mal etwa 250 Daten dar, das Zeitintervall beträgt 0,5 s. Dies soll den Zusammenhang der dynamischen Darstellung garantieren. Zuletzt erfolgt ein Ergebnis der Darstellung verschiedener Vital Daten wie in Abbildung 6-15 zu sehen.

3. Service-Beschreibung für HBS Module

Service: LiveViewService

Resource: Inbound-Endpoint: JDBC, Outbound-Endpoint: HTTP

Message: String to XML

Interface:

```
public interface LiveViewService {
    public void getData(IoSession session);
    public void drawData(IoSession session, Object message);
    public void updateData(IoSession session, Object message);
    public void updateView(IoSession session) throws Exception;
    public void exceptionCaught(IoSession session, Throwable cause);
}
```

6.3.4 Studien-Management

1. Modulfunktion

Das Studien-Management setzt hauptsächlich die inhaltliche Verwaltung der Untersuchungsaufgaben um. Um sich den Anforderungen verschiedener Untersuchungen anzupassen, wird in dieser Arbeit eine wiederverwendbare Study-Struktur entwickelt, die auf CDA XML Schema basiert (siehe Anhang 1).

Eine Studie beinhaltet die folgenden drei Teile: *Patient*, *medizinische Einrichtungen* und *Untersuchungsevent*. Ein *Untersuchungsevent* setzt sich aus verschiedenen Datenerfassungen wie Herzfrequenzerfassung, Temperaturerfassung, EKG und Belschleunigungserfassung zusammen. Das IPM-ZS kann durch das Hinzufügen, Ändern und Löschen verschiedener Erfassungsmodule und CDA-Studien-Beschreibungen die Anforderungen verschiedener Untersuchung anzupassen.

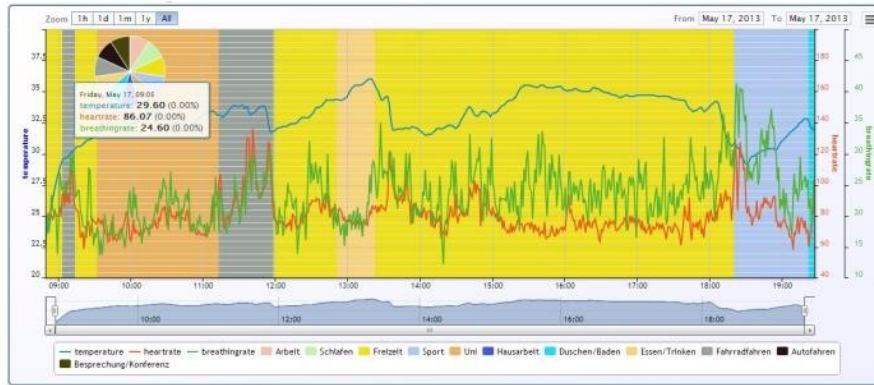


Abbildung 6-15 Diagramm der erfassten physiologischen Parameter

2. Implementierung

Abbildung 6-16 zeigt die Hauptbedienungsfläche des Study-Manager Moduls. Über den Button „Subject Matrix“ gelangt man in die Untersuchungsdetails. Auf der linken Seite der Werkzeugleiste wird die gegenwärtig ausgewählte Studie: „Test“ dargestellt. Die Hauptbedienungs-Oberfläche verwendet eine Matrix für die Darstellung aller laufenden Untersuchungen in „Test“ Study. Zur Auswahl einer konkreten Untersuchung wird auf „Ausführlichen Inhalt betrachten“ geklickt. Daraufhin können die erfassten physiologischen Daten in die spezifische Untersuchung eingesehen und exportiert werden.

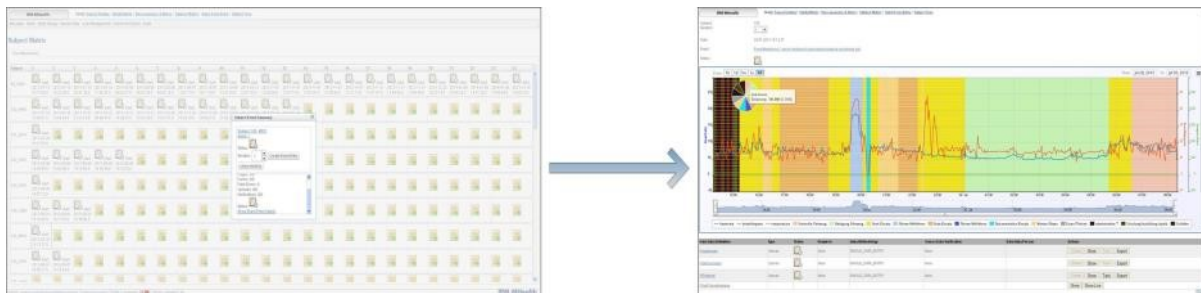


Abbildung 6-16 Hauptbedienungsfläche des Studienmanagers

3. Service-Beschreibung für HSB Module

Service: StudyManagerService

Resource: Inbound-Endpoint: Study Object, Outbound-Endpoint: SOAP

Message: XML to Study Object

Interface:

```
public interface StudyManagerService {
    public void createStudy( IoSession session );
    public void addEvent( IoSession session );
    public void addSubject( IoSession session );
    public void editStudy( IoSession session );
    public void deleteStudy( IoSession session );
    .....
    public void exceptionCaught( IoSession session, Throwable cause );
}
```

6.3.5 Datenmanagement

Im System des IPM-ZS führen häufig mehrere Benutzer Operationen im Datenbanksystem durch. Um die Leistung zu verbessern, wird „*StoredProcedure*“ verwendet [142]. Die *StoredProcedure* muss nur beim Aufbau kompiliert werden, wodurch die Effizienz des Systems gesteigert wird. Der Zugriff auf das Datenbanksystem über *StoredProcedure* verläuft wie folgt:

- 1) Etablieren eines Verbindungs-String, Instanziierung des SqlConnection-Objects:
- 2) Öffnen der Verbindung: myConnection.Open().
- 3) Verwenden ADO-Objekte [143] als Instanz für Operationen des Datenbanksystems, wie beispielsweise:

```
SqlCommand cmd = CreateProcCommand(procName, null);
cmd.ExecuteNonQuery();
```

- 4) Schließen der Datenbankverbindung nachder Operation: myConnection.Close().

Folgender Abbildung 6-17 beschreibt das Speichervorgang für die Funktion: Add_New_Patient:

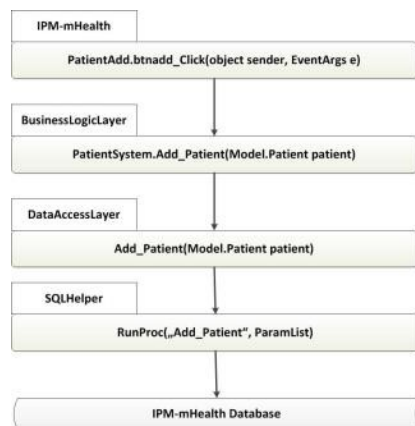


Abbildung 6-17Speichervorgang für die Funktion: Add_New_Patient

6.3.5.1 Struktur der Datenbank

Entsprechend der Analyse der Anforderungen einer Systemdatenbank und den zuvor geplanten Modulen, wird der Entwurf einer Datenbank im Hinblick auf die Umsetzung des Business-Szenarios vorgestellt. Wie die nachfolgende Abbildung 6-18 zeigt.

In der Tabelle „*Study*“ werden die Merkmalbeschreibungen der verschiedenen Untersuchung(Event) abgespeichert, wie Zeitpunkt der Untersuchung, Verantwortlicher der Untersuchung und Untersuchungsmethode, welche die Untersuchung beschreiben.

Tabelle „*Responsible*“ beschreibt die Informationen des Verantwortlichen der Untersuchungen.

Tabelle „*Patient*“ stellt die Merkmale des entsprechenden Patienten dar, darunter Name, Geschlecht, Benutzername im System, etc.

In Tabelle „*Event*“ sind einzelne Aufgaben bei einer Untersuchung aufgeführt (wie die Messung bestimmter Daten). Konkrete Informationen umfassen die jeweiligen Geräte, den jeweiligen Ablauf, etc.

Tabelle „*Schedule*“ beschreibt, wie die einzelnen Events konkret angeordnet werden.

In der Tabelle „*Device*“ sind die konkreten Merkmalsinformationen der Geräte enthalten, wie beispielsweise Hauptelemente der Bedienungsanleitung, Beschreibung der Gerätetreiber, etc. Konkrete Feldinformationen.

Tabelle „*MeasurementType*“ beschreibt den konkreten Typ der Messungen.

Die Informationen in der „*Measurement*“ Tabelle beinhalten den konkreten Inhalt der übermittelten Informationen, wie z.B. den Zeitpunkt der Messung, den Patienten, konkrete Messwerte, etc.

Measurement

Measurement ist die umfangreichste Tabelle in diesem System. Die physiologischen Informationen von einem Patienten können von einigen hundert bis zu Millionen Zeilen umfassen [144]. Ist eine Tabelle verhältnismäßig umfangreich, so beeinflusst dies häufig die Geschwindigkeit der Datenabfrage, weswegen einige Verbesserungen an den Datenblättern vorgenommen werden müssen. Dies geschieht durch die Einrichtung eines Index. Die Index-Technologie ist in der Lage, dieses Problem einfach zu lösen. Den Tabellen der physiologischen Daten wird ein zweistufiger Index hinzugefügt. Die erste Stufe ist die ID, die zweite Stufe ist der Type. Wenn nun die physiologischen Informationen eines bestimmten Patienten gesucht werden, können die entsprechenden Aufzeichnungen schneller lokalisiert werden. Auf diese Weise wird die Zugriffsgeschwindigkeit der Datenbank erhöht.

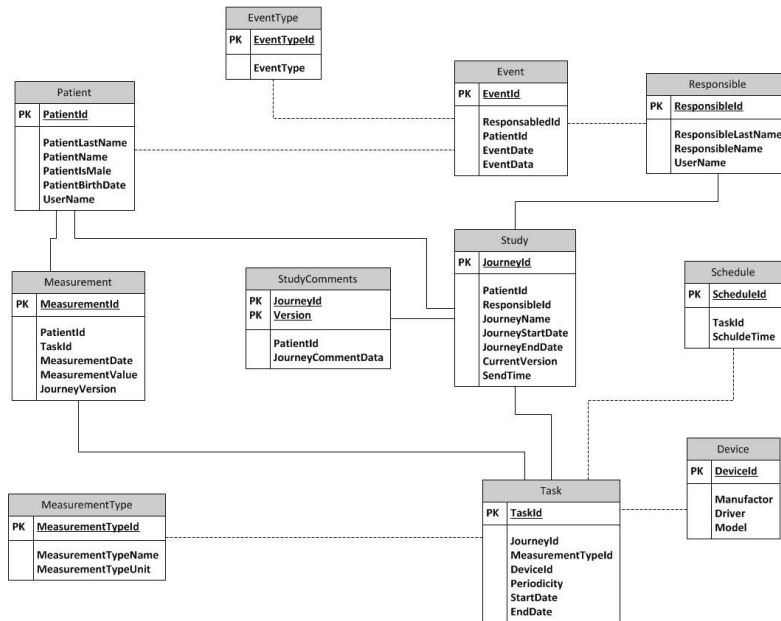


Abbildung 6-18 Tabellenstruktur in der IPM-ZS Datenbank

Tabelle 6-2 Tabelle Measurement

Field	Type	Description
ID	Index	Measurement ID
Type	Index	Measurement Type ID
Patient ID	Char[10]	Patient ID
Value	Char[20]	Measurement Vaule
Time	Datetime	Read Time

3. Service-Beschreibung für HSB Module

Service: DataManagerService

Resource: Inbound-Endpoint: JDBC, Outbound-Endpoint: SOAP

Message: String to Java Object

Interface:

```
public interface DataManagerService {
    public void getConnection (IoSession session);
    public void getDBHelper (IoSession session, Object message);
    .....
    public void exceptionCaught (IoSession session, Throwable cause);
}
```

6.4 Implementierung der Services

6.4.1 HSB Service-Integration

Im obigen Teil finden sich die Einzelheiten der Interaktion zwischen den verschiedenen Services im Mess-Szenario. Diese bieten jeweils verschiedene Schnittstellen, wie JMS, SOAP, Webservice, HL7, etc. In diesem Abschnitt wird auf der Grundlage des Beispiels dieses

Szenarios erläutert, wie im HSB vorhandene Module als Monitoring-Zentrum-Services gehostet, verbunden und das Message-Format umgewandelt werden kann. Die Verknüpfung der Service-Module zu einem einheitlichen telemedizinischen Dienst wird über die Konfigurationsdatei erreicht. Die Konfigurationseinstellungen des HSB sind Aufgabe des Entwicklers. Sie beinhalten:

- 1) Definition des Service-Moduls;
- 2) Welcher Endpunkt eine Service Message empfängt;
- 3) Welcher Transformer für die Messages vor dem Übergang in das Service-Modul verwendet wird;
- 4) Das Ziel der vom Endpunkt ausgehenden Messages;
- 5) Die Routing-Informationen der Message, mit welcher sie an das nächste Service-Modul gesandt wird.

6.4.2 Anwendungsszenarien

Durch den Entwurf der Systemarchitektur und die Erörterung der Systemanforderungen wird das Funktionsmodell des Systems deutlich, es ist notwendig, die Dienste und die Interaktion zwischen den Modulen wie in Abbildung 6-19 zu realisieren.

1. Der Nutzer will sich im System anmelden, es müssen daher zuerst der Nutzernamen und das Passwort vom Identification Service verarbeitet und die Parameter an IPM-ZS übermittelt werden.
2. Nachdem die Überprüfung abgeschlossen ist, wird kontrolliert, ob für diese ID neue Konfigurationsinformationen vorliegen. Wenn die eingegebenen Informationen nicht ordnungsgemäß sind, wird der Alarm Service gestartet.
3. Falls dabei festgestellt wird, dass für den entsprechenden Patienten neue Konfigurationsinformationen vorliegen, werden die entsprechenden Messkonfigurationsdaten geändert und die Informationen in der Datenbank abgespeichert. Falls das Speichern fehlschlägt, wird der Alarm-Service gestartet.
4. Die Messung beginnt, wenn die vorgeschriebene Messzeit überschritten.
5. Nach Beendigung der Messung werden die Rohdaten in der Datenbank abgespeichert.
6. Visualisierungsdienste lesen die Daten und senden dem User eine http-Webseite als Response zurück.
7. Letztlich werden die Rohdaten an die Auswertungsmodule weitergeleitet. (Die Auswertungsmethode werden im nächsten Kapitel detailliert erklärt).

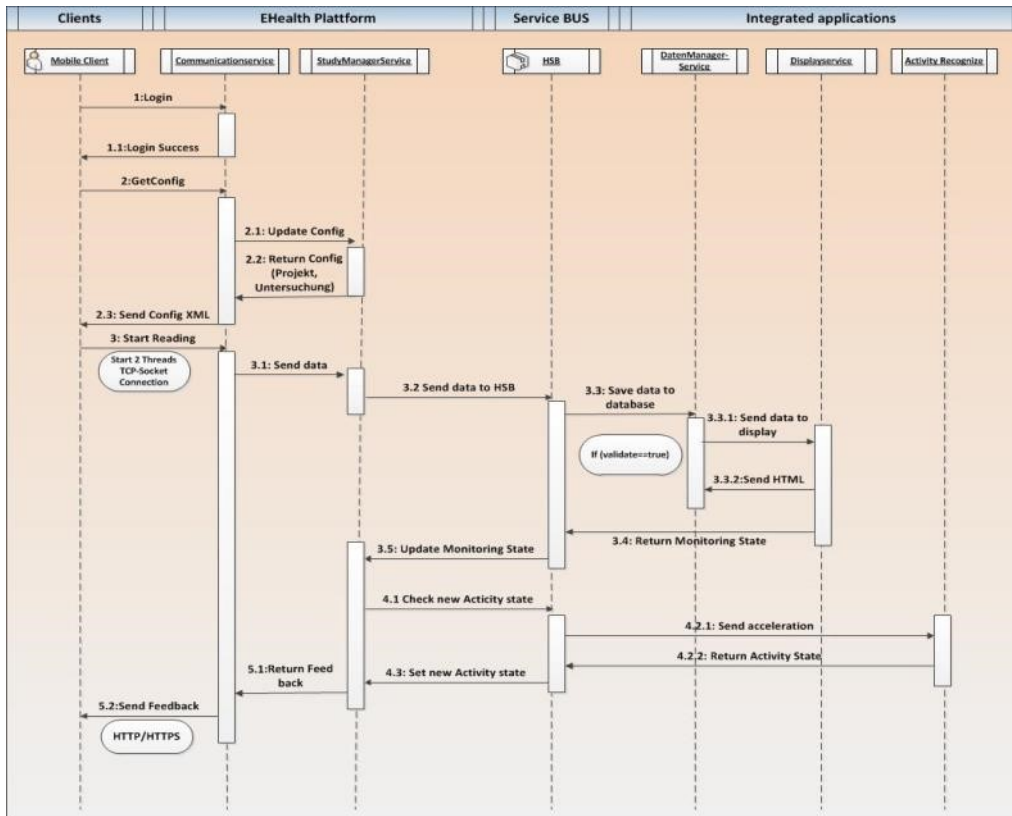


Abbildung 6-19 Messszenario im IPM-mHealth System

Der gesamte Verlauf des Prozesses muss der Konfiguration von hsb-config.xml entsprechen. Jedes Modul ist in HSB ein „mule-descriptor“. Wie die Messages empfangen werden, wird über den Konfigurationsendpoint des inbound-router-Knotens definiert und die Richtung der Messages beim Ausgang wird über den Endpoint der Konfiguration des outbound-router-Knotens eingestellt. Das Folgende ist die Konfiguration der IPM-Server-Umgebung:

Tabelle 6-3 IPM-Server-Umgebung

AMD-based 2.6 GHz 4 GB RAM, and Intel-based 3 GHz 4 GB RAM servers
Windows Server 2008
Java Run-time Environment 1.7.0_45
Mule current general availability package
A JMS provider server, IP: 139.30.204.203 ,Port: 7222
A Apache application server, IP: 139.30.204.203 ,Port: 8080
A Socket Kommunikation server, IP: 139.30.204.203 ,Port: 5598

Zunächst müssen am Anfang der Konfigurationsdatei einige Transformer definiert werden, welche die im Datenfluss übermittelten Informationen in die verschiedenen, von den unterschiedlichen Modulen verwendbaren Formate umwandeln. Beispielsweise müssen von HTTP Request stammende Informationen vom String-Format ins HL7 Format konvertiert werden. Um diese Art von Information zu konvertieren, wird wie beim in Abbildung 6-20 gezeigten Konvertierungsmodul verfahren.

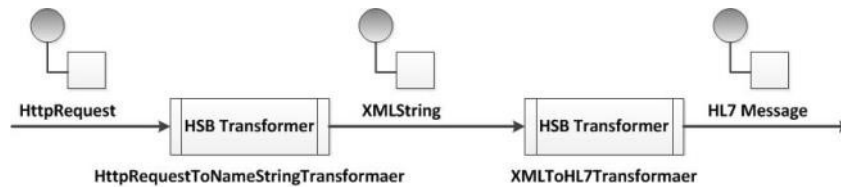


Abbildung 6-20 HSB Transformer-Modell

Nach dem Empfang der Informationen wird diese Art von Informationen zunächst von `HttpRequestToNameString` konvertiert. Der Eingang des Konvertierungsmoduls ist `HttpRequest`, dieses wird in ein `String`-Objekt konvertiert. Nachdem das `String`-Objekt durch das XSLT Konvertierungsmodul hindurch ist (Template bereits festgelegt CDA Schema), wird die Information dann in das benötigte HL7-Format umgewandelt.

```
<mule-hsb-configuration id="Enterprise-Applications-Mashup" version="2.0">
<custom-transformer name="ByteArrayToString" class="org.hsb.mhealth. ByteArrayToString" />
<custom-transformername="StringToByteArray" class="org.hsb.mhealth. StringToByteArray" />
<custom-transformername="StringToNameString" class="org.hsb.mhealth.StringToNameString"/>
<custom-transformername="NameStringToChatString" class=" org.hsb.mhealth.NameStringToChatString"/>
<custom-transformername="ChatStringToString" class="org.hsb.mhealth.ChatStringToString"/>
<custom-transformername="HttpRequestToNameString" class="org.hsb.mhealth.HttpRequestToNameString"/>
<custom-transformername="ExceptionToString" class="org.hsb.mhealth.ExceptionToString"/>
<custom-transformername="StringToHttpResponse" class="org.hsb.mhealth. StringToHttpResponse" />
<custom-transformer name=" XMLToHL7" class="org.hsb.mhealth.XMLToHL7" doc:name="Java"/>
<custom-transformer name=" HL7ToXml" class="org.hsb.mhealth.HL7ToXml" doc:name="Java"/>
```

Nach der in Abschnitt 6.1 beschriebenen Vorgehensweise wird das Modul definiert. Jeder darin enthaltene Service gehört zu den zuvor beschriebenen medizinischen Diensten.

```
<modelname="MeasureModel">
<servicename="CommunicationService">
<inbound>
```

1. Das erste Modul `CommunicationService` wird für die Kommunikation mit dem Client verwendet. Zunächst wird ein Entrypoint definiert um vom Client eingehende `Http` Informationen zu empfangen und der zuvor definierte Transformer verwendet, um `ByteArray` in `String`-Format umzuwandeln.

```
<inbound>
<tcp:inbound-endpoint host="139.30.204.203" port="5958" transformer-refs="HttpRequestToName-
String" synchronous="true"/>
</inbound>
```

An dieser Stelle verwenden wir die in Abschnitt 6.2.1 geschilderte Kommunikation-Service-Schnittstelle, das Kommunikationsmodul kann hier als Blackbox dienen. Seine konkrete Umsetzungsmethode ist unabhängig vom HSB, die Schnittstelle beinhaltet die Eingangsmethode der Business Logik des Kommunikationsmoduls.

```
<componentclass="org.hsb.mhealth.CommunicationService.CommunicationServiceImp"/>
<outbound>
```

Outbound ist der Kanal, über welchen das Modul Antwortdaten nach außen sendet; CommunicationService überprüft die vom Client übermittelten Message-Formate und übermittelt das Message-Ergebnis zurück an den HSB. An dieser Stelle werden zwei Message-Routings definiert, um die vom CommunicationService zurückgesendeten Informationsformate zu beurteilen. Wenn die Messages über die in Abschnitt 6.2.2 geschilderte Message-Struktur bewertet werden, so gehen sie in das IdentificationService-Modul über und der Client wird verifiziert.

```
<filtering-router>
<vm:outbound-endpoint path=" IdentificationServieRouter " synchronous="true"/>
<payload-type-filterexpectedType="org.hsb.mhealth.hello.NameString"/>
</filtering-router>
```

Andernfalls liegt eine fehlerhafte Information vor und die Informationsübermittlung wird abgebrochen.

```
<filtering-router>
<vm:outbound-endpoint path=" AlarmServiceRouter " synchronous="true"/>
<payload-type-filterexpectedType="java.lang.Exception"/>
</filtering-router>
</outbound>
```

An dieser Stelle wird das Modul zur Behandlung als unbekannter Fehler definiert.

```
<default-service-exception-strategy>
<vm:outbound-endpoint path="systemErrorHandler"/>
</default-service-exception-strategy>
</service>
```

2. Die Message wird an das IdentificationService-Modul weitergeleitet und der vom Client gesendete Benutzername und das Passwort werden überprüft.

```
<servicename="IdentificationServie">
<inbound>
<vm:inbound-endpoint path="IdentificationServieRouter" transformer-refs="NameStringTo
ChatString" responseTransformer-refs="ChatStringToString" synchronous="true"/>
</inbound>
<componentclass=" org.hsb.mhealth.IdentificationServie.IdentificationServieImp "/>
<outbound>
```

Über das Übermitteln der empfangenen Informationen erhält der IdentificationService die Parameter und führt eine Identitätsüberprüfung durch. Die vom Bus an den IdentificationService übermittelten Typen werden beurteilt und falls die Überprüfung bestanden ist, wird mit dem Empfang der vom Client gesendeten Daten begonnen und die Daten in dem vom Bus übermittelten nächsten Modul abgespeichert.

```
<filtering-router>
<vm:outbound-endpoint path="StudyManagerServiceRouter " synchronous="true"/>
<payload-type-filterexpectedType="org. hsb.mhealth.hello.NameString"/>
</filtering-router>
```

Andernfalls wird der Warn-Service gestartet.

```
<filtering-router>
<vm:outbound-endpoint path="AlarmServiceRouter" synchronous="true"/>
<payload-type-filterexpectedType="java.lang.Exception"/>
</filtering-router>
</outbound>
</service>
```

3. Die Message wird an den StudyManagerService übermittelt. Vor dem Beginn der Messung wird die Patientenkonfiguration überprüft.

```
<servicename="StudyManagerService">
<inbound>
<vm:inbound-endpoint path="StudyManagerServiceRouter" transformer-refs="NameStringToChatString" responseTransformer-refs="ChatStringToString" synchronous="true"/>
</inbound>
<componentclass="org. hsb.mhealth.StudyManagerService.StudyManagerServiceImpl"/>
<outbound>
```

Beim Übermitteln der empfangenen Informationen überprüft der StudyManagerService, ob neue Einstellungen beim Patienten vorhanden sind. Falls dies so ist, werden die Informationen erneuert und es wird damit begonnen, die empfangenen Messdaten in der Datenbank abzuspeichern.

```
<filtering-router>
<vm:outbound-endpoint path="DatenManagerRouter" synchronous="true"/>
<payload-type-filterexpectedType="org. hsb.mhealth.hello.NameString"/>
</filtering-router>
```

Wenn eine Änderung fehlschlägt, wird der Fehler-Warndienst gestartet.

```
<filtering-router>
<vm:outbound-endpoint path="AlarmServiceRouter" synchronous="true"/>
<payload-type-filterexpectedType="java.lang.Exception"/>
```

```
</filtering-router>
```

```
</outbound>
```

```
</service>
```

4. Die Message wird an den DatenManagerService weitergeleitet und der Empfang und die Abspeicherung der Messdaten werden gestartet. Das String-Format wird in das vom Datenmanagement-Modul verwendbare sql-Format umgewandelt.

```
<servicename="DatenManagerService">
```

```
<inbound>
```

```
<vm:inbound-endpoint path="DatenManagerRouter" transformer-refs="NameStringToChatString"
responseTransformer-refs="ChatStringToString" synchronous="true"/>
```

```
</inbound>
```

```
<componentclass="org. hsb.mhealth.DatenManagerService.DatenManagerServiceImpl"/>
```

```
<outbound>
```

Durch die übermittelten Informationen erhält der DatenManagerService die Parameter und untersucht, ob die Werte der Norm entsprechen. Sind sie korrekt, so werden die Daten abgespeichert und der Datenanzeige-Service gestartet.

```
<filtering-router>
```

```
<vm:outbound-endpoint path="LiveViewServiceRouter" synchronous="true"/>
```

```
<payload-type-filterexpectedType="org. hsb.mhealth.hello.NameString"/>
```

```
</filtering-router>
```

Weichen die Daten von der Norm ab, so wird der Warn-Service gestartet.

```
<filtering-router>
```

```
<vm:outbound-endpoint path="AlarmServiceRouter" synchronous="true"/>
```

```
<payload-type-filterexpectedType="java.lang.Exception"/>
```

```
</filtering-router>
```

```
</outbound>
```

```
</service>
```

5. Die Message wird an das LiveViewService-Modul weitergeleitet und die Daten ausgelesen. Auf einer HTTP-Seite werden die Daten angezeigt.

```
<servicename="LiveViewService">
```

```
<inbound>
```

```
<vm:inbound-endpoint path="LiveViewServiceRouter" transformer-refs="NameStringToChatString"
responseTransformer-refs="ChatStringToString" synchronous="true"/>
```

```
</inbound>
```

```
<componentclass="org. hsb.mhealth. LiveViewService.LiveViewServiceImpl"/>
```

```
<outbound>
```

Über die übermittelten Daten puffert LiveViewService die Parameter. Die Daten werden in http Response abgespeichert und an die Client-Seite zurückgeschickt. An dieser Stelle wird String in das HTTP Response_Format umgewandelt.

```
<filtering-router>
```

```
<payload-type-filter expectedType="org.hsb.mhealth.hello.NameString" responseTransformer-refs="StringToHttpResponse"/>
```

```
</filtering-router>
```

Falls ungewöhnliche Daten entdeckt werden, wird der Warn-Service gestartet.

```
<filtering-router>
```

```
<vm:outbound-endpoint path="AlarmServiceRouter" synchronous="true"/>
```

```
<payload-type-filter expectedType="java.lang.Exception"/>
```

```
</filtering-router>
```

```
</outbound>
```

```
</service>
```

```
<servicename="AlarmService">
```

```
<inbound>
```

```
<vm:inbound-endpoint path="AlarmServiceRouter" responseTransformer-refs="ExceptionToString" synchronous="true"/>
```

```
</inbound>
```

```
</service>
```

Die Fehlermeldung wird ausgedruckt und der HSB verlassen

```
<servicename="SystemErrorHandler">
```

```
<inbound>
```

```
<vm:inbound-endpoint path="systemErrorHandler" synchronous="true"/>
```

```
</inbound>
```

```
<pass-through-router>
```

```
<outbound>
```

```
<pass-through-router>
```

```
<stdio:outbound-endpoint system="ERR"/>
```

```
</pass-through-router>
```

```
</outbound>
```

```
</service>
```

```
</hsb>
```

7 Entwurf der automatisierten Aktivitätserkennung

In diesem Kapitel werden beispielhaft für weitere Anwendungen das Design und die Umsetzung einer automatisierten Aktivitätserkennungs-Methode auf der Grundlage von Beschleunigungssensoren erläutert. Zunächst wird der für diese Arbeit eingesetzte Equivalital Sensor näher erläutert. Im Anschluss werden die neuen Abläufe und Methoden dargelegt, wie mittels von Beschleunigungsdaten Patientenverhalten erkannt werden kann. Bewegungserkennung kann als Klassifizierungsproblem des maschinellen Lernens definiert werden [145]. Der grundlegende Ablauf der Bewegungserkennung lautet wie folgt: (1) Zunächst werden die Beschleunigungsdaten erfasst und geglättet, (2) danach werden spezifische Sensorwerte innerhalb eines definierten Zeitfensters extrahiert (*Mittelwert, Maximalwert, Standardabweichung* etc.). Das WEKA-Werkzeug wird verwendet, um die Messdaten unter den verschiedenen Zuständen zu klassifizieren. (3) Zuletzt wird mit dem Nächsten-Nachbar-Klassifikation-basierten Bewegungserkennungsalgorithmus die Bewegungserkennung des Patienten vervollständigt.

7.1 Vorstellung des Equivalital Eq01 Multiparameter-Sensors

In diesem Projekt wurde der Equivalital-Multiparameter-Sensor (siehe Abbildung 7-1,7-2) zum Sammeln der Beschleunigungsdaten verwendet. Das Sensormodul Equivalital mit seinem speziellen Brustgurt eignet sich für die Aufzeichnung der benötigten Messwerte. Es bietet die Möglichkeit, Herzfunktionen (über die integrierten faserbasierten Elektroden mit Silbermantel), Lungenfunktion (über einen integrierten Dehnmessstreifen) und die Körpertemperatur (über einen integrierten Thermistor) zu erheben.



Abbildung 7-1 Hidalgo Equivalital Multi-parameter Sensor[146]

Das Sensormodul selbst verfügt ebenfalls über einen integrierten dreiachsigen Beschleunigungssensor ADXL330 [147] und erlaubt Aussagen über die Ausrichtung (z.B. aufrecht, liegend, Rückenlage). Zusätzlich berechnet das Gerät intern die Herz- und Atemrate aus den Rohdaten und sendet Benachrichtigungen, falls bestimmte kritische Werte überschritten werden. Die gesamten Daten (Rohdaten, berechnete Daten und Benachrichtigungen) können beständig und kabellos vom Sensormodul über eine Bluetooth-Schnittstelle übermittelt werden.



Abbildung 7-2 Equivalital Sensor Brustgurt [148]

Die Sensorlösung ermöglicht selbst unter voller Auslastung einen 24h-Betrieb. Aufgrund des geringen Gesamtgewichts von 175g (Elektronikmodul des Sensors 75g und Gurt 100g) und der Schutzklasse IP 67 (0,4m=30 min.) kann das mHealth-System beinahe überall und von jedem eingesetzt werden.

7.1.1 ADXL330 3-Axis Accelerometer

Der ADXL330 ist ein kleiner, dünner, kompletter dreiachsiger Beschleunigungssensor mit niedrigem Energieverbrauch und signalgesteuertem Spannungsausgang auf einem einzelnen IC. Das Produkt misst die Beschleunigung mit einem Minimalwert von ± 3 g. Es ist in der Lage, die statische Beschleunigung der Gravitation in neigungssensitiven Anwendungen zu messen, genauso wie die dynamische Beschleunigung als Ergebnis von Bewegung, Erschütterung oder Vibration.

Der Sensor besteht aus einer mikrobearbeiteten Struktur mit Polysilizium-Oberfläche auf einer Siliziumscheibe (Abbildung 7-3). Polysilizium-Federn fixieren die Struktur über der Scheibe und bieten Widerstand gegen Beschleunigungskräfte. Die Abweichung der Struktur

wird mittels eines Differentialkondensators gemessen, welcher aus unabhängigen Platten und Platten, welche an der sich bewegenden Masse befestigt sind, besteht. Die festen Platten werden von 180° phasenverschobenen Rechteckschwingungen betrieben. Die Beschleunigung verschiebt die sich bewegende Masse und bringt den Differentialkondensator aus dem Gleichgewicht, wodurch eine Sensorausgabe entsteht, deren Amplitude proportional zur Beschleunigung ist. Daraufhin werden phasensensitive Demodulationstechniken verwendet, um die Stärke und Richtung der Beschleunigung zu bestimmen.

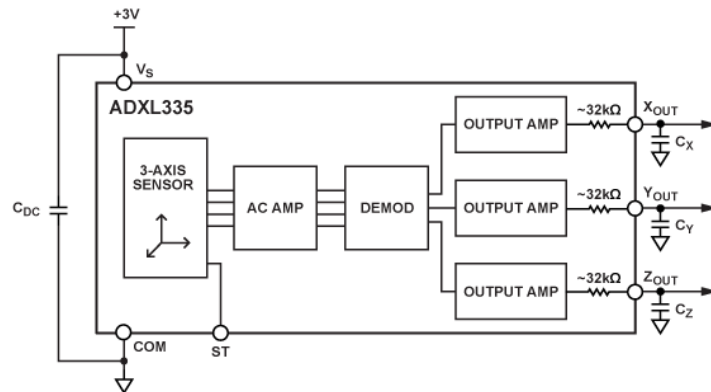


Abbildung 7-3 ADXL330 3-Axis Accelerometer [149]

Der Benutzer wählt die Bandbreite des Beschleunigungssensors aus, durch die Verwendung der CX, CY und CZ Kompensatoren an den Anschlüssen XOUT, YOUT und ZOUT. Die Bandbreite kann entsprechend der Anwendung gewählt werden und reicht von 0,5 Hz bis 1600 Hz für die X- und Y-Achse und von 0,5 Hz bis 550 Hz für die Z-Achse.

Der ADXL330 ist in einem kleinen, unauffälligen Plastikgehäuse von $4 \text{ mm} \times 4 \text{ mm} \times 1.45 \text{ mm}$, 16-lead erhältlich.

Der ADXL330 verwendet eine einzige Struktur, um die X-, Y- und Z-Achse zu erfassen. Aus diesem Grund sind die Richtungen stark orthogonal und es gibt nur eine geringe Empfindlichkeit zwischen den Achsen. Eine mechanische Fehlstellung des Sensors ist der Hauptgrund für die Empfindlichkeit zwischen den Achsen. Die mechanische Fehlstellung kann selbstverständlich auf der Systemebene durch Kalibrierung ausgeglichen werden.

7.1.2 Dreiachsige Beschleunigungssensoren

Dreiachsige Beschleunigungssensoren sind eine Art von Sensoren, welche in Echtzeit die Einzelwerte zur X-, Y- und Z Achse ermitteln. Über einen Beschleunigungswert können die Sensoren einen Geschwindigkeitswert und die Verschiebung messen. Wie Abbildung 7-4 zeigt, verläuft die X-Achse im System auf der Ebene des Geräts und von links nach rechts, die

Y-Achse verläuft von unten nach oben und die Z-Achse verläuft von unter dem Bildschirm in Richtung der Oberfläche.

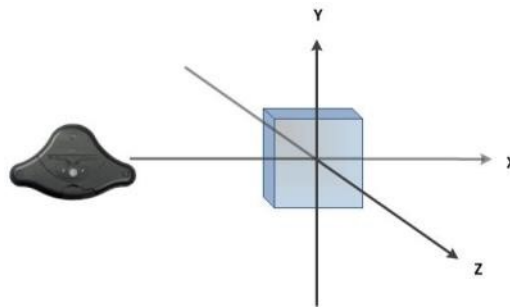


Abbildung 7-4 Direction of the EQ01 3-Axis Accelerometer

Der Equivital-Sensor sammelt die Beschleunigungsdaten, nachdem dieser mit IPM-MC eingebunden wurde, hauptsächlich von 3 Parametern: value[0], value[1], value[2] (Einheit m/s) value[0]: Der Wert der X-Achsen-Beschleunigung minus der Fallbeschleunigung entlang der X-Achse. Value[1]: Der Wert der X-Achsen-Beschleunigung minus der Fallbeschleunigung entlang der Y-Achse. Value[2]: Der Wert Z-Achse der Beschleunigung minus der Fallbeschleunigung entlang der Z-Achse. Wenn also beispielsweise das Gerät flach auf eine Oberfläche gelegt wird, ist die Beschleunigung die Fallbeschleunigung g . Über $0-g$ (die Fallbeschleunigung g ist in entgegengesetzter Richtung zum Verlauf der Z-Achse und daher ein negativer Wert) erhält man den Wert g für value[2]. Schiebt man den Sensor horizontal nach rechts, so ergibt dies eine Beschleunigung in X-Richtung und damit einen positiven Wert für value[0]. Wenn der Sensor vertikal mit einer Beschleunigung von a m/s bewegt wird, so ergibt dies für value[2] den Wert $(a+g)$ m/s, welcher sich aus der Beschleunigung a minus der Fallbeschleunigung $-g$ ergibt.

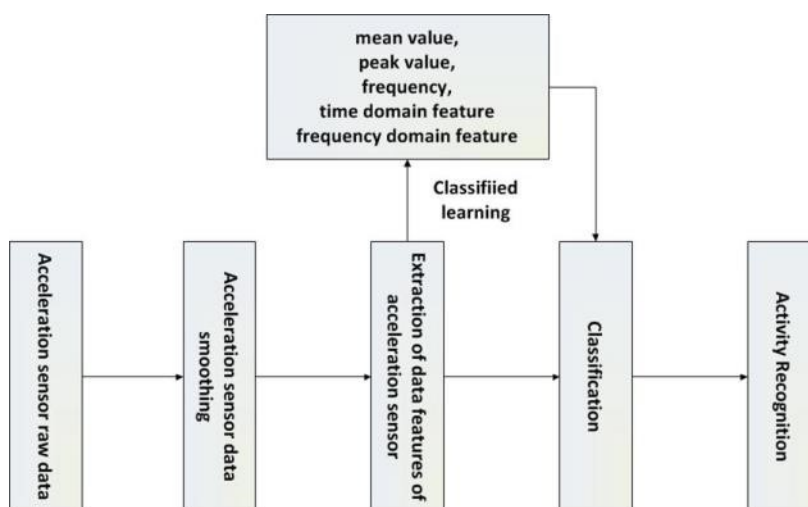


Abbildung 7-5 Verlauf der Bewegungserkennung

7.2 Entwurf des Bewegungserkennungsalgorithmus

Der grundlegende Verlauf der Bewegungserkennung ist folgender:

1. Zunächst werden Beschleunigungssensordaten eingeholt und geglättet.
2. Danach werden spezifische Sensorwerte innerhalb eines definierten Zeitfensters extrahiert (Mittelwert, Maximalwert, Standardabweichung etc.).
3. Das WEKA-Werkzeug wird verwendet, um die Messdaten unter den verschiedenen Zuständen zu klassifizieren [150].
4. Zuletzt wird mit dem Nächsten-Nachbar-Algorithmus die Bewegungserkennung des Patienten vervollständigt (siehe Abbildung 7-5).

7.2.1 Erfassung der Beschleunigungssensordaten

Über den Aufruf der im IPM-MC angebotenen Sensorkommunikationsmodule können die Rohdaten der Equivital-Beschleunigungssensoren erfasst werden. Es wird eine eigene Klasse `SensorEventListenerAccelerometer` eingerichtet. Diese Klasse erbt die Klasse `SensorEventListener` von Android API. Die Klasse `SensorEventListenerAccelerometer` bietet folgende Funktionen: Registrierung von Sensor-Services, Widerrufen von Sensor-Services, Änderung von Sensordaten und Änderung der Genauigkeit des Sensors. Der Aufrufprozess entspricht Abbildung 7-6. Hier erfolgt die gesamte Arbeit der Datenerfassung innerhalb der Methode `onSensorChanged()` (der Änderung der Sensordaten).

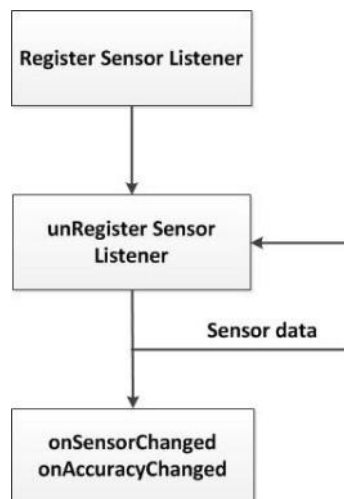


Abbildung 7-6 Aufrufprozess der Beschleunigungssensordaten

An dieser Stelle ist es notwendig, die Funktion `onSensorChange` erneut zu realisieren: Nachdem die Sensordaten erkannt wurden, können die Sensordaten der drei Richtungen alle über den Aufruf von `SensorManager.Data_X`, `SensorManager.Data_Y`, `SensorManager.Data_Z`, etc. erlangt werden. Da die konkrete Veränderung der Daten der drei

Richtungen den späteren Bewegungserkennungsalgorithmus nicht stark beeinflusst, wird hier eine Verarbeitung durchgeführt. Bei jeder Veränderung der Werte der Richtungsbeschleunigung wird der Vektor mit folgender Formel ermittelt:

$$\mathbf{Data} = \sqrt{(\mathit{Acc_X})^2 + (\mathit{Acc_Y})^2 + (\mathit{Acc_Z})^2}$$

An dieser Stelle werden die Daten und die jeweilige Zeit (in Millisekunden) nach jeder Zustandsänderung der Beschleunigungsdaten getrennt in zwei Arrays `time_series[]` und `data_series[]` gespeichert.

Nach der Erfassung erfolgt eine Zeitreihenanalyse der Rohdaten wie in Abbildung 7-7 zu sehen. Horizontal verläuft die Zeitachse, also `time_series[]`, vertikal werden die Beschleunigungsdaten der Sensoren angezeigt, also `data_series[]`.



Abbildung 7-7 Zeitreihenanalyse der Beschleunigungssensordaten

Die Reihenfolge der Aktionen in der obigen Abbildung 7-7 ist: (1) *Gehen*, (2) *3x Springen*, (3) *Laufen*, (4) *langames Gehen*. Aus den Ursprungsdaten lässt sich erkennen, dass sich unterschiedliche Aktionen in verschiedenen Datenmerkmalen wie Mittelwert, Maximalwert und Frequenz offensichtlich unterscheiden.

7.2.2 Glättung der Beschleunigungssensordaten

Da die vom IPM-MC verwendete Schnittstelle zum Sammeln der Sensordaten `OnSensorChange()` diese passiv übermittelt, wird nur bei einer Änderung des Zustands des Sensors die Funktion aufgerufen. Aus diesem Grund belegen die gesammelten Daten nicht gleichmäßig Zeitpunkte. Wenn die Frequenz der Überwachung auf `SENSOR_DELAY_FASTEST` (die höchste Frequenz) eingestellt wird, beträgt der zeitliche Zwischenraum beim Sammeln der Daten durchschnittliche 40 ms. Um den zeitlichen Zwischenraum zwischen den gesammelten Daten gleichmäßig zu gestalten, wird in dieser Arbeit auf Grundlage der bereits gesammelten Rohdaten in `time_series[]` und `data_series[]` folgender Algorithmus zur Glättung verwendet, welcher einen mittleren Datenwert in einem zeitlichen Zwischenraum von 40 ms berechnet:

Bei einem gesuchten Zeitpunkt T_0 werden die beiden nächstgelegenen Zeitpunkte in `time_series[]` als T_1 und T_2 bezeichnet. Aus T_1 und T_2 erhält man die korrespondierenden Daten in `data_series[]` D_1 und D_2 . Wird davon ausgegangen, dass zwischen T_1 und T_2 die Beschleunigung des Sensors eine lineare Veränderung erfährt, so kann der Datenwert D_0 von T_0 mittels folgender Methode des gewichteten Mittelwerts bestimmt werden:

$$D_0 = \frac{D_1 * (T_2 - T_0) + D_2 * (T_0 - T_1)}{T_2 - T_1}$$

Mit dem obigen Algorithmus lässt sich eine neue `time_series[]` und `data_series[]` berechnen und abstrahieren.

7.2.3 Extraktion der Merkmale der Beschleunigungssensordaten

Um im Stande zu sein, Aktionen zuverlässig zu erfassen und zu unterscheiden, kann die Größe des Fensters des Datensets auf $256 * 40$ eingestellt werden, also 10240 ms. Damit können in der Array `data_series[]` 256 Daten abgelegt werden. Nachdem die Array voll ist, wird sie geleert und wieder von `data_series[0]` gespeichert. Nachdem die Array voll ist, anschließend wird erneut eine Datenextraktion von Zeitfeldmerkmal und Frequenzfeldmerkmal durchgeführt.

Zeitfeldmerkmale

Das Zeitfeldmerkmal ist der Wert der Datenmerkmale in einem bestimmten Zeitfenster. Da einen Schritt zuvor eine Glättung der Daten durchgeführt wurde, handelt es sich bei den in `data_series[]` abgelegten Daten der Beschleunigungssensoren um in einem Intervall von 40 ms erfasste Daten. Daher können direkt anhand der Mittelwerte, Standardabweichung, Maximalwerte und Minimalwerte die Zeitfeldmerkmale der Beschleunigungssensoren in 10240 ms (ungefähr 10 s) ermittelt werden.

Frequenzfeldmerkmale

Da die Zeitfeldmerkmale nicht in der Lage sind, die bedeutenden Frequenzfeldmerkmale der Bewegung widerzuspiegeln, müssen noch mittels einer Fast-Fourier-Transformation [151] die Frequenzfeldmerkmale der Daten ermittelt werden. Hier wird zunächst für die Array `data_service[]` eine Fast-Fourier-Transformation (FFT) durchgeführt. Die FFT ist ein schneller Algorithmus der diskreten Fourier-Transformation. Durch diesen Algorithmus können die Daten effizient vom Zeitbereich in den Frequenzbereich übertragen werden. Bei der schnellen Fourier-Transformation für `data_series[]` (Die Größe der `data_series[]` Array beträgt 256, sie kann direkt mit der FFT berechnet werden) erhält man die komplexe Array

des Raumes des Frequenzbereichs. In der Array beinhaltet jede komplexe Zahl einen Realteil (real) und einen virtuellen Teil (image). Hier kann mittels folgender Beziehungen der korrespondierende Fourier-Wert jeder Frequenz im Frequenzfeld ermittelt werden, also die Matrix dieser komplexen Zahl.

$$\mathbf{Mod} = \sqrt{\mathbf{real}^2 + \mathbf{image}^2}$$

Der in dieser Arbeit angeforderte Frequenzfeldmerkmalwert Energiewert (energy) ist das Quadrat des Fournierwerts im Fenster und des Größenverhältnisses des Fensters. Also:

$$\mathbf{Energy} = \sum_{k=0}^w \frac{(\mathbf{Mod}[k])^2}{w}$$

Es muss darauf geachtet werden, dass die hier eingerichtete Fenstergröße w 256 beträgt. Die Frequenzfeldenergie spiegelt die Periodizität der Daten wider. Bei Bewegungen mit relativ hoher Periode (wie Laufen) ist die Frequenzfeldenergie ebenfalls deutlich höher.

7.2.4 Analyse des Klassifikationslernprozesses

Nachdem der Datenerfassungsprozess abgeschlossen ist, werden die Daten im IPM-MC abgelegt. Das in der Arbeit verwendete Testgerät ist ein Sony Experia Mobiltelefon. Nach dem Start des Prozesses vollführte die Testperson der Reihenfolge nach die folgenden Aktionen: (1) *Gehen*, (2) *Springen*, (3) *Laufen*, (4) *Stehen*, (5) *Treppen hinaufsteigen*, (6) *Treppen hinabsteigen*.

Jede Aktion dauerte 30 bis 50 Sekunden, um sicherzustellen, dass für jede Aktion 3 bis 5 Gruppen von Eigenwerten gesammelt werden konnten. Nach dem Ende wurde der Prozess beendet. Dabei wurden durch den IPM-MC automatisch die Eigenwerte auf den IPM-ZS hochgeladen.

Unter Verwendung des oben genannten Datenerfassungsprozesses wurden 45 Bewegungszustände-Sample erfasst. Danach wurde das WEKA-Werkzeug verwendet, um für diese 45 Testdaten getrennt Machine Learning durchzuführen und danach eine Grundlage für die Bewegungserkennung zu schaffen.

Die erfassten Zustandsinformationen umfassten die sechs Aktionen *Gehen*, *Springen*, *Stehen*, *Laufen*, *Treppen hinaufsteigen* und *Treppen hinabsteigen*. Durch den Test konnte festgestellt werden, dass verschiedene Algorithmen die Datenmenge unterschiedlich einteilten. Mittels

einer Confusion Matrix (s.u.) sollen die unterschiedlichen Erscheinungen der Algorithmen beschrieben werden.

Confusion Matrix

Die Confusion Matrix [152] ist eine N-stufige Matrix, wobei N die Zahl der Zielklassifikationen ist. Jede Spalte im Array steht für die Zahl jeder prognostizierten Klassifikation und jede Reihe in der Matrix steht für die Zahl der tatsächlichen Klassifikation. Wenn sich alle Beispiele in der Diagonalen der Matrix befinden, so liegt die Präzision dieses Klassifikators bei 100%. Weichen die Beispiele in der Matrix von der Diagonalen ab, so ist die Präzision des Algorithmus niedrig. Die Confusion Matrix kann auch dafür verwendet werden, um den Grad der Konfusion zwischen zwei Klassifikationen zu analysieren. Je größer die Anzahl der korrespondierenden Matrix-Items zweier Klassen, desto größer ist der Grad der Konfusion zwischen den beiden Klassen, demzufolge ist die Präzision der Unterscheidung der beiden Klassen durch den Klassifikator niedriger.

Tabelle 7-1 Confusion Matrix der Nächster-Nachbar-Klassifikation (IB1)

Activity	Classification					
	Jump	Walk-ing	Stand-ing	Down-stairs	Up-stairs	Running
Jump	7	0	0	0	0	1
Walking	0	11	0	0	0	0
Standing	0	1	11	0	0	0
Down-stairs	0	0	0	6	0	0
Upstairs	0	0	0	0	4	0
Running	0	0	0	0	0	4

Aus der Confusion Matrix lässt sich erkennen, dass die Genauigkeit der Klassifikation vom Nächsten-Nachbar-Algorithmus [153] dennoch verhältnismäßig hoch ist. Insgesamt liegt die Präzision bei 95.556%. Eine „Springen“-Aktion wurde im Beispiel fälschlicherweise als „Laufen“ klassifiziert und eine „Stehen“-Aktion wurde als „Gehen“ klassifiziert. Dies zeigt, dass diese beiden Aktionen beim Klassifikationsalgorithmus eventuell zu Konfusionsproblemen führen können.

Bayes-Klassifikators

Die Confusion Matrix des Naiven Bayes-Klassifikators [154] befindet sich in Tabelle 7-2. Aus der Confusion Matrix lässt sich erkennen, dass auch die Präzision des Naiven Bayes-Klassifikators verhältnismäßig hoch ist. Insgesamt liegt die Präzision bei 93.333%. Allerdings zeigt die Matrix, dass der Naive Bayes-Klassifikator Stehen und Gehen drei Mal verwechselt,

dabei hat er in einem Fall Gehen als Stehen klassifiziert und in zwei Fällen Stehen als Gehen. Dies zeigt für die Analyse des Algorithmus, dass die verschiedenen Merkmale der beiden Aktionen Gehen und Stehen nahe beieinander liegen und die Merkmale nicht gut von einander getrennt werden können. Wenn der Grad der Korrelation der drei Merkmale Maximalwert, Minimalwert und Mittelwert verhältnismäßig groß ist, so kann dies zu Verwechslungen beim naiven Bayes-Klassifikator führen.

Tabelle 7-2 Confusion Matrix der Bayes-Klassifikation

Activity	Classification					
	Jump	Walk-ing	Stand-ing	Down-stairs	Up-stairs	Running
Jump	8	0	0	0	0	1
Walking	0	10	1	0	0	0
Standing	0	2	10	0	0	0
Down-stairs	0	0	0	6	0	0
Upstairs	0	0	0	0	4	0
Running	0	0	0	0	0	4

Entscheidungsbaum-Klassifikationsalgorithmus

Die Confusion Matrix des Entscheidungsbaum-Klassifikationsalgorithmus [155] findet sich in Tabelle 7-3. Aus der Confusion Matrix lässt sich erkennen, dass die Präzision der Klassifikation des C4.5 Entscheidungsbaum-Algorithmus verhältnismäßig niedrig ist und nur 80% erreicht. Beim Entscheidungsbaum-Algorithmus kommt es bei allen Aktionen zu Verwechslungen. Hier ist der Grund für die niedrige Präzision des Entscheidungsbaum-Algorithmus, dass die Beispieldatensets nicht ausreichen. Der Entscheidungsbaum-Algorithmus benötigt mehr Daten, um die Informationsmerkmale der verschiedenen Attribute präzise zu berechnen. Die nicht ausreichenden Beispiele führen zu einem zu starken „Rauschen“ der Datensets und reduzieren so die Klassifikationsfähigkeit des Entscheidungsbaum-Algorithmus.

Was die Zeiteffizienz angeht, so ist von den Klassifikatoren derjenige mit dem längsten Zeitaufwand für das Training mit dem Training-Sample der Entscheidungsbaum-Algorithmus, derjenige mit dem geringsten Zeitaufwand ist die Nächste-Nachbar-Klassifikation. Bei der Klassifikation des Test-Samples benötigte allerdings der naive Bayes-Klassifikator den höchsten Zeitaufwand, den geringsten Zeitaufwand dagegen benötigte der Entscheidungsbaum-Algorithmus.

Tabelle 7-3 Confusion Matrix der Entscheidungsbaum-Klassifikationsalgorithmus

Activity	Classification
----------	----------------

	Jump	Walk- ing	Stand-ing	Down- stairs	Up-stairs	Running
Jump	6	0	1	0	0	1
Walking	0	10	1	0	0	0
Standing	0	1	11	0	0	0
Down-stairs	1	0	0	3	1	1
Upstairs	0	0	0	0	3	1
Running	0	0	0	0	1	3

Um den Einfluss verschiedener Merkmale von Datensets auf die Klassifikation der Datensets zu vergleichen, werden hiermit die Attribute in den Datensets entfernt. Nachdem die Entfernung verschiedener Merkmale erfolgte, wird die Veränderung der Präzision des Klassifikators bei der Einteilung der Datensets verglichen. Tabelle 7-4 zeigt, dass der Einfluss auf die Einteilung der Daten durch verschiedene Klassifikatoren bei Entfernen von verschiedenen Merkmalen, auf die Einteilung der Daten durch verschiedene Klassifikatoren unterschiedlich ausfällt. Die Nächster-Nachbar-Methode ist bei Minimalwerten relativ empfindlich, während die Frequenzfeldenergie und der Mittelwert nur geringen Einfluss haben. Der naive Bayes-Klassifikator ist bei Standardabweichungen und Minimalwerten empfindlich. Dagegen hat die Entfernung eines Merkmals keinen Einfluss auf die Einteilungsgenauigkeit des Entscheidungsbaum-Algorithmus.

Tabelle 7-4 Einfluss verschiedener Merkmale auf die Einteilung der Daten

Klassifi- kators algorithmus	Einfluss bei Entfernen von verschiedener Merkmale					
	Mittel- wert	Standard- abweichung	Max.- Wert	Min.- Wert	Frequenz- energiewert	Alle Eigen- schaften
nearest- neighbor	95.55%	93.20%	95.55%	91.11%	94.16%	95.55%
Bayes	93.37%	91.11%	93.33%	88.11%	93.33%	93.33%
Decision tree	82.10%	80%	80%	80%	82.33%	80%

Durch den Vergleich und die zusammenfassende Analyse der verschiedenen Algorithmen ergibt sich, dass die Nächster-Nachbar-Methode im Vergleich zu den anderen Klassifikationsalgorithmen eine höhere Genauigkeit und verhältnismäßig hohe Effizienz besitzt.

7.2.5 Analyse des Bewegungserkennungsprozesses

Das WEKA Data-Mining-Werkzeug bietet ein Java-Develop-Paket. WEKA wird mit unserem HSB verbunden, um die in WEKA integrierte Nächster-Nachbar-Methode zu verwenden, womit das Verhalten und der Zustand des Patienten in Echtzeit analysiert und differenziert werden sollen. Der Prozess der Bewegungserkennung verläuft wie in Abbildung 7-8 zu sehen:

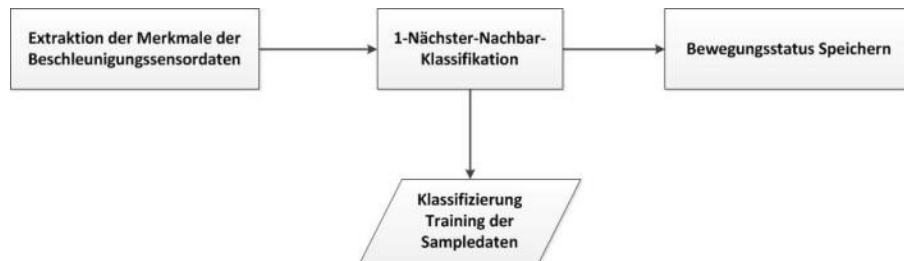


Abbildung 7-8 Prozess der Bewegungserkennung

Zunächst muss ein Training-Sample beschafft werden. Das im oberen Abschnitt im IPM-MC getestete Training-Sample wird in das Monitoring-System überführt und in *trainfile.arff* benannt. Damit der Klassifikator das Training-Sample lesen kann, muss der Dateifluss von *trainfile.arff* im WEKA-Dateifluss als Instanz eingefügt werden.

Nach dem Lesen des Training-Samples wird ein neuer Klassifikator *rn_classifier* erstellt, und seine Instantiation als Nächster-Nachbar-Algorithmus durchgeführt. Der Parameter *k* des *ffik()* Klassifikators beträgt in der Defaultaufforderung 1, also ein Klassifikator der Nächster-Nachbar-Methode. Danach wird das Training-Sample von *trainfile.arff* zum Training verwendet.

Nach dem Training können die Eigenwerte der Echtzeit-Beschleunigungssensoren dazu verwendet werden, das Verhalten und den Zustand des Patienten zu erkennen. Zunächst werden Informationen wie Merkmale, Merkmaltypen und Zielmerkmale des WEKA-Beispiels definiert, um die Bewegungsunterscheidungsprozesse für das IPM-MC zu analysieren. Die Testperson ist mit einem Android-Mobiltelefon ausgestattet und soll nach dem Start des IPM-MC-Prozesses eine Reihe von Bewegungen durchführen, wobei ein Beobachter gleichzeitig die jeweiligen im Zeitraum vom Tester durchgeführten Bewegungen aufzeichnet. Der IPM-MC-Prozess unterscheidet die Bewegungen und lädt diese gleichzeitig auf den Server, wo sie in der Datei *Action-Recognition.csv* abgespeichert werden. Nach Abschluss des Tests wird im System des Monitoring-Zentrums diese Datei geöffnet und die Versuchsergebnisse können verglichen werden. Die Ergebnisse der Testabschnitte sind Tabelle 7-5 zu entnehmen:

Aus den Versuchsergebnissen wird ersichtlich, dass die Exaktheit der Erkennung von Stehen und Springen verhältnismäßig hoch ist, auch Treppen hinaufsteigen und Treppen hinabsteigen kann mit verhältnismäßig hoher Genauigkeit bestimmt werden, aber Laufen, Gehen, etc. sind bei der Erkennung fehleranfällig.

Die Versuchsergebnisse zeigen zudem, dass wenn der Tester fortlaufend mehrere Bewegungen durchführt, die Erkennung des Übergangs von verschiedenen Bewegungen extrem fehleranfällig ist. Der Grund für diese fehlerhafte Erkennung liegt in der Auswahl des Zeitfensters für das Sammeln der Testdaten. In dieser Arbeit wurde ein Zeitfenster von 10 Sekunden gewählt, also führt der IPM-MC für den Benutzer alle 10 Sekunden eine Bewegungserkennung durch. Wenn in den 10 Sekunden danach eine Änderung des Verhaltens des Benutzers erfolgt, können innerhalb der 10 Sekunden-Datenabschnitte die korrespondierenden Beschleunigungsdaten verschiedener Bewegungen vermischt werden, wodurch die Bewegungen des Benutzers und die Merkmale der Daten nicht übereinstimmen, was letzten Endes zu einer fehlerhaften Erkennung der Bewegungen führt. Dieses Problem kann durch die Auswahl eines vergleichsweise kurzen Zeitfensters gelöst werden. Das kurze Zeitfenster kann möglicherweise zu geringeren Problemen mit der Übereinstimmung zwischen den Bewegungen und den korrespondierenden Datenmerkmalen führen.

Tabelle 7-5 Testergebnisse

Zeit	Benutzerverhalten	Erkennungsergebnisse	Ergebnis
11:04:02	Stehen	Stehen	Korrekte
11:04:13	Stehen	Stehen	Korrekte
11:04:25	Springen	Springen	Korrekte
11:04:36	Springen	Springen	Korrekte
11:08:36	Laufen	Springen	Fehler
11:08:46	Laufen	Laufen	Korrekte
11:08:57	Laufen	Laufen	Korrekte
11:09:07	Gehen	Springen	Fehler
11:09:18	Gehen	Gehen	Korrekte
11:09:28	Gehen	Gehen	Korrekte
11:09:39	Treppen hinaufsteigen	Treppen hinaufsteigen	Korrekte
11:09:50	Treppen hinaufsteigen	Treppen hinaufsteigen	Korrekte
11:10:00	Stehen	Gehen	Fehler
11:10:11	Treppen hinabsteigen	Treppen hinabsteigen	Korrekte
11:10:22	Treppen hinabsteigen	Treppen hinabsteigen	Korrekte
11:10:32	Treppen hinabsteigen	Treppen hinabsteigen	Korrekte

8Evaluierung und Anwendung

Untersuchung zur Auslastung des mobilen Clients und des Serversystems

Das in dieser Arbeit entwickelte mHealth-System unterstützt einen zentralen Server und verbindet zugleich mehrere Clients. Zudem empfängt es in Echtzeit die physiologischen Parameter des Patienten. In diesem Kapitel wird zunächst die Belastung der Datenübertragung des Systems getestet. Durch den Test soll des Weiteren auch die maximale zu verbindende Anzahl von Clients bestätigt werden. Um Stabilität und Zuverlässigkeit des Systems zu testen, werden drei Tests durchgeführt.

Testumgebung und Versuchsgeräte

Die Tests werden auf einem Smartphone mit Android 2.3 Betriebssystem durchgeführt. Die Testgeräte umfassen Sony Experia Arcs, Hidalgo Equivital Multiparameter Sensor, Wahoo Bewegungssensor und einen zentralen Server mit Windows 2008 Betriebssystem (siehe Abbildung 8-1).



Abbildung 8-1 Testumgebung und Versuchsgeräte

8.1 Aufbau und Durchführung der Untersuchung

Das Hauptziel der Untersuchung ist es, die fundamentalen Funktionen jedes Moduls zu testen und zu erkennen, ob das Programm vom Client den Fernzugriff auf die Datenbank erreichen kann. Das eingepackte Android-Applikationsprogramm wird auf dem verwendeten Smartphone installiert. Das Smartphone und der Server befinden sich in unterschiedlichen LANs. Die Tests werden von drei verschiedenen Berechtigungsbenutzern wie Patienten, Ärzte und Administrator durchgeführt. Durch eine umfassende Bedienung kann sich das System mit dem Fernserver verbinden. Bei der Bedienung der Datenbank haben verschiedene Nutzer unterschiedliche Berechtigungen, und der Betrieb jedes Moduls entspricht der ursprünglichen Bedürfnisanalyse.

8.1.1 Untersuchung der Auslastung des IPM Servers

Um die Auslastung des IPM-ZS zu untersuchen, wurde ein besonders hierfür entwickeltes Java Programm verwendet. Es kann die Übernahme der Simulation des Übertragungs-

prozesses eines Clients von einem Desktop-PC aus realisieren. Hierzu wurden alle simulierten Clients durch einen eigenen PC verwirklicht. Auf einer Seite heißt dies, dass vom Kommunikationsserver keine doppelten IPs zwischen den Clients zugelassen sind, und auf der anderen Seite, dass die simulierten Clients die Ressourcen nicht teilen müssen. Jedoch war im Laufe der ersten Testdurchläufe des Untersuchungsaufbaus zu bestimmen, dass die Auswahl der PCs Einfluss auf die Zeitmessung zwischen dem Versand und der Bestätigung eines Datensatzes hat. Deshalb wurde zur Untersuchung ein homogener Rechnerpool (Intel Core 2 Quad 3GHz, 8GB RAM) eingesetzt, damit die Ergebnisse auch in Hinblick auf die verschiedenen Client-Anzahlen vergleichbar werden. Während des verwendeten Versuchsaufbaus liegen der Kommunikationsserver und die Datenbank in einem kollektiven Serversystem (Intel Xeon 2GHz, 8GB RAM). Hierbei ist zu erwarten, dass die Datenbank gerade bei hochfrequenten und großen Abfragen die Auslastung des Serversystems und somit den Archivierungsprozess des Kommunikationsservers stark beeinflussen wird. Somit erfolgt die Durchführung des Untersuchungsablaufs je einmal ohne und einmal mit anderthalbsekündlicher Datenbankabfrage durch die eingegangenen Daten der jeweils letzten 10 Minuten.

Im Laufe der Implementierung des Untersuchungsablaufes erfolgt die Kombination der Zahl der Clients (1, 2, 4, 6, 8, 10) mit unterschiedlichen Datenraten. Die Datenraten waren hierzu in der Simulationssoftware durch ein Sendeintervall (in Millisekunden) einzustellen. Um die Versuche durchzuführen, wurden die Sendeintervalle 1000, 800, 600, 500, 400, 300, 200, 150, 100, 50, 25, 12ms bestimmt. Dies entspricht einer Datenrate von 1 – 83,3 Datensätzen pro Sekunde. Um die Prozessorzeiten zu messen, wurden die *Microsoft Management Console*[156] (Gesamtauslastung der CPU) und der *Process Explorer*[157] eingesetzt. Dieser macht die getrennte Messung der Auslastung des Kommunikationsservers möglich. Der Nachteil bei der Anwendung des *Process Explorers* besteht darin, dass er die Prozessorauslastung gering auflöst; sie wird lediglich in 0,81%-Schritten ausgegeben.

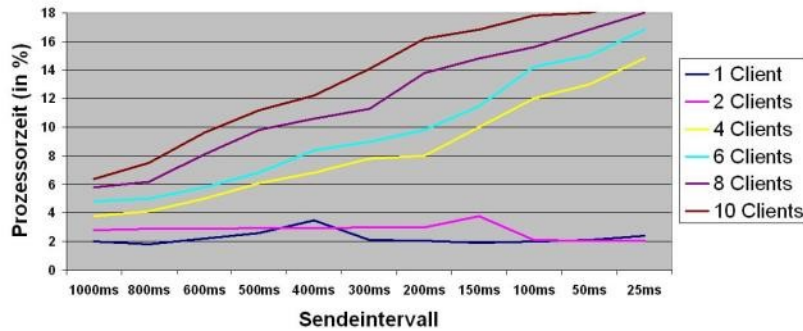


Abbildung 8-2 Prozessorzeiten des gesamten Serversystems bei verschiedenen Belastungen (keine fortlaufende Datenbankabfrage)

8.1.2 Ergebnisse der Untersuchung

Wie zu erwarten war, zeigte sich durch eine höhere Belastung des Kommunikationsservers (höhere Datenraten bzw. Anzahl übertragender Clients) auch eine entsprechend höhere Auslastung der CPU. Dabei verhält sich die Gesamtlast des Serversystems (mit und ohne Datenbankabfrage) mit einer entsprechenden Differenz annähernd proportional zur Last des Kommunikations-Servers (Teil der Gesamtlast). Es zeigte sich besonders bei der Gesamtlast mit zusätzlicher Datenbankabfrage eine starke Steigerung der CPU-Auslastung, was auf die *MySQL*-Datenbankanwendung zurückzuführen ist. Eine entsprechende Optimierung der Datenbankkonfigurationen wurde dabei nicht berücksichtigt. In den folgenden Abbildungen sind die Prozessorzeiten des Serversystems ohne (Abbildung 8-2) und mit fortlaufender Datenbankabfrage (Abbildung 8-3) sowie die des Kommunikationsservers (Abbildung 8-4) bis zur Auslastung dargestellt. Die Teilbelastung des Prozessors durch den Kommunikationsserver zeigte keine Abhängigkeit von den Datenbankabfragen. Außerdem konnte aufgrund der geringen Auflösung bei niedriger Client-Anzahl kaum eine Veränderung in der genutzten Prozessorzeit festgestellt werden.

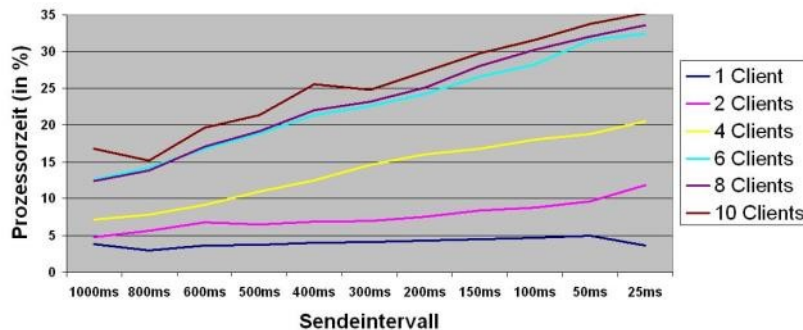


Abbildung 8-3 Prozessorzeiten des gesamten Serversystems bei verschiedenen Belastungen (mit fortlaufender Datenbankabfrage)

Die Teilbelastung des Prozessors über den Kommunikationsserver stellte keine Abhängigkeit von den Abfragen der Datenbank dar. Zudem erfolgt auch die Registration der Wartezeiten zwischen der Bestätigung eines Datensatzes und dem Versand sowie den erfolgten Verarbeitungs- und Übertragungsfehlern. Hierbei wird das Erscheinen von Fehlern als geeigneter Indikator, um die Auslastungsgrenze einzuschätzen, angesehen. Meistens waren die Daten nach einer gewissen Datenmenge nicht richtig in die Datenbank einzutragen und verursachten somit entsprechende Fehlermeldungen bei sämtlichen Clients. Jedoch entstanden fehlende Empfangsbestätigungen, welche normalerweise durch nochmalige Übertragungen ausgeglichen und damit wie übliche Übertragungsfehler ohne Probleme zu behandeln waren. Gerade bei größeren Belastungen herrscht eine große Proportionalität bei der Erscheinung der mangelnden Bestätigungen zwischen den Clients. Das heißt, solch ein Fehler bei einem Client führt normalerweise auch zu einem Fehler bei den anderen Clients. Der Grund für diese Wirkung sind erste Auslastungserscheinungen; dieser wurde erst bei der Überschreitung von 1% als Auslastungsgrenze eingeschätzt. Die Wartezeiten zwischen der Bestätigung des Datensatzes und dem Versand werden als nicht aufschlussreich angesehen. Die Wartezeiten stellten Abhängigkeiten von erscheinenden Fehlern sowie von ungewissen Einflüssen dar. Der Grund hierfür liegt in der unterschiedlichen Netzwerkbelastung. Die Abhängigkeit von den erscheinenden Fehlern ist hierbei mit der notwendigen Fehlerbehandlung über das Simulationsprogramm verbunden. Unter Berücksichtigung der geschilderten Konditionen und Kriterien entstehen die äußersten Belastungsgrenzen aus der Untersuchung für den Kommunikationsserver und die Datenbank auf dem angewendeten Serversystem. Die Verhältnisse zwischen Anzahl und Datenrate der transferierenden Clients für die maximale Belastung sind in Abbildung 8-5, mit und ohne zusätzliche Belastung durch eine ständige Datenbankabfrage, dargestellt.

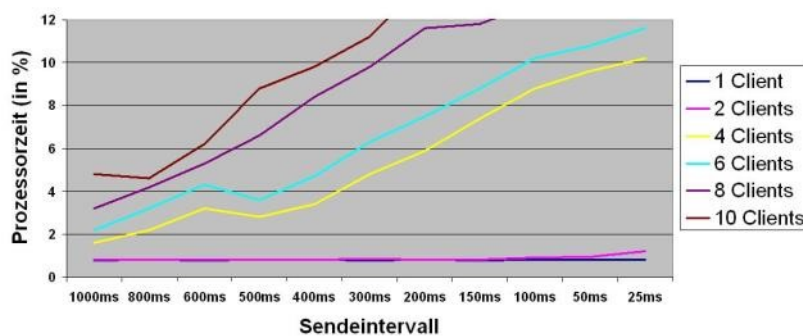


Abbildung 8-4 Prozessorzeiten des Kommunikationsservers

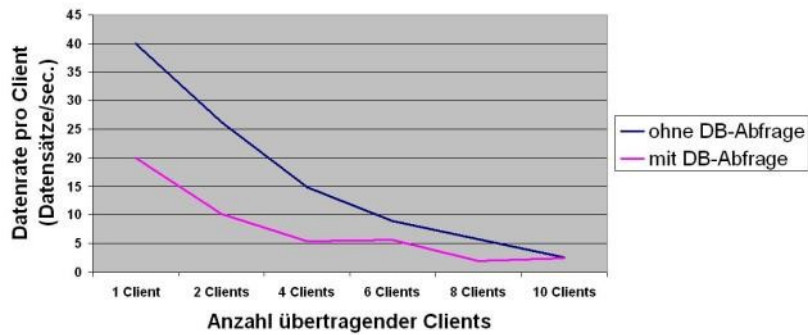


Abbildung 8-5 Verhältnis zwischen Datenrate und Anzahl transferierender Clients für maximale Auslastung des Serversystems

8.1.3 Echtzeit-Datenübertragungstest

Das Hauptziel dieses Tests liegt darin, die Echtzeit-Leistung, die Wellformqualität und die Genauigkeit der physiologischen Parameter zu testen. Der Patient verwendete als Test-User die echtzeitige Überwachungsfunktion. Er trägt den Sensor Equivital Sensor an der Brust und bündelt den Bewegungssensor am Bein. Danach wird das IPM-MC eingeschaltet, um physiologische Signale zu sammeln. Die gesammelten Daten sind in Abbildung 8-6 dargestellt.

Überwachung ID	Client	Interval	Data Entry	Ablesung	Subject
344420	198020014 15 00 50 442	600	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344421	198020014 15 00 50 500	613	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344440	198020014 15 00 50 910	587	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344441	198020014 15 00 50 104	587	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344442	198020014 15 00 50 201	608	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344443	198020014 15 00 51 320	608	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344444	198020014 15 00 52 029	590	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344445	198020014 15 00 52 042	608	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344446	198020014 15 00 53 080	625	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344447	198020014 15 00 53 093	608	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344448	198020014 15 00 54 097	642	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344449	198020014 15 00 54 110	644	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344450	198020014 15 00 55 046	620	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344451	198020014 15 00 55 050	644	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344452	198020014 15 00 57 167	652	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344453	198020014 15 00 57 309	644	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344454	198020014 15 00 58 013	628	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344455	198020014 15 00 58 042	628	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344456	198020014 15 00 58 086	620	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344457	198020014 15 00 58 141	640	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344458	198020014 15 00 58 390	652	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344459	198020014 15 00 59 024	640	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344460	198020014 15 00 59 124	587	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344461	198020014 15 00 59 152	718	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344462	198020014 15 00 59 362	708	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344463	198020014 15 00 59 394	740	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344464	198020014 15 00 59 583	687	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344465	198020014 15 00 59 590	684	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344466	198020014 15 00 59 597	682	138220907483	1 Körperteil: Fühlerzug/Gewicht	102
344467	198020014 15 00 59 599	688	138220907483	1 Körperteil: Fühlerzug/Gewicht	102

Abbildung 8-6 Tabellarische Darstellung der physiologischen Daten

8.1.4 Stabilitätstest

Das Hauptziel dieses Tests ist es, unter der Bedingung des lang andauernden Betriebs die Stabilität des gesamten IPM-mHealth-Systems zu testen. In diesem Test wird dieselbe Serie von Sensorenknoten eingesetzt, unter der Bedingung des fixierten Kommunikationsabstands und der gleichbleibenden Kommunikationsfrequenz wird die Datenverlust-Rate bei der Überwachung von physiologischen Signalen jeweils für 1,4, und 8 Stunden analysiert. In Tabelle 8-1 sind die Versuchsergebnisse dargestellt, die besagen, dass im Laufe der

Verlängerung der Kommunikationszeit die Signale nicht unterbrochen werden, das System nicht abstürzt und die Paketverlustrate keine deutliche Zunahme aufweist.

Tabelle 8-1 Versuchsergebnisse

Dauer der Verbindung (H)	Kommunikationsabstand (M)	Kommunikationsfrequenz (Hz)	Theoretische Daten	Erfassene Daten	Paketfehler r-rate
1	5	100	360,000	358,107	0,68%
4	5	100	1,440,000	1,430,272	0,89%
8	5	100	2,880,000	2,859,680	1,01%

8.1.5 Leistungsaufnahme des Smartphones

Das in diesem Projekt eingesetzt Handy Sony xperia arcs ist mit einem 3,7V Lion Akku ausgestattet. Das gesamte Stromvolumen erreicht 1500mAh. Dessen Wartezeit beträgt in der Regel zwei Tage. Bei dem Test des Stromverbrauchs im Prozess der Datenübertragung werden alle Applikationsprogramme auf dem Handy entleert, um die Genauigkeit der Testergebnisse zu garantieren. Wenn das Stromvolumen 100% des maximalen Stromvolumens erreicht, beträgt die Übertragungshäufigkeit 5mal pro Stunde. Nach einer Stunde Arbeit erreicht das Stromvolumen 99%; nach zwei Stunden Arbeit 98%; nach drei Stunden Arbeit 96%; nach vier Stunden Arbeit 94%; nach 24 Stunden Arbeit beträgt das Stromvolumen 46%. Im Lauf der Datenübertragung sind einige Systemservices notwendig. Daher liegt kein lineares Verhältnis zwischen Stromverbrauch und Zeit vor.

8.2 Anwendung in der Praxis

Am Institut für Präventivmedizin wurden bereits einige medizinische Untersuchungen im Zusammenhang mit dem entwickelten mHealth-System durchgeführt (siehe Tabelle 8-2). Nachfolgend werden hierzu exemplarisch mehrere Untersuchungen detailliert dargestellt. Die Anwendung des Gesamtsystems erfolgt, um 150 Untersuchungen zu registrieren, aufzuzeichnen und zu speichern. Über 80 verschiedene Probanden wurden untersucht. Hierbei wurden unterschiedliche physiologische Messgrößen wie Herzfrequenz, Atemgasparameter und Aktivität der Probanden berücksichtigt. In Bezug auf die meisten Untersuchungen erfolgte die Aufzeichnung der Tätigkeitsprofile durch die Probanden selbst [158], [159].

Tabelle 8-2 Durchgeführt Study List

Study	Zeitraum	Probanden	Dauer	Sensor
Interne Tests	Dzember 2012-März 2013	30	24 Stunden	Equivital / Zephyr / Ant

Study I: „E-Health“	März 2013	10	2 Stunden	Equivital
Study II: “Celisca Lab”	Mai - Juni 2013	10	24 Stunden	Equivital
Study III: Sensorvergleich-Life Science	Juli – Oktober 2013	24	24 Stunden	Zephyr BH3
Study IV Mercury Analysis	August – September 2013	12	8 Stunden	Equivital + Ant Speed Sensor

8.2.1 Untersuchung physiologischer Parameter während Langzeitfelduntersuchungen

Eine Untersuchung wurde am Institut für Präventivmedizin im Rahmen des Forschungsprojekts „E-Health“ durchgeführt [160]. Im Laufe dieser Studie fand die Untersuchung der Belastungen des Studentenalltags sowie der physischen und mentalen Beanspruchungsreaktionen von 10 Probanden über 24 Stunden statt. Zugleich wurde die Tauglichkeit des Datenerfassungs- und Verarbeitungssystems geprüft, um psychophysiologische Daten täglich aufzuzeichnen und auszuwerten. Die Aufzeichnung der Herzschlagintervalle erfolgte zur Analyse der Herzfrequenz. Die Belastungen wurden durch das Aufzeichnen der Tätigkeiten mit dem IPM-MC Client registriert. Im Zuge der Protokollierung wurde die Erhebung des Nasa-TLX [161] und weiterer Fragebögen zur Untersuchungsmethodik erreicht. Um die HRV auszuwerten, wurde die Analyse der Datensätze durch die systemintegrierte Fuzzy Methode von M. Kumar [162] durchgeführt. Die Daten zur statistischen Auswertung wurden durch die systemintegrierte Datenausgabe und -aufbereitung zusammengestellt. Die Messwerte verschiedener Messgrößen wurden durch die Bestimmung quasistationärer Bereiche (hier 5 min) zeitlich synchronisiert. Die Ermittlung und der Vergleich physiologischer Daten waren durch unterschiedliche Tätigkeiten erreichbar. Der Vergleich von Schlaf- und Wachphasen führte zu einer leicht verkleinerten Herzschlagfrequenz. Die Parameter der Herzfrequenzvariabilität wiesen eine große Erhöhung im hohen Frequenzbereich um zwei Drittel beim Schlafen auf. Das geänderte Verhältnis LF/HF stellte die zunehmende parasympathische Herrschaft im Schlaf im Vergleich zum ruhigen Wachzustand dar[163].

Durch die Bewertung der Kurzfragebögen für die aktuelle Beanspruchung wurde die subjektiv empfundene Beanspruchung in Hinblick auf die verschiedenen Tätigkeiten berücksichtigt. Obwohl das Probandenkollektiv relativ klein war, waren signifikante Differenzen zwischen einzelnen Tätigkeiten auszumachen und eine Rangfolge mit der subjektiven Beanspruchung konnte erstellt werden. Die Herzschlagfrequenz und die

Parameter der HRV stellen ebenfalls große Differenzen während unterschiedlicher Tätigkeitskategorien dar. Die Reproduzierbarkeit der subjektiven Beanspruchung und der physiologischen Messgrößen war mit einem Vergleich der zwei Untersuchungstage pro Proband darzustellen.

Tabelle 8-3 Mittelwerte der normalisierten HRV-Parameter (Liegen = 1; Schlafen = 2)

Kategorie	N	Mittelwert	Standardabweichung	Standardfehler des Mittelwertes
HF% 1	139	,3340	,19014	,01492
2	1382	,5126	,44298	,01159
LF% 1	139	1,0661	,28726	,02273
2	1382	,9438	,36114	,00951
VLF% 1	139	,9233	,38892	,03082
2	1382	,7358	,43029	,01129
TP% 1	139	9418,1096	7074,30345	561,02898
2	1382	8963,7349	8091,30038	212,78159
HR_HRV% 1	139	133,9034	18,64351	1,47791
2	1382	120,5042	15,35289	,40379
LF/HF 1	139	4,7510	3,44489	,27319
2	1382	3,7308	3,66393	,09629

Die Untersuchung der Methodik des eingesetzten Systems ist eines der Hauptziele der Untersuchungsreihe. Die Untersuchungsperiode wurde als repräsentativ angesehen; 78% der Probanden war dieser Meinung (siehe Abbildung 8-7). Die Messung der Herzschlagfrequenz durch einen Brustgurt wurde in der Nachtphase als negativ betrachtet, vereinzelt entstanden wegen der Messmethode Messfehler oder Unterbrechungen der Aufzeichnung. Die Angaben in Bezug auf die Frequenz der Erhebung der Fragebögen weisen keinen deutlichen Trend auf. Die Selbstprotokollierung verursachte eine zunehmende Selbstbeobachtung bei subjektiv nicht beeinflusstem Verhalten.

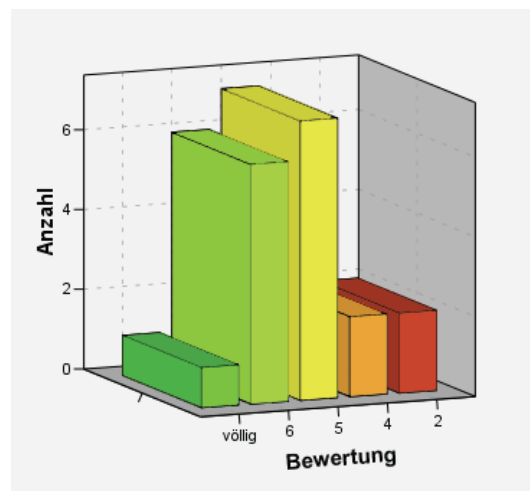


Abbildung 8-7 Häufigkeitsverteilung der Antworten auf die Frage: Waren die vergangenen 24 Stunden repräsentativ für Sie?

Zusammenfassend gelten das Untersuchungsinventar und das angewendete System als geeignet für Langzeituntersuchungen unter Feldbedingungen. Die Probandenrückmeldungen weisen eine gute Verwendbarkeit und niedrige Kreativität auf. Besonders die integrierte Datenanalyse und die Datenaufbereitung galten als positiv.

8.2.2 Beanspruchung von Chemikern, technischen Assistenten und Biologen des Center for Life Science Automation (Celisca)

Anschließend wurde die zweite Untersuchung durchgeführt. Dabei wurde die Beanspruchung von technischen Assistenten, Chemikern und Biologen des Zentrum for LifeScience Automation (celisca) untersucht [164]. Hierfür wurde das mHealth-System von den Probanden eine Arbeitswoche lang getragen, und zwartäglich 8 bis 9 Stunden, um die physiologischen Parameter und das subjektive Beanspruchungsempfinden aufzuzeichnen. Hierbei arbeiteten die Probanden an einem bestimmten Standort; teilweise waren sie gebäudeübergreifend tätig. Der Fokus dieser Testphase bestand darin, die Stabilität des Systems bei der nicht ortsabhängigen Beaufsichtigung der Probandendaten zu prüfen. Die hierbei entstehende Fehlerquote bei dem Transfer war kleiner als 0,08%. Sie war durch einen erneuten Transfer des Datensatzes zu berichtigen. In Abbildung 8-8 ist die Online-Visualisierung mit den Beispieldaten eines Probanden durch den Arbeitstag mit den richtigen Belastungskategorien dargestellt.

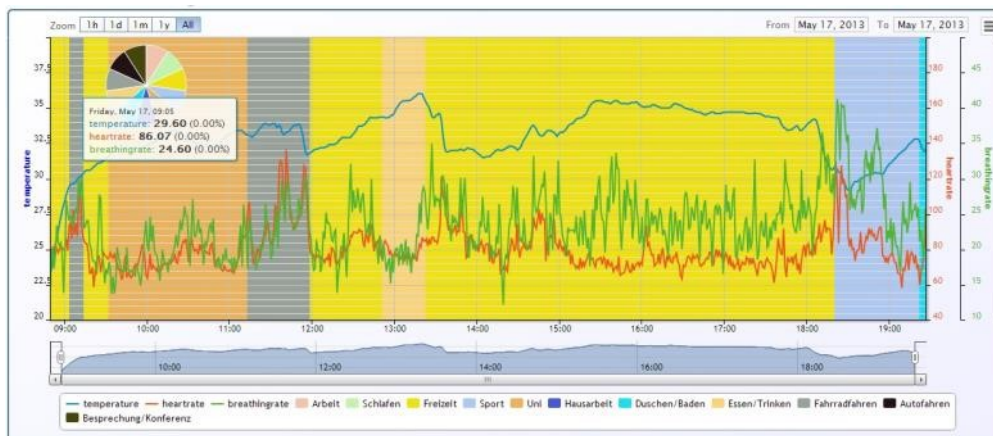


Abbildung 8-8 Online-Visualisierung mit den Beispieldaten eines Probanden im Verlauf eines Arbeitstages mit den richtigen Belastungskategorien

Bei diesen Untersuchungen beeinflusst die physische Komponente den Probanden zusätzlich zu der psychomentalen und der emotionalen Komponente. In Abbildung 8-9 wird zum Beispiel die Beziehung der körperlichen Belastung zur Atemfrequenz gezeigt. Wegen der unterschiedlichen Tätigkeiten und der verwendeten Muskelarbeit zeigt die Atemfrequenz große Differenzen zwischen den unterschiedlichen Tätigkeiten.

Im Laufe der Untersuchungen waren unterschiedliche Effekte zum Handling des mobilen Rechnersystems zu ermitteln. Sie ließen sich durch entsprechende Maßnahmen entfernen oder in ihrem Ausmaß begrenzen. Diesbezüglich wurden zum Beispiel nicht gebrauchte Buttons nach Möglichkeit nicht aktiviert, um eine unabsichtliche Betätigung und die hiermit verknüpfte Aktivierung anderer Anwendungen bzw. die nicht gewollte Bestätigung von Eingaben zu vermeiden. Zudem war eine mangelnde Alarmierung des Probanden bei unabsichtlichem Liegenlassen des Rechnersystems zu ermitteln. Jedoch ließ sich als entgegenwirkende Maßnahme an der Stelle lediglich eine Alarmierung mittels des vergessenen mobilen Rechnersystems verwirklichen. Dies genügt jedoch unter gewissen Bedingungen nicht. Die geeignetere Alarmierung über das Sensorsystem kann durch die komplizierte Detektion eines nicht gewollten Abbruchs mit hohen Kosten durch den Produzenten des Sensormoduls erreicht werden.

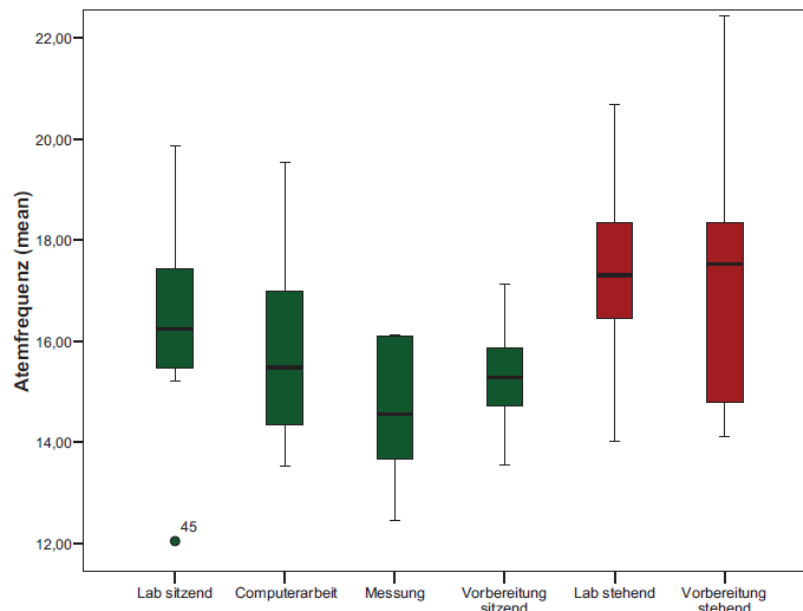


Abbildung 8-9 Einfluss unterschiedlicher Tätigkeiten (sitzend und stehend) auf die Atemfrequenz

Die Einschränkungen in Hinblick auf die kleinen Akkulaufzeiten der mobilen Rechnersysteme waren durch zwei Maßnahmen auszugleichen. Auf der einen Seite wurde die Übertragungskomponente der IPM-MC als Dienst gestartet und im Hintergrund des Systems laufen gelassen. Die IPM-MC liegt seit langer Zeit im Ruhestand, sofern der Proband keine Interaktion mit IPM-MC benötigt. Auf der anderen Seite sind so genannte Notladegeräte einzusetzen (verwendete Kapazitäten: 2680mAh, 1700mAh). Sie machen eine Verdopplung der Kapazität des Original-Akkus möglich.

9 Zusammenfassung und Ausblick

9.1 Zusammenfassung

Das Ziel dieser Arbeit war die Entwicklung eines mobilen Online-Erfassungssystems, das neue Möglichkeiten für Untersuchungen nach dem Belastungs-Aktivitäts-Konzept bietet. Dabei stand die Echtzeitüberwachung von physiologischen Parametern, Bewegungen und diversen Eingabedaten unabhängig vom Standort eines Probanden und des Untersuchers im Mittelpunkt.

Zur Umsetzung erfolgte die Durchführung einer Recherche gemäß geeigneter Systemkomponenten, welche die notwendigen Eigenschaften in Hinblick auf die Datenerfassung und -übertragung zur Verfügung stellen. Hierbei konnte das mHealth System durch die folgenden Komponenten festgesetzt werden: Sensorsystem, mobiles Rechnersystem, Serversystem, Auswertungssystem.

Die Basiskomponente IPM-MC der von einem Probanden mitgeführten Ausrüstung ließ sich durch mobile Rechnersysteme wie zum Beispiel Smartphones verwirklichen. Über sie sind das Empfangen von Sensormessdaten, das Eingeben von Beanspruchungs- und Belastungsdaten sowie das Weiterleiten der erfassten Daten an das Serversystem zu implementieren. Hierzu fand auch die Entwicklung der Steuersoftware statt, welche den notwendigen Einfluss des Probanden auf den Bedienungsprozess auf ein Minimum sinken ließ. Über ein speziell für die Anwendung entwickeltes Protokoll können die Mess- und Dialogdaten sowie die für die Steuerprozesse der Serversoftware benötigten Informationen übertragen werden. Das Protokoll stellt unterschiedliche Mechanismen zur Verfügung, um unsichere Netzzustände zu kompensieren, welche für einen völligen Transfer der Daten nötig sind. Die Anwendung von Mobilfunkstandards macht hierbei den standortunabhängigen Transfer vom Client (wie z.B. mobiles Rechnersystem und Sensorsystem) auf das Serversystem möglich.

Das Serversystem IPM-ZS gilt als das zentrale Element des Gesamtsystems und lässt das parallele oder quasiparallele Empfangen von Messdaten einiger mobiler Clients zu. Das automatisierungstechnisch konzipierte und realisierte Serversystem führt zu einer erheblichen Verbesserung in der Versuchsdurchführung und -auswertung bezüglich der arbeitsphysiologischen Forschung und Diagnose. Es verfügt über vereinheitlichte Strukturen für die Datenerfassung, die Verwaltung und Methoden, um die Auswertung und Analyse zu

erreichen. Die modulare Systemstruktur trennt Datenerfassung, -speicherung und -Auswertung voneinander und ermöglicht so eine gezielte Erweiterung des Systems um neue Funktionalitäten. Die arbeitsmedizinischen Experten legen den Fokus auf Versuchsplanung und -auswertung, und das medizinisch-technische Personal ist vorrangig verantwortlich für die Datenakquisition.

In die Auswertungsmodule wurde in dieser Arbeit beispielhaft eine automatisierte Aktivitätsanalysemethode entwickelt. Ziel dieser Komponente war die Auswertung von telemedizinisch erhobenen Vital-Daten in Verbindung mit einer Bewegungsklassifizierung. Diese bietet qualitativ neue Möglichkeiten für Langzeituntersuchungen verschiedener Vitalfunktionen. Aktivitätsprofile können einerseits eigenständig als Grundlage diagnostischer Aussagen dienen, andererseits in Kombination mit anderen Messgrößen deren Aussagekraft maßgeblich erweitern. Die Analyse erlaubt eine Online-Klassifikation verschiedener Bewegungsformen. Typischerweise werden alltägliche Aktivitäten wie Gehen, Laufen, Treppensteigen und Liegen klassifiziert. Die Struktur der eingesetzten Klassifikationsverfahren ermöglicht allerdings problemlos die Erweiterung der Auswertung bezüglich weiterer Bewegungsmuster. Anhand der Aktivitätsanalyse kann eine Abschätzung des Energieverbrauchs aufgrund der Bewegung des Patienten durchgeführt werden. Grundlage dafür bildet die Zuordnung von energieäquivalenten zu verschiedenen körperlichen Aktivitäten. Bei diesen Berechnungen wird selbstverständlich das Körpergewicht des Patienten berücksichtigt.

9.2 Ausblick

Im zweiten Kapitel werden zunächst Definition und Struktur des medizinischen drahtlosen Sensornetzwerks dargelegt. Danach wird das neue Entwicklungsframework SOTH für Telemonitoring Applikationen vorgestellt sowie der Design- und Validierungsprozess des SOTH-Framework detailliert erläutert. Auf Basis des „*Layer-Oriented-Designs*“ der SOTH-Struktur werden eine Reihe von Modellierungsmethoden zur Beschreibung des „Remote Healthcare-Dienstes“ vorgebracht. Der Schwerpunkt liegt darauf, wie die Struktur eingesetzt werden kann, um die Integration verschiedener medizinischer Komponenten in ein einheitliches mHealth-System zu vereinfachen.

Kurz gesagt, bietet die SOTH-Lösung einen integrierten Telemedizin-Service, der den Prozess von der Datensammlung bis hin zur Informationslieferung als ein „Computing Utility“ abbildet. Diese Implementierung beinhaltet mehrere praktische Vorteile, wie z.

B., „always-on“ und eine Echtzeit-Datenerfassung. Dies beseitigt die manuelle Erfassungsarbeit sowie Tippfehler erleichtert den Einführungsprozess, weil es bei einer drahtlosen Vernetzung keine Notwendigkeit für eine Verkabelung oder andere physikalische Setup gibt.

Aus Sicht der Softwareentwicklung fördert das vorgeschlagene Design eine Wiederverwendbarkeit durch die Verwendung von Standard-Dienstleistungen, die mit Hilfe einer Plattform as a Service (PaaS) [165] umgesetzt und eingesetzt werden. Außerdem nutzt es anderen Gesundheitseinrichtungen, um Dienste über eine Software als ein Service (SaaS) [166], ohne Investitionen für Hardware -oder Softwarelizenzen, in Anspruch zu nehmen.

Die Entwicklung solcher Systeme hat auch den Vorteil, dass verschiedene Programmiersprachen und verschiedene Arten von Geräten verwendet werden können und einfach miteinander verbunden sind. Es bietet eine hohe Flexibilität in Bezug auf Programmiersprachen und Geräte, jedoch mit dem Nachteil der Heterogenität, die häufig zu mehr Komplexität führt.

Das IPM-mHealth System wird in aktuellen arbeitsmedizinischen Untersuchungen für eine psychomentele Beanspruchung bereits erfolgreich eingesetzt. Die Probanden waren hierzu mit dem Erfassungssystem ausgestattet. Dies macht es dem Untersuchungsführer möglich, standortunabhängig den Untersuchungsverlauf zu überwachen.

In Zukunft werden folgende Forschungspunkte zu berücksichtigen sein:

1. Tragbares Design mit geringem Energieverbrauch: Das Sensor System sollte miniaturisiert werden, um ein lückenloses Design zu realisieren; die Probleme wie Instabilität der Signale, geringe Bequemlichkeit und kurze Lebensspanne sollten gelöst werden.
2. Weitere Optimierung des Systems: Das ausgereifte Bluetooth 4.0 [167] Chip sollte die Bluetooth 2.1 - Technologie ersetzen, der Server der Datenbank und die Funktion zur Daten-Live-Visualisierung sollten verteilt werden.
3. An der Clientseite sollte die Funktion zur Healthcare-Überwachung mittels der ins Handy eingebetteten Multi-Sensoren erweitert werden, wie z.B. GPS-System, Umgebungswahrnehmung usw.
4. Ein weiterer Fokus liegt auf Cloud-Computing; beim Server sollte Cloud-Computing eingesetzt werden, um eine große Menge von gesundheitsbezogenen Daten zu bearbeiten.

Die Cloud-Computing-Umgebung selbst bietet eine skalierbare Infrastruktur, die es sehr attraktiv macht [168]. Der Hauptengpass in dieser Architektur ist die Fernkommunikation zwischen den drahtlosen Knotenpunkten der Sensornetzwerkkommunikation und der Cloud-Computing-Umgebung. Die lokale Berechnung innerhalb der Cloud-Computing-Umgebung ist in der Regel schnell, jedoch ist die Fernkommunikation zwischen dem Knotenpunkt und der Cloud-Computing-Umgebung ist meist noch viel langsamer, was im Vorfeld berücksichtigt und in Zukunft weiter untersucht werden sollte (siehe [169], [170]).

Literaturverzeichnis

1. Graf, J., Martin, H., Hörsöki, M., Schellong, S., Niederlag, W., Heinig, A., Holland H.J., Kaiser, J. SmartVital - Telemetrische EKG-Auswertung Und Bewegungsklassifikation e-Health (2012) , pp. 261-267.
2. Clark, R.A., Inglis, S.C., McAlister, F.A., Cleland, J.G.F., Stewart, S. Telemonitoring or structured telephone support programmes for patients with chronic heart failure: Systematic review and meta-analysis (2007) British Medical Journal, 334 (7600), pp. 942-945.
3. Pantelopoulos, A., Bourbakis, N.G. A survey on wearable sensor-based systems for health monitoring and prognosis (2010) IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 40 (1), art. no. 5306098, pp. 1-12.
4. Wurm, E.M.T., Hofmann-Wellenhof, R., Wurm, R., Soyer, H.P. Telemedicine and teledermatology: Past, present and future (2008) JDDG - Journal of the German Society of Dermatology, 6 (2), pp. 106-112.
5. Estrin, D., Sim, I. Open mHealth architecture: An engine for health care innovation (2010) Science, 330 (6005), pp. 759-760.
6. Jovanov, E., Milenkovic, A., Otto, C., De Groen, P.C. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation (2005) Journal of NeuroEngineering and Rehabilitation, pp. 2-3.
7. Zandi, F. A bi-level interactive decision support framework to identify data mining-oriented electronic health record architectures (2014) Applied Soft Computing Journal, 18, pp. 136-145.
8. Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E. Wireless sensor networks: A survey (2002) Computer Networks, 38 (4), pp. 393-422.
9. Weinstein, R.S., Lopez, A.M., Joseph, B.A., Erps, K.A., Holcomb, M., Barker, G.P., Krupinski, E.A. Telemedicine, telehealth, and mobile health applications that work: Opportunities and barriers (2014) American Journal of Medicine, 127 (3), pp. 183-187.
10. Clark, R.A., Inglis, S.C., McAlister, F.A., Cleland, J.G.F., Stewart, S. Telemonitoring or structured telephone support programmes for patients with chronic heart failure: Systematic review and meta-analysis (2007) British Medical Journal, 334 (7600), pp. 942-945.
11. Asua, J., Orruño, E., Reviriego, E., Gagnon, M.P. Healthcare professional acceptance of telemonitoring for chronic care patients in primary care (2012) BMC Medical Informatics and Decision Making, 12 (1), art. no. 139, .
12. Martin, S. Telemedicine in Germany [Telemedizin in Deutschland] (2012) Diabetologe, 8 (4), p. 274.

13. Liu, L., Wang, H., Liu, X., Jin, X., He, W., Wang, Q., Chen, Y. Greencloud: A new architecture for green data Zentrum (2009) Proceedings of the 6th International Conference Industry Session on Autonomic Computing and Communications Industry Session, ICAC-INDST'09, pp. 29-38.
14. Jha, A.K., Desroches, C.M., Campbell, E.G., Donelan, K., Rao, S.R., Ferris, T.G., Shields, A., Rosenbaum, S., Blumenthal, D. Use of electronic health records in U.S. Hospitals (2009) New England Journal of Medicine, 360 (16), pp. 1628-1638.
15. Henderson, T.R., Katz, R.H. Transport Protocols for Internet-Compatible Satellite Networks (1999) IEEE Journal on Selected Areas in Communications, 17 (2), pp. 326-343.
16. Otsuka, Y., Ogawa, T., Saito, A., Tsugawa, T., Fukao, S., Miyazaki, S. A new technique for mapping of total electron content using GPS network in Japan (2002) Earth, Planets and Space, 54 (1), pp. 63-70.
17. Croner, C.M., Sperling, J., Broome, F.R. Geographic information systems (GIS): New perspectives in understanding human health and environmental relationships (1996) Statistics in Medicine, 15 (17-18), pp. 1961-1977.
18. Medjahed, H., Istrate, D., Boudy, J., Baldinger, J.-L., Dorizzi, B. A pervasive multi-sensor data fusion for smart home healthcare monitoring (2011) IEEE International Conference on Fuzzy Systems, art. no. 6007636, pp. 1466-1473.
19. De Pauw, B., Walsh, T.J., Donnelly, J.P., Stevens, D.A. Revised definitions of invasive fungal disease from the European Organization for Research and Treatment of Cancer/Invasive Fungal Infections Cooperative Group and the National Institute of Allergy and Infectious Diseases Mycoses Study Group (EORTC/MSG) Consensus Group (2008) Clinical Infectious Diseases, 46 (12), pp. 1813-1821.
20. Wolf, B. Cellular radio-based medical knowledge base with sensory interactive mobile telephones (2005) Biomedizinische Technik. Biomedical engineering, 50 (5), pp. 156-158.
21. Van Dam, T., Langendoen, K. An adaptive energy-efficient MAC protocol for wireless sensor networks (2003) SenSys'03, Proceedings of the First International Conference on Embedded Networked Sensor Systems, pp. 171-180.
22. Baronti, P., Pillai, P., Chook, V.W.C., Chessa, S., Gotta, A., Hu, Y.F. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards (2007) Computer Communications, 30 (7), pp. 1655-1695.
23. Yang, L., Giannakis, G.B. Ultra-wideband communications (2004) IEEE Signal Processing Magazine, 21 (6), pp. 26-54.
24. Zhang, W., Passow, P., Jovanov, E., Stoll, R., Thurow, K. A secure and scalable telemonitoring system using ultra-low-energy wireless sensor interface for long-term monitoring in life science applications (2013) IEEE International Conference on Automation Science and Engineering, art. no. 6653979, pp. 617-622.
25. Varshney, U. Pervasive healthcare and wireless health monitoring (2007) Mobile Networks and Applications, 12 (2-3), pp. 113-127.

26. Krishna, S., Boren, S.A., Balas, E.A. Healthcare via cell phones: A systematic review (2009) *Telemedicine and e-Health*, 15 (3), pp. 231-240.
27. Eysenbach, G. CONSORT-EHEALTH Group CONSORT-EHEALTH: improving and standardizing evaluation reports of Web-based and mobile health interventions. (2011) *Journal of medical Internet research*, 13 (4), pp. e126.
28. Gerd S. Grundlagen der Vermittlungstechnik. R. v. Decker; Heidelberg; 1992
29. International Telecommunication Union <http://www.itu.int/en/Pages/default.aspx>
30. Telemedicine Alliance Project Telemedicine 2010: Visions for a personal medical network (2004) European Space Agency (Brochure) ESA BR, (229), 39 p.
31. Europa Space Agency <http://www.esa.int/ESA>
32. Telemedizin Standardization Coordination Group <http://www.who.int/ehscg/en/>
33. Sapsford, S.M., Faure, M.A., Begg, S.M. Boundary-Mesh-Motion: Eine neue Vernetzungstechnik zur Einbindung der Strömungsdynamiksimulation flüssiger Körper (1999) *MTZ Motortechnische Zeitschrift*, 60 (2), pp. 80-93.
34. Clark, R.A., Inglis, S.C., McAlister, F.A., Cleland, J.G.F., Stewart, S. Telemonitoring or structured telephone support programmes for patients with chronic heart failure: Systematic review and meta-analysis (2007) *British Medical Journal*, 334 (7600), pp. 942-945.
35. Dolin, R.H., Alschuler, L., Beebe, C., Biron, P.V., Boyer, S.L., Essin, D., Kimber, E., Lincoln, T., Mattison, J.E. The HL7 clinical document architecture (2001) *Journal of the American Medical Informatics Association*, 8 (6), pp. 552-569.
36. Mildenerger, P., Eichelberg, M., Martin, E. Introduction to the DICOM standard (2002) *European Radiology*, 12 (4), pp. 920-927
37. Häyrynen, K., Saranto, K., Nykänen, P. Definition, structure, content, use and impacts of electronic health records: A review of the research literature (2008) *International Journal of Medical Informatics*, 77 (5), pp. 291-304.
38. Dodge, C., Schneider, R., Sander, C. The HSSP database of protein structure-sequence alignments and family profiles (1998) *Nucleic Acids Research*, 26 (1), pp. 313-315.
39. Milenković, A., Otto, C., Jovanov, E. Wireless sensor networks for personal health monitoring: Issues and an implementation(2006) *Computer Communications*, 29 (13-14), pp. 2521-2533.
40. Noel, H.C., Vogel, D.C., Erdos, J.J., Cornwall, D., Levin, F. Home telehealth reduces healthcare costs. (2004) *Telemedicine journal and e-health : the official journal of the American Telemedicine Association*, 10 (2), pp. 170-183.

41. Otal, B., Alonso, L., Verikoukis, C. Highly reliable energy-saving mac for wireless body sensor networks in healthcare systems (2009) *IEEE Journal on Selected Areas in Communications*, 27 (4), art. no. 4909290, pp. 553-565.
42. Akyildiz, I.F., Wang, X., Wang, W. Wireless mesh networks: A survey (2005) *Computer Networks*, 47 (4), pp. 445-487.
43. Orgun, B., Vu, J. HL7 ontology and mobile agents for interoperability in heterogeneous medical information systems (2006) *Computers in Biology and Medicine*, 36 (7-8), pp. 817-836.
44. Celli, B.R., MacNee, W., Agusti, A., Anzueto, A., Berg, B., Buist, A.S., Calverley, P.M.A., Chavannes, N., Dillard, T., Fahy, B., Fein, A., Heffner, J., Lareau, S., Meek, P., Martinez, F., McNicholas, W., Muris, J., Austegard, E., Pauwels, R., Rennard, S., Rossi, A., Siafakas, N., Tiep, B., Vestbo, J., Wouters, E., ZuWallack, R. Standards for the diagnosis and treatment of patients with COPD: A summary of the ATS/ERS position paper (2004) *European Respiratory Journal*, 23 (6), pp. 932-946.
45. Martínez, I., Escayola, J., De Bobadilla, I.F., Martínez-Espronedada, M., Serrano, L., Trigo, J., Led, S., García, J. Optimization proposal of a standard-based patient monitoring platform for ubiquitous environments (2008) *Proceedings of the 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 1813-1816.
46. Atkins, D., Cullen, T. The future of health information technology: Implications for research (2013) *Medical Care*, 51 (3 SUPPL. 1), pp. S1-S3.
47. Testa, M.A., Simonson, D.C. Assessment of quality-of-life outcomes (1996) *New England Journal of Medicine*, 334 (13), pp. 835-840.
48. Ye, W., Heidemann, J., Estrin, D. An energy-efficient MAC protocol for wireless sensor networks (2002) *Proceedings - IEEE INFOCOM*, 3, pp. 1567-1576.
49. Lorussi, F., Rocchia, W., Scilingo, E.P., Tognetti, A., De Rossi, D. Wearable, redundant fabric-based sensor arrays for reconstruction of body segment posture(2004) *IEEE Sensors Journal*, 4 (6), pp. 807-818.
50. Baronti, P., Pillai, P., Chook, V.W.C., Chessa, S., Gotta, A., Hu, Y.F. Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards (2007) *Computer Communications*, 30 (7), pp. 1655-1695.
51. Gungor, V.C., Hancke, G.P. Industrial wireless sensor networks: Challenges, design principles, and technical approaches (2009) *IEEE Transactions on Industrial Electronics*, 56 (10), pp. 4258-4265.
52. Ekici, E., Gu, Y., Bozdog, D. Mobility-based communication in wireless sensor networks (2006) *IEEE Communications Magazine*, 44 (7), pp. 56-62.
53. Sikora, A., Groza, V.F. Coexistence of IEEE802.15.4 with other systems in the 2.4 GHz-ISM-band (2005) *Conference Record - IEEE Instrumentation and Measurement Technology Conference*, 3, art. no. 1604479, pp. 1786-1791.

54. Cook, B.W., Molnar, A., Pister, K.S.J. Low power RF design for sensor networks (2005) Digest of Papers - IEEE Radio Frequency Integrated Circuits Symposium, art. no. RTU1A-2, pp. 357-360.
55. William Atwood, J., Jiang, X., Das, K.C. Establishing Multimedia Sessions between IP version 6 Hosts and IP Version 4 Hosts (2003) Proceedings of the IASTED International Conference on Wireless and Optical Communications, 3, pp. 538-542.
56. Satyanarayanan, M. Pervasive computing: Vision and challenges (2001) IEEE Personal Communications, 8 (4), pp. 10-17.
57. ANT Alliance. What is ANT <http://www.thisisant.com/consumer/ant-101/what-is-ant/>, (2013)
58. Wyld, D.C. RuBee: Applying low-frequency technology for retail and medical uses (2008) Management Research News, 31 (7), pp. 549-554.
59. Stoll, R., Thurow, K. Center for Life Science Automation / Universität Rostock: Life Sciences Automation & Ergonomics – Labor – (2014)
60. Lorincz, K., Malan, D.J., Fulford-Jones, T.R.F., Nawoj, A., Clavel, A., Shnayder, V., Mainland, G., Welsh, M., Moulton, S. Sensor networks for emergency response: Challenges and opportunities (2004) IEEE Pervasive Computing, 3 (4), pp. 16-23.
61. Jovanov, E., Raskovic, D., Price, J., Chapman, J., Moore, A., Krishnamurthy, A. Patient monitoring using personal area networks of wireless intelligent sensors (2001) Biomedical Sciences Instrumentation, 37, pp. 373-378.
62. Henderson, D.A., Inglesby, T.V., Bartlett, J.G., Ascher, M.S., Eitzen, E., Jahrling, P.B., Hauer, J., Layton, M., McDade, J., Osterholm, M.T., O'Toole, T., Parker, G., Perl, T., Russell, P.K., Tonat, K. Smallpox as a biological weapon: Medical and public health management (1999) Journal of the American Medical Association, 281 (22), pp. 2127-2137.
63. Zito, D., Pepe, D., Mincica, M., Zito, F., Tognetti, A., Lanatà, A., De Rossi, D. SoC CMOS UWB pulse radar sensor for contactless respiratory rate monitoring (2011) IEEE Transactions on Biomedical Circuits and Systems, 5 (6), art. no. 6104396, pp. 503-510.
64. Rahal, Y., Pigot, H., Mabillean, P. Location estimation in a smart home: system implementation and evaluation using experimental data (2008) International journal of telemedicine and applications, pp. 21-30.
65. Winkler, S., Schieber, M., Lücke, S., Heinze, P., Schweizer, T., Wegerteder, D., Scherf, M., Nettelau, H., Henke, S., Braecklein, M., Anker, S.D., Koehler, F. A new telemonitoring system intended for chronic heart failure patients using mobile telephone technology - Feasibility study (2011) International Journal of Cardiology, 153 (1), pp. 55-58.
66. Dahlman, E., Beming, P., Knutsson, J., Ovesjö, F., Persson, M., Roobol, C. WCDMA-The radio interface for future mobile multimedia communications (1998) IEEE Transactions on Vehicular Technology, 47 (4), pp. 1105-1118.

67. Dolin, R.H., Alschuler, L., Boyer, S., Beebe, C., Behlen, F.M., Biron, P.V., Shabo, A. HL7 clinical document architecture, release 2 (2006) Journal of the American Medical Informatics Association, 13 (1), pp. 30-39.
68. Potlapally, N.R., Ravi, S., Raghunathan, A., Jha, N.K. Analyzing the Energy Consumption of Security Protocols (2003) Proceedings of the International Symposium on Low Power Electronics and Design, pp. 30-35.
69. Jha, A.K., Desroches, C.M., Campbell, E.G., Donelan, K., Rao, S.R., Ferris, T.G., Shields, A., Rosenbaum, S., Blumenthal, D. Use of electronic health records in U.S. Hospitals (2009) New England Journal of Medicine, 360 (16), pp. 1628-1638.
70. Neubert, S., Behrendt, S., Rieger, A., Weippert, M., Kumar, M., Stoll, R. Echtzeit-Telemonitoring-System in der Präventivmedizin, In: Duesberg, F. (Hrsg.), E-Health 2012 - Informationstechnologien und Telematik im Gesundheitswesen. Medical Future Verlag, Solingen 2011, pp. 249-252
71. [Http://developer.android.com/](http://developer.android.com/) (2014)
72. Park, G.Y., Yun, J.H., Choi, Y.L., Kim, J.-W. Development of android application for wireless control of omnidirectional biped walking of humanoid robot (2014) Journal of Institute of Control, Robotics and Systems, 20 (2), pp. 223-231.
73. Milenković, A., Otto, C., Jovanov, E. Wireless sensor networks for personal health monitoring: Issues and an implementation (2006) Computer Communications, 29 (13-14), pp. 2521-2533.
74. Passow, P., Stoll, N., Junginger, S., Thurow, K. A Wireless Sensor Node for Longterm Monitoring in Life Science Applications. Proceedings, IEEE Instrumentation and Measurement Technology Conference, Minneapolis (USA), 06.-09.05.2013, 2013, art. no. 6555545, pp. 898-901.
75. Hidalgo Limited, Equivital™ EQ-01 Hidalgo Ltd., (2006).
76. Wireless Bluetooth Heart Rate Sensor, Zephyr Technology Corp., (2007)
77. BioHarness 3: Wireless Professional Heart Rate Monitor Zephyr Technology Corp., (2012).
78. Mio alphaheart rate sport watch, MIO Global (Physical Enterprises Inc.), (2014).
79. Herzfrequenz-sensoren H7 bluetooth smart (4.0) , Polar Electro.,(2014)
80. Wahoo Fitness., <Http://eu.wahoofitness.com/wahoo-cycling-speed-cadence-sensor.html>, (2013).
81. Wahoo Fitness ., <Http://eu.wahoofitness.com/devices/hr.html>(2013).
82. Garmin-Tempe, Tempe_Instructions_WEB_ML14, Garmin Ltd (2013).

83. Xu, H., Zhao, G.-J. Wireless monitoring system for elevator on Android platform (2014) Lecture Notes in Electrical Engineering, 237 LNEE, pp. 373-381.
84. HC-06 Bluetooth module, Wireless Bluetooth RF Transceiver.
85. Dynastream Innovations Inc., (2013)
86. Talasila, M., Curtmola, R., Borcea, C. Collaborative Bluetooth based location authentication on smart phones (2014) Pervasive and Mobile Computing, . Article in Press.
87. Haartsen, J.C. Bluetooth radio system (2000) IEEE Personal Communications, 7 (1), pp. 28-36.
88. Oliveira, L.M.L., Rodrigues, J.J.P.C., Elias, A.G.F., Zarpelão, B.B. Ubiquitous monitoring solution for Wireless Sensor Networks with push notifications and end-to-end connectivity (2014) Mobile Information Systems, 10 (1), pp. 19-35.
89. ANT Message Protocol and Usage. Dynastream Innovations Inc., Rev 4.1 edition, July 2007.
90. ANT protocol brief technical description. Technical description, Dynastream Innovations Inc., June 2007.
91. Liu, X.P., Zhou, Y.B. Design of an intelligent home system based on the android system (2014) Applied Mechanics and Materials, 443, pp. 576-579.
92. Sung, W.-T., Chen, J.-H., Chang, K.-W. Mobile physiological measurement platform with cloud and analysis functions implemented via IPSO (2014) IEEE Sensors Journal, 14 (1), art. no. 6588926, pp. 111-123.
93. Su, C.-J., Chiang, C.-Y. Pervasive community care platform: Ambient intelligence leveraging sensor networks and mobile agents (2014) International Journal of Systems Science, 45 (4), pp. 778-797.
94. Pugh, K. Interface-Oriented Design (Pragmatic Programmers), Pragmatic Bookshelf, 21. Juli 2006
95. Gribok, A., Rumpler, W., Hoyt, R., Buller, M. Functional regression for data fusion and indirect measurements of physiological variables collected by wearable sensor systems and indirect calorimetry (2013) 2013 IEEE International Conference on Body Sensor Networks, BSN 2013, art. no. 6575464, .
96. Walters, V., Fierstine, H.L. Measurements of swimming speeds of yellowfin tuna and wahoo [43] (1964) Nature, 202 (4928), pp. 208-209.
97. Hidalgo Limited, Equivital™ SDK Instructions For Use. Dokumentation, HidalgoLtd., (2007).
98. ANT+ Device ProfileStride Based Speed and Distance MonitorDynastream Innovations Inc.

99. Su, T.-H., Tsai, H.-J., Yang, K.-H., Chang, P.-C., Chen, T.-F., Zhao, Y.-T. Reconfigurable vertical profiling framework for the android runtime system (2014) Transactions on Embedded Computing Systems, 13 (2 SUPPL.), art. no. 59.
100. Weeramanthri, T., Jancey, J. Fly-in-fly-out (FIFO) work in Australia: The need for research and a health promotion framework (2013) Health Promotion Journal of Australia, 24 (1), pp. 5-6.
101. Sánchez, M.G., López, R.N., Gay-Fernández, J.A. Real time health remote monitoring systems in rural areas (2014) IFMBE Proceedings, 41, pp. 1473-1476.
102. Maglogiannis, I., Kazatzopoulos, L., Delakouridis, K., Hadjiefthymiades, S. Enabling location privacy and medical data encryption in patient telemonitoring systems (2009) IEEE Transactions on Information Technology in Biomedicine, 13 (6), art. no. 4757292, pp. 946-954.
103. Anacleto, R., Figueiredo, L., Almeida, A., Novais, P. Server to Mobile Device Communication: A Case Study (2013) Advances in Intelligent Systems and Computing, 219, pp. 79-86.
104. Chen, D., Han, X., Wang, W. Use of SQLite on embedded system (2010) Proceedings - 2010 International Conference on Intelligent Computing and Cognitive Informatics, ICICCI 2010, art. no. 5565996, pp. 210-213.
105. Papazoglou, M.P., Van Den Heuvel, W.-J. Service oriented architectures: Approaches, technologies and research issues (2007) VLDB Journal, 16 (3), pp. 389-415.
106. Wang, M.-G., Mao, Y.-J., Li, W. The application of telemedicine technology (2014) Lecture Notes in Electrical Engineering, 269 LNEE, pp. 3261-3268.
107. Benharref, A., Serhani, M.A. Novel cloud and SOA-based framework for E-health monitoring using wireless biosensors (2014) IEEE Journal of Biomedical and Health Informatics, 18 (1), art. no. 6515329, pp. 46-55.
108. Chen, L.-C., Chen, C.-W., Weng, Y.-C., Shang, R.-J., Yu, H.-C., Chung, Y., Lai, F. An information technology framework for strengthening telehealthcare service delivery (2012) Telemedicine and e-Health, 18 (8), pp. 596-603.
109. Abousharkh, M., Mouftah, H. Service oriented architecture-based framework for WBAN enabled patient monitoring system (2011) Proceedings of the 2nd Kuwait Conference on e-Services and e-Systems, KCESS'11, art. no. 18, .
110. Kästner, C., Apel, S., Kuhlemann, M. Granularity in software product lines (2008) Proceedings - International Conference on Software Engineering, pp. 311-320.
111. Carbone, M., Honda, K., Yoshida, N. Structured communication-centred programming for web services (2007) Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4421 LNCS, pp. 2-17.
112. Skogan, D., Grønmo, R., Solheim, I. Webservicecomposition in UML (2004) Proceedings - IEEE International Enterprise Distributed Object Computing Workshop, EDOC, pp. 47-57.

113. Fu, X., Bultan, T., Su, J. Analysis of interacting BPEL Web services (2004) Thirteenth International World Wide Web Conference Proceedings, WWW2004, pp. 621-630.
114. Hui, S.C., Fong, A.C.M., Jha, G. A web-based intelligent fault diagnosis system for customer service support (2001) Engineering Applications of Artificial Intelligence, 14 (4), pp. 537-548.
115. [Http://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml](http://www.iana.org/assignments/uri-schemes/uri-schemes.xhtml) (2014)
116. Clemens, S., Dominik, G., Stephan, M. Component Software- BeyondObjectOriented Programming SecondEdition (2002),Addison-Wesley/ACM Press.
117. Demirkan, H., Kauffman, R.J., Vayghan, J.A., Fill, H.-G., Karagiannis, D., Maglio, P.P. Service-oriented technology and management: Perspectives on research and practice for the coming decade (2008) Electronic Commerce Research and Applications, 7 (4), pp. 356-376.
118. Boonserm, P., Jitsupap, A., Thongsamrith, N., Achalakul, T. E-framework for virtual collaborative community (2008) 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, ECTI-CON 2008, 1, art. no. 4600397, pp. 161-164.
119. Bartetzko, D., Fischer, C., Möller, M., Wehrheim, H. Jass-Java with assertions (2001) Electronic Notes in Theoretical Computer Science, 55 (2), pp. 103-117.
120. Diaz, M., Juanole, G., Courtiat, Jean, P. Observer-a concept for formal on-line validation of distributed systems (1994) IEEE Transactions on Software Engineering, 20 (12), pp. 900-913.
121. Gerdson, F., Müller, S., Jablonski, S., Probsch, H.-U. Standardized exchange of clinical documents: Towards a shared care paradigm in glaucoma treatment (2006) Methods of Information in Medicine, 45 (4), pp. 359-366.
122. Abousharkh, M., Mouftah, H. XMPP-enabled SOA-driven middleware for remote patient monitoring system (2012) 2012 International Conference on Information Technology and e-Services, ICITeS 2012, art. no. 6216689, .
123. Sutherland, L., Igras, E., Ulmer, R., Sargious, P. A laboratory for testing the interoperability of telehealth systems. (2000) Journal of telemedicine and telecare, 6 Suppl 2, pp. S74-75.
124. Gonçalves, P., Torres, J., Sobral, P., Moreira, R. Remote patient monitoring in home environments (2009) Proceedings of the 1st International Workshop on Mobilizing Health Information to Support Healthcare-Related Knowledge Work, MobiHealthInf 2009 In Conjunction with BIOSTEC 2009, pp. 87-96.
125. Martinez-Sarriegui, I., Zhu, H., Shan, L., García-Sáez, G., Gómez, E.J., Hernando, M.E. Modelling and formal specification of a multiagent telemedicine system for diabetes care (2009) ICAART 2009 - Proceedings of the 1st International Conference on Agents and Artificial Intelligence, pp. 507-512.

126. Aguilar, K.M., Campbell, R.S., Fiester, A., Simpson, R.L., Hertel, C. Bringing care home: How Telemonitoring can expand population health management beyond the hospital (2014) *Nursing Administration Quarterly*, 38 (2), pp. 166-172.
127. Warren, S., Lebak, J., Yao, J., Creekmore, J., Milenkovic, A., Jovanov, E. Interoperability and security in wireless body area network infrastructures (2005) *Annual International Conference of the IEEE Engineering in Medicine and Biology - Proceedings*, 7 VOLS, art. no. 1615297, pp. 3837-3840.
128. Papazoglou, M.P., Van Den Heuvel, W.-J. Service oriented architectures: Approaches, technologies and research issues (2007) *VLDB Journal*, 16 (3), pp. 389-415.
129. Yin, J., Chen, H., Deng, S., Wu, Z., Pu, C. A dependable ESB framework for service integration (2009) *IEEE Internet Computing*, 13 (2), pp. 26-34.
130. Mulesoft: <http://www.mulesoft.org> (2013)
131. Apache CXF: An Open-Source Services Framework: <http://cxf.apache.org/> (2013)
132. Stoicu-Tivadar, V., Stoicu-Tivadar, L., Puşcoci, S., Berian, D., Topac, V. WebService based solution for an intelligent telecare system (2012) *Studies in Computational Intelligence*, 378, pp. 383-408.
133. Park, I.-S., Shin, S.-J. WS security of XBRL financial documents encoded by SOAP (2013) *International Journal of Security and its Applications*, 7 (4), pp. 177-184.
134. Kyriacou, E., Pavlopoulos, S., Berler, A., Neophytou, M., Bourka, A., Georgoulas, A., Anagnostaki, A., Karayiannis, D., Schizas, C., Pattichis, C., Andreou, A., Koutsouris, D. Multi-purpose healthcare telemedicine systems with mobile communication link support (2003) *BioMedical Engineering Online*, 2, art. no. 7, 12 pp. 18-24.
135. Kannan, S.S., Varsha, N.S. Wireless telemonitoring of wearable device for recording biopotentials using embedded terminal platform (2013) *International Conference on Communication and Signal Processing, ICCSP 2013 - Proceedings*, art. no. 6577190, pp. 918-921.
136. Vergari, F., Gazzarata, R., Morandi, F., Parodi, V., Naso, S., Arezza, A., Salmon Cinotti, T., Giacomini, M. A standardized middleware as the core of a telemonitoring European Project (2012) *Studies in Health Technology and Informatics*, 180, pp. 497-501.
137. Pepe, G., Frumento, E., Garlaschelli, A. A service for remote screening of physiological data and follow up of patients using a home gateway and domestic wireless technologies (2004) *Studies in Health Technology and Informatics*, 103, pp. 299-306.
138. Carter, J.L., Wegman, M.N. Universal classes of hash functions (1979) *Journal of Computer and System Sciences*, 18 (2), pp. 143-154.
139. Trigo, J.D., Martinez-Espronedada, M., Eguzkiza, A., Serrano, L. An HTML5-based ECG viewer (2013) *Computing in Cardiology*, 40, art. no. 6713543, pp. 979-982.

140. Valky, G., Lehocki, F. Modern approach in multiple patients ECG monitoring (2012) Proceedings - IEEE-EMBS International Conference on Biomedical and Health Informatics: Global Grand Challenge of Health Informatics, BHI 2012, art. no. 6211526, pp. 131-134.
141. De Ridder, M., Constantinescu, L., Bi, L., Jung, Y.H., Kumar, A., Kim, J., Feng, D.D., Fulham, M. A web-based medical multimedia visualisation interface for personal health records (2013) Proceedings of CBMS 2013 - 26th IEEE International Symposium on Computer-Based Medical Systems, art. no. 6627787, pp. 191-196.
142. Moratilla, J.M., Alonso-Calvo, R., Molina-Vaquero, G., Paraiso-Medina, S., Perez-Rey, D., Maojo, V. A data model based on semantically enhanced HL7 RIM for sharing patient data of breast cancer clinical trials (2013) Studies in Health Technology and Informatics, 192 (1-2), p. 971.
143. Liu, J., Sun, Q. Information processing based on SQLite for target tracking system (2013) Applied Mechanics and Materials, 278-280, pp. 2065-2068.
144. Cruz, J., Brooks, D., Marques, A. Home telemonitoring in COPD: A systematic review of methodologies and patients' adherence (2014) International Journal of Medical Informatics, 83 (4), pp. 249-263.
145. Wang, W., Liu, Z. A learning method for pedestrian activity classification (2014) Chinese Journal of Electronics, 23 (1), pp. 129-134.
146. Hidalgo Limited, Equivital™ EQ-01 Vital Signs Monitor Health Care Practitioner Guide. Dokumentation, Hidalgo Ltd., (2006).
147. Huang, B., Wu, X. Pedometer algorithm research based-matlab (2012) Advances in Intelligent and Soft Computing, 168 AISC (VOL. 1), pp. 81-86.
148. Hidalgo Limited, Equivital™ EQ-01 Vital Signs Monitor User Application Guide. Dokumentation, Hidalgo Ltd., (2006).
149. Small, Low Power, 3-Axis ± 3 g MEMS Accelerometer Analog Devices, Inc., (2007).
150. Abe, B.T., Olugbara, O.O., Marwala, T. Classification of hyperspectral images using machine learning methods (2014) Lecture Notes in Electrical Engineering, 247 LNEE, pp. 555-569.
151. Frigo, M., Johnson, S.G. The design and implementation of FFTW3 (2005) Proceedings of the IEEE, 93 (2), pp. 216-231.
152. Ying, Y., Garrett, J.H., Oppenheim, I.J., Soibelman, L., Harley, J.B., Shi, J., Jin, Y. Toward data-driven structural health monitoring: Application of machine learning and signal processing to damage detection (2013) Journal of Computing in Civil Engineering, 27 (6), pp. 667-680.
153. Nauman, T.W., Thompson, J.A. Semi-automated disaggregation of conventional soil maps using knowledge driven data mining and classification trees (2014) Geoderma, 213, pp. 385-399.

154. Mitchell, J.B.O. Machine learning methods in chemoinformatics (2014) Wiley Interdisciplinary Reviews: Computational Molecular Science, . Article in Press.
155. Eastwood, M., Jayne, C. Evaluation of hyperbox neural network learning for classification (2014) Neurocomputing, 133, pp. 249-257.
156. Finkelstein, J., Wood, J. Designing pervasive telemedicine applications using various gaming platforms (2012) Proceedings of the IADIS International Conference e-Health 2012, EH 2012, Part of the IADIS Multi Conference on Computer Science and Information Systems 2012, MCCSIS 2012, pp. 97-104.
157. Neubert, S. Mobiles Online-Erfassungssystem für telemedizinische Anwendungen in der arbeits- und präventivmedizinischen Forschung, Dissertation 2011
158. Stoll, R., Kreuzfeld, S., Weippert, M. Vilbrandt, R., Stoll, N. System for Flexible Field Measurement of Physiological Data of Operators Working in Automated Labs, JALA 12(2), pp.110-114.
159. Vilbrandt, R., Kreuzfeld, S., Stoll, R. Flexible Erfassung von Belastungs- und Beanspruchungsparametern bei arbeitsmedizinischen Felduntersuchungen, Arbeitsmed Sozialmed Umweltmed. 41(10), pp.457-462.
160. Neubert, S., Behrendt, S., Rieger, A., Weippert, M., Kumar, M., Stoll, R. Echtzeit-Telemonitoring-System in der Präventivmedizin. In: Duesberg, F. (Hrsg.): E-Health 2012 - Informationstechnologien und Telematik im Gesundheitswesen. Medical Future Verlag, Solingen 2011, pp. 249-252
161. Kumar, M., Stoll, N., Thurow, K., Stoll, R. On the combination of fuzzy models. IEEE Conference on Automation Science and Engineering (CASE), Trieste, Italy, August 24-27, 2011, ISBN: 978-1-4577-1730-7, art. no. 6042461 pp. 322-326
162. Kumar, M., Stoll, N., Stoll, R. Stationary Fuzzy Fokker-Planck Learning and Stochastic Fuzzy Filtering. IEEE Transactions on Fuzzy Systems 19 (2011) 5, pp. 873-889 [SSCi-IF 2,695]
163. Vilbrandt, R., Kreuzfeld, S., Weippert, M., Stoll, R. „Flexible Erfassung von Beanspruchungsparametern bei arbeitsmedizinischen Felduntersuchungen“, Abstracts der 45. Jahrestagung der Deutschen Gesellschaft für Arbeitsmedizin und Umweltmedizin e. V. in Bochum 6.-9. April 2005, Arbeitsmed.Sozialmed.Umweltmed 40 (2005) 3, 206.
164. Stoll R., Arndt D., Kreuzfeld S., Weippert M., Thurow K. Stress-Monitoring in hoch automatisierten Umgebungen der Life Science. BioSpectrum 14(3), 258-261 (2008).
165. Hayes, G., El-Khatib, K., McGregor, C. Supporting health informatics with platform-as-a-service cloud computing (2014) Lecture Notes in Electrical Engineering, 260 LNEE, pp. 1149-1158.
166. Chang, L., Guan, C., Xu, J., Xu, H. Paas-platform for mobile health (2013) Proceedings - 5th International Conference on Computational Intelligence and Communication Networks, CICN 2013, art. no. 6658055, pp. 549-5

167. Xu, J., Zhang, T., Lin, D., Mao, Y., Liu, X., Chen, S., Shao, S., Tian, B., Yi, S. Pairing and authentication security technologies in low-power Bluetooth (2013) Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCoM 2013, art. no. 6682199, pp. 1081-1085.
168. Shah, S.H., Iqbal, A., Shah, S.S.A. Remote health monitoring through an integration of wireless sensor networks, mobile phones & Cloud Computing technologies (2013) Proceedings of the 3rd IEEE Global Humanitarian Technology Conference, GHTC 2013, art. no. 6713719, pp. 401-405.
169. Douglas, F., Loosley, C. High-Performance Client/Server, John Wiley and Sons Inc., 1997.
170. Denning, J.P. The Locality Principle, Communications of the acm, vol. 48, no. 7, pp. 19-24, July 2005.

Anhangsverzeichnis

Anhang A: Spezifikationen Sensorsysteme

A.1 Hidalgo Ltd. – Equivital – Technische Spezifikationen

Device Classification		
Shock Protection:	Type BF Applied Part , Internally Powered Equipment	
Environmental Protection:	IP67 (0.4m/30mins)	
Flammable Gas Protection:	Unprotected	
Mode of Operation:	Continuous	
FDA Device Classification:	Class II	
EU Device Classification:	Class IIb	
Chest Harness /Belt		
Size (circumference):	Small:	40mm x 108mm
	Medium:	40mm x 116mm
	Large:	40mm x 124mm
	Extra Large:	40mm x 131mm
Weight:	100g	
Duration of Use:	24 HRs	
Operating temperature:	-10°C to +55°C	
Operating Humidity:	10% to 75% RH Non-Condensing	
Storage Temperature:	-20°C to +65°C	
Storage Humidity:	5% to 90% RH Non-Condensing	
Air Pressure:	570hPA - 1060hPA	

A.2 Garmin Speed/Cadence

Der Garmin Speed/Cadence Sensor wird zur Messung der Pedaltrittfrequenz pro Minute und der Umdrehungsgeschwindigkeit von Fahrradreifen verwendet. Der Sensor ist selbstkalibrierend und besteht aus dem Sensorarm sowie zwei kleinen Magneten, die jeweils

an einem Pedal und einer Radspeiche monitert werden und so den Sensorarm am Rahmen passieren. Nach einem Reset durch den weien Knopf blinkt die LED rot beim Passieren des Pedalmagneten und grün beim Passieren des Speichenmagnets. Die LED blinkt jedoch nur 60 Sekunden nach einem Reset mit. In dieser Arbeit kann dieser Sensor beispielsweise auch bei einem Rollstuhl verwendet werden. Wie bereits im Abschnitt 4 erklärt, kann jedes ANT-kompatible Gerät über eine speizische Kanal ID einem ANT-Kanal zugewiesen werden. Die Kanal ID des hier verwendeten Garmin Speed/Cadence Sensor wird in der folgenden Tabelle dargestellt.

Cadence Sensor	Geräte Nummer(Device Number)	Gerätetyp(Device Type)	Übertragungsart(Transmission Type)
	0x06 0xAF	0x79	0x01

Anhang B: Spezifikation mobile Handhelds

A.1 Sony Ericsson xperia arc S

Prozessor	
Prozessor-Name:	ARM Cortex A8
Prozessor-Geschwindigkeit:	1,4 GHz
Anzahl Prozessor-Kerne:	Single-Core
Betriebssystem:	Android 2.3 Gingerbread
Maße	
Größe:	125 x 63 x 9 mm
Gewicht:	117 g
SAR-Wert:	0,66 W/kg (mittel)
Akku	
Akku:	Li-Ionen 1500 mAh
Standby:	19,2 Tage
Sprechzeit:	7,5 Stunden
Datentransfer	

Bluetooth:	JA
USB:	JA
GPS:	JA
ANT	JA
Speicher	
Speicherkapazität:	320 MB interner Speicher
Arbeitsspeicher (RAM):	512 MB
Steckplatz weitere Speicherkarte:	JA
Art der Speicherkarte:	microSDHC
Mobiles Internet	
GPRS:	JA
WLAN:	JA
EDGE:	JA
UMTS:	JA
HSDPA:	JA
HSUPA:	JA
LTE:	NEIN

Anhang C: ANT Message Protokoll and Programming Library

A.1 Output

Die Zusammenstellung der zu übertragenden Daten erfolgt identisch auf dem Arduino als auch in der Software-Frontend Library. Nur in der Ausgabe besteht ein Unterschied aufgrund der gegebenen Hardware.

A.1.1 Rechner

Am PC wird zur Kommunikation mit dem ANT.-Netzwerk der ANT. USB Stick verwendet. Die Verbindung zwischen dem nRF24AP1-Chip und dem Software- Frontend wird durch den FT232RL-Chip von FTDI realisiert. Dieser USB-Serial-Konverter kann durch eine API von

FTDI konfiguriert werden. Somit müssen die Byte-Arrays auf folgende Weise am PC übertragen werden.

```

1 // Globally defined
2 FT_HANDLE ftHandle ; // Variable for access of the FT232RL
3 FT_STATUS ftStatus ; // Variable for the status of the FT232RL
4 ...
5 void Output ( byte * setup , int iBytes )
6 {
7     DWORD dwBytesWritten = 0;
8
9     // Write data through the FT232RL - Chip to the Transceiver
10    ftStatus = FT_Write ( ftHandle , &setup , iBytes , & dwBytesWritten );
11    CheckError ( ftStatus ); // Check if the FTDI returns any error
12
13    Sleep (200); // Wait 200 ms
14 }

```

A.1.2 Arduino

Am Arduino wird zur Kommunikation mit dem ANT.-Netzwerk der nRF24AP1-Transceiver verwendet. Der Transceiver, bestehend aus dem nRF24AP1-Chip und Antenne, ist mit dem Arduino über die emulierte UART-Schnittstelle verbunden. Somit erfolgt die Kommunikation über die Soft-Serial-Schnittstelle des Arduino.

```

1 // Soft Serial Setup for the ANT Transceiver
2 # define rxPinANT 5
3 # define txPinANT 6
4
5 NewSoftSerial antSerial ( rxPinANT , txPinANT , false );
6...
7 void Output ( char * setup , int iBytes )
8 {
9     for(int i = 0 ; i < iBytes ; i++)
10    antSerial . print ( setup [i]);
11
12    delay (200); // Wait 200 ms
13 }

```

A.2 ANT. Befehle

Im Folgenden erfolgt eine kurze Übersicht der in dieser Arbeit verwendeten ANT.-Befehle mit explizitem Code ihrer Verwendung. Eine komplette Dokumentation aller Befehle sowie detaillierteren Informationen wird in Dyn [76] gegeben.

Diese Befehle dienen dazu die Parameter einer ANT.-Verbindung zu konfigurieren. Dazu werden Byte-Arrays entsprechend initialisiert und nach Abschnitt A.1 übertragen. Alle diese Kommandos geben eine Rückmeldung (Response) zurück.

A.2.1 Assign Kanal

Konfiguriert die Art der Übertragung eines ANT.-Kanals und weist den Kanal einem Network zu

Parameter		
Kanal Number	Kanal Type	Network Number
Nummer des zu konfigurierenden ANT.- Kanals	0x00 - Receive Kanal 0x10 - Transmit Kanal 0x20 - Shared Receive Kanal 0x30 - Shared Transmit Kanal	Spezifiziert zu welchem Network dieser Kanalgehört. Null für das Default Public Network

```

1 void AssignChannel (int iChannel )
2 {
3 byte setup [7];
4
5 setup [0] = 0xa4; // SYNC
6 setup [1] = 0x03; // MSG LENGTH
7 setup [2] = 0x42; // MSG ID = Assign Channel
8 setup [3] = iChannel ; // Channel Number
9 setup [4] = 0x00; // Channel Type ; RX (0 x00 ), Shared RX (0 x20 )
10 setup [5] = 0x01; // Network Number
11 setup [6] = CreateChecksum (setup , 6); // Returns the checksum
12
13 Output (setup , 7);
14 }

```

A.2.2 Unassign Kanal

Deaktiviert den gegebenen ANT.-Kanal.

Parameter
Kanal Number
Nummer des zu deaktivierenden ANT.- Kanals.

```

1 void UnAssignChannel (int iChannel )
2 {
3 byte setup [5];
4
5 setup [0] = 0xa4; // SYNC
6 setup [1] = 0x01; // MSG LENGTH
7 setup [2] = 0x41; // MSG ID = Assign Channel
8 setup [3] = iChannel ; // Channel Number
9 setup [4] = CreateChecksum (setup , 4);
10
11 Output (setup , 5);
12

```

```
13 delay (200);
14 }
```

A.2.3 Kanal ID

Die Kanal ID, bestehend aus 2 Byte Device Number, 1 Byte Device Type und 1 Byte Transmission Type, wird durch den Sender definiert und jeder Nachricht automatisch angehängt. Am Empfänger kann durch das Setzen einer Kanal ID die Verbindung zu einem bestimmten Sender hergestellt werden. Wird als Kanal ID Null (Wildcard) gesetzt, kann der Empfänger sich mit jedem Sender verbinden.

Parameter			
Kanal Number	Device Number	Device Type ID	Transmission Type
Nummer des ANT-Kanals	Individuelle Nummer jedes Senders	Art des Senders	Art der Übertragung

```
1 void SetChannelID (int iChannel , int iDevice )
2 {
3 byte setup [9];
4
5 setup [0] = 0xa4; // SYNC
6 setup [1] = 0x05; // MSG LENGTH
7 setup [2] = 0x51; // MSG ID = SetChannelID
8 setup [3] = iChannel ; // Channel Number
9
10 // Set Channel ID depending on given device
11 // setup [4] -- Device Number
12 // setup [5] -- Device Number
13 // setup [6] -- Device Type ID
14 // setup [7] -- Transmission Type
15
16 switch ( iDevice )
17 {
18 case WILDCARD : // All zero parameters to search for any device
19 setup [4] = 0x00;
20 setup [5] = 0x00;
21 setup [6] = 0x00;
22 setup [7] = 0x00;
23 break ;
24 case HRM: // Parameters for the Garmin HRM
25 setup [4] = 0x88;
26 setup [5] = 0xA5;
27 setup [6] = 0x78;
28 setup [7] = 0x01;
29 break ;
```

```
30 case FP: // Parameters for the Garmin Foot Pod
31 setup [4] = 0xAA;
32 setup [5] = 0xF0;
33 setup [6] = 0x7C;
34 setup [7] = 0x01;
35 break ;
36 case SC: // Parameters for the Garmin Speed Cadence
37 setup [4] = 0x06;
38 setup [5] = 0xAF;
39 setup [6] = 0x79;
40 setup [7] = 0x01;
41 break ;
42 default : // In the default case use the wildcard to find any device
43 setup [4] = 0x00;
44 setup [5] = 0x00;
45 setup [6] = 0x00;
46 setup [7] = 0x00;
47 break ;
48}
49
50 setup [8] = CreateChecksum (setup , 8);
51
52   Output (setup , 9);
53 }
```

Kurzreferat / Abstract

Das Ziel dieser Arbeit war die Entwicklung eines Telemonitoring-Systems, das eine effektivere Durchführung von Untersuchungen vornehmlich im Bereich der Arbeits- und Präventivmedizin ermöglicht. Die Aufzeichnung authentischer Daten in den entsprechenden Umfeldern der Probanden und die kontinuierliche Überwachung des Untersuchungsverlaufs trotz einer örtlichen Trennung der Beteiligten bieten neue Möglichkeiten in Belastungs-Beanspruchungs-Untersuchungen.

In den mobile clients wurde eine Middleware konzipiert um verschiedene Nahbereich-Kommunikationstechnologien zu unterstützen. Die Middleware erfasst die aktuelle interne und externe Rechnerumgebung des Geräts und gibt diese Informationen an den oberen Anwendungs-Layer über eine einheitliche Schnittstelle weiter. Auf diese Weise wurde auf Hardware-Ebene das Ziel der flexiblen Anbindung mit heterogener Sensorik erreicht.

Im Rahmen dieses Systems wurde ein SOA (Service Oriented Architecture) basiertes offenes Telemonitoring-Framework (SOTH) konzipiert und eingesetzt. SOTH verwendet den SOA-Ansatz zur Erzeugung eines strukturellen Rahmens für Ingenieure und Softwareentwickler, um ihnen Hilfestellungen bei der Entwicklung von Komponenten einer Mobile-Health-Plattform zu geben. In dieser Plattform werden die medizinischen Dienstleistungen als das Ergebnis der Kombination von verschiedenen abstrakten Ebenen betrachtet, was zur losen Koppelung der medizinischen Leistungen und zur Verbesserung der Interoperabilität zwischen den Software-Dienst-Modulen führt.

Für die Auswertungsmodule wurde in dieser Arbeit eine automatisierte Aktivitätsanalyse-methode entwickelt. Ziel dieser Komponente war die Auswertung von telemedizinisch erhobenen Vital-Daten in Verbindung mit einer Bewegungsklassifizierung. Diese bietet beispielhaft qualitativ neue Möglichkeiten für Langzeituntersuchungen verschiedener Vitalfunktionen. Aktivitätsprofile können einerseits eigenständig als Grundlage diagnostischer Aussagen dienen, andererseits in Kombination mit anderen Messgrößen deren Aussagekraft maßgeblich erweitern.

Als Ergebnis der Arbeit ist ein mobiles Online-Erfassungssystem entstanden, das die standortunabhängige und kontinuierliche Untersuchung mehrerer Probanden parallel ermöglicht. Die Standortunabhängigkeit ist dabei sowohl für den Probanden als auch für den Untersucher gegeben.

The aim of this thesis was the development of a telemonitoring system, which allows a more effective execution of examinations in the field of occupational health and preventivemedicine. The acquisition of authentic data in the familiar environments of the subjects andthe continuous monitoring of the examination process in spite of different locations of theparticipants offer new possibilities in examinations of stress and strain.

A SOA (Service Oriented Architecture) based open telemonitoring Framework (SOTH) was designed and used in the develop of the IPM-mHealth system. SOTH uses the SOA approach to create a structural framework for engineers and software developers to give them assistance in the development of components of a Mobile Health Platform. In this platform, the medical service units are considered as the result of the combination of different abstract levels, which leads to loose coupling of medical services and to improve interoperability between software service modules.

In the mobile client, a middleware was designed to support different short-range communication technologies. The middleware detects the current internal and external computing environment of the device and passes this information on to the upper application layer via a uniform interface. It separates the collection of sensor information and the development of upper-layer applications. Therefore, the object of the flexible connection is achieved with a heterogeneous sensor on the hardware level.

In the analysis Module an online activity recognition algorithms based on 3-axis acceleration sensorswas developed in this work. The monitor's purpose is to evaluate the physiological parameter combined with the human body status, and detect an anomalous event through the context information.

The result of this work is a mobile online acquisition system: IPM-mHealth, which allows the continuous parallel examination of several subjects regardless of location. The system has been successfully used during several studies and research projects.

Thesen

1. Die Verwendung kabelloser Übertragungsstandards verbessert wesentlich den Komfort für den Probanden und schafft ein Netzwerk, das sicher und flexibel erweiterbar ist.
2. Die Entwicklung eines flexiblen und mobilen Telemonitoring-Systems für medizinische Anwendungen ist an den Einsatz mobiler Rechnersysteme mit entsprechender Konnektivität (wie z. B. mobile Handhelds, SmartPhones) gebunden.
3. Die anwendungsspezifische und flexible Datenübertragung des Erfassungssystems erfordert die Konzeption eines eigenen, entsprechend anpassungsfähigen Übertragungsprotokolls.
4. Die konsequente Trennung von Datenerfassung, -übertragung, -speicherung und –auswertung ermöglicht es, das System auch zukünftig modularartig zu erweitern, ohne jeweils das Gesamtsystem anpassen zu müssen.
5. Die grafische Visualisierung von arbeitsphysiologischen Parametern synchron zu den relevanten Arbeitsaufgaben oder Tätigkeiten ermöglicht eine schnelle Vorauswertung und Plausibilitätsprüfung der aufgenommenen Messdaten.
6. Um einen umfassenden Effekt bei der Rationalisierung der Arbeiten in der Arbeitsphysiologie zu bewirken, reicht es nicht aus, vordergründig anwendbare und sofort einsetzbare Methoden der Automatisierungstechnik und Informationstechnologie zu benutzen, sondern man muss die Erfordernisse des Einsatzgebietes analysieren und spezielle Lösungen finden bzw. kombinieren.
7. Serverbasierte Auswertemethoden ermöglichen den einfachen Multi-User-Zugriff auf diese Routinen bei gleichzeitigem urheberrechtlichem Schutz der zugrunde liegenden Algorithmen.
8. Bei der Datenauswertung wird ein automatisierte Aktivitätsanalyse Algorithmus auf der Grundlage von Beschleunigungssensoren entwickelt. Aktivitätsprofile können

einerseits eigenständig als Grundlage diagnostischer Aussagen dienen, andererseits in Kombination mit anderen Messgrößen deren Aussagekraft maßgeblich erweitern.

9. Durch die definierten einfachen Schnittstellen von mHealth System ist es möglich, zukünftig weitere Parameter bzw. Messgeräte mit minimalem Aufwand in das System zu integrieren.
10. Bei der parallelen Untersuchung mehrerer Probanden kann die Echtzeitfähigkeit des Systems nur durch eine dezentrale Datenvorverarbeitung auf dem mobilen Rechnersystem ermöglicht werden.
11. Die Middleware erfasst die aktuelle interne und externe Rechnerumgebung des Geräts und gibt diese Informationen an den oberen Anwendungs-Layer über eine einheitliche Schnittstelle weiter. Auf diese Weise wurde auf Hardware-Ebene das Ziel der flexiblen Anbindung mit heterogener Sensorik erreicht.