



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Masterarbeit

Frank Hardenack

- Das intelligente Bett -
Sensorbasierte Detektion von Schlafphasen

Frank Hardenack

- Das intelligente Bett -
Sensorbasierte Detektion von Schlafphasen

Masterarbeit eingereicht im Rahmen der Masterprüfung
im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter: Prof. Dr. rer. nat. Gunter Klemke

Abgegeben am 31. Oktober 2011

Frank Hardenack

Thema der Masterarbeit

Das intelligente Bett - Sensorbasierte Detektion von Schlafphasen

Stichworte

Intelligente Wohnumgebungen, Schlafphasen, kapazitive Sensorik, Living Place Hamburg

Kurzzusammenfassung

Diese Masterarbeit beschäftigt sich mit der kontaktlosen Erkennung von Bewegungs- und somit auch angenommenen Leichtschlafphasen im Kontext von Smart Homes am Beispiel des Living Place Hamburg. Für diese Erkennung wurde ein mehrschichtiges System zur Interpretation von Sensordaten entworfen und implementiert. Die für die Erfassung notwendigen Sensoren wurden als Teil dieser Arbeit ebenfalls entwickelt und im Bett des Living Place Hamburg eingebaut.

Die Ergebnisse der Interpretation werden zur weiteren Verarbeitung auf dem ActiveMQ des Living Place Hamburg abgelegt und sollen die Basis für weitere Anwendungen darstellen.

Frank Hardenack

Title of the paper

The smart Bed - Sensor-based detection of sleep phases

Keywords

Smart Homes, Sleep Phases, capacitive sensing, Living Place Hamburg

Abstract

This master thesis deals with the contactless detection of movement- and therefore assumed light-sleep-phases in the context of smart homes using the example of the Living Place Hamburg. To interpret the sensor data a multi-layer system was developed and implemented for this detection. There was also a need to develop suitable sensors within this thesis. The sensors were integrated in the bed placed in the Living Place Hamburg.

The results of the interpretation are being published to the ActiveMQ in the Living Place Hamburg for further processing by other applications.

Inhaltsverzeichnis

1	Einleitung	1
2	Grundlagen	4
2.1	Der menschliche Schlaf	4
2.2	Kapazitive Sensorik	5
2.3	ActiveMQ	6
2.4	Smart Furniture	6
2.5	Der Living Place Hamburg	7
3	Existierende Ansätze	8
3.1	Kommerzielle Ansätze	8
3.2	Das klinische Schlaflabor	9
3.3	Ansätze in der Forschung	10
4	Analyse	18
4.1	Die Idee des intelligenten Bettes	18
4.2	Einsatzszenarien	20
4.3	Diskussion der Laborausstattung	21
4.3.1	Vergleich verschiedener Schwingkreiserzeuger	21
4.3.2	Evaluierung geeigneter Sensoren	22
4.3.3	Prototyp des Bettes	30
4.4	Aufgaben und Ziele	31
5	Design	32
5.1	Vorgehen bei der Verarbeitung der Sensordaten	32
5.2	Interpretationsebenen und deren Bedeutung	33
5.3	Interpretationsverfahren	36
5.4	Bestehende Visualisierungssoftware	39
6	Implementierung und technische Realisierung	42
6.1	Realisierung der Hardware	42
6.2	Erfassung von Testdaten	42
6.3	Implementierung der Software	43
6.4	Kommunikation der Interpretationsergebnisse	53
6.4.1	Format der Interpretationsergebnisse	53
7	Evaluierung	54
8	Fazit	56
9	Ausblick	57
	Glossar	58
	Literaturverzeichnis	59

A Messreihen	I
A.1 Aufbau 1	I
A.2 Aufbau 2	II
A.3 Aufbau 3	II
A.4 Aufbau 4	III
A.5 Aufbau 5	III
A.6 Aufbau 6	IV
A.7 Aufbau 7	IV
B Messwerte einer ganzen Nacht	VI
C Konfiguration der Software	XV
D Publizierung der Interpretationsergebnisse	XX
Tabellenverzeichnis	XXVI
Abbildungsverzeichnis	XXVI
Listings	XXVIII
Index	XXIX

Danksagung

An dieser Stelle möchte ich mich bei den Menschen bedanken, ohne die diese Masterarbeit nicht in dieser Form entstanden wäre.

Zuallererst geht ein riesiges Dankeschön an meine Eltern, die mich während meines gesamten Studiums bestmöglich unterstützt haben und mir immer mit Rat und Tat zur Seite standen.

Meiner Freundin danke ich für ihr Verständnis und dafür, dass sie mich in der Zeit der Masterarbeit unterstützt, umsorgt und letztlich auch ertragen hat.

Der Firma Rode Melder, speziell Herrn Grimmer, danke ich für die leihweise zur Verfügung gestellten Plattenelektroden und die sehr informativen Gespräche.

Für die Unterstützung in Bezug auf Ideen und Umsetzung im Rahmen der Masterarbeit sowie die wegweisenden Gespräche danke ich meinen beiden Prüfern, Kai von Luck und Gunter Klemke.

Zuletzt danke ich allen, die ich in der vorangegangenen Aufzählung vergessen habe.

1 Einleitung

„Das Forschungsvorhaben folgt der Vision, dass technische Systeme der Zukunft Companion-Systeme sind - kognitive technische Systeme, die ihre Funktionalität vollkommen individuell auf den jeweiligen Nutzer abstimmen: Sie orientieren sich an seinen Fähigkeiten, Vorlieben, Anforderungen und aktuellen Bedürfnissen und stellen sich auf seine Situation und emotionale Befindlichkeit ein. Dabei sind sie stets verfügbar, kooperativ und vertrauenswürdig und treten ihrem Nutzer als kompetente und partnerschaftliche Dienstleister gegenüber.“ (Universität Ulm, 2010)

In dieser Vision vom Sonderforschungsbereich (SFB) 62 Transregio, an dem unter anderem die Universität Ulm beteiligt ist, wird der Computer als eine Art Kamerad für den Menschen personifiziert. Die aktuelle Debatte über den Computer als Weggefährten zieht im Umfeld der Informatik große Kreise und hat einen sehr umfassenden Anspruch an die Technik und die Interpretation der Daten, lässt aber noch viel Interpretationsspielraum. In diesem Rahmen findet auch der Übergang vom Computer als Stand-Alone-Gerät zum Smart Environment, dem Computer als ein ubiquitärer Dienst in intelligenten Umgebungen im Sinne der Vision von Mark Weiser statt (Weiser, 1991).

„Sie orientieren sich an seinen Fähigkeiten [...] und stellen sich auf seine Situation und emotionale Befindlichkeit ein.“ (Universität Ulm, 2010)

Betrachtet man diesen Teil des Zitats genauer, so wird deutlich, dass Smart Environments in intelligenten Wohnumgebungen vom Grundgedanken bereits einige Punkte dieser „Buddy“-Metapher erfüllen. Eine konkrete Ausprägung für ein Smart Environment ist das intelligente Bett, welches sich zwar nicht direkt der Situation und Befindlichkeit eines Nutzers anpasst, aber anderen Anwendungen im Umfeld eines Smart Homes dazu verhelfen soll.

Ein weiterer Punkt zur Motivation für diese Arbeit war das eigene Interesse an intelligenten Wohnumgebungen, der damit verbundenen Context Awareness, sowie dem Schlafverhalten des Einzelnen. Es ist dabei sicherlich jedem bekannt, dass es Tage gibt, an denen das Aufstehen leichter fällt als an anderen Tagen. Die Abhängigkeit des Befindens nach dem Aufwachen vom Weckzeitpunkt und der gerade vorliegenden Schlafphase sind bereits bekannt, ebenso existieren bereits Ansätze, den Weckzeitpunkt dahingehend zu optimieren. Diesen Ansätzen fehlt aber bisher die Integration in die Schlafumgebung des Nutzers sowie in Smart Homes.

Wenn man die Vision des SFB 62 nun mit dem Living Place Hamburg (im Weiteren Living Place) kombiniert, dann bietet sich die Voraussetzung, ein intelligentes Bett in einem Smart Home so zu integrieren, dass es dem Living Place ermöglicht, sich auf die Situation und Befindlichkeit des Nutzers in Abhängigkeit der Schlafqualität einzustellen. Unter der Voraussetzung, dass die Technologie für die Erfassung geeignet und die zu entwickelnde Anwendung ausreichend adaptiv ist, bietet sich die Möglichkeit, den Einsatzbereich später auch auf den Bereich des Ambient Assisted Living (AAL) und damit auch in den Bereich der (Schlaf-)Medizin auszudehnen.

Ziel dieser Arbeit

Das Ziel dieser Arbeit ist die Entwicklung und Erstellung eines Systems zur Detektion von Schlafphasen durch Interpretation von Messwerten geeigneter Sensoren. Die dazu erforderlichen Sensoren werden als Teil dieser Arbeit ebenfalls entwickelt, da die Verwendung bildgebender Verfahren im Schlafbereich zur Wahrung der Privatsphäre nicht gewollt ist. Das zu erstellende System soll die Bewegungen eines Menschen im intelligenten Bett kontaktlos erfassen, erkennen, verarbeiten, mit geeigneten Verfahren interpretieren und eine Smart Home Umgebung wie den Living Place zu einem adäquaten Verhalten provozieren.

Eine einfache Bedienbarkeit, Konfigurierbarkeit und eine sehr gute Integration in bestehende Wohnumgebungen sind an das System gestellte Anforderungen, um eine möglichst hohe Akzeptanz bei dem Nutzer zu erzeugen.

Die durch die Interpretation erlangten Informationen sollen dem Living Place sowie den darin befindlichen Anwendungen (Wecker, Lichtinstallation, Belüftungssteuerung etc.) zur Verfügung stehen, damit diese in Abhängigkeit des Kontextes, in dem sich der Nutzer befindet (im Bett, gerade aufgestanden etc.) mit diesem interagieren können.

In dieser ersten Version des intelligenten Bettes wird der Schwerpunkt auf dem Lifestyle-Aspekt der Schlafphasenerkennung liegen, da an diesen bei Weitem nicht so hohe Anforderungen in Bezug auf Ausfall- und Erkennungssicherheit sowie Fehlinterpretationen gestellt werden wie an die zuerst angedachte Verwendung im medizinischen- oder AAL-Kontext.

Eine Veröffentlichung dieser Arbeit in Kurzform findet sich in der Zeitschrift „Schlafmagazin“, einer medizinischen Fachzeitschrift rund um das Thema Schlaf und Schlafstörungen. Die Ausgabe erscheint voraussichtlich im November 2011.

Vorarbeiten zur Arbeit

Im Rahmen der Veranstaltungen

- Erste Annäherung an das Thema (Hardenack, 2010b),
- Vergleich und Abgrenzung zu aktuellen, bestehenden Arbeiten (Hardenack, 2010c),
- Experimente (1) (Hardenack, 2010a),
- Thesis Outline (Hardenack, 2011a) und
- Experimente (2) (Hardenack, 2011b)

entstanden grundlegende Vorarbeiten zu dieser Masterarbeit, die hier ebenfalls Verwendung finden.

Dabei wurde in der **ersten Annäherung an das Thema** versucht, die anstehende Arbeit in der Informatik näher einzuordnen und einen Bereich der Informatik, der in der Arbeit Verwendung finden soll, abzustecken.

Im **Vergleich zu aktuellen Arbeiten** wurden ähnliche aktuelle Forschungsprojekte näher betrachtet und Gemeinsamkeiten sowie interessante Aspekte herausgearbeitet.

Die **Thesis Outline** beschreibt eine grobe Vorüberlegung über den Ablauf der Masterarbeit und ist als eine Thesis-Outline zu betrachten.

Während der ersten Praxisphase, die in **Experimente (1)** dokumentiert ist, wurde der erste Prototyp eines Lattenrostensors in mehreren Iterationen entwickelt sowie ein Vergleich verschiedener Schwingkreiserzeuger vorgenommen.

In **Experimente (2)** wurde der entwickelte Sensor-Prototyp verbessert und ein erstes Bett für Testzwecke damit ausgestattet. Des Weiteren wurde der letzte Schwingkreiserzeuger auf Einsetzbarkeit im intelligenten Bett getestet.

2 Grundlagen

In diesem Kapitel werden die nötigen Grundlagen zu dieser Arbeit und existierende Rahmenbedingungen des Living Place näher erläutert.

2.1 Der menschliche Schlaf

Der Schlaf eines Menschen richtet sich nach dem circadianen Rhythmus (Bear u. a., 2009) und lässt sich in fünf unterschiedliche Stadien unterteilen (vgl. Abb. 1, S. 5), die aufgrund ihrer charakteristischen Eigenschaften mit geeigneten Überwachungsmethoden unterscheidbar sind.

- Einschlafphase (Stadium 1)
- Leichter Schlaf (Stadium 2)
- Mittlerer Schlaf (Stadium 3)
- Tiefer Schlaf (Stadium 4)
- REM-Schlaf („Traumschlaf“, Stadium 5)

Im **Stadium 1 (Einschlafphase)** ist der Muskeltonus mittelhoch bis hoch, es können Einschlafzuckungen der Muskulatur auftreten und das Wachbewusstsein schwindet.

Das **Stadium 2 (Leichtschlaf)** zeichnet sich dadurch aus, dass das Wachbewusstsein komplett ausgeschaltet, der Muskeltonus aber noch mittelhoch bis hoch ist. In diesem Stadium ist die Weckbarkeit noch kaum vermindert.

Im **Stadium 3 (Mittlerer Schlaf)** ist die Weckbarkeit schon deutlich vermindert, der Muskeltonus ist nur noch niedrig.

Das **Stadium 4 (Tiefschlaf)** hat, im Vergleich zu den anderen Stadien, nur einen geringen Anteil am Nachtschlaf. Trotzdem ist es eines der wichtigsten Stadien, da der Tiefschlaf für die Erholung des Körpers von großer Bedeutung ist. Im Tiefschlaf nimmt der Muskeltonus weiter ab, die Weckbarkeit ist auf ein Minimum reduziert.

Träume werden im **REM-Schlaf (Traumschlaf)** erlebt. In diesem Stadium ist der Muskeltonus nahezu aufgehoben, damit der Schlafende die Bewegungen in seinen Träumen nicht wirklich ausführt und sich verletzt (Lund und Clarenbach, 1995).

Die verschiedenen Stadien werden während des Schlafs zyklisch durchlaufen, wobei ein Zyklus eine Dauer von 80 bis 110 Minuten besitzt. Bei einem Erwachsenen werden bei normalem Nachtschlaf (8 Stunden) vier bis fünf vollständige Zyklen durchlaufen (Boss u. a., 1993, Seite 1483), (Bear u. a., 2009). Im Schlafverlauf nimmt die Schlaftiefe und damit der Anteil des Tiefschlafs je Zyklus ab und der Anteil des Traumschlafs nimmt zu. Der Übergang vom REM-Schlaf in den Leichtschlaf führt, ebenso wie der umgekehrte Weg, zu (Fast-)Wachmomenten, in denen die schlafende Person sich unbewusst bewegt („Decke zurechtziehen“, „Schlafposition ändern“). Neben diesen (Fast-) Wachmomenten kann es in der Leichtschlafphase durch den mittelhohen bis hohen Muskeltonus zu leichten Muskelzuckungen (speziell bei der Gesichtsmuskulatur) kommen (Speckmann, 2008, Seite 263). Verglichen mit den restlichen Schlafphasen, in denen sich die

schlafende Person nahezu gar nicht bewegt, sind die unbewussten Bewegungen und Muskelzuckungen ein Indikator für (Fast-)Wachmomente und Leichtschlafphasen.

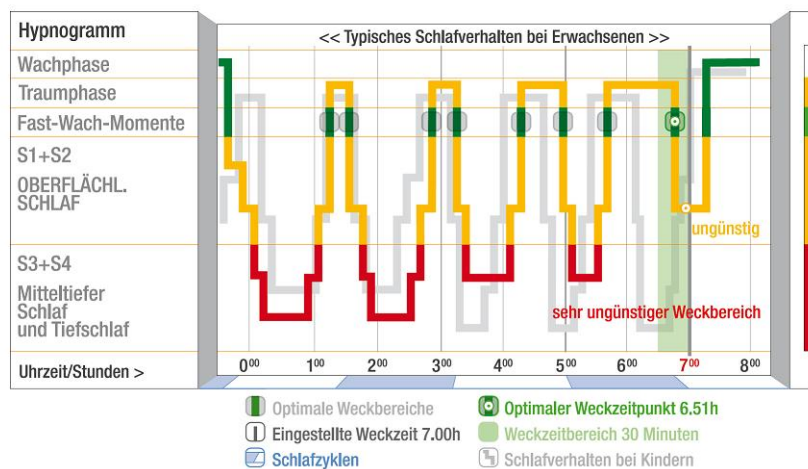


Abbildung 1: Hypnogramm (Quelle: www.axbo.com)

2.2 Kapazitive Sensorik

In der kapazitiven Sensorik bedient man sich der Funktionsweise und der Eigenschaften eines Kondensator, um berührungslos Abstände und Bewegungen von Objekten zu messen. Dazu wird eine der Kondensatorplatten durch ein leitfähiges, zu überwachendes Objekt (bzw. den Sensor) ersetzt. Nun wird die Sensorfläche mit einer positiven Ladung aufgeladen und kann eine Annäherung bzw. Entfernung von (geerdeten) Objekten gegenüber der Sensorfläche durch die implizite Beeinflussung der Kapazität des dadurch gebildeten Kondensators detektieren.

Die Messung von Annäherungen kann auf verschiedene Art und Weise erfolgen. Dabei basieren die in dieser Arbeit vorgestellten Schwingkreiserzeuger zwei unterschiedliche Messtechniken:

Bei der von RSI und Rode Melder verwendeten Messmethode wird durch einen Schwingkreiserzeuger ein Wechselfeld erzeugt, das als Bezugspotential das normale Erdpotential nutzt. Gemessen wird hierbei die Kapazität des Wechselfeldes zwischen Sensorfläche und Erdpotential. Findet nun eine Annäherung an die Sensorfläche statt, so verändert dies die gemessene Kapazität des Feldes und die Annäherung wird detektiert (Fraden, 2004).

Das Capsense Toolkit bedient sich eines als Oszillator verwendeten Timer-IC. Die Elektrode wird durch den Timer-IC kontinuierlich geladen und, wenn die Spannung der Elektrode einen definierten Wert überschreitet, entladen. Dabei wird die Zeit zwischen dem Umschalten und dem Erreichen der Spannungsgrenze gemessen. Diese Zeit variiert in Abhängigkeit von einer Annäherung an die Elektrode und der damit verbundenen Kapazitätsänderung (Wimmer u. a., 2007).

2.3 ActiveMQ

In (Snyder u. a., 2011) wird ActiveMQ von der Apache Software Foundation frei aus dem Englischen übersetzt als eine „*Open Source, JMS 1.1 kompatible, nachrichtenorientierte Middleware*“ beschrieben. Diese in Java implementierte Messaging-Middleware soll die Kommunikation zwischen einzelnen Anwendungen ausfallsicher, performant, skalierbar und verlässlich als Message-Broker sicherstellen und ist eine Art Blackboard, wie es in ersten Ansätzen schon in (Ermann und Lesser, 1980) beschrieben und verwendet wurde. Durch die Lizenzierung unter der Apache License sowie die standardisierten Kommunikationsschnittstellen und die Unterstützung vieler gängiger Programmiersprachen für den Zugriff auf die Nachrichten (Java, C++, C#, Ruby u.a.) eignet sich der ActiveMQ sehr gut als Messaging-System für die Kommunikation der einzelnen Anwendungen des Living Place unter einander.

Detaillierte Informationen zur Integration des ActiveMQ in den Living Place sowie die Entwicklung des HAW-spezifischen ActiveMQ-Wrappers werden in (Voskuhl und Otto, 2010) und (Voskuhl und Otto, 2011) erläutert.

2.4 Smart Furniture

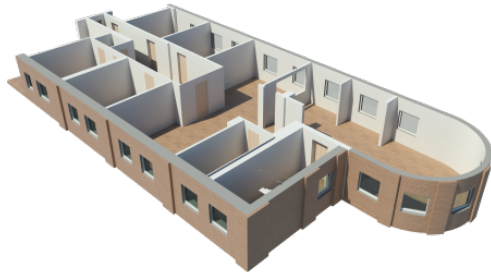
Smart Furniture, oder auch Computational- bzw. Intelligent Furniture, beschreiben alltägliche Möbelstücke, denen mittels Sensorik eine gewisse „*Intelligenz*“ gegeben wird, um den Nutzer bei alltäglichen Tätigkeiten zu unterstützen (Dreschke, 2009). Die erwähnte Intelligenz beschreibt dabei nicht eine Art künstlicher Intelligenz, vielmehr ist es das Wissen über das Umfeld und die Umgebung, die das Möbelstück intelligenter werden und auf äußere Einflüsse reagieren lässt.

Im einfachsten Fall wertet das Smart Furniture die erfassten Daten selbst aus und agiert in Abhängigkeit der vorliegenden Situation (Kontext / Context Aware Computing) direkt mit dem Nutzer.

Komplizierter wird es, wenn Smart Furnitures nur einen Teil des Gesamtsystems wie einer dienstorientierten Wohnung bilden. Das Smart Furniture muss dann Daten erfassen, sie gegebenenfalls auswerten, eventuell auf Informationen anderer Smart Furniture oder Technologien zurückgreifen und erweiterte, semantisch angereicherte Informationen für andere Anwendungen zur Verfügung stellen. Dies setzt eine vollkommene Integration in das Gesamtsystem voraus.

Ein weiterer, wichtiger Aspekt von Smart Furniture ist die bestmögliche Integration der Computertechnik in das Möbelstück, so dass dem Nutzer äußerlich kein Unterschied zu einem „normalen“ Möbelstück auffällt. Diese Integration ist dabei sehr wichtig für die Akzeptanz des Nutzers gegenüber dem Möbelstück, da die Computertechnik, wie schon im Grundgedanken des Ubiquitous Computing als *allgegenwärtig aber verborgen beschrieben*, keinesfalls eine Einschränkung oder unnötige Veränderung des gewohnten Umfelds darstellen darf.

2.5 Der Living Place Hamburg



Im Rahmen des *Living Place Hamburg* entsteht an der HAW Hamburg eine intelligente, dienstorientierte Wohnung, die der Forschung in den Bereichen des Ambient Assisted Living (AAL) und des Ubiquitous Computing (UbiComp) dienen soll. Inspiriert wurde dieses Projekt durch die Vision von Mark Weiser aus dem Jahr 1991 (Weiser, 1991). Vorarbeiten, die im Rahmen der iFlat (Stegelmeier u. a., 2009) entstanden sind, sowie bereits entwickelte Technologien können im Living Place unter realistischen Bedingungen getestet werden (Probewohnen, Einsatz in einer *echten* Wohnung).

Abbildung 2: 3D Modell des Living Place (Quelle: (LPHH Homepage, 2011))

Die gesamte Wohnfläche wird für Usability-Untersuchungen von mehreren Kameras und Mikrofonen überwacht, aus einer Steuerzentrale

kann so jede Bewegung in der Wohnung beobachtet und ausgewertet werden. (von Luck u. a., 2010)



3.1: Küche



3.2: Wohnbereich



3.3: Essbereich



3.4: Schlafbereich

Abbildung 3: Impressionen aus dem Living Place (Quelle: (LPHH Homepage, 2011))

3 Existierende Ansätze

Die Interpretation von Schlafphasen ist sowohl in der Forschung als auch in bestehenden, kommerziellen Produkten vom Grundsatz her nichts Neues. So gibt es für den Heimanwender bereits eine Auswahl an Schlafphasenweckern, die der Ermittlung des optimalen Weckzeitpunkts dienen. Ein professioneller Einsatz der Schlafphasenanalyse wird in medizinischen Schlaflaboren zur Diagnose von Schlafstörungen vorgenommen. Im Bereich der Forschung gibt es verschiedene innovative Projekte, bei denen aus einer Menge an Sensor-Rohdaten mittels semantischer Interpretation ein Kontext für den Nutzer erzeugt und damit eine Aussage über die vorliegende Situation getroffen wird.

3.1 Kommerzielle Ansätze

Aktuell gibt es mehrere verschiedene Systeme unterschiedlicher Anbieter zur Schlafüberwachung für den Heimanwenderbereich. Bei diesen Produkten handelt es sich um sogenannte *Schlafphasenwecker*, die zur Ermittlung der Leichtschlafphasen (vgl. Abschnitt 2.1) Bewegungssensoren verwenden. Die eingesetzte Technologie ist zwar wesentlich einfacher als die in den Schlaflaboren verwendete Technik (vgl. Abschnitt 3.2), aber dafür für den Heimanwender komfortabler in der Anwendung. Diese vereinfachte Technik wirkt sich zwar negativ auf die Genauigkeit der Messergebnisse aus, ist aber für den vorgesehenen Einsatzbereich vollkommen ausreichend.

Sleeptracker



Der Sleeptracker ist ein Schlafphasenwecker in Form einer normalen Armbanduhr, der seinen Träger während des Schlafs permanent auf Leichtschlafphasen und (Fast-) Wachmomente überwacht. Sollte eine Leichtschlafphase in ein Zeitfenster zwischen dem Weckzeitpunkt und einem bestimmaren Zeitpunkt davor fallen, so wird der Träger geweckt, spätestens jedoch zur eingestellten Weckzeit. Eine Auswertung der gesammelten Schlafdaten am PC ist, nach der Übertragung via USB, ebenfalls möglich. (Innovative Sleep Solutions, 2009)

Abbildung 4: Sleeptracker (Quelle: www.sleeptracker.de)

Funktionsweise: Der Sleeptracker erkennt (Fast-)Wachmomente und Leichtschlafphasen anhand der vermehrten Bewegung des Trägers (durch den höheren Muskeltonus in der Leichtschlafphase). Die Bewegungen werden durch einen Bewegungssensor im Sleeptracker erfasst und auf charakteristische Bewegungsmuster und -stärke analysiert. Fällt eine Leichtschlafphase in das Weckzeitfenster, so wird der Träger geweckt, spätestens jedoch zur eingestellten Weckzeit.

Schlafphasen-Wecker (aXbo)

Ein weiterer Schlafphasenwecker ist der aXbo-Schlafphasenwecker. Er bedient sich der gleichen Funktionsweise wie der Sleeptracker und wertet das Schlafverhalten des Nutzers auf Leichtschlafphasen aus. Zur Bestimmung der Leichtschlafphasen wird auch bei diesem Produkt ein Armband verwendet. Neben der Bestimmung des optimalen Weckzeitpunkts ermöglicht der aXbo das Aufzeichnen von Schlafdaten für zwei Personen über einen Zeitraum von 14 Tagen. Die Daten können mit einer Zusatzsoftware am PC ausgewertet und somit das persönliche Schlafverhalten analysiert werden (infactory innovations & trade GmbH, 2009).



Abbildung 5: aXbo-Wecker
(Quelle: www.axbo.com)

Sleep-Cycle (iPhone-App)



Abbildung 6: Logo der Sleep-Cycle App (Quelle: www.lexwarelabs.com)

Ein innovativer Ansatz zur kontaktlosen Erfassung und Interpretation von Schlafphasen wird mit der iPhone-App *Sleep Cycle Alarm Clock* von LexWare Labs verfolgt. Zur Erfassung wird das iPhone am Kopfende des Betts auf die Matratze gelegt und die App gestartet. Die Erfassung der Schlafphasen erfolgt mittels der im iPhone verbauten Beschleunigungssensoren, die die Bewegungen des Nutzers über die Matratze erfassen. Die Funktionsweise ähnelt der des Sleeptrackers.

Bewertung der existierenden Ansätze

Die beiden vorgestellten Schlafphasenwecker wurden nicht selbst getestet, jedoch ist das Anlegen eines Armbands bzw. einer Art Armbanduhr gleichzusetzen mit einer *Verkabelung* des Nutzers. Dies trägt bestimmt nicht zu einem ruhigeren Schlaf bei und ruft mit Sicherheit Akzeptanzprobleme hervor, da man sich nicht so *frei* zu Bett begeben kann wie ohne die Verwendung eines Schlafphasenweckers. Eine Aussage über die tatsächliche Funktionstüchtigkeit der Schlafphasenwecker ist durch die fehlende Testmöglichkeit nicht machbar.

Die Applikation für das iPhone wurde persönlich getestet, hat aber nicht das gewünschte Ergebnis geliefert. So klingelte der Wecker bei allen Tests erst zur eingestellten Weckzeit. Dazu kommt, dass ein Handy direkt neben dem Kopf den Schlafenden unnötigerweise einem elektrischen Feld aussetzt, was, wie die Armbänder der Schlafphasenwecker, zu Akzeptanzproblemen bei den Nutzern führen kann.

3.2 Das klinische Schlaflabor

Ein Schlaflabor dient der Untersuchung von Schlafstörungen und Schlafkrankheiten wie beispielsweise Narkolepsie und Schlafapnoe. In diesen Laboren werden Patienten mit

einer Vielzahl von Elektroden versehen, bevor sie in der Klinik eine Nacht zur Erfassung von Schlafdaten verbringen (vgl. Abb. 7, S. 10). Die erfassten Daten werden nach Abschluss der Untersuchung analysiert und geben Aufschluss über den physischen Zustand des Schlafenden. Aus diesen Messreihen lassen sich Rückschlüsse auf die Schlafqualität und die Schlafphasen ziehen.

Die erwähnten Elektroden dienen dabei der...

- ... Überwachung der Atmung
- ... Messung der Herzfrequenz (**ElektroKardioGramm**)
- ... Messung der Augenbewegung (**ElektroOkuloGraphie**)
- ... Messung der Muskelspannung (**ElektroMyoGraphie**)
- ... Messung der Hirnströme (**ElektroEnzephaloGraphie**)

Des Weiteren werden in diesem Zusammenhang oft die Körpertemperatur, die Beinbewegung und die Körperlage überwacht.

Die verwendeten Elektroden erlauben bei der Auswertung der Daten eine exakte Festlegung, wann sich der Patient in welcher Schlafphase befunden hat. Neben den in der Grafik abgebildeten verwendeten Elektroden wird der Schlafende häufig über eine Infrarot-Kamera aufgezeichnet.

Eine Schlafüberwachung dieser Art in einem Schlaflabor wird Polysomnographie genannt.

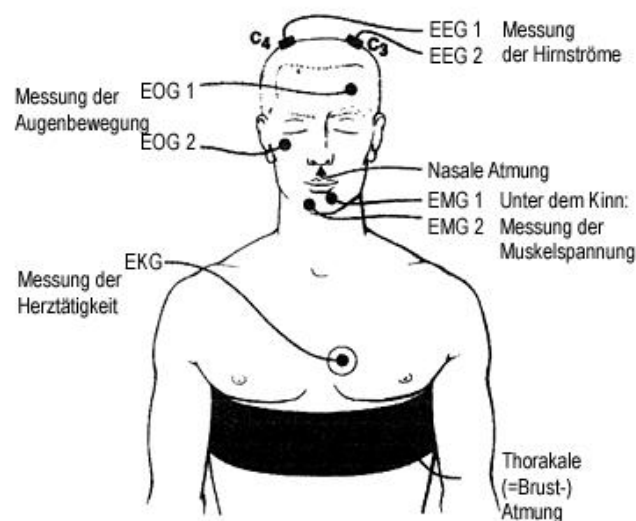


Abbildung 7: Elektroden zur Schlafanalyse im Schlaflabor (Quelle: www.asklepios.com)

3.3 Ansätze in der Forschung

Die Vielzahl der aktuellen Projekte, bei denen es darum geht, Sensordaten semantisch zu interpretieren, hat die Auswahl der vorzustellenden Projekte deutlich erschwert. Die

Wahl fiel auf die folgenden Projekte, da diese interessante und eventuell im späteren Verlauf weiter verwertbare Informationen zur semantischen Interpretation liefern. Leider war kein aktuelles Projekt ausfindig zu machen, das sich direkt mit der Interpretation von Schlafphasen beschäftigt.

ConText

Das ConText-Projekt ist ein von der EU gefördertes Forschungsprojekt, an dem neben verschiedenen europäischen Forschungseinrichtungen wie Philips Research oder dem Textilforschungsinstitut Thüringen-Vogtland auch Hochschulen (TU Berlin, Katholieke Universiteit Leuven (Belgien)) beteiligt waren (Langereis u. a., 2007). Das Projekt fand bereits Mitte 2008 seinen Abschluss. Ziel dieses Projekts war die Entwicklung einer „intelligenten Weste“ (Abbildung 9), die zur Früherkennung von Muskel-Skelett-Erkrankungen¹ eingesetzt werden kann.

Entwicklung eines geeigneten EMG-Sensors

Neben der Entwicklung der Weste war zuerst die Entwicklung eines textilen EMG²-Sensors notwendig. Die Anforderung war dabei die mögliche Kombination mit Textilien, um den Sensor bestmöglich in die Kleidung (in diesem Fall die Weste) zu integrieren sowie eine Erfassung der Muskelspannung ohne direkten Hautkontakt. Durch die Verwendung von EKG³-Elektroden wurde eine Erfassung ohne direkten Hautkontakt ermöglicht. Die Integration in Textilien wurde durch die Verwendung elektrisch leitfähiger Stoffe und Garne realisiert. Daraus entstand ein robuster, leicht in Kleidung vernäher Sensor zur Erfassung der Muskelspannung, der in ersten Versuchen vielversprechende Ergebnisse lieferte und durch seine Integration in die Kleidung die Akzeptanzprobleme beim Nutzer minimierte.

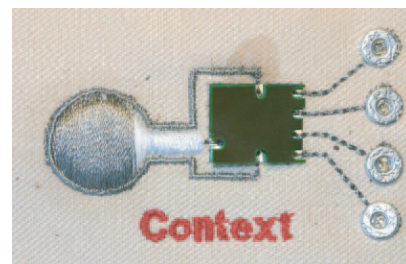


Abbildung 8: ConText EMG-Sensor (Quelle: (Taelman u. a., 2006))

Funktionsweise

Die Idee zur Früherkennung von Muskel-Skelett-Erkrankungen liegt in der Erfassung des physischen Stresslevels und der rechtzeitigen Benachrichtigung des Trägers der Weste, wenn dieses Stresslevel zu hoch wird, was wiederum auf eine muskuläre Überlastung schließen lässt. Dazu wird ein EMG durch die Messung der elektrophysiologischen Signale der überwachten Muskelgruppen erstellt und nach einem Algorithmus der katholischen Universität Leuven (Belgien) ausgewertet. Neben der gemessenen Muskelspannung werden noch weitere Attribute des menschlichen Körpers zur Betrachtung herangezogen. Dazu gehören unter anderem die Körpertemperatur und die Herzfrequenz des

¹Muskelzerrungen, allgemeine Schäden am Bewegungsapparat

²ElektroMyoGraphie - Messung der Muskelspannung

³ElektroKardioGramm - Messung der Herzfrequenz

Nutzers. Nach der Auswertung der erfassten Messdaten wird dem Nutzer gegebenenfalls eine Rückmeldung über eine drohende Überlastung des Bewegungsapparats gegeben. Durch diese Rückmeldung soll dem Nutzer eine nötige Pause oder, am Beispiel des Fliesenlegers aus Abbildung 9, eine Veränderung der Arbeitsposition oder Tätigkeit zur Entlastung des Schulterbereichs signalisiert werden.



9.1: Beispiel: Fliesenleger



9.2: Beispiel: Informatiker

Abbildung 9: Weste des ConText-Projekts (Quelle: (ConText, 2006))

Fazit

Mit dem Hintergrund der Früherkennung von Schäden am Bewegungsapparat entstand im Rahmen des ConText-Projekts ein vernäherbarer Sensor zur Messung der Muskelspannung. Die vollständige Integration in ein textiles Produkt ist dabei unter Verwendung leitfähiger Garne und Druckknöpfe als Kontakte sehr gut gelungen. Für die Überwachung des physischen Stresslevels bei körperlicher Arbeit wurde eine Weste entwickelt, in der diese Sensoren vernäht sind. Neben dem Einsatz in der entwickelten Weste ist eine Verwendung zur Sportlerüberwachung ebenfalls angedacht (ConText, 2008). Durch die bereits bei der Entwicklung berücksichtigten verschiedenen Einsatzkontexte stellt sich die Frage, ob dieser Sensor, integriert in ein Kissen, die Matratze oder ein Bettlaken auch für den Einsatz im intelligenten Bett geeignet ist und in wie weit die erfasste Muskelspannung Rückschlüsse auf die Schlafphase des Anwenders zulässt.

sWBAN

Das sWBAN-Projekt (**s**emantic **W**ireless **B**ody **A**rea **N**etwork) beschäftigt sich mit der Entwicklung einer Methode zur Interpretation und semantischen Anreicherung von Bodymonitoring-Daten. Die entwickelte Methode wird am Beispiel der Früherkennung von Herzrhythmusstörungen durch Interpretation der EKG-Messwerte näher erläutert. Hierbei sollen die Symptome interpretiert und dem Nutzer gegebenenfalls Hinweise auf

ein weiteres Vorgehen gegeben werden. Sollte ein akuter Notfall vorliegen, so wird der Rettungsdienst oder eine andere Instanz zur Hilfeleistung benachrichtigt. Zur Erfassung der EKG-Messwerte wurde ein wireless EKG-Sensor entwickelt (Nimmala u. a., 2008). Dieser Sensor überträgt die Messwerte kabellos an einen PC, der die Auswertung und Interpretation vornimmt.

Entwicklung des wireless EKG-Sensors

Für die Erfassung des Herzschlags wurde ein sehr leichter und flexibler Sensor entwickelt. Dabei wurde darauf geachtet, dass der Sensor wenig Energie verbraucht, um auch eine Langzeitüberwachung des Herzschlags von bis zu einer Woche zu ermöglichen. Auf einer flexiblen Kunststoffolie ist neben einem Standard-EKG-Sensor ein Microcontroller, die Stromversorgung, sowie eine Sendeeinheit zur Datenübertragung (2.4 Ghz) untergebracht. Durch die Verwendung einer Kunststoffolie als Trägermaterial passt sich der Sensor der Anatomie des Trägers an. Insgesamt wiegt der Sensor gerade einmal ≈ 15 Gramm und stellt durch die flexible Bauform und das niedrige Gewicht nur eine geringe Einschränkung des Nutzers dar.

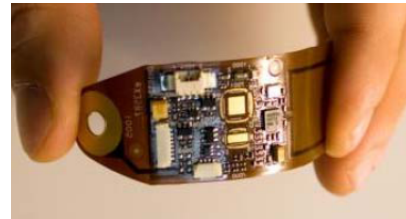


Abbildung 10: sWBAN Sensor
(Quelle: (Nimmala u. a., 2008))

ID	Feature	Class	value	Unit
001002	HR	Normal	$50 < HR < 100$	bpm
001001	HR	Low	< 50	bpm
001003	HR	High	> 100	bpm
002002	QRS width	Normal	$80 < qrswidth < 120$	ms
002001	QRS width	Low	< 80	ms
002003	QRS width	High	> 120	ms
003002	QRS amplitude	Normal	$1.0 < qrsheight < 2.0$	mV
003001	QRS amplitude	Low	< 1.0	mV
003003	QRS amplitude	High	> 2.0	mV
004002	QT interval	Normal	$360 < qtint < 440$	ms
004001	QT interval	Low	< 360	ms
004003	QT interval	High	> 440	ms
005002	P width	Normal	$90 < pwidth < 130$	ms
005001	P width	Low	< 90	ms
005003	P width	High	> 130	ms
006002	P amplitude	Normal	$0.1 < pheight < 0.2$	mV
006001	P amplitude	Low	< 0.1	mV
006003	P amplitude	High	> 0.2	mV
007002	PQ/PR interval	Normal	$120 < pqint < 200$	ms
007001	PQ/PR interval	Low	< 120	ms
007003	PQ/PR interval	High	> 200	ms

Abbildung 11: sWBAN Lookup-Tabelle (Quelle: (Nimmala u. a., 2008))

Funktionsweise

Abbildung 12 zeigt den Aufbau eines sWBAN. Sollen nun Sensordaten interpretiert oder semantisch angereichert werden, so werden die einzelnen Stufen von unten nach oben durchlaufen.

Zu erst werden durch den EKG-Sensor Messdaten erfasst. In der nächsten Stufe, der **Feature Extraction**, werden charakteristische Merkmale (Herzfrequenz, QRS-Komplexe⁴) des EKG-Signals mittels einer diskreten Wavelet Transformation aus den Messwerten extrahiert. Im hier vorliegenden Beispiel findet die Feature Extraction direkt auf dem entwickelten EKG-Sensor statt. Die Ergebnisse werden zur weiteren Verarbeitung an einen PC gesendet.

In der **semantischen Assoziation** werden die extrahierten Merkmale klassifiziert. Im einfachsten, hier vorliegenden Fall wird die Klassifikation anhand einer Lookup-Tabelle (z.B. nach der Herzfrequenz) vorgenommen, aber auch der Einsatz von maschinellen Verfahren zur Klassifikation (z.B. Support Vector Machines) ist möglich. Durch die Lookup-Tabelle wird ein Merkmal einer semantisch assoziierten Klasse, die einen Wertebereich über ein Merkmal abbildet, zugeordnet (Vgl. Lookup-Tabelle in Abbildung 11). Die klassifizierten Merkmale sind nun semantisch assoziierte Attribute und werden Features genannt.

Auf die semantische Assoziation folgt die **semantische Anreicherung**. Dabei werden die einzelnen, extrahierten Features zusammen als ein Ganzes betrachtet. Bei dieser ganzheitlichen Betrachtung werden inkompatible Kombinationen der Features, die in dieser Form nicht auftreten können und auf Messfehler oder falsche Anreicherung hinweisen, nach vorher definierten Regeln herausgefiltert und aus der Menge der semantischen Attribute entfernt.

Der letzte Schritt der Datenverarbeitung ist die **semantische Interpretation**, bei der durch eine Semantik-Engine ermittelt wird, welche Diagnose am besten auf die Ergebnisse der Vorverarbeitung passt. Dabei werden die semantisch angereicherten Daten mit medizinischem Fachwissen und kontextbezogenen Daten wie der Krankenakte oder bekannten vorangegangenen Erkrankungen kombiniert. Die Semantik-Engine ermittelt die möglichen Interpretationen und versieht diese mit einer Punktzahl (in Prozent), die angibt, welche Übereinstimmung die erzeugten Features mit dieser Interpretation haben. Das Ergebnis wird dem Nutzer in Form eines Feedbacks vermittelt und das sWBAN agiert in Abhängigkeit des Ergebnisses mit dem Nutzer.

Fazit

Mit dem Ziel der Entwicklung eines intelligenten Health-Monitoring-Systems entstand im sWBAN-Projekt ein Konzept zur Entwicklung eines semantischen Netzwerks zur Interpretation von Sensordaten am Beispiel des EKG. Die schichtweise Aufbereitung der Daten liefert dabei Ansätze für die Verarbeitung anfallender Messwerte in einem Netzwerk aus Bodymonitoring-Applikationen. Durch diese Schichtung entsteht ein leicht um weitere Sensoren erweiterbares Konzept, das sich auf den jeweiligen Anwendungsfall (z.B. EKG) durch Festlegen der Kriterien für die semantische Assoziation, Anreicherung und Interpretation anpassen lässt. Die Verwendung eines maschinellen Lernverfahrens

⁴Der positive Ausschlag im EKG-Signal ist ein sogenannter QRS-Komplex

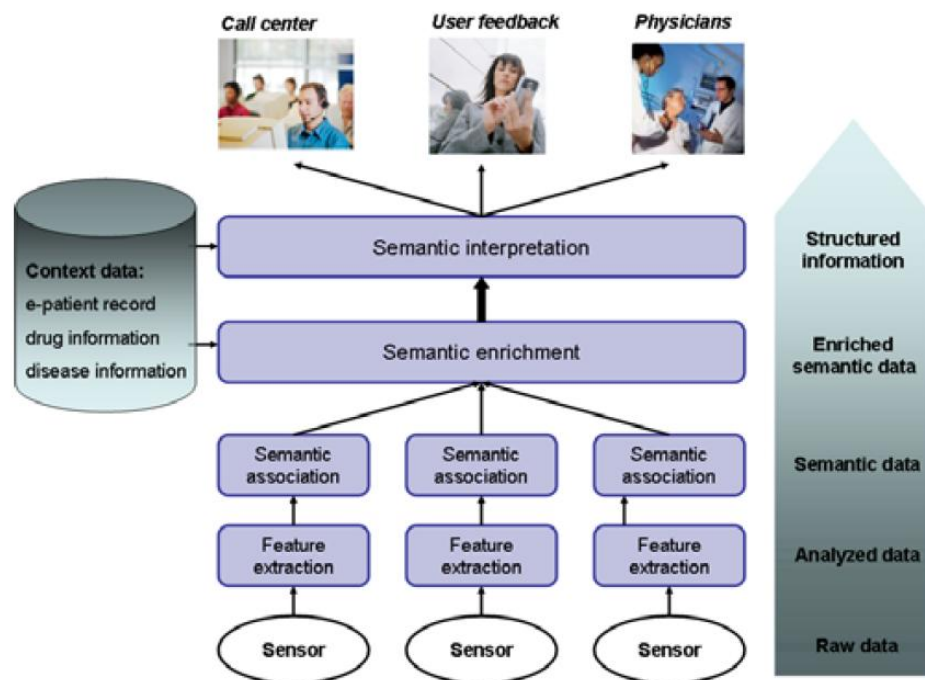


Abbildung 12: Übersicht der Funktionsweise des sWBAN (Quelle: (Nimmala u. a., 2008))

würde die semantische Assoziation der Daten im Vergleich zu einer Lookup-Tabelle noch verbessern.

„Human Status Awareness“

Das Projekt *Enabling Human Status Awareness in Assistive Environments based on Advanced Sound and Motion Data Classification* (kurz: Human Status Awareness) ist ein Projekt der griechischen Universität der Ägäis in Karlovasi (Samos). Die Zielsetzung ist hierbei die Erfassung des Zustands von Menschen, speziell die Sturzerkennung im Bereich des Ambient Assisted Living (AAL). Ein Sturz soll anhand seiner charakteristischen Eigenschaften wie auffälligen Geräuschen beim Aufschlagen auf den Boden sowie der damit verbundenen Beschleunigungskurve mit einem plötzlichen Abbruch der Beschleunigung erkannt werden. Um eine Erkennung unabhängig vom Nutzer und von der Umgebung zu gewährleisten werden die erfassten Daten mit einem maschinellen Lernverfahren, in diesem Fall mit einer Support Vector Machine, klassifiziert (Doukas und Maglogiannis, 2008).

Der entwickelte Sensor

Zur Erfassung der benötigten Daten wurde ein tragbarer Sensor entwickelt, den der Nutzer am Körper tragen muss, damit die Daten korrekt erfasst werden. Zur Erfassung der Daten kommen als Komponenten zwei 3-Achsen Beschleunigungssensoren zum Messen der Beschleunigungen sowie auf einem optionalen Erweiterungsmodul ein Mikrofon zur Erfassung der Umgebungs- und Sturzgeräusche zum Einsatz. Der Sensor bietet Schnittstellen zur Erweiterung durch andere Komponenten wie einen Temperaturfühler oder Helligkeitssensor. Die Übertragung der erfassten Daten zum PC kann sowohl über die vorhandene serielle Schnittstelle (RS232) als auch via ZigBee stattfinden.

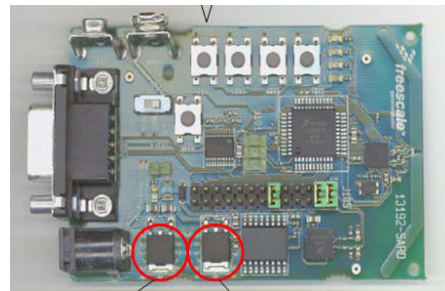


Abbildung 13: Sensor zur Sturzerkennung (Quelle: (Doukas und Maglogiannis, 2008))

Funktionsweise

Die Funktionsweise des entwickelten Ansatzes wird in Abbildung 14 schematisch dargestellt. Der entwickelte Sensor erfasst die Beschleunigung, der der Körper des Nutzers unterliegt und das eingebaute Mikrofon die Umgebungs- sowie eventuelle Sturzgeräusche. Die erfassten Daten werden über ZigBee an einen sogenannten Monitoring Node übermittelt, der eine Verarbeitung der Daten vornimmt. Zuerst werden die Daten des Mikrofons vorverarbeitet, indem diese einen Rauschfilter durchlaufen und charakteristische Sequenzen mittels Feature Extraction herausgearbeitet werden. In Kombination mit den erfassten Beschleunigungen entsteht daraus ein Feature Vektor, der an das Klassifikationsmodul weitergegeben wird.

Für die Klassifikation der Daten wird eine Support Vector Machine (SVM) verwendet, die durch einen Satz Trainingsdaten⁵ vorher angelernt wurde. Die SVM klassifiziert das durch den Feature Vektor repräsentierte Ereignis und gibt einen Status des Nutzers in Bezug auf einen Sturz sowie die damit verbundenen Daten aus. Auf Basis dieses Status und der Daten könnten dann beispielsweise Angehörige, Rettungs- oder Pflegekräfte informiert werden.

Fazit

Das Human Status Awareness Projekt verfolgt einen interessanten Ansatz, bei dem unter Zuhilfenahme eines Mikrofons neben den Bewegungsdaten des Beschleunigungssensors auch die Umgebungsgeräusche mit in die Auswertung einfließen. Durch diesen Ansatz soll es auch möglich sein, das Stresslevel in der Stimme des Nutzers zu bestimmen, um die fehlerhaften Klassifikationen zu minimieren. Die Ergebnisse der Klassifikation werden bereitgestellt und können durch andere Applikationen weiter verarbeitet oder genutzt werden. Dadurch ist eine Integration in eine altersgerechte Wohnumge-

⁵Für die Erfassung der Trainingsdaten wurden Testpersonen mit dem Sensor ausgestattet und haben Sturzereignisse auf verschiedenen Untergründen simuliert

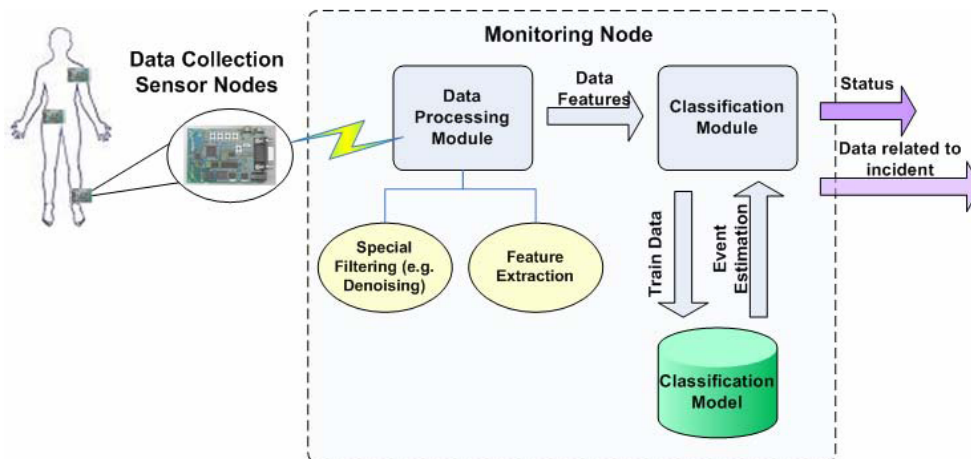


Abbildung 14: Übersicht der Funktionsweise des Human Status Awareness Projekts (Quelle: (Doukas und Maglogiannis, 2008))

bung gut realisierbar. Wenn der Sensor in der Bauform noch optimiert wird, dann stellt dieses Projekt eine sinnvolle Erweiterung der Überwachung des Gesundheitszustands von Menschen dar. Die Erweiterbarkeit des Sensors um weitere Komponenten zur Datenerfassung bietet vielfältige Möglichkeiten, die Präzision der Dateninterpretation zu erhöhen oder durch eine Vielzahl verschiedener Komponenten die Ausgangsdaten für unterschiedliche Überwachungsfunktionen zu liefern.

Somnus Schlafhemd

Das junge Startup-Unternehmen Nyx Devices⁶ entwickelt aktuell in Kooperation mit dem Massachusetts Institute of Technology (MIT) an einem Smart Shirt, das die Atemaktivität des Trägers durch eingewebte Sensoren überwacht. Da sich die Atmung in Tiefe, Frequenz und Pausen je nach Schlafphase unterscheidet, soll so durch die erfassten Atemmuster ein Rückschluss auf die vorliegende Schlafphase und die Schlafqualität möglich sein. Dabei sind REM-Schlafphasen durch die variierende Atemtiefe und -frequenz von den Tiefschlafphasen, die durch eine sehr gleichmäßige, fast sinusartige Atmung charakterisiert sind, abgrenzbar. Das Somnus Schlafhemd soll dabei gleichermaßen für Heimanwender sowie medizinische Einrichtungen geeignet sein.

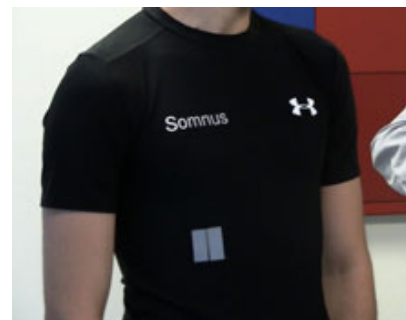


Abbildung 15: Somnus Schlafhemd (Singer, 2011)

Sollten die aktuell anstehenden Versuche zufriedenstellende Ergebnisse liefern, so will das Unternehmen das Produkt ab Mitte 2012 in den Handel bringen (Singer, 2011).

⁶<http://nyxdevices.com>

4 Analyse

Im Rahmen der Analyse soll erarbeitet werden, wie das intelligente Bett aussehen und was es leisten soll. Dabei geht es nicht um konkrete Anforderungen, vielmehr steht ein Überblick über das gesamte System von der Idee eines Smart Furniture über die Metapher des Smart Environments und der Ambient Intelligence im Mittelpunkt. Das intelligente Bett soll hierbei weniger auf die Belange des Ambient Assisted Living eingehen sondern vielmehr als ein interdisziplinäres Projekt im Rahmen von Smart Environments zur Integration in eine dienstorientierte Wohnung wie den Living Place (Vgl. Abschnitt 2.5, S. 7) mit vielen, unterschiedlichen Projekten (Intelligentes Sofa, Smart Textiles, Indoor Lokalisierung, Sturzerkennung etc.) gesehen werden.

4.1 Die Idee des intelligenten Bettes

Die grundlegende Idee des intelligenten Bettes ist die Vereinigung von neuen Technologien und alltäglichen Gegenständen in Form eines Smart Furniture (Vgl. auch Abschnitt 2.4). Besonderes Augenmerk wird darauf gelegt, dass sich das Bett als solches für den Nutzer nicht verändert und die eingebrachte Technologie sich optimal in das Bett integriert. Für den Ansatz zur Erfassung der Bewegungen im Schlaf orientiert sich das intelligente Bett an den bereits bestehenden und in Abschnitt 3 dargestellten kommerziellen und momentan in der Erforschung befindlichen Ansätzen.

Die Hypothese, die dabei im Kern steht, ist, dass das intelligente Bett mit wenig Aufwand und unter Verwendung geeigneter Sensoren (siehe Abschnitt 4.3.2) in der Lage sein soll, die Schlafphasen des Schlafenden, anders als in den Ansätzen in Abschnitt 3.1 und 3.3 beschrieben, ohne direkten Körperkontakt zu erfassen und Informationen unterschiedlicher Interpretationsstufen für andere Anwendungen im Living Place zur Verfügung stellen. Dies kennzeichnet das intelligente Bett als einen Dienstleister (Kontext-Provider) im Gesamtsystem des Living Place und soll anderen Anwendungen ermöglichen, in Abhängigkeit der vom Bett gelieferten Informationen ihrerseits den Kontext, in dem sich der Bewohner befindet, besser einschätzen und so gezielter agieren zu können. Hierfür muss festgelegt werden, welche Dienstleistungen das Bett genau erbringen soll und wieviele Ebenen der Interpretation in einem Bottom-up-Ansatz nur aus den Sensordaten des Bettes realisierbar sind, da die semantische Anreicherung der Daten von Ebene zu Ebene zunimmt. Abbildung 16 zeigt die Visualisierung von Messwerten aus einem Zeitabschnitt mit eingezeichnetem Informationsgehalt zum Anfang und Ende einer Umdrehbewegung. Die Veränderung der Messwerte ist sensortechnisch erkennbar, mit bloßem Auge lassen sich die Zeitpunkte und die Art der Messwertveränderungen bestimmen. Diese Bestimmung sollte folglich auch programmtechnisch durch eine zu entwickelnde Software möglich sein. Denkbar sind an diesem Beispiel vier Ebenen der Interpretation, die sich aufeinander aufbauend wie folgt gliedern:

Ebene 0 - Sensor-Rohdaten zur Interpretation durch andere Anwendungen: Die erfassten Messwerte werden anderen Anwendungen über eine zentrale Vermittlungsarchitektur, in diesem Fall ActiveMQ, zur Verfügung gestellt, damit diese Anwendungen ihrerseits mit den Rohdaten arbeiten können.

Ebene 1 - Erkennbare Bewegung: Anhand der Messwerte wird ermittelt, ob sich der Schlafende bewegt oder kürzlich bewegt hat. Die erkannte Bewegung wird dem Living Place zur Verfügung gestellt.

Ebene 2 - Schlafphasenbestimmung des Schlafenden: Bewegt der Schlafende sich über eine gewisse Dauer oder wurden in einem definierten Zeitraum mehrere Bewegungen erfasst, so ist eine Leichtschlafphase anzunehmen. Auch diese Information wird dem Living Place mitgeteilt.

Ebene 3 - Dauer und Anzahl der Leichtschlaf- und Nicht-Leichtschlafphasen: Über die Nacht hinweg wird ermittelt, wie häufig und wie lange sich der Schlafende in einer Leichtschlafphase mit vermehrter Bewegung der Extremitäten oder Drehbewegungen befunden hat.

Ebene 4 - Bestimmung der Schlafqualität anhand der einzelnen Phasen: Aus den ermittelten Leichtschlafphasen und deren Auftrittshäufigkeit sowie unvermittelter Bewegungen durch unruhigen Schlaf lässt sich ableiten, wieviele Schlafzyklen durchlaufen werden und wie die allgemeine Schlafqualität des Schlafenden über die gesamte Nacht zu bewerten ist.



Abbildung 16: Informationsgehalt einer Drehbewegung anhand der Rohdaten

Dafür werden Sensorwerte erfasst, aufbereitet und im weiteren Verlauf interpretiert, um eine Aussage über den Kontext des Nutzers treffen zu können. Eine schichtweise Interpretation soll dabei dafür sorgen, dass die anderen Anwendungen im Living Place wahlweise uninterpretierte, teilinterpretierte oder vollends interpretierte Daten des Bettes nutzen können.

Dabei wird unter anderem auf die Privatsphäre des Einzelnen besonders großer Wert gelegt, weshalb auf kamerabasierte Verfahren vollständig verzichtet werden soll.

Um auch in Zukunft um weitere Sensoren erweiterbar zu sein, muss die Software für das

intelligente Bett eine Möglichkeit zur Einbindung neuer Sensoren zur Verfügung stellen. Dabei sollen die neuen Sensoren ausgelesen und in das zu implementierende Verfahren zur Bestimmung der Schlafphase integriert werden können.

4.2 Einsatzszenarien

Da der Ansatz der Ambient Intelligence auf Szenarien basiert, wären für das intelligente Bett beispielhaft die nachfolgenden Szenarien denkbar. An diesen Szenarien sind unter Umständen mehrere unabhängige Anwendungen des Living Place beteiligt.

Für den Bezug auf eine Person werden die Szenarien von der durch Mark Weisers Publikationen bekannten *Sal* erlebt (Weiser, 1991).

Bestimmung des optimalen Weckzeitpunkts

- Sal hat morgens um 8⁰⁰ Uhr einen Termin und hat als Weckzeit 7⁰⁰ Uhr angegeben.
- Sal schläft und ihr Schlaf wird kontinuierlich auf Leichtschlafphasen überwacht, in denen sie besonders leicht zu wecken ist.
- Sal hat um 6³⁸ Uhr eine Leichtschlafphase.
- Das intelligente Bett sendet eine Benachrichtigung an eine zentrale Stelle des Living Place, dass sich Sal in einer Leichtschlafphase befindet.
- Sal wird in der Leichtschlafphase geweckt und kann erholter und frischer in den Tag starten als wenn sie aus einer Tiefschlafphase geweckt würde.

Interpretation des gesamten Nachtschlafs

- Sal legt sich abends schlafen.
- Sal schläft schlecht, hat viele unruhige Schlafphasen mit vermehrter Bewegung durch mangelnde Tiefschlafphasen (Lund und Clarenbach, 1995, S. 22 f.).
- Die unruhigen Schlafphasen werden registriert.
- Sal's Schlafverlauf wird analysiert, interpretiert und dem Living Place über eine zentrale Stelle mitgeteilt, so dass sich andere Anwendungen, die auf Ereignisse der Schlafphaseninterpretation „hören“, darauf einstellen können.
- Sal steht morgens auf, der Living Place ist darauf eingestellt, dass Sal schlecht geschlafen hat. Es könnte beispielsweise...
 - ...das Licht gedimmt sein.
 - ...die Musik in der Küche leiser gestellt sein, es werden nur ruhige Titel gespielt.

4.3 Diskussion der Laborausstattung

Die Laborumgebung zur Umsetzung des intelligenten Bettes wurde analysiert und verschiedene Versuchsreihen zur Bestimmung der physikalischen Eigenschaften sowie der Einbettbarkeit unterschiedlicher Materialien evaluiert. Ebenso wurde ein sinnvoller, prototypischer Aufbau diskutiert.

4.3.1 Vergleich verschiedener Schwingkreiserzeuger

Durch die sehr frühzeitig selbst definierte Anforderung an kontaktlose Erfassung der Schlafphasen fiel die Wahl schon zu Beginn auf eine kapazitive Erfassungsmethode (Vgl. auch Abschnitt 2.2). Dabei wird neben einem geeigneten Material zur Verwendung als Sensorfläche und einer Software zum Auslesen und Verarbeiten der Sensordaten auch ein Schwingkreiserzeuger benötigt. Die zur Verfügung stehenden Schwingkreiserzeuger wurden auf ihre Einsetzbarkeit im intelligenten Bett getestet und miteinander verglichen. Verglichen wurden:

- Der EMC von Rode Melder
- Der Cx-1 von RSI
- Das CapSense Toolkit⁷

RSI Cx-1

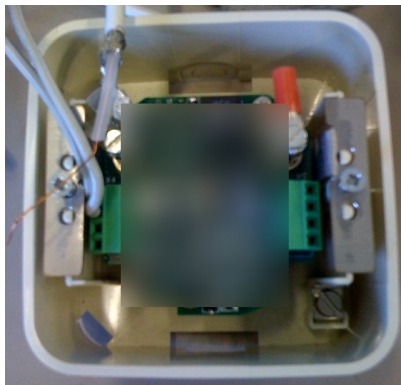


Abbildung 17: Cx-1 von RSI

Der Cx-1 von RSI erlaubt in der vorliegenden Ausstattung ebenfalls nur das Anschließen eines Sensors zur Zeit. Der Messwert dieses Sensors wird durch die aufgespielte Softwareversion laut RSI allerdings nur alle ≈ 100 ms ausgegeben. Die angelegte Frequenz driftet bei dem Produkt von RSI nicht, was auch eine Verwendung des absoluten Messwerts erlaubt. Durch die Bauform kann der Schwingkreiserzeuger in einer handelsüblichen Unterputz-Dose in die Wand eingelassen werden und integriert sich dadurch sehr gut in einen Wohnbereich.

Durch eine bereits in den Schwingkreiserzeuger integrierte Vorbereitung der Daten erhält man Messwerte besonders hoher Güte ohne störende Ausreißer, die auf einen Messfehler hindeuten könnten. Dieser Aspekt sowie die hohe Ausfallsicherheit haben dazu geführt, dass der Cx-1 für das intelligente Bett verwendet wird.

⁷<http://www.capsense.org>

EMC (Rode Melder)



Abbildung 18: EMC von Rode Melder

Der EMC ermöglicht durch die anliegende Spannung eine (bei entsprechend großer Sensorfläche) hohe Reichweite bei der Detektion von Annäherung. In der uns zur Verfügung stehenden Ausstattung ist das Anschließen von nur einem Sensor zur Zeit möglich, dessen Messwerte laut Aussage von Rode Melder alle ≈ 20 ms abgefragt werden. Die angelegte Frequenz driftet, je nach Sensortyp mehr oder weniger, kontinuierlich leicht in einen höheren Frequenzbereich.

CapSense Toolkit

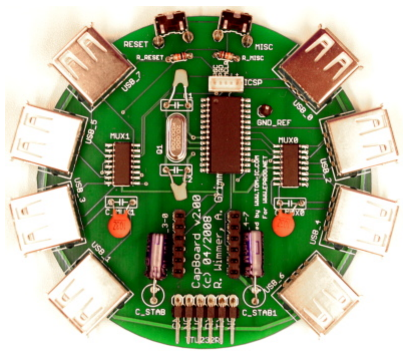


Abbildung 19: CapSense Board (<http://www.capsense.org>)

Das CapSense-Toolkit nutzt intelligente Sensoren. Das verwendete Messverfahren für die Annäherung an eine Sensorfläche funktioniert dabei mittels eines Timer-IC (LMC555), der in Abhängigkeit von der Kapazität eines angeschlossenen Kondensators eine nichtlineare Frequenz bis 3 MHz ausgibt. Dadurch wird ein elektrisches Feld rund um die Elektrode erzeugt, durch welches eine Annäherung von Objekten an die Elektrode detektiert werden kann, da sich die Frequenz des Timers durch eben diese Annäherung verändert (Wimmer u. a., 2007).

Sehr interessant ist beim CapSense-Toolkit der vergleichsweise geringe Anschaffungspreis sowie die komplett quelloffene Firmware sowie das Protokoll.

Leider ist anzumerken, dass nicht immer alle Messwerte vom CapSense-Toolkit an den PC übertragen werden, ebenso wie die Tatsache, dass das gezielte Ein- und Ausschalten einzelner Sensoren nur bedingt funktioniert. Damit erfüllt das CapSense-Toolkit zwar nicht die Anforderungen, die an die Produkte aus der Sicherheitstechnik gestellt werden, reicht aber für den Einsatz im Bereich der Smart Furniture vollkommen aus.

4.3.2 Evaluierung geeigneter Sensoren

Zur Bestimmung der physikalischen Eigenschaften und der Eignung zum Einsatz im intelligenten Bett wurden in enger Zusammenarbeit mit Oliver Dreschke (Dreschke, 2011) verschiedene Versuchsreihen mit unterschiedlichen Materialien und unterschiedlichem Aufbau durchgeführt und anschließend evaluiert. Diese Evaluierung stellt eine Analyse des geplanten Erfassungsverfahrens, der Eignung verschiedener Materialien für die Verwendung in Kombination mit Schwingkreiserzeugern sowie der Möglichkeiten zur Wahrnehmung von Bewegungen im Bett dar. Sie verlief in mehreren Schritten, bei denen

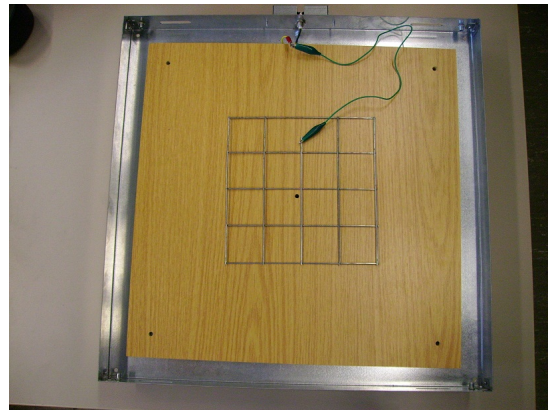
verschiedene Aspekte unterschiedlich intensiv betrachtet wurden. Neben der Flexibilität des Sensors wurde versucht, den Sensor abzuschirmen, damit der Nutzer nicht über Gebühr einem elektrischen Feld ausgesetzt ist. Die Schirmung ist speziell beim Einsatz im Bett wichtig, da sich ein elektrisches Feld negativ auf die Schlafqualität auswirken kann und des Weiteren zu Akzeptanzproblemen bei den Nutzern führen könnte. Als eine wichtige Voraussetzung stellte sich die Verwendung von geschirmten Kabeln heraus, um unverfälschte Messergebnisse zu erhalten und den Drift im Schwingkreis zu minimieren.

Aufbau 1 - Originale Plattenelektrode von Rode Melder

Die von Rode Melder zur Verfügung gestellte Plattenelektrode (Abbildung 20.1) stammt aus dem Bereich der Sicherheitstechnik. Die Bauform ist recht stabil und zur Montage hinter Gemälden oder in Wänden geeignet. Die Elektrode ist auf nicht leitenden Abstandhaltern in einem geerdeten Rahmen aus Stahlblech montiert, der eine Schirmung zur Rückseite und den Seiten bewirkt. Mit beiden kommerziellen Schwingkreiserzeugern lieferte die Elektrode gute Messwerte (Siehe Anhang A.1), war äußerst sensibel und driftete nur minimal. Durch die unflexible, große Bauform ist die Elektrode für eine Verwendung im intelligenten Bett jedoch nicht geeignet.



20.1: Plattenelektrode von Rode Melder (Aufbau 1)



20.2: Modifizierte Elektrode (Aufbau 2)

Abbildung 20: Elektrode von Rode Melder

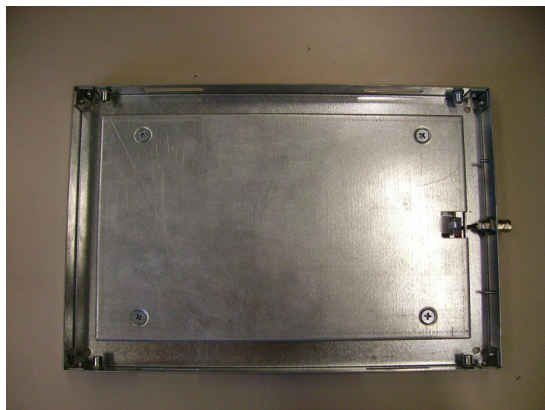
Aufbau 2 - Modifizierte Plattenelektrode von Rode Melder

Um ein Gefühl dafür zu entwickeln, welche Materialien für die Verwendung als Sensor im Schwingkreis einsetzbar sind, wurde die Elektrode aus dem Schirmungsrahmen entfernt und eine Holzplatte eingepasst. Dank dieser Veränderungen konnten verschiedene Materialien in dem geschirmten Rahmen getestet werden. In Abbildung 20.2 wurde ein Stück Estrichmatte aus dem Baumarkt auf ihre Tauglichkeit überprüft, aber auch Lochbleche, Kaninchendraht und elektrisch leitfähige Stoffe (Vgl. Abschnitt 4.3.2) wurden erprobt. Dabei war festzustellen, dass gelochte Materialien (z.B. Estrichmatte) neben der reinen Annäherung auch Bewegungen bei auf der Elektrode befindlichen Gegenständen bzw. Personen erfassen können (Anhang A.2). Durch die Verwendung von anderen Ma-

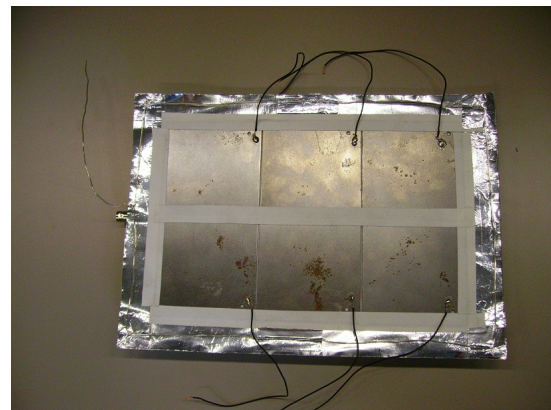
terialien wäre die Herstellung einer Elektrode nach der hier gezeigten Bauform günstig, der Nachteil der unflexiblen Bauform besteht weiterhin.

Aufbau 3 - Abschirmmatrix über einer Plattenelektrode

Ein weiterer Ansatz war der Einsatz einer Abschirmmatrix, um eine einzelne Plattenelektrode von Rode Melder (Abbildung 21.1) in mehrere Sensorfelder zu segmentieren, um bei der Erfassung der Daten eine höhere Auflösung zu erreichen. Dazu wurden mehrere mit einem Kabel versehene Metallplatten berührungsfrei nebeneinander auf eine nicht leitende Trägerplatte aufgebracht und ohne leitenden Kontakt auf der Plattenelektrode befestigt (Abbildung 21.2). Die einzelnen Metallplatten werden geerdet und schirmen somit die Elektrode ab. Wird nun bei einer Metallplatte der Kontakt zur Erdung entfernt, so kann die darunter liegende Elektrode in diesem Segment eine Annäherung detektieren. Das Prinzip der segmentweisen Schirmung der Elektrode würde die Erfassung von Annäherung bzw. Bewegung über der Elektrode in einzelnen oder gleichzeitig in mehreren Segmenten ermöglichen. Durch das Hinzu- bzw. Abschalten von Segmenten ergaben sich Spitzen in den Messwerten (Anhang A.3), die im späteren Verlauf zu Fehlinterpretationen führen könnten.



21.1: Rode Melder Plattenelektrode (klein)



21.2: Abschirmmatrix

Abbildung 21: Elektrode von Rode Melder (Aufbau 3)

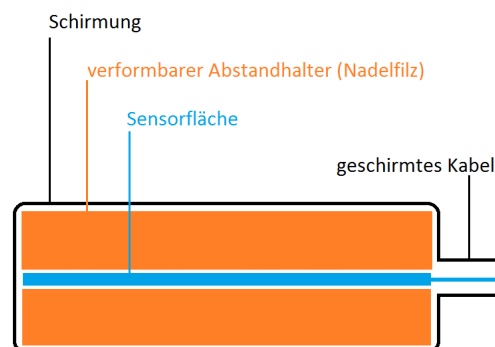
Aufbau 4 - Sensor in Sandwich-Bauweise

Der Ansatz mit der Abschirmmatrix (Abschnitt 4.3.2) ergab, dass sich die Messwerte auch durch die Annäherung der geerdeten Abschirmung an die im Schwingkreis befindliche Fläche der Plattenelektrode (also durch Druck) veränderten. Diese Eigenschaft war der Grundgedanke für den Ansatz eines eigenen Sensors in sogenannter Sandwich-Bauweise (Abbildung 22.1). Dabei wird die Sensorfläche zu beiden Seiten mit verformbaren, nicht leitenden Abstandhaltern (in diesem Fall mit Nadelfilz) gegenüber der umschließenden Schirmung verkleidet (Abbildung 22.2). Wird nun von außen Druck auf den Sensor ausgeübt, so verformt sich der Nadelfilz und die Schirmung wird der Sensorfläche angenähert. Der Sensor reagiert äußerst sensibel auf von außen einwirkenden Druck, bereits das Ablegen eines Stifts auf der Sensorfläche wird deutlich registriert,

während Annäherung an den Sensor durch die komplette Schirmung nicht mehr erfasst wird. Für den Einsatz im intelligenten Bett ist dieser Sensor allerdings zu sensibel und die Bauweise ist aufgrund der Verwendung von Aluminiumfolie für die Schirmung und die Sensorfläche zu anfällig für äußere mechanische Belastungen. Eine Messreihe zu Versuchen mit diesem Aufbau findet sich in Anhang A.4.



22.1: Gesamtansicht



22.2: Aufbau

Abbildung 22: Sensor in Sandwich-Bauweise & Aufbau (Aufbau 4)

Aufbau 5 - Sensor in Sandwich-Bauweise 2

Der durch die erste Version des Sensors in Sandwich-Bauweise aufgetretene Nachteil der starken Anfälligkeit gegenüber mechanischen Einwirkungen von außen führte zu einer stabileren Weiterentwicklung mit der gleichen Funktionsweise (Abbildung 23.1 - 23.3). Die Sensorfläche (1mm Lochblech) ist dabei wieder zwischen zwei Lagen Nadelfilz als verformbarer, nicht leitender Abstandhalter eingefasst und wird von 1mm starkem Stahlblech geschirmt. Für die nötige Stabilität sorgen je eine Holzplatte auf der Ober- und Unterseite. Die gesamte Konstruktion ist miteinander verschraubt und kann so modulweise eingesetzt werden. Durch die Verwendung von Stahlblech und Holzplatten ist der entwickelte Sensor zum Einen aufgrund der großen und unflexiblen Bauform, zum Anderen durch das hohe Gewicht im intelligenten Bett nicht einsetzbar. Für die im Kontext des Living Place geplante Entwicklung einer Sturzerkennung (Siehe auch (Teske, 2011a) und (Teske, 2011b)) könnte dieser Sensor jedoch hilfreich sein. In Anhang A.5 wird eine Messreihe zu diesem Aufbau dargestellt.

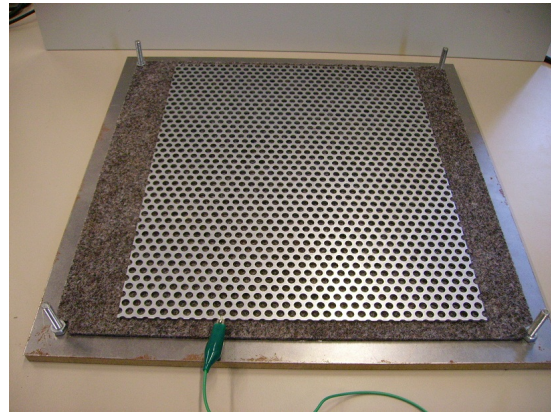
Aufbau 6 - Modifizierter Lattenrost

Nachdem bereits verschiedene Varianten analysiert wurden, lag nun der Fokus auf einen Sensor, der sich besser in ein Bett integrieren lässt. Durch die geplante Indirektionsstufe der kontaktlosen Erfassung und Schirmung gegenüber dem Nutzer entstand die Idee, die Biegung des Lattenrosts mit einem geeigneten Sensor abzugreifen und für die Ermittlung der Schlafphase zu verwenden.

Mit der Idee, den Sensor mit einem bereits bestehenden Bauteil des Bettes zu verbinden wurde in Aufbau 6 ein Sensor analysiert, der eine Modifikation einer einzelnen Latte



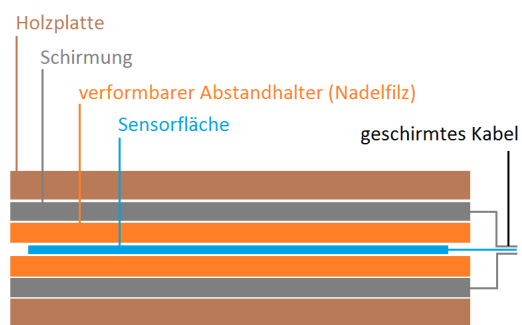
23.1: Gesamtansicht



23.2: Gesamtansicht (offen)



23.3: Übersicht Aufbau

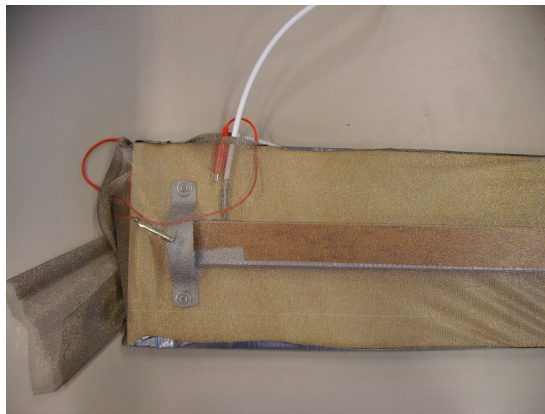


23.4: Aufbau

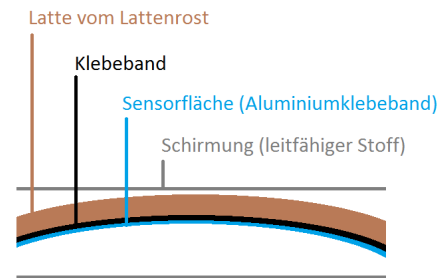
Abbildung 23: Sensor in Sandwich-Bauweise 2 & Aufbau (Aufbau 5)

des Lattenrosts darstellt (Abbildung 24.1). Im abgebildeten Testaufbau wurde die Latte auf einem Holzbrett fixiert, um damit einen intakten Lattenrost zu simulieren. Auf die konkave Unterseite der Latte ist eine Schicht aus elastischem Klebeband aufgebracht, auf der wiederum eine Schicht Aluminiumklebeband⁸ aufgebracht ist (Abbildung 23.4). Um den gesamten Testaufbau wurde elektrisch leitfähiges Gewebe (Abschnitt 4.3.2) gespannt, um eine Detektion der Annäherung an die Latte auszuschließen. Wird die Latte nun durchgebogen, so verändert sich auch die Biegung des aufgetragenen Aluminiumklebebands. Diese Veränderung der Biegung ist in den Messergebnissen deutlich sichtbar (siehe Messreihe in Anhang A.6). Ein großer Vorteil dieses Sensors ist neben der einfachen Integration in eine Wohnumgebung, dass die Latte des Lattenrosts nach dem Durchbiegen immer in ihre ursprüngliche Position zurückkehrt. Diese Eigenschaft ist wichtig, um eine Verfälschung der Messwerte durch anhaltende Verformung zu vermeiden.

⁸Aluminiumklebeband aus dem Baumarkt (findet bei der Verarbeitung von aluminiumkaschierter Dämmwolle sowie im Heizungsbau Verwendung)



24.1: Draufsicht



24.2: Übersicht Aufbau

Abbildung 24: Modifizierter Lattenrost & Aufbau (Aufbau 6)

Aufbau 7 - Verbesserung des entwickelten Lattenrost-Sensors

Der Sensor in Form einer modifizierten Latte des Lattenrosts war noch nicht ausgereift genug, um im Prototyp des Bettes verwendet zu werden. Die über und unter der Latte angebrachte Schirmung aus leitfähigem Gewebe verformte sich bei mehrfachem Durchbiegen des Lattenrosts nachhaltig und führte so zu stark schwankenden Messergebnissen.

Um die Schirmung besser aufzubringen, wurde diese ebenfalls aus leitfähigem Aluminiumklebeband gefertigt. Dadurch entstand der in Abbildung 25 von oben links nach unten rechts dargestellte, siebenschichtige Aufbau aus Aluminiumklebeband und Isolierband. Der Anschluss des Sensors erfolgt nach wie vor über ein Koaxialkabel.

Der verbesserte Aufbau hat den Vorteil, dass die Schirmung nun immer den gleichen Abstand zur Sensorfläche behält und der aus der locker aufliegenden Schirmung resultierende Seiteneffekt behoben wurde. Durch das Aufbringen des Sensors auf der Unterseite des Lattenrosts verschwindet dieser nun auch optisch „unter“ dem Bett und fügt sich noch besser als bisher in ein bestehendes Wohnumfeld ein. Diese Integration in den Wohnbereich sowie die komplette Schirmung wirken eventuellen Akzeptanzproblemen späterer Nutzer entgegen. Zum einen trägt die neue Bauform gar nicht mehr gegenüber einem Standard-Lattenrost auf, zum anderen wird Argumentationen gegen ein den Körper während der Nacht umgebendes elektrisches Feld durch die komplette Schirmung präventiv entgegengewirkt.

Die ersten Ergebnisse waren dabei sehr vielversprechend, wie auch in Anhang A.7 zu sehen ist.

Elektrisch leitfähige Stoffe

Durch die Recherche im Internet ergab sich ein Kontakt zur Firma Biologa.net⁹, die Produkte zur Abschirmung von nieder- und hochfrequenten Feldern anbietet. Darunter befinden sich auch Gewebe und Gewirke mit einer sehr weichen Haptik, die aufgrund einer vollflächigen Versilberung der Einzelfasern elektrisch leitfähig sind. Versuche, diese Stoff-

⁹<http://biologa.net>

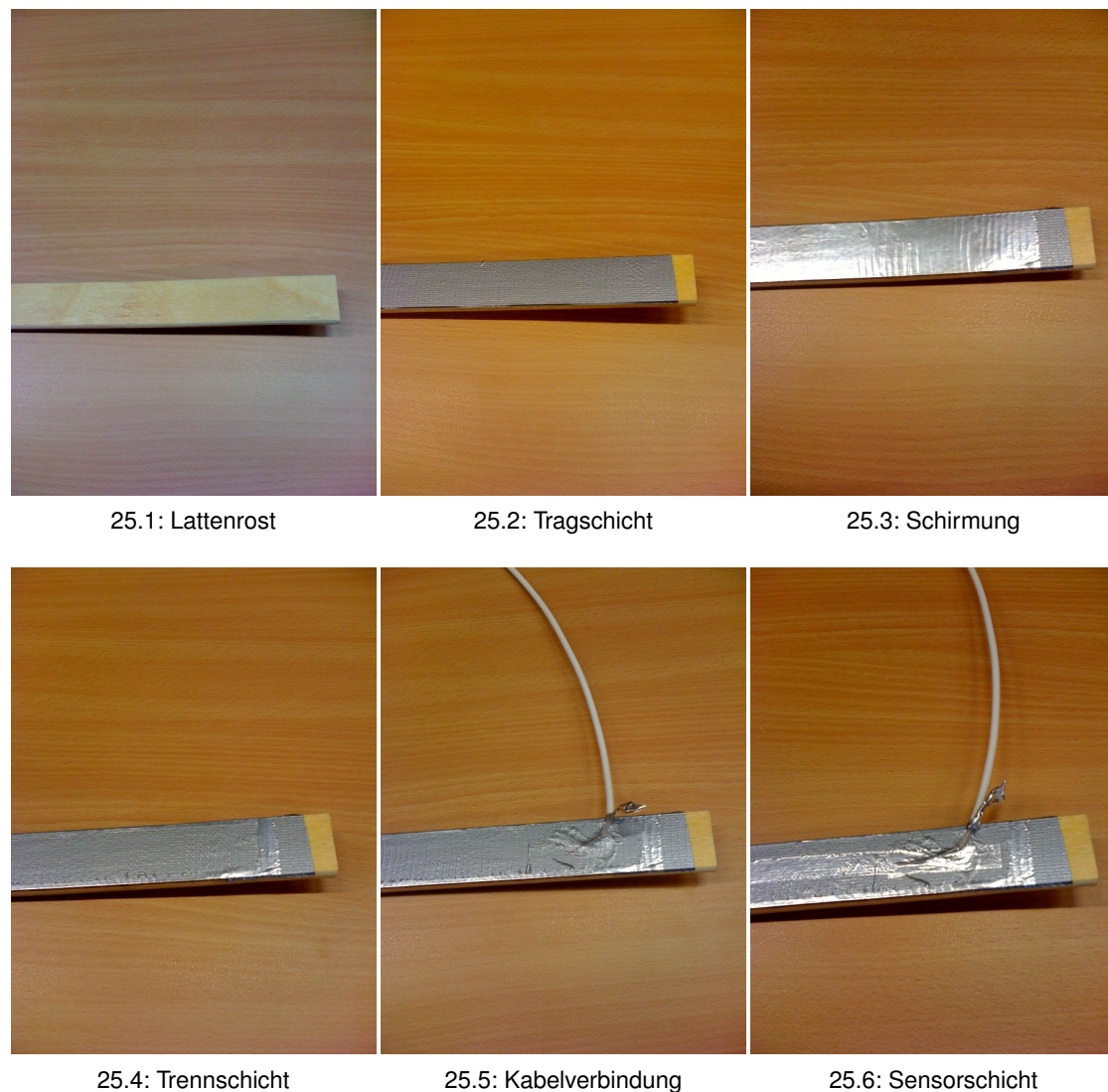


Abbildung 25: Entwickelter Lattenrost-Sensor (Aufbau 7)

fe für einen Sensor im intelligenten Bett zu verwenden, scheiterten an den Verformungseigenschaften der Stoffe, da diese nicht in die Ausgangsposition zurückfielen und damit die Messergebnisse nachhaltig verfälschten. Ebenso führte eine Verschiebung der einzelnen Stoffschichten gegeneinander zu verfälschten Messergebnissen. Zur Schirmung der Sensoren gegenüber dem Nutzer sind die Stoffe jedoch sehr gut geeignet, da ihre Verarbeitung der von normalen Stoffe in nichts nachsteht und dadurch eine Integration in eine Wohnumgebung im höchsten Maß gegeben ist (Details zur Verwendung leitfähiger Textilien sowie deren Verarbeitung werden in der Bachelorarbeit von Ann-Kathrin Weiss (Weiss, 2011) näher erläutert.).

Fazit der Evaluierung

Durch die Evaluierung der verschiedenen Varianten des Sensoraufbaus ergaben sich aufbauspezifische Vor- und Nachteile, anhand derer eine Entscheidung für den zu ver-

wendenden Sensor getroffen wurde. Die Kriterien für eine Entscheidung sind die allgemeine Sensorik (Sensibilität, Stabilität der Messwerte etc.), die Verarbeitung (wie wird der Sensor hergestellt, wie lässt er sich verarbeiten etc.) sowie die entstehenden Kosten. Der bestmögliche Trade Off aus diesen Anforderungen soll im intelligenten Bett Verwendung finden.

Zur detaillierten Bewertung der verschiedenen Aufbauten wurden wichtige Eigenschaften herausgearbeitet, nach denen die Bewertung erfolgen soll. Diese Eigenschaften lassen sich dabei in die aus der nachfolgenden Tabelle 1 ersichtlichen Bereiche unterteilen. Bewertet werden folgende Eigenschaften:

Schirmung:

Die Schirmung des Sensors gegenüber äußeren Störeinflüssen durch andere elektrische Felder und ungewollter Detektion von Bewegungen und Annäherung an die Sensorfläche.

Drift:

Das Driftverhalten der Messwerte, die mit dem Sensor erfasst werden. Wichtig hierbei ist die Stärke des Drifts, da sich eine stärkere Drift verfälschend auf Berechnungen wie beispielsweise Varianz oder Mittelwert auswirkt.

Sensibilität:

Die Sensibilität des Aufbaus beschreibt die Schwelle, ab der man in den Messwerten subjektiv von einer deutlichen Erfassung einer Annäherung oder Druckbelastung des Sensors sprechen kann.

Datenqualität:

Die Qualität der erfassten Rohdaten in Bezug auf Erkennbarkeit einer Annäherung oder Druckbelastung. Dazu zählt auch die Gleichmäßigkeit des Messwerts in Ruheposition, Rückkehr zum Ausgangswert nach einem Ausschlag sowie der Rauschfaktor.

Integration:

Die Möglichkeit, den Sensor in eine intelligente Wohnumgebung, im Speziellen in ein Bett, zu integrieren, so dass dieser sich dem Blickfeld des Nutzers vollständig entzieht.

Flexibilität:

Die mögliche Anpassbarkeit des Sensors an neue Gegebenheiten. Dies umfasst eine neue oder andere Positionierung des Sensors sowie dessen Verwendung in einem anderen Kontext.

Abschirmung:

Die Abschirmung des Aufbaus gegenüber dem Nutzer, um das elektrische Feld vom Nutzer fern zu halten und Akzeptanzproblemen vorzubeugen.

Materialkosten:

Die Kosten für das Material, das benötigt wird, um den Sensor herzustellen, oder im Fall von fertig produzierten Sensoren der Kaufpreis des Sensors.

	Aufb. 1	Aufb. 2	Aufb. 3	Aufb. 4	Aufb. 5	Aufb. 6	Aufb. 7		
Sensorik									
Schirmung	-	-	-	o	o	+	++		
Drift	++	-	--	-	-	+	++		
Sensibilität	++	+	o	++	o	++	++		
Datenqualität	++	o	--	-	-	o	++		
Verarbeitung									
Integration	--	--	-	--	--	++	++		
Flexibilität	++	++	++	+	+	--	--		
Abschirmung	--	--	--	o	o	++	++		
Kosten & Aufwand									
Materialkosten	--	++	--	+	o	-	++		
Herstellungsaufwand	++	+	o	--	--	o	+		
Ergebnis	o (2,7)	o (3,0)	- (3,9)	o (3,2)	- (3,6)	+	(2,4)	+	(1,6)
[++] sehr gut (1), [+] gut (2), [o] mittelmäßig (3), [-] schlecht (4), [- -] sehr schlecht (5)									

Tabelle 1: Bewertung evaluierter Sensoren

Herstellungsaufwand:

Der Aufwand und die Arbeit, den Sensor herzustellen und die benötigten Materialien zu verarbeiten.

Aus der vorgenommenen Bewertung geht hervor, dass der in Aufbau 7 (Abschnitt 4.3.2) evaluierte Sensor im intelligenten Bett Verwendung finden wird.

4.3.3 Prototyp des Bettes

Um die Software und die Sensoren so realitätsnah wie möglich testen zu können und nicht immer vor Ort im Living Place arbeiten zu müssen ist ein Prototyp des intelligenten Bettes unerlässlich. Aus diesem Grund wurde ein Testsetup im Projekttraum aufgebaut, bei dem es sich um nicht viel mehr als einen Lattenrost und eine Matratze handelt.

In diesen Prototyp wurden drei Sensoren integriert, deren Positionierung in Abschnitt 4.3.3 näher erläutert wird. Mit diesem Prototyp können im Live-Modus die Messwerte am Computer angezeigt und so die Sensoren bezüglich der Sensibilität getestet werden.

Abbildung 26: Prototyp im Projekttraum

Positionierung der Sensoren

Die unter Punkt drei der nebenstehenden Abbildung gezeigte Liegeposition zeigt zum einen die optimale Liegeposition mit einer geraden Wirbelsäule, zum anderen aber auch zwei Körperschwerpunkte auf Schulter- und Hüfthöhe. An eben diesen Schwerpunkten wurden zwei Latten des Lattenrosts mit dem modifizierten Sensor aus Abschnitt 4.3.2 versehen. Bewegt sich ein Mensch im Bett, oder dreht er sich um, so findet an diesen beiden Schwerpunkten, die sich am tiefsten in die Matratze drücken, die stärkste Belastung des Lattenrosts statt. Diese Tatsache wird genutzt, um eine möglichst starke Auslenkung der Messwerte der Lattenrost-Sensoren zu erreichen. Eine dritte Latte wurde auf Höhe der Knie beziehungsweise der Unterschenkel modifiziert, um auch Bewegungen der Beine zusätzlich zu den Bewegungen an den beiden Körperschwerpunkten erfassen zu können. Mit dieser Aufteilung der Sensoren wurden bereits erste, sehr zufriedenstellende Versuche zur Bewegungserkennung durchgeführt (Abbildung 32 zeigt die Veränderung der Messwerte der einzelnen Sensoren bei einer Drehung auf der Matratze um 90° von der Seitenlage in die Rückenlage).

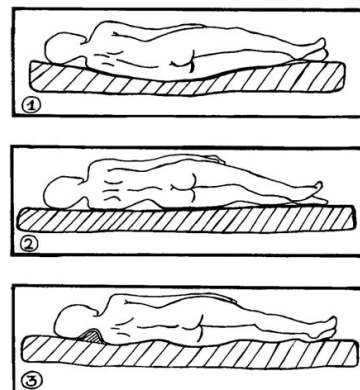


Abbildung 27: Liegepositionen
(Quelle: www.wickerklinik.de)

4.4 Aufgaben und Ziele



Abbildung 28: Bett im Living Place
(Quelle: (LPHH Homepage, 2011))

Nach der Analyse ergaben sich die in den nachfolgenden Kapiteln umzusetzenden Aufgaben zur Verarbeitung der über die Sensoren erfassten verschiedenen Datenströme des intelligenten Bettes. Wird eine Aufteilung der Sensoren im Bett gewählt, wie sie in Abschnitt 4.3.3 beschrieben wurde, dann ergeben sich sechs Datenströme (einer je Sensor, drei Sensoren je Lattenrost). Diese Datenströme müssen vorverarbeitet und anschließend kohärent interpretiert werden, um einen Rückschluss auf die Schlafphase des Nutzers zu erhalten. Dadurch, dass die Ströme alle auf einem Computer zusammenlaufen und auf diesem auch

verarbeitet werden sollen, entfällt die Notwendigkeit der Synchronisation verschiedener Uhren. Eine weitere Herausforderung wird die performante Umsetzung der Interpretation der verschiedenen Datenströme sowie eine softwaretechnisch saubere und gut um neue Sensoren erweiterbare Anwendung sein.

5 Design

Nachdem in Kapitel 4 die Analyse der Laborausstattung sowie die Idee des intelligenten Bettes vorgenommen wurde, beschäftigt sich dieses Kapitel nun mit dem konkreten Design der Anwendung.

5.1 Vorgehen bei der Verarbeitung der Sensordaten

Eine Interpretation von Sensordaten ist nur möglich, wenn die Rohdaten vorbereitet und im Rahmen einer Vorverarbeitung markante Merkmale (Features) aus den Daten extrahiert wurden. Abbildung 29 zeigt das methodische Vorgehen zur Verarbeitung der Sensordaten mit den einzelnen Stufen der Bereinigung bis hin zu den aufeinander aufbauenden Ebenen der Interpretation. Nun muss eine geeignete Methode gefunden werden, diese Interpretation, aufbauend auf dem Filtering und der Feature Extraction, vorzunehmen. Dafür werden im weiteren Verlauf in Abschnitt 5.3 drei unterschiedliche Ansätze genauer betrachtet und das Ergebnis dieser Betrachtung dargestellt.

Filtering

Die Rohdaten müssen zuerst gefiltert werden, um eventuelle Ausreißer, Messfehler oder verlorene Messwerte auszugleichen (Abbildung 29 Übergang 1). Diese Filterung lässt sich beispielsweise über einen Tiefpass, die Nutzung eines Mittelwerts, der Streuung, der Standardabweichung oder der Varianz über ein *floating window* der Messwerte realisieren. Nach der Filterung erhält man am Beispiel von Abbildung 30, im Vergleich zu den Ausgangsdaten (blaue Kurve), eine geglättete Kurve (rote Kurve), anhand der sich markante Merkmale leichter extrahieren lassen. Diese Filterung hat auch zur Folge, dass die Erfassung der Messwerte von Körpereigenschaften des Menschen, wie beispielsweise dem Gewicht, losgelöst wird, da nur noch mit (stochastisch) vorverarbeiteten, relativen Werten gearbeitet wird. Ein erster Ansatz zum Filtering ist die Verwendung stochastischer Verfahren wie Mittelwert, Standardabweichung und Varianz.

Feature Extraction

Nach dem Filtering werden die Features aus den Daten extrahiert, anhand derer die Interpretation und Klassifikation ausgeführt werden soll. Dabei ist die Qualität und die Wahl der Features die Grundlage für die Qualität der späteren Interpretation, weshalb die Wahl der Features wohl überlegt sein sollte.

Je nach Sensortyp und Verwendungszweck der Sensordaten wären mögliche Features

- das Überschreiten eines Schwellwerts,
- die Anzahl der Überschreitungen eines Schwellwerts für ein Zeitintervall,
- die Streuung oder die Varianz der Messwerte oder
- in einer Ableitung der Sensorwerte nach der Zeit die Änderung des Sensorwerts (ansteigend oder abfallend) über die Zeit.

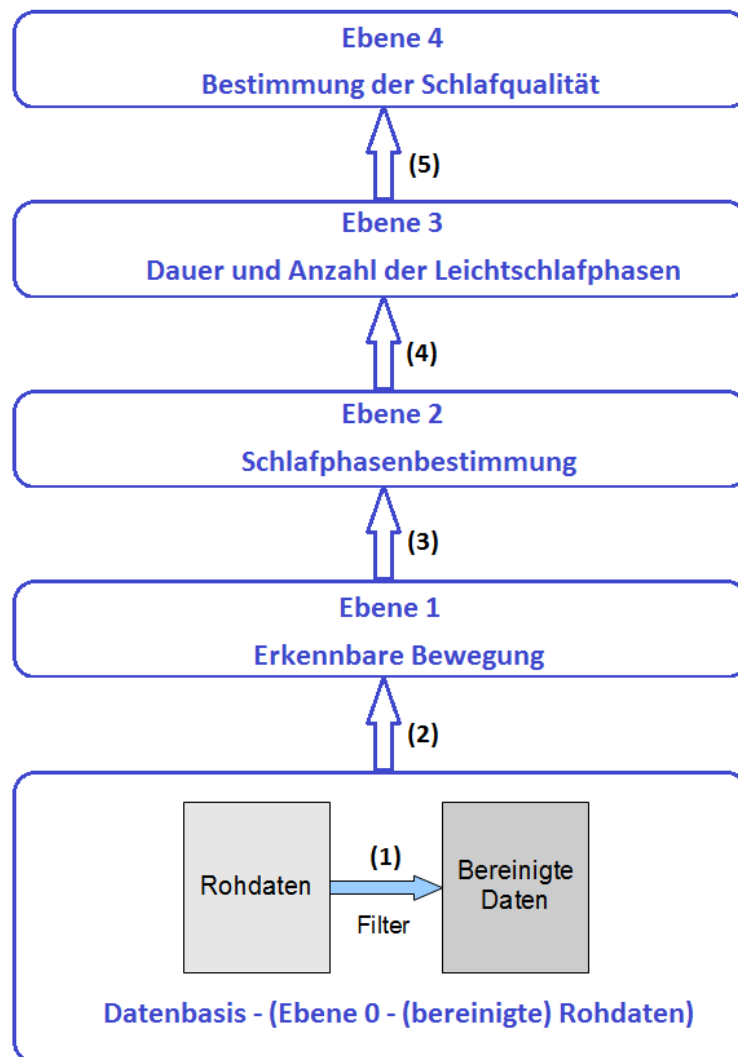


Abbildung 29: Verarbeitung der Sensordaten

In der Abbildung 30 wurde als beispielhaftes Feature das Überschreiten eines gewissen Schwellwerts (dargestellt durch die grüne Gerade) gewählt. Für die verschiedenen möglichen Ebenen der Interpretation werden die Ergebnisse der darunter liegenden Ebene als Feature mit einbezogen. Unter Umständen bilden die Ergebnisse der darunter liegende Ebene die einzige Informationsquelle (Datenbasis) zur Interpretation auf einer höheren Ebene.

5.2 Interpretationsebenen und deren Bedeutung

Die im Kapitel 4 angesprochenen Ebenen der Dateninterpretation bauen aufeinander auf (Vgl. Abbildung 29) und nutzen die Informationen der jeweils unter ihnen liegenden Ebene. Nun muss betrachtet werden, was genau die aus den Messwerten zu ziehenden Informationen sind, die zur Interpretation in den einzelnen Ebenen nötig sind. Je Ebene

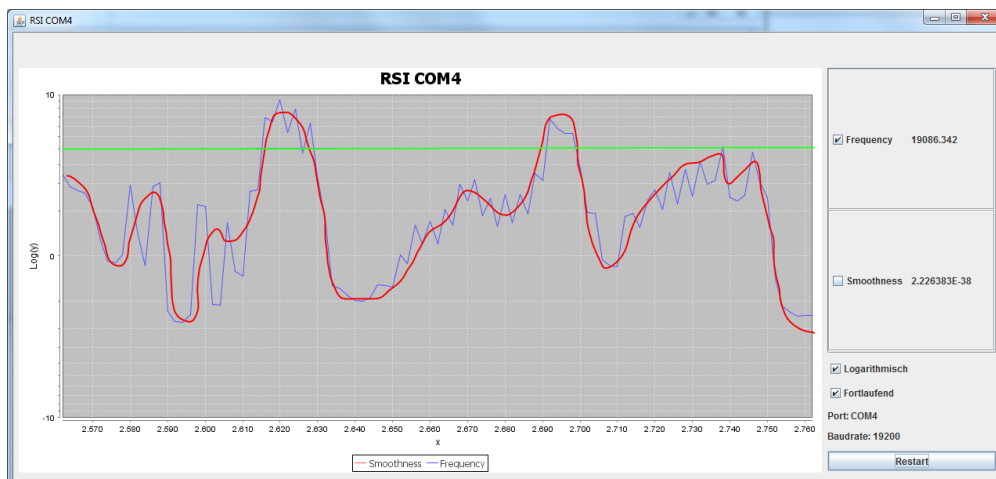


Abbildung 30: Darstellung der Sensordaten

nimmt dabei die Aussagekraft einer Interpretation zu.

In der Beschreibung der einzelnen Ebenen der Interpretation ist bewusst von *Leichtschlafphasen* und *Nicht-Leichtschlafphasen* die Rede, da sich mit den gegebenen Mitteln nur Leichtschlafphasen anhand der vermehrten Bewegung des schlafenden Nutzers erkennen lassen werden. Für eine Unterscheidung der Nicht-Leichtschlafphasen wäre eine Unterstützung durch ein EEG (Vgl. Abschnitt 3.2) notwendig.

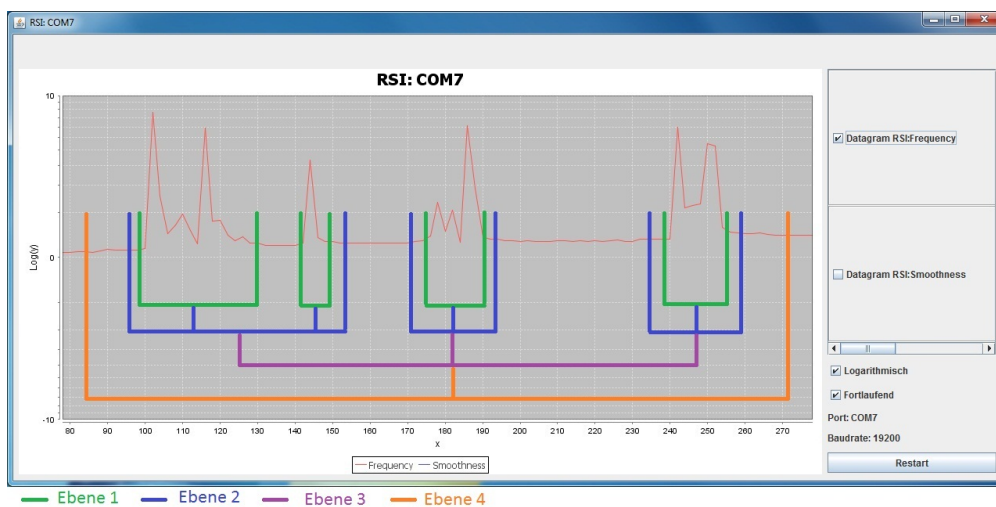


Abbildung 31: Beispiel für die verschiedenen Ebenen der Interpretation

Ebene 0 - Rohdaten

Die erfassten Rohdaten haben keine große Aussagekraft und werden erfasst, um anderen Anwendungen zur Verfügung gestellt werden zu können. Dabei hilft das Filtering, eine gewisse Vorbereitung zu erlangen, mit der eventuelle Ausreißer abgeschwächt werden. Da Ebene 0 die Repräsentation der Rohdaten der einzelnen Sensoren und so-

mit die unterste Ebene darstellt, kann nicht auf andere Ebenen als Interpretationsgrundlage zurückgegriffen werden.

Ebene 1 - Erkennbare Bewegung

Auf Basis der bereinigten Rohdaten aus Ebene 0 lässt sich, beispielsweise durch die Überschreitung eines Schwellwertes durch die Standardabweichung, Streuung oder Varianz, eine Bewegung erkennen. Dabei ist die Anzahl der Messwerte, über die die Streuung oder das Ergebnis einer vergleichbaren Berechnung bestimmt wird, ausschlaggebend dafür, ob der Schwellwert schnell (bei wenigen Elementen) oder langsam (bei vielen Elementen) überschritten wird. Im konkreten Fall des intelligenten Bettes wird eine Bewegung angenommen, wenn die Standardabweichung über die letzten 25 Messwerte größer 1 ist. Wichtig hierbei ist noch die Definition, wann eine Bewegung beginnt und wann eine Bewegung endet. Der Beginn einer Bewegung ist durch das Überschreiten eines festgelegten Wertes definiert und dauert so lange an, wie der Wert nicht wieder unter diesen Schwellwert sinkt. Ein kurzzeitiges Absinken unter den Schwellwert für eine Zeit kleiner als 30 Sekunden wird, sofern der Schwellwert danach wieder überschritten wird, nicht als Bewegungsende, sondern vielmehr als eine Pause in einer Bewegungsphase gewertet. In Abbildung 31 werden durch die grünen Markierungen beispielhaft erkennbare Bewegungsphasen dargestellt, wie sie während einer Leichtschlafphase auftreten können.

Ebene 2 - Schlaf-/Bewegungsphasenbestimmung

Aufbauend auf der Erkennung von Bewegungen des Nutzers kann eine Einordnung der momentanen Schlafphase in Leicht- bzw. Nicht-Leichtschlafphasen vorgenommen werden. Ein Indiz dafür ist das mehrfache Überschreiten eines Schwellwerts in einem bestimmten Zeitintervall (oder auch mehrfache Bewegung in einem Zeitfenster). Dafür werden die erkannten Bewegungen der Ebene 1 als Datenbasis herangezogen. Abbildung 31 zeigt anhand der blauen Markierungen eine beispielhafte Einordnung der Messreihe nach Schlafphasen. Hierbei sind die markierten Bereiche angenommene Leichtschlafphasen.

Ebene 3 - Dauer und Anzahl der Leichtschlafphasen

Mit dem Wissen aus Ebene 2 über die Schlafphasen lässt sich in Ebene 3 eine Aussage über die Dauer und Anzahl der Leicht- bzw. Nicht-Leichtschlafphasen treffen. Anhand der ermittelten Dauer und Anzahl der unterschiedlichen Schlafphasen lässt sich damit bestimmen, wieviele Schlafzyklen während des Schlafens vom Nutzer durchlaufen wurden. Anhand der in Abbildung 31 dargestellten Leichtschlafphasen (blaue Markierung) kann in Ebene 3 deren Häufigkeit und Dauer bestimmt werden (violette Markierung).

Ebene 4 - Schlafqualität

Als oberste Ebene der Interpretation der Sensordaten soll eine beispielhafte Aussage über die Schlafqualität möglich sein. Dafür müssen selbst definierte Rahmenbedingungen über den Nachtschlaf hinweg erfüllt werden. Diese könnte lauten:

- Mindestens fünf komplette Schlafzyklen in einer Nacht (Vgl. Abschnitt 2.1)
- Mindestens 80 % Nicht-Leichtschlafphasen
- Weniger als 10 detektierte Bewegungsphasen in der Nacht

Auf Basis der in Abbildung 31 ermittelten Dauer und Häufigkeiten der Leichtschlafphasen und unter Abgleich mit den hinterlegten Erkenntnissen über einen guten Schlaf kann die Qualität des Schlafs ermittelt werden (orangene Markierung).

5.3 Interpretationsverfahren

Die Abbildung 32 zeigt die anfallenden Rohdaten der drei jeweils auf Schulter-, Hüft- und Kniehöhe im Prototyp des Bettes eingebauten Sensoren, dargestellt durch die in Zusammenarbeit mit Oliver Dreschke entwickelte Visualisierungssoftware (Dreschke, 2011). Unverarbeitet haben diese Rohdaten wenig Aussagekraft und ein Umdrehereignis ist anhand der Kurven nicht bestimmbar. Diese Rohdaten müssen zur weiteren Verarbeitung einer Vorverarbeitung zur Bereinigung unterzogen werden, um beispielsweise korrupte Datensätze zu entfernen und Schwankungen zu glätten. Liegen diese bereinigten Daten vor, so kann man verschiedene Ansätze verfolgen, um aus diesen Daten Informationen über die Schlafphase des Nutzers zu gewinnen. Diese Ansätze haben spezifische Vor- und Nachteile, die nachfolgend erläutert werden.

Variante 1 - „Hard coded“

Bestimmte Bewegungen im Bett (Umdrehbewegung, Bewegung der Beine etc.) erzeugen eine bestimmte Abfolge von Messwerten, vergleichbar mit einer Art Choreographie. Diese Abfolgen von Messwerten für bestimmte Bewegungen beziehungsweise Ereignisse werden als Vergleichsmuster hinterlegt und die aktuellen Messwerte gegen diese Muster verglichen und auf Übereinstimmungen überprüft. Dabei könnte man als Muster die Veränderung der Sensorwerte zueinander wählen.

Ein Risiko besteht in dem individuellen Schlafverhalten des Einzelnen, wodurch sich durch unterschiedliche Positionen, die beim Schlafen im Bett eingenommen werden, individuelle Veränderungen der Messwerte je Schlafposition ergeben könnten. Man könnte versuchen, diese personen- und gewichtsabhängigen Messwerte mittels Normalisierung zu vereinheitlichen, das Risiko der Vielzahl verschiedenster Messwertkombinationen, die es als Vergleichsmuster abzubilden gilt, bleibt jedoch bestehen. Um dieses Risiko zu reduzieren, könnten von außen einstellbare Parameter als *Stellschrauben* für die Schlafdynamik des Einzelnen dienen.

Variante 2 - Modellbasiert

Bei dem modellbasierten Ansatz handelt es sich um einen analytischen Ansatz, bei dem sowohl das Bett als auch die schlafende Person simuliert werden. Zur Vereinfachung sollte die Person jedoch nicht menschnah, sondern als Skelettmodell simuliert werden. Anhand des modellierten Bettes und der modellierten Person werden Bewegungen und Schlafphasen simuliert und die Simulationsergebnisse in einer Modell-Datenbank abgelegt. Dies hat den Vorteil, dass bei neuen Simulationsergebnissen nur die Modell-Datenbank ausgetauscht werden muss, der Rest der Anwendung hingegen unangetastet bleibt. Werden nun echte Messwerte erfasst, so kann man diese mit den Werten in der Modell-Datenbank vergleichen und auf das passende Modell (oder das Modell mit der größten Übereinstimmung) matchen.

Dieser Ansatz ist alles andere als trivial und es besteht das Risiko, dass man sich in der Erstellung der Modelle (Bett und Person) bereits verläuft. Allein die Modellierung des Bettes und der darin schlafenden Person stellen vom Arbeits- und Zeitaufwand eine eigene „kleine“ Abschlussarbeit dar.

Eine weitere Möglichkeit zur Nutzung modellbasierter Ansätze wäre der Einsatz von Rule Engines wie beispielsweise der java-basierten Jess Rule Engine¹⁰. Dies würde bedeuten, dass man auf Basis der vorbereinigten Messwerte eine Reihe an Regeln aufstellt, die definieren, wann beispielsweise eine Leichtschlafphase vorliegt. Eine Regel ist dabei einem Kausalsatz gleichzusetzen (Beispiel: „*Wenn Messwert1 größer 100, dann liegt eine Bewegung im Schulterbereich vor*“). Diese Regeln können wiederum untereinander verknüpft werden, so dass aus mehreren einfachen Regeln immer komplexere Regelwerke entstehen.

Durch die Erstellung komplexer Regeln bietet sich die Möglichkeit, die verschiedenen Ebenen der Interpretation durch unterschiedliche Verknüpfungen der Regeln untereinander abzubilden. Des Weiteren ist das Regelsystem zur Einbindung anderer Sensoren leicht durch die Definition zusätzlicher Regeln erweiterbar.

Variante 3 - Maschinelle Lernverfahren

Mit einem empirischen Ansatz durch den Einsatz maschineller Lernverfahren bietet sich bei der hier vorliegenden Klassifikationsaufgabe die Verwendung einer Support Vector Machine (SVM) an. Im einfachsten Fall ist eine binäre Klassifikation von verschiedenen, in Verbindung gebrachten Sensorwerten notwendig, um eine Leichtschlafphase zu bestimmen. Damit eine SVM die eingehenden Featurevektoren mit den einzelnen Features der Sensordaten klassifizieren kann, muss sie trainiert werden. Dafür eignen sich zwei unterschiedliche Methoden.

Teaching:

Im Teaching-Modus werden, ähnlich dem Modell-Ansatz aus Abschnitt 5.3, charakteristische Schlafszene durch freiwillige Probanden im intelligenten Bett nachgespielt und die Messwerte als Trainingsdaten eingespielt (und für spätere Trai-

¹⁰<http://www.jessrules.com>

ningsdurchläufe gespeichert). Wichtig ist hierbei die Auswahl der Probanden, damit man die Größen- und Gewichtsunterschiede in der Gesellschaft möglichst gut abdeckt. Ausserdem müssen charakteristische Schlafszeneen zuerst ermittelt und dann die dafür relevanten Bewegungen erkannt und nachspielbar dargestellt werden. Dafür sind Aufzeichnungen eines professionellen medizinischen Schlaflabors unerlässlich.

Das größte Risiko bei diesem Ansatz zum Trainieren der SVM ist das falsche Nachspielen der Schlafszeneen durch die Probanden, da dadurch die Klassifikation nach dem Training nachhaltig verfälscht würde.

Trainingsdaten:

Eine zweite Variante um die SVM zu trainieren ist das Einspielen von gesammelten Trainingsdaten. Dafür muss der Prototyp eingesetzt werden um Trainingsdaten zu sammeln. Die gesammelten Daten müssen nachträglich verarbeitet und die Klassenzugehörigkeit der Messwerte bestimmt werden. Dieser Prozess ist durch den doppelten Zeitaufwand für die Erfassung und die anschliessende Nachbearbeitung sehr aufwändig und benötigt die Unterstützung eines Spezialisten bei der Zuordnung der Messdaten zu den Schlafphasen. Neben einem Spezialisten muss die Datenerfassung am besten im Umfeld eines Schlaflabors geschehen, um die Zuordnung der einzelnen Schlafphasen durch die dort zusätzlich eingesetzte Technik optimal vornehmen zu können.

Der Vorteil dieser Trainingsmethode liegt in der höheren Präzision gegenüber dem Teaching-Modus und der Vergleichbarkeit der Genauigkeit gegenüber der in einem Schlaflabor eingesetzten Technik.

Unabhängig vom gewählten Trainingsmodell besteht das Risiko des falschen Trainingszustands bei einer SVM. Die Ergebnisse der Klassifikation in verschiedenen Trainingszuständen ist in Abbildung 33 dargestellt.

Ist eine SVM nicht ausreichend trainiert, so findet keine korrekte Klassifikation statt (Underfitting) und die trennende Hyperebene verläuft nicht an der Klassengrenze zweier zu trennender Klassen (Abbildung 33, oben links).

Bei einem zu trainierten Zustand kann man im Extremfall von „Auswendiglernen“ der SVM sprechen, da nur noch nahezu gleiche Objekte in einer Klasse liegen und als Mitglied einer solchen klassifiziert werden. Die trennende Hyperebene verläuft sehr nah an den Objekten einer Klasse entlang (Overfitting), ein gutes Klassifikationsergebnis ist hier nicht mehr zu erwarten (Abbildung 33, oben rechts).

Im optimalen Fall erreicht man mit dem Trainieren einer SVM einen Zustand, bei dem die Hyperebene einen möglichst großen Abstand zu den am nächsten liegenden Objekten der getrennten Klassen hat (Trade Off). In diesem Zustand sind gute Klassifikationsergebnisse zu erwarten (Abbildung 33, unten).

Verwendungsmöglichkeiten der Interpretationsverfahren

Die Verwendung von Interpretationsverfahren muss je Ebene der Interpretation neu diskutiert werden. Dabei stellt sich vor allem die Frage, ob es Sinn macht, jedes Mal ein komplexes Interpretationsverfahren zu wählen oder ob ein pragmatischer Ansatz unter

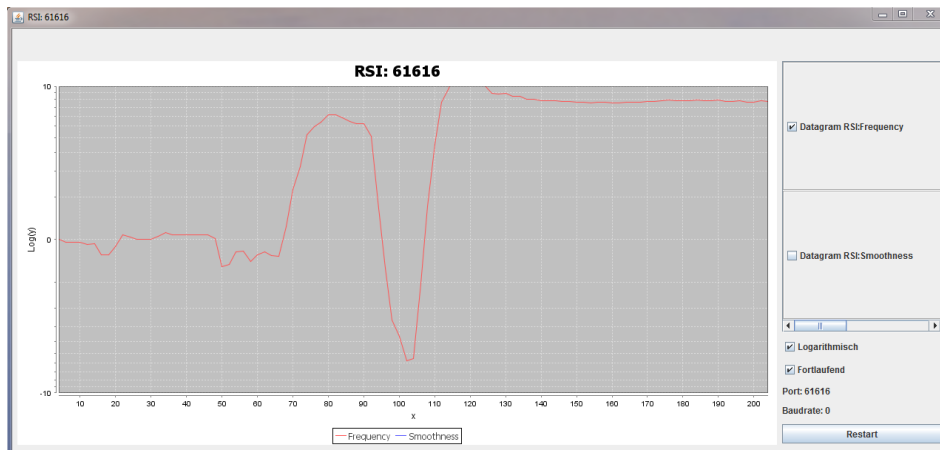
Umständen sinnvoller ist. Betrachten wir die unter Abschnitt 5.2 beschriebenen Ebenen genauer, so macht es Sinn, die Interpretationen für Ebene 0 und 1 durch einen „Hard coded“ Ansatz umzusetzen (ähnliche Ansätze wären die Verwendung eines Kalman-Filters (Schwanecke, Prof. Dr. Ulrich, 2007) oder eine Abwandlung des *Mexican Hat* (Bohnert und Kiesenbauer, 2003), einem kombinierten Laplace- und Gauss-Filter aus der Bildbearbeitung), da es sich hierbei nur um die einfache Überschreitung eines Schwellwertes handelt. Die Ebenen 2 bis 4 setzen eine komplexere Interpretation voraus, bei der es, aufbauend auf den darunterliegenden Ebenen, um eine Aggregation der gelieferten Informationen geht. Diese Aggregation geschieht nach gewissen Regeln, weshalb sich eine Rule Engine wie Jess anbietet.

5.4 Bestehende Visualisierungssoftware

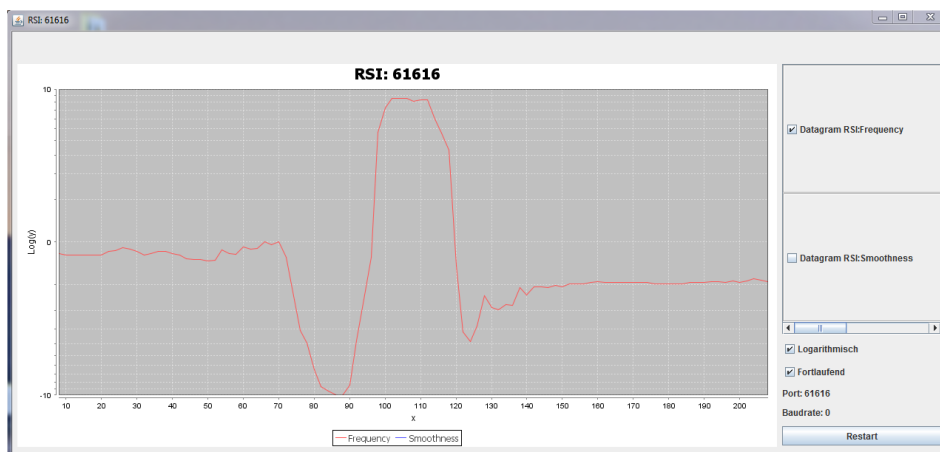
Die in Kooperation mit Oliver Dreschke entwickelte Visualisierungssoftware (beschrieben in (Dreschke, 2011)) ist eine performante Anwendung zum Auslesen der Messwerte verschiedenster Schwingkreiserzeuger und anderer Sensortypen. Dabei erfolgt die Darstellung der Messwerte logarithmisch aufgetragen in einem fortlaufenden Diagramm oder in einem 3D-Modell des intelligenten Sofas (Vgl. Abbildung 34). Die Verbindung zu einem laufenden ActiveMQ ist ebenfalls möglich, wodurch die Messwerte anderen Anwendungen leicht zur Verfügung gestellt werden können.

Mit dieser Software ist es möglich, die Messwerte der im intelligenten Bett verwendeten Schwingkreiserzeuger von RSI auszulesen und über einen lokal laufenden ActiveMQ der zu entwickelnden Anwendung zur Detektion von Bewegungen und Schlafphasen zur Verfügung zu stellen. Diese Funktionalität soll im Rahmen dieser Arbeit auch Verwendung finden, damit die Kommunikation mit den RSI Schwingkreiserzeugern nicht erneut implementiert werden muss.

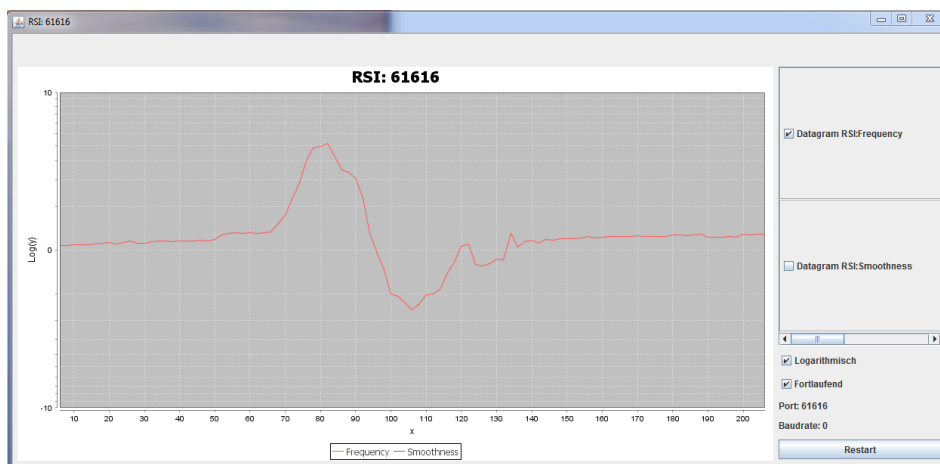
Zur Erfassung von Testdaten soll die Software dahingehend abgeändert werden, dass das Senden der Daten an den ActiveMQ mit dem Living Place Wrapper geschieht, um ein persistiertes Abbild der Daten in einer MongoDB zu erzeugen. Dafür muss die Kommunikation mit dem ActiveMQ grundlegend überarbeitet werden.



32.1: Schulter



32.2: Hüfte



32.3: Knie

Abbildung 32: Rohdaten einer Umdrehbewegung (Seiten- in Rückenlage, logarithmisch aufgetragen)

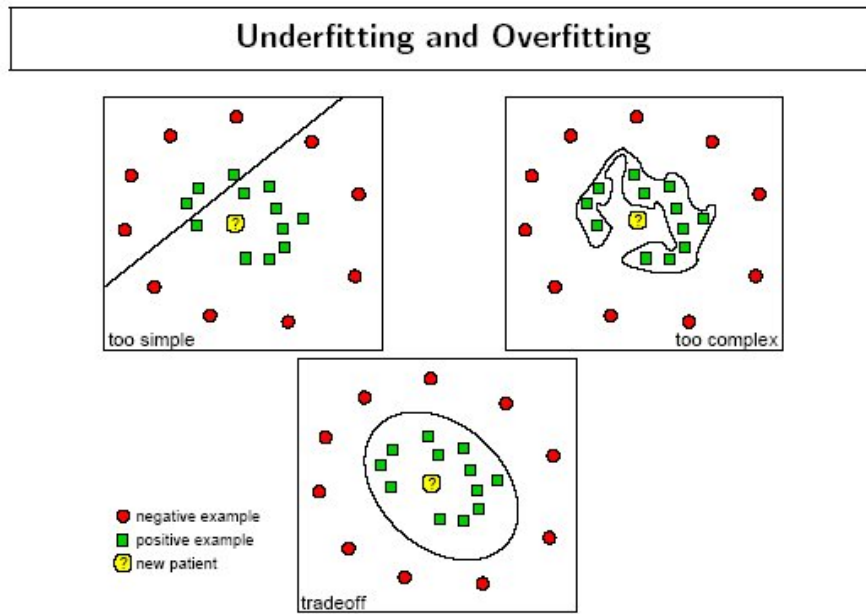
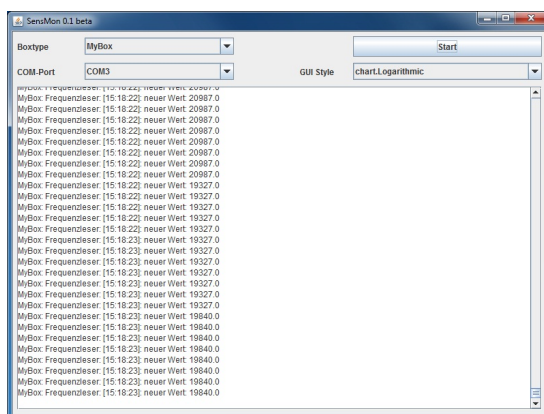
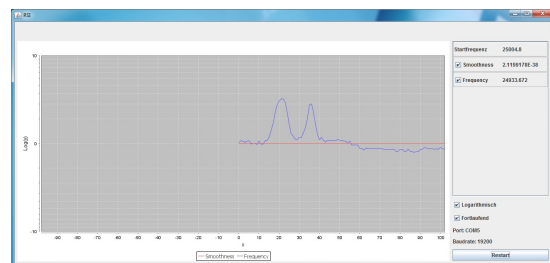


Abbildung 33: Over- und Underfitting (Quelle: <http://www.dtreg.com>)



34.1: Log-Fenster



34.2: Chart-Fenster

Abbildung 34: Visualisierungssoftware

6 Implementierung und technische Realisierung

Nach der Analyse und dem Design der Software für das intelligente Bett wird in diesem Abschnitt die technische Realisierung, die verwendeten Softwarekomponenten sowie die konkrete Architektur der Software beschrieben. Dabei unterteilt sich dieser Abschnitt in die Realisierung der Hardware und die Implementierung der Software.

6.1 Realisierung der Hardware

Durch die Vorarbeiten aus Abschnitt 4.3 ergab sich, dass für die Datenerfassung eigens entwickelte Sensoren aus verschiedenen Schichten leitender und nicht-leitender Folie in Kombination mit Schwingkreiserzeugern von RSI im Bett verwendet werden. Dafür wurden sechs Latten (je drei Latten pro Lattenrost) der Lattenroste im Bett des Living Place mit dem entwickelten Sensor an den in Abschnitt 4.3.3 festgelegten Positionen versehen und jeweils an einen Schwingkreiserzeuger angeschlossen. Die Übertragung der erfassten Messwerte vom Schwingkreiserzeuger zum Computer (Apple MacMini Server der 2. Generation) erfolgt über USB, wobei ein FT-DI Breakout Board die Schnittstelle zwischen dem Schwingkreiserzeuger und dem Computer bildet.



Abbildung 35: Bett mit Sensoren

Konfiguration der Hardware

Um für den Einsatz im intelligenten Bett nutzbar zu sein, mussten sowohl der MacMini Server als auch die RSI Schwingkreiserzeuger vor dem Einbau entsprechend konfiguriert werden.

Bei den Schwingkreiserzeugern wurde dafür die Auslöseschwelle angepasst und die Arbeitsfrequenz (im „Leerlauf“, ohne angeschlossene Sensoren) auf die laut Handbuch optimale Frequenz von ≈ 25200 Hz eingestellt.

Die Konfiguration des MacMini Servers war recht aufwändig, da auf diesem für die Java-Bibliotheken zur seriellen Kommunikation ein Windows 7 32bit installiert und die Bibliotheken eingebunden werden mussten. Um den MacMini auch später im laufenden Betrieb warten zu können, wurde die Remotedesktop-Verbindung aktiviert und eine statische IP im Living Place Netzwerk vergeben.

6.2 Erfassung von Testdaten

Zum Testen der Software und zur Demonstration der Funktionsweise des intelligenten Bettes war es unerlässlich, eine geeignete Menge von Testdaten zu erfassen. Dafür wurde auch das private Bett mit den entwickelten Sensoren versehen. Für die Erfassung der Testdaten werden die Messwerte durch die Visualisierungssoftware von Oliver Dreschke (Abschnitt 5.4 und (Dreschke, 2011)) von den Schwingkreiserzeugern ausgelesen und

durch die Verwendung des ActiveMQ-Wrappers ((Voskuhl und Otto, 2010) und (Voskuhl und Otto, 2011)) sowohl auf einem ActiveMQ als auch in einer MongoDB abgelegt. Der Datenbestand der MongoDB kann später nach Belieben mit dem angepassten MongoDB-Player von Jens Ellenberg in unterschiedlichen Geschwindigkeiten auf einem ActiveMQ abgelegt werden.

Die Testdaten einer erfassten Nacht werden in Anhang B dargestellt.

6.3 Implementierung der Software

Bei der Implementierung der SmartBed-Software zur Interpretation der Messwerte sollten von Anfang an Aspekte wie

- Wartbarkeit
- Erweiterbarkeit
- Konfigurierbarkeit
- Performanz

berücksichtigt werden. Die Entwicklung fand, aufbauend auf den erlernten Techniken und der vorliegenden Berufserfahrung, in Java statt. Entstanden ist eine Multithreaded-Anwendung mit einer Swing GUI, deren genauer Aufbau nachfolgend beschrieben wird.

Allgemein

Die entwickelte Software verfügt über einige allgemeine Entwicklungen, die nachfolgend dargestellt werden:

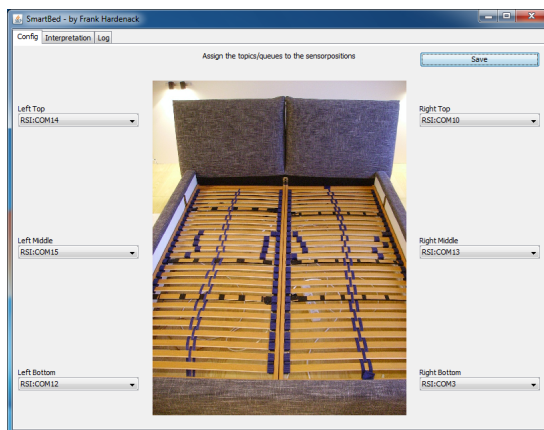
Threads: Die Consumer und Interpreter laufen in eigenen Threads und leiten von der eigens implementierten Klasse *SmartbedThread* ab.

Threadverwaltung: Eine zentrale Thread-Verwaltung ermöglicht den Zugriff auf die einzelnen Threads von jeder anderen Klasse aus.

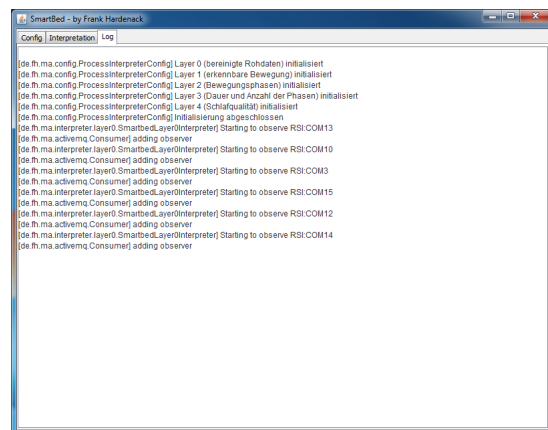
Debugging: Zum Ausgeben von Messwerten oder Interpretationsergebnissen kann man für die einzelnen Interpretationsebenen per Debug-Flag die Logausgaben aktivieren. Diese werden auf den Standard-Output (Meist die Command Line oder die Eclipse Console) geschrieben.

Logging: Die Anwendung verfügt über ein Logfenster, welches über einen selbst implementierten Logger bedient wird.

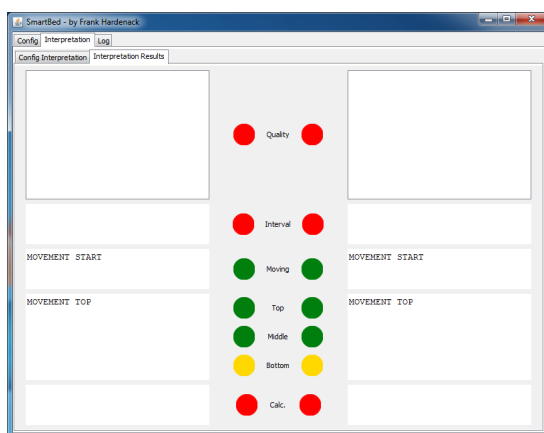
Reset-Funktionalität: Um die Anwendung nicht für jede Vorführung bzw. jeden Test neu starten zu müssen, wurde eine Reset-Funktionalität implementiert, mit der die Interpretationsebenen sowie die Consumer zum Verarbeiten der Daten vom lokalen ActiveMQ auf die initialen Einstellungen zurückgesetzt werden.



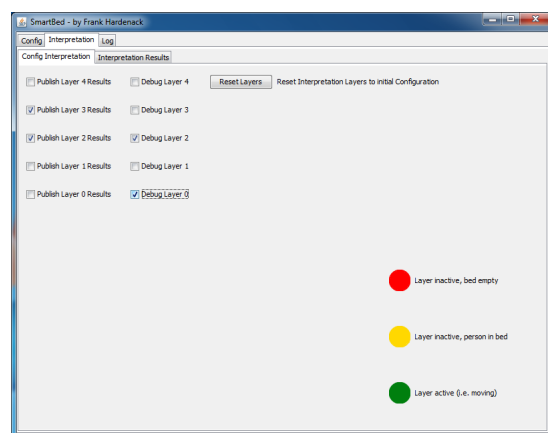
36.1: Auswahl der Sensoren



36.2: Das Log-Fenster



36.3: Live-Darstellung der Ergebnisse



36.4: Konfiguration der Interpretationsebenen

Abbildung 36: Die Smartbed-Software

Publishing: Es kann für jede Ebene eingestellt werden, ob die Ergebnisse auf den konfigurierten ActiveMQ publiziert werden sollen, oder nicht.

Ergebnisdarstellung: Die Ergebnisse der Interpretationsebenen werden in einer Art Ampelsystem dargestellt (Abbildung 36.3).

- Bei **rot** ist das Bett leer, es ist keine Aktivität zu verzeichnen.
- **Gelb** besagt, dass das Bett nicht leer ist, aber auch keine Aktivität in Form von Bewegungen vorliegt.
- Liegt eine Bewegungs-/Aktivitätsphase vor, so leuchtet die Ampel für die jeweilige Ebene **grün**.

Konfigurierbarkeit

Um die Software konfigurierbar zu gestalten, damit diese auch weiterhin in der sich dynamisch entwickelnden Struktur des Living Place eingesetzt werden kann, gibt es sowohl für die Consumer der Rohdaten als auch für die Interpretationsebenen jeweils ei-

ne .csv-Datei mit einer initialen Konfiguration. Diese Dateien werden beim Programmstart mit einem Open-Source-Framework zur Verarbeitung von .csv-Dateien (SuperCSV¹¹) eingelesen und die Konfigurationsparameter in das Programm übernommen. Durch diese Konfigurationsdateien lässt sich auf die Herkunft der Sensor-Rohdaten, die Vorbereitung und die „Stellschrauben“ der Interpretationsebenen Einfluss nehmen. Eine detaillierte Beschreibung der Konfigurationsmöglichkeiten der Software findet sich in Anhang C.

Konkrete Verfahren zur Interpretation

Die in Abschnitt 5.2 beschriebenen Ebenen zur Interpretation und semantischen Anreicherung der Messwerte bilden eine Art Pipe-Konzept ab, bei dem alle Ebenen aufsteigend durchlaufen werden und wurden wie nachfolgend beschrieben in die SmartBed-Software implementiert. Abbildung 37 zeigt dabei die einzelnen Ebenen und die von den Ebenen gelieferten Informationen über den Schlafzustand des Nutzers. Die Konfiguration der einzelnen Ebenen geschieht über .csv-Dateien, die Optionen zur Konfiguration werden in Anhang C detailliert beschrieben.

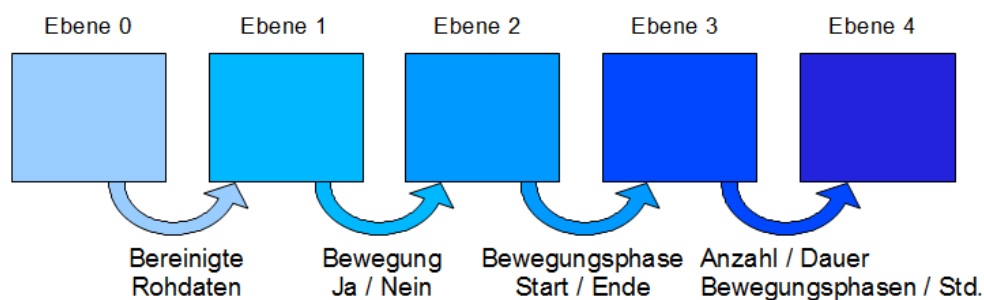


Abbildung 37: Informationsfluss zwischen den einzelnen Interpretationsebenen

Ebene 0 - Rohdaten

Auf der untersten Ebene der Interpretationsebenen werden die Rohdaten von allen zu einem Lattenrost gehörenden Sensoren mittels implementierter Berechnungsverfahren aufbereitet. Dabei bekommt die unterste Ebene (*SmartbedLayer0Interpreter*) von den zugeordneten Consumern über das Observer-Pattern mitgeteilt, dass neue Messwerte vorliegen.

Da eine Drehbewegung deutlich länger als 100ms dauert und die RSI-Boxen jeweils alle 100ms einen neuen Messwert liefern, gibt es eine Option, mit der (konfigurierbar viele) Messwerte abgewartet wird, bis die nächste Berechnung durchgeführt wird. Die abgewarteten Messwerte werden in der nächsten Berechnung mit berücksichtigt. Wird die Aufbereitung der Daten durch eine Benachrichtigung über neue Messwerte angestoßen, so wird ein Snapshot der letzten Messwerte (Anzahl ist konfigurierbar) eines Sensors in Form eines Ringpuffers in das ausgewählte Berechnungsverfahren gegeben. Das Ergebnis des jeweils eingesetzten Verfahrens wird an die darüberliegende Ebene über das

¹¹<http://supercsv.sourceforge.net>

SmartbedLayer0Observable kommuniziert. Es wurden dafür folgende Berechnungsverfahren implementiert:

- Mittelwert
- Standardabweichung
- Varianz
- Streuung

Bereits auf der Ebene der Consumer wurde eine Filterung auf große Messfehler realisiert. Dafür wird in den einzelnen Consumern ein Mittelwert über ein floating window der letzten Messwerte vorgehalten. Sollte ein erfasster Messwert größer als das 1,1-fache oder kleiner als das 0,9-fache des vorgehaltenen Mittelwerts sein, so wird dieser Messwert verworfen, da mit sehr großer Wahrscheinlichkeit ein Messfehler vorliegt.

Listing 1: Algorithmus der Ebene 0 (Pseudocode)

```
1 interprete (value) {  
2   if (++count % leaveOutMeasurements == 0) {  
3     //Only calculate every n measurements  
4     value = useSelectedCalculationMethod ();  
5     notifyLayer1 (value);  
6   }  
7 }
```

Ebene 1

Die zweite Ebene ist, nach der Bereinigung der Rohdaten, die erste Ebene mit einer definierten Interpretation der Messwerte. Wird der *SmartbedLayer1Interpreter* durch das beobachtete *SmartbedLayer0Observable* über neue, interpretierbare Informationen benachrichtigt, so wird das *SmartbedLayer0Result* aus dem Observable ausgelesen und die dort abgelegten Messwerte können weiter verarbeitet werden.

Da sich auf dieser Ebene mit der Detektion von Bewegungen sowie der Erkennung von Hinlege- und Aufstehereignissen beschäftigt wird, musste zuerst eine Unterscheidung zwischen Hinlege-/Aufstehereignissen sowie einfachen Bewegungen (z.B. durch eine Drehung) implementiert werden. Mit den aus der darunterliegenden Ebene zur Verfügung stehenden Informationen in Form der Standardabweichung (in der initialen Konfiguration) der Messwerte sowie der jeweils aktuellen Messwerte selbst stehen dafür ausreichend Informationen zur Verfügung.

Um eine Unterscheidung zwischen einem Hinlege-/Aufstehereignis und einer simplen Bewegung vornehmen zu können, wird beim Start der Anwendung eine Kalibrierung vorgenommen, bei der ein initialer Mittelwert je Sensor über eine konfigurierbare Anzahl an Werten bestimmt wird. Dies ist möglich, da der verwendete Sensor im unbelasteten Zustand immer wieder in seine Ruhelage zurückkehrt (siehe Abschnitt 4.3.2). Mit diesem initialen Mittelwert sowie den Informationen aus Ebene 0 kann man folgende Informationen aus den Messwerten erhalten:

Hinsetzereignis: Ein Hinsetzereignis liegt vor, wenn das Bett vorher leer war und sich der Messwert eines Sensors nicht innerhalb des Intervalls vom initialen Mittelwert \pm einer konfigurierbaren Auslöseschwelle befindet. Liegt diese Kombination vor, so wird das Bett als nicht mehr leer angenommen, es hat sich jemand auf das Bett gesetzt.

Hinlegeereignis: Ein Hinlegeereignis liegt vor, wenn sich die Messwerte von mindestens zwei Sensoren nicht innerhalb des Intervalls vom initialen Mittelwert \pm einer konfigurierbaren Auslöseschwelle befinden. Liegt diese Kombination vor, so wird das Bett als nicht mehr leer angenommen, es hat sich jemand in das Bett gelegt.

Aufstehereignis: Ist das Bett nicht leer und nähert sich der Messwert eines Sensors seinem initialen Mittelwert aus der Kalibrierungsphase an (d.h. der Messwert liegt in dem Intervall vom initialen Mittelwert des Messwerts \pm der konfigurierbaren Auslöseschwelle), so liegt ein Aufstehereignis vor und das Bett wird als leer angenommen, die Person ist aus dem Bett aufgestanden.

Bewegung: Ist das Bett nicht leer und liegt der Messwert nicht in dem Intervall vom initialen Mittelwert des Messwertes \pm der Auslöseschwelle, so hat sich die schlafende Person nur im Bett bewegt.

Für alle vier möglichen Ereignisse ist die Überschreitung einer konfigurierbaren Auslöseschwelle durch die Standardabweichung aus Ebene 0 eine notwendige Voraussetzung. Es kommt vor, dass das Aufstehereignis aus den Messwerten nicht korrekt erkannt wird. Um diesem Fehler entgegenzuwirken wurde ein konfigurierbarer Wert eingeführt, mit dem festgelegt wird, wieviele Messungen sich jeder Sensor in der Ruhelage (also nahe dem initialen Mittelwert) befinden muss, bis das bis dahin als belegt eingestufte Bett als leer angenommen wird.

Die Ergebnisse der Bewegungserkennung werden in ein *SmartbedLayer1Result* verpackt, welches in das *SmartbedLayer1Observable* geschrieben wird und darüber mittels Observer-Pattern die Interpretation der Ebene 2 anstößt.

Listing 2: Algorithmus der Ebene 1 (Pseudocode)

```
1 interprete (value) {
2   if (!calibrated) {
3     initialAverage = calibrateSensor (value);
4   } else {
5     if (value > measurementTrigger) {
6       if (positionEmpty) {
7         if (!measurementIsNearInitialAverage (value)) {
8           if (bedIsEmpty ()) {
9             notifyLayer2 ("Sitdown");
10          } else {
11            if (oneSensorIsNotEmpty ()) {
12              notifyLayer2 ("Laydown");
13            } else {
14              notifyLayer2 ("Movement");
15            }
16          }
17        }
18      }
19    }
20  }
21 }
```

```
16     }
17   }
18   positionEmpty = false;
19 } else {
20   if (measurementIsNearInitialAverage(value)) {
21     if (oneSensorIsNotEmpty()) {
22       notifyLayer2("Standup");
23     }
24     positionEmpty = true;
25   } else {
26     notifyLayer2("Movement");
27   }
28 }
29 } else {
30   if (!bedsEmpty() && measurementIsNearInitialAverage(value)) {
31     incrementAssumedEmptyCount();
32   }
33   if (allPositionsAssumeBedEmpty()) {
34     notifyLayer2("Standup");
35   } else {
36     notifyLayer2("NoMovement");
37   }
38 }
39 }
40 }
```

Ebene 2

Mit den Informationen aus Ebene 1, die eine Aussage darüber zulassen, ob sich jemand in das Bett gelegt hat, bewegt hat oder aufgestanden ist, lassen sich nun Bewegungs- und Nicht-Bewegungsphasen (oder genauer: Angenommene Leichtschlaf- und Nicht-Leichtschlafphasen) bestimmen. Dafür werden die Informationen, die im *SmartbedLayer2Interpreter* durch das *SmartbedLayer1Result* aus dem *SmartbedLayer1Observable* zur Verfügung stehen, ausgelesen und entsprechend verarbeitet.

Mit einem empfangenen Hinlegeereignis beginnt die Erfassung von Bewegungsphasen, mit einem Aufstehereignis endet sie. Da es in einer Bewegungsphase auch zu kurzen Momenten der Ruhe ohne eine Bewegung kommen kann, werden dicht beieinander liegende, aber durch Ruhemomente unterbrochene Bewegungen zu einer Bewegungsphase zusammengefasst. Die Anzahl der Messwerte ohne erkennbare Bewegung, die zwischen zwei zu einer Bewegungsphase gehörenden Bewegungen liegen dürfen, ist konfigurierbar.

Damit ein Ausreißer bei den Messwerten eines Sensors nicht fälschlicherweise als der Beginn einer Bewegungsphase interpretiert wird, müssen innerhalb eines konfigurierbaren Intervalls mindestens zwei Sensoren eine Bewegung melden. Geschieht dies nicht und es meldet nur ein Sensor eine Bewegung, so wird von einem fehlerhaften Messwert ausgegangen.

Listing 3: Algorithmus der Ebene 2 (Pseudocode)

```
1 interprete (value) {
2   if (notificationFromLayer1 == "Laydown") {
3     if (!personInBed) {
4       personInBed = true;
5       notifyLayer3 ("Laydown");
6     }
7     return;
8   }
9
10  if (personInBed) {
11    if (notificationFromLayer1 == "Standup") {
12      if (movementPhase) {
13        movementPhase = false;
14        notifyLayer3 ("MovementEnd");
15      }
16      notifyLayer3 ("Standup");
17      personInBed = false;
18    } else if (notificationFromLayer1 == "Movement") {
19      resetNoMovementCount ();
20      if (movementPhase && !movementPhaseActive) {
21        movementPhaseActive = true;
22        notifyLayer3 ("MovementStart");
23      }
24    } else if (notificationFromLayer1 == "NoMovement") {
25      incrementMovementWaitingCount ();
26      if (movementPhaseActive) {
27        incrementNoMovementCount ();
28        if (movementPhaseAssumedOver ()) {
29          movementPhase = false;
30          movementPhaseActive = false;
31          notifyLayer3 ("MovementEnd");
32        }
33      }
34    }
35  }
36 }
```

Ebene 3

Durch die in Ebene 2 erkannten Bewegungs- und Nicht-Bewegungsphasen kann in Ebene 3 deren Dauer und zeitliche Abfolge näher betrachtet werden. Die Betrachtung beginnt bei einem Hinlegeereignis und endet mit einem Aufstehereignis oder dem Ende des konfigurierbaren Betrachtungsintervalls. Endet ein Intervall vor dem Aufstehereignis, so wird das bisherige Ergebnis der Betrachtung an die nächste Ebene weitergereicht und ein neues Betrachtungsintervall begonnen. In dem Ergebnis des Intervalls wird für jede Phase festgehalten, ob es eine Bewegungsphase war, wann sie begonnen und wann sie geendet hat. Ist eine Phase zum Ende eines Intervalls nicht abgeschlossen, so wird für

das Ende keine Zeit eingetragen, da diese Phase auch im nächsten Betrachtungsintervall noch aktiv ist. Es sind dementsprechend fünf verschiedene Aktionen möglich:

Hinlegen: Legt sich jemand in das Bett, so wird ein Betrachtungsintervall gestartet. Die mit dem Hinlegen verbundenen Bewegungen werden bereits als Bewegungsphase gewertet.

Bewegungsphase: Beginnt eine Bewegungsphase, so wird die vorher aktive Phase in der Ergebnisliste als abgeschlossen vermerkt und der Ergebnisliste ein weiterer Eintrag hinzugefügt.

Ruhephase: Beginnt eine Ruhephase, so wird die vorher aktive Phase in der Ergebnisliste als abgeschlossen vermerkt und der Ergebnisliste ein weiterer Eintrag hinzugefügt.

Aufstehen: Steht die Person auf, so wird die aktive Phase beendet und die Ergebnisliste an die nächste Interpretationsebene weitergereicht. Die mit dem Aufstehen verbundenen Bewegungen werden als Bewegungsphase gewertet.

Intervallende: Alle $\frac{\text{Segmentlänge}}{100}_s$ wird überprüft, ob die maximale Länge für das Segment erreicht ist. Wurde die maximale Länge erreicht, so wird das Segment beendet und die Ergebnisliste an die nächste Interpretationsebene weitergereicht, die aktive Phase jedoch nicht beendet.

Um die Ergebnisse in der Reihenfolge ihres Auftretens an die nächste Interpretationsebene weiterzureichen, wurde für die Ergebnisliste eine einfach verkettete Liste implementiert (Siehe Anhang D, Listing 7).

Listing 4: Algorithmus der Ebene 3 (Pseudocode)

```

1  interprete () {
2    if (notificationFromLayer2 == "Laydown") {
3      if (!personInBed) {
4        personInBed = true;
5        createNewLinkedList ();
6        renewInterval ();
7        addEntryToLinkedList (starttime , noMovement);
8      }
9    }
10   if (personInBed) {
11     if (notificationFromLayer2 == "Movement_Start") {
12       closeActiveListEntry ();
13       addEntryToLinkedList (starttime , movement);
14     } else if (notificationFromLayer2 == "Movement_End") {
15       closeActiveListEntry ();
16       addEntryToLinkedList (starttime , noMovement);
17     } else if (notificationFromLayer2 == "Standup") {
18       closeActiveListEntry ();
19       notifyLayer4 (linkedList);
20       createNewLinkedList ();

```

```
21     personInBed = false;
22   }
23 }
24 }
25
26 checkInterval() {
27   while (interpreterActive) {
28     if (intervalEndTimeReached) {
29       notifyLayer4(linkedList);
30       createNewLinkedList();
31       renewInterval();
32     }
33   }
34 }
```

Ebene 4

Ist in Ebene 3 das Ende eines Intervalls erreicht, oder wird die Erfassung von Daten durch ein Aufstehereignis beendet, so wird jeweils eine *SmartbedLinkedList* an Ebene 4 gesendet und dort vorgehalten. Die Interpretation des Nachtschlafs erfolgt erst beim Empfangen eines Aufstehereignisses, welches mit der letzten übertragenen Liste ebenfalls gesendet wird. Es werden alle empfangenen Listen sequentiell in der Reihenfolge ihres Eintreffens verarbeitet und die Informationen über Bewegungs- bzw. Nicht-Bewegungsphasen sowie deren Dauer aggregiert. Da es vorkommen kann, dass eine Phase beim Ende eines Intervalls noch nicht beendet ist, müssen diese über zwei Listen verteilten „halben“ Phasen für die Bewertung der Schlafqualität wieder zusammengeführt werden.

Für die Ermittlung der Schlafqualität werden unterschiedliche, konfigurierbare Werte herangezogen (Siehe Tabelle 9 in Anhang C), die die durchschnittliche und minimale Schlaflänge, die Zeit zwischen zwei Bewegungsphasen sowie eine maximale Anzahl von Bewegungsphasen als Vergleichsbasis für guten Schlaf definieren. Wurden alle Ergebnislisten aus Ebene 3 verarbeitet, so werden die ermittelten Werte mit den vorkonfigurierten Werten verglichen und eventuelle Abweichungen gewichtet bewertet. Diese gewichtete Bewertung betrachtet die nachfolgend aufgelisteten, möglichen Ereignisse und vergibt dabei die als Gewichtung angegebenen Punkte:

- Unterschreitung der durchschnittlichen Schlaflänge - Da zu wenig Schlaf ein Indiz für eine nicht erholsame Nacht und somit schlechtes Befinden des Nutzers ist. (Gewichtung: 1-fach)
- Unterschreitung der minimalen Schlaflänge - Deutlich zu wenig Schlaf ist ein klares Indiz für eine nicht erholsame Nacht und somit schlechte Schlafqualität. (Gewichtung: zusätzlich zur durchschnittlichen Schlaflänge 2-fach)
- Überschreitung der maximal angegebenen Bewegungsphasen - Deutet auf einen unruhigen und damit schlechten Schlaf hin. (Gewichtung: 2-fach)

- Anzahl der Unterschreitungen der angenommenen Zeit zwischen zwei Bewegungsphasen - Wird die vorkonfigurierte Zeit zwischen zwei Bewegungsphasen unterschritten, so ist dies ein Anzeichen für unruhigen Schlaf, da Bewegungsphasen nur beim Übergang von und in den Leichtschlaf angenommen werden (Siehe Abbildung 1 in Abschnitt 2.1).
 - Weniger als 2 Unterschreitungen: 1-fache Gewichtung.
 - Häufigere Unterschreitung: 2-fache Gewichtung.

Nach der Gewichtung erfolgt eine Zuordnung des evaluierten Schlafs zu einer der drei Ergebnisklassen (Siehe Tabelle 14 in Anhang D) in Abhängigkeit der gewichteten Ergebnisse. Die Zuordnung geschieht durch folgende Einteilung:

- 0 Punkte: Guter Schlaf
- <5 Punkte: Mittelmäßiger Schlaf
- ≥ 5 Punkte: Schlechter Schlaf

Listing 5: Algorithmus der Ebene 4 (Pseudocode)

```
1  interprete(list) {
2    addListToListMemory(list);
3    if (notificationFromLayer3 == "Standup") {
4      interpreteInternal();
5      resetInterpretationLayers(leftSide);
6    }
7  }
8
9  interpreteInternal() {
10   startTime = 0;
11   previousPhaseIncomplete = false;
12   for (SmartbedLinkedList list : listMemory) {
13     while (list != null) {
14       if (list.isPhaseCompleted()) {
15         duration = 0;
16         if (previousPhaseIncomplete) {
17           duration = list.getEndTime() - startTime;
18         } else {
19           duration = list.getDuration();
20         }
21         addToSleepDuration(duration);
22         if (list.isMovementPhase()) {
23           movements++;
24         } else {
25           if (duration < timeBetweenMovements) {
26             timeBetweenMovementsNotOk++;
27           } else {
28             timeBetweenMovementsOk++;

```



```
29     }
30   }
31   } else {
32     startTime = list.getStartTime();
33     previousPhaseIncomplete = true;
34   }
35   list = list.getChild();
36 }
37 }
38 evaluateResult();
39 }
```

6.4 Kommunikation der Interpretationsergebnisse

Die Ergebnisse der einzelnen Interpretationsebenen dienen zum einen als Datenbasis für die darüberliegende Interpretationsebene, zum anderen sollen diese Ergebnisse dem Living Place und damit auch anderen, potentiellen Verbrauchern (Consumern) dieser Informationen zur Verfügung gestellt werden. Die Ergebnisdatenströme werden beispielhaft im Living Place auf einen dort eingerichteten ActiveMQ abgelegt. Dafür verfügt jede Interpretationsebene über einen eingebauten Producer, der die Ergebnisse mit Gson¹² serialisiert und auf einem konfigurierbaren ActiveMQ ablegt.

Bei den konkreten Ergebnissen handelt es sich um die in Abbildung 37 dargestellten, zunehmend aggregierten Informationen.

6.4.1 Format der Interpretationsergebnisse

Um ein einheitliches Format für die Ergebnisse der einzelnen Ebenen sicherzustellen wurde die generische Klasse *SmartbedPublishingResult* implementiert, die ein boolesches Flag zur Identifikation des linken oder rechten Lattenrosts sowie eine Map für die hinzugefügten Ergebnisse enthält.

Eine detaillierte Auflistung der publizierten Ergebnisse, ein Listing der Klasse *SmartbedPublishingResult* sowie ein kurzer Beispielcode zum Deserialisieren der Ergebnisse finden sich in Anhang D.

¹²<http://code.google.com/p/google-gson/>

7 Evaluierung

Im Rahmen der Masterarbeit wurde ein Konzept entwickelt und eine Software implementiert, mit der eine Unterscheidung von gutem und schlechtem Schlaf möglich ist. Das Konzept sowie die Software entstanden im Kontext von intelligenten Wohnumgebungen am Beispiel des Living Place und wurden als eine Erweiterung der vielfältigen Projektlandschaft in den Living Place integriert.

Zusätzlich zu der Erstellung des Konzepts sowie der Implementierung war es notwendig, geeignete Sensoren für die Datenerfassung zu entwickeln. Nach mehreren Versuchen wurde eine kapazitive Messmethode unter Verwendung von Schwingkreiserzeugern aus der Sicherheitstechnik sowie selbst entwickelten Sensoren gewählt. In mehreren Durchgängen fiel die Entscheidung hier auf einen mehrschichtigen Folien-Sensor, da dieser in den Vergleichstests (Siehe Tabelle 1) am besten abgeschnitten hat und sich mit geringem Aufwand und geringen Kosten fertigen lässt. Durch die Montage auf dem Lattenrost war die Integration in eine bestehende Wohnumgebung bestmöglich gegeben. Das langfristige Verhalten des Sensors und der verwendeten Materialien ist allerdings noch nicht erprobt und muss über die Zeit beobachtet werden.

Für die Interpretation der erfassten Messwerte musste ein geeignetes Verfahren ermittelt werden, mit dem eine qualitativ gute Interpretation möglich war. Hierfür wurden verschiedene Ansätze vorgestellt (Siehe Abschnitt 5.3). Für die unteren Ebenen (Ebene 0 und Ebene 1) war von Anfang an ein „Hard-Coded“-Ansatz geplant, der in dieser Form auch umgesetzt wurde und über Parameter an das individuelle Schlafverhalten sowie äußere Gegebenheiten anpassbar ist. Die grundsätzliche Idee, für die darüber liegenden Ebenen eine Rule-Engine wie Jess zu verwenden, wurde aufgrund der Komplexität des Regelwerks sowie der auslaufenden Lizenz für Jess verworfen und stattdessen ein pragmatischer Ansatz ähnlich der unteren Ebenen gewählt. Der Ansatz, die Interpretation auch auf den höheren Ebenen über konfigurierbare „Hard-Coded“-Algorithmen umzusetzen, liefert dabei sowohl am Bett im Living Place als auch am prototypischen Aufbau im Projektraum sehr zufriedenstellende Ergebnisse. Die Parametereinstellungen der einzelnen Interpretationsebenen sollen dabei einer natürlichen Interpretation möglichst nahe kommen, die optimale Einstellung der Parameter ist an dieser Stelle jedoch noch offen und muss im Rahmen weiterer Untersuchungen geklärt werden.

Da das Bett von Anfang an als ein passiver Dienstleister im Gesamtprojekt des Living Place angesiedelt war (Siehe Abschnitt 4.1), werden die Ergebnisse der Interpretation nicht nur über die entwickelte Software visualisiert (Siehe Abbildung 36), sondern auch auf einem konfigurierbaren ActiveMQ abgelegt, damit weitere Anwendungen (Wecker, Beleuchtung) auf Basis der Interpretationsergebnisse mit dem Living Place oder dem Nutzer interagieren können.

Durch die vollständige Integration in den Living Place ist das intelligente Bett als Smart Environment ein Bestandteil einer intelligenten Umgebung (Smart Home). Die Erkennung von Bewegungsphasen ist bis hin zu einer ersten Annahme über die Schlafqualität vollständig umgesetzt und wird, kombiniert mit der Lichtinstallation und der Lüftungssteuerung, zur Beschreibung eines Aufsteh-Szenarios betrieben. Die Ergebnisse des

intelligenten Bettes provozieren dabei durch Benachrichtigung der Lichtinstallation und der Ansteuerung der Oberlichter bei einem Aufstehereignis eine Veränderung der Beleuchtung sowie der Belüftung. Dadurch erfolgt eine implizite Steuerung des Living Place durch den Nutzer.

Rückblickend auf den Verlauf der Masterarbeit haben sich die Entscheidung zur Verwendung eines kapazitiven Messverfahrens sowie der selbst entwickelte Sensor zur Applikation auf dem Lattenrost als richtige Entscheidungen herausgestellt. Auch das Pipe-Konzept bei der Implementierung der einzelnen Interpretationsebenen, kombiniert mit einer auf einander aufbauenden Observer-Struktur, bietet gute Erweiterungsmöglichkeiten, ist leicht verständlich und gut wartbar. Die Verwendung selbst entworfener, konfigurierbarer Algorithmen als maßgeschneiderte Lösung für die Detektion von Bewegungs- und Schlafphasen liefern zuverlässige Ergebnisse und lassen sich zudem individuell konfigurieren.

Durch weitere Tests, am besten in Kooperation mit Medizinern und Schlaflaboren, muss sich zeigen, ob die in Abschnitt 5.2 getroffenen Annahmen über die Dateninterpretation der einzelnen Ebenen soweit korrekt sind. Ebenso muss der Bezug von Bewegungsphasen zu Leichtschlafphasen medizinisch noch weiter verifiziert werden.



38.1: Das intelligente Bett



38.2: Person im Bett, Licht aus



38.3: Person aufgestanden, Licht an

Abbildung 38: Impressionen vom intelligenten Bett

8 Fazit

In dieser Arbeit wurde im Kontext intelligenter Wohnumgebungen (Smart Homes) auf Basis der Annahme, dass nächtliche Leichtschlafphasen mit Bewegungsphasen einhergehen (Abschnitt 2.1), ein Konzept zur kontaktlosen Erkennung von Bewegungen im intelligenten Bett entwickelt und eine Interpretations-Software implementiert. Als Grundgedanke dienten unter anderem bereits verfügbare, kommerzielle Produkte zur Erkennung von Bewegungsphasen sowie aktuelle Forschungsprojekte und medizinische Erkenntnisse (Abschnitt 3). Für ein intelligentes Bett im Umfeld einer Smart Home Umgebung wie dem Living Place waren jedoch besondere Maßstäbe an Integrierbarkeit, Usability und die Art der Sensoren zu stellen, um ein optimales, für den Nutzer in erster Linie unsichtbares Ergebnis zu erzielen.

Für die Erfassung fiel die Wahl unter Beachtung verschiedener Gesichtspunkte wie Integrierbarkeit, Schirmung und Fertigungsaufwand auf kapazitive Sensorik. Die zur Erfassung der Daten notwendige Hardware stammt aus der Sicherheitstechnik und wird mit selbst entwickelten Sensoren kombiniert. Diese entstanden durch die Evaluierung geeigneter Sensoren (Abschnitt 4.3.2) in mehreren Schritten und bestehen aus sieben Schichten Folie (leitend und nicht leitend). Die Sensoren sind durch eine integrierte Schirmung gegenüber dem Nutzer komplett abgeschirmt, so dass um die Sensoren kein den Nutzer potentiell belastendes Feld auftritt. Die fertigen Sensoren wurden in einem prototypischen Aufbau des intelligenten Bettes im Projektraum mit vielversprechenden Ergebnissen getestet, bevor sie auf jeweils drei Latten je Lattenrost aufgebracht wurden. Die Erfassung erfolgt über das Auslesen der Sensoren mit der in Abschnitt 5.4 beschriebenen Visualisierungssoftware, die die Daten ausliest und unter Verwendung des Living Place Wrappers auf einem lokalen ActiveMQ ablegt, von dem aus die Daten von der Smartbed-Software weiter verarbeitet werden.

Die auf diese Weise erfassten Messwerte werden durch die SmartbedSoftware vom lokalen ActiveMQ ausgelesen, zur Beseitigung von Messfehlern vorbereitet und über fünf Interpretationsebenen immer weiter aggregiert und interpretiert. Die Interpretation beginnt mit den bereinigten („geglätteten“) Messwerten auf Ebene 0, darauf aufbauend folgt auf Ebene 1 eine Erkennung von Hinlege-, Aufsteh- und Bewegungsereignissen. Auf Ebene 2 werden die Ergebnisse aus Ebene 1 zu Bewegungs- und Nicht-Bewegungsphasen zusammengefasst, welche in Ebene 3 in zeitlichen Bezug zueinander gesetzt werden und deren einzelne Dauer bestimmt wird. Schließlich folgt auf Ebene 4 eine Aussage über die angenommene Schlafqualität. Die Interpretation erfolgt mit Hilfe selbst entwickelter Algorithmen, die über Parameter für das individuelle Schlafverhalten konfigurierbar sind. Die Quelle der für die Interpretation herangezogenen Messwerte der RSI-Boxen ist dabei in der Software konfigurierbar, um die Software auch mit anderen ActiveMQ-Servern verbinden zu können.

Die Ergebnisse der Interpretationen werden zum einen in der Smartbed-Software dargestellt, zum anderen auf dem ActiveMQ des Living Place zur Verwendung durch andere Anwendungen abgelegt. Zur Publizierung der Ergebnisse wurde ein Format entworfen, mit dem die Ergebnisse in geeigneter Form dargestellt werden können (Anhang D).

9 Ausblick

Das intelligente Bett ist ein erster Ansatz zur Interpretation von Schlafphasen auf Basis von Bewegungs- und Ruhephasen. Inwieweit dieser Ansatz und die aufgestellten Hypothesen interindividuell zu nutzen sind, muss noch im Rahmen weiterer Forschungen geklärt werden. Die anfangs erwähnte „Buddy“-Metapher aus der Vision des SFB 62 (Universität Ulm, 2010) ist für das intelligente Bett aufgrund der passiven, informationserzeugenden und nicht direkt mit dem Nutzer interagierenden Funktionsweise nur teilweise zutreffend, da sich erst andere Anwendungen (Lichtsteuerung, Wecker (Ellenberg, 2011)) auf Basis der Daten, die vom intelligenten Bett zur Verfügung gestellt werden, auf die Situation und Befindlichkeiten des Nutzers einstellen. Im weiteren Verlauf des Projekts „Living Place Hamburg“ wäre eine Möglichkeit zur direkten Interaktion mit dem Nutzer (beispielsweise das Einspielen leiser, beruhigender Musik in einem unruhigen Abschnitt der Nacht oder das Beeinflussen der Beleuchtung bei einem erkannten Aufstehereignis) eine sinnvolle Erweiterung für das intelligente Bett, um dieses zu einem echten Companion-System zu machen.

Das im intelligenten Bett verwendete kapazitive Messverfahren zur Wahrnehmung von Bewegungen kann in vielen weiteren Szenarien Verwendung finden und zum Beispiel die Annäherung an den Multitouch-Küchentresen oder Stürze in einer entsprechend ausgestatteten Wohnumgebung detektieren. Die mehrschichtige Interpretation der Messwerte bis hin zu einer Aussage über die Schlafqualität erschafft einen Kontext, in dem sich der Nutzer befindet (beispielsweise im Bett liegend, gerade aufgestanden oder unruhig schlafend). Dieser Kontext kann dabei durch intendierte Gesten wie das Aufstehen aus dem Bett oder nicht intendierte Gesten wie einen Sturz beeinflusst werden. In wie weit dieser erschaffene Kontext bei der Interaktion von Mensch und Computer (Smart Environment) hilft und wie die Akzeptanz des Nutzers durch das Agieren des ihn umgebenden Smart Environments beeinflusst wird muss noch evaluiert werden. Sicher ist, dass das intelligente Bett einen kleinen Beitrag zum Companion- und Context-Aware-Ansatz im Living Place darstellt.

Als Ergebnis dieser Forschungsarbeit im Rahmen des Masterstudiums liefert das intelligente Bett durch kontaktlose Sensorik gute Ansätze und Ergebnisse zur Detektion von Bewegungsphasen. Mit diesen Ergebnissen wird die Forschung an dem Bett weiter vorangetrieben, um das Potential für einen zukünftigen Einsatz im medizinischen oder AAL-Umfeld zu überprüfen.

Glossar

Ambient Assisted Living	Technologien und Produkte zur Verbesserung der Lebensqualität älterer Menschen, 1
Ambient Intelligence	dt. Umgebungsintelligenz, Vernetzung von Sensoren, Computern und Gegenständen, Kommunikation untereinander über eine Art Vermittlungs-Infrastruktur (Blackboard), 18
Circadianer Rhythmus	Täglicher Zyklus von Tageslicht und Dunkelheit, nach dem sich fast alle Landtiere richten (Bear u. a., 2009)., 4
Context Aware Computing	Das Verhalten von Anwendungen und Computern mit dem Wissen über die Situation bzw. den Kontext des Nutzers., 6
FTDI	Wandler von USB zu Seriell, um Daten einer seriellen Schnittstelle über USB auslesen zu können, 42
Muskeltonus	Spannungszustand eines Muskels (Boss u. a., 1993, Seite 1137), 4
Narkolepsie	Krankheit mit anfallsweisem, nicht überwindbarem Schlafzwang während des Tages (Boss u. a., 1993, Seite 1159), 10
REM-Schlaf	Schlafstadium, das sich durch schnelle Augenbewegungen des schlafenden (R apid- E ye- M ovement) auszeichnet, 4
Schlafapnoe	Atemstillstand während des Schlafs, kann in Kombination mit Blutdruckerhöhung und Herzrhythmusstörungen auftreten (Boss u. a., 1993, Seite 1483), 10
Ubiquitous Computing	Die Allgegenwärtigkeit von Computern, auch in Alltagsgegenständen. Dabei rückt die Computertechnik immer mehr in den Hintergrund und bleibt dem Nutzer verborgen., 1

Literatur

- [Bear u. a. 2009] BEAR, Mark F. ; CONNORS, Barry W. ; PARADISO, Michael A.: *Neurowissenschaften - Ein grundlegendes Lehrbuch für Biologie, Medizin und Psychologie*. Auflage 3. Spektrum Akademischer Verlag, 2009. – ISBN 978-3-8274-2028-2
- [Bohnert und Kiesenbauer 2003] BOHNERT, Fabian ; KIESENBAUER, Dieter: Elementare Bildverarbeitungsoperationen - Kantenerkennung - / Universität Ulm. URL <http://www.mathematik.uni-ulm.de/stochastik/lehre/ss03/seminarss03/seminar/vortrag9b.pdf>, 2003. – Forschungsbericht. [Seminar Simulation und Bildanalyse mit Java, Sommersemester 2003]
- [Boss u. a. 1993] BOSS, Dr. med. Norbert ; JÄCKLE, Dr. med. Renate ; WANGERIN, Dr. med. Günter: *Roche Lexikon Medizin*. Auflage 3. Urban & Schwarzenberg, 1993. – ISBN 3-541-11213-1
- [ConText 2006] CONTEXT: ConText project develops intelligent clothing for preventing muscular disorders and sharpening athletics performance. URL http://www.hitech-projects.com/euprojects/context/downloads/ConText_Press_Release_Sept06.pdf, 2006. – Forschungsbericht. [Pressekundgebung über den Projektstart]
- [ConText 2008] CONTEXT: Newsletter 3. URL http://www.hitech-projects.com/euprojects/context/downloads/Context_newsletter_03.pdf, 2008. – Forschungsbericht. [Newsletter 3]
- [Doukas und Maglogiannis 2008] DOUKAS, Charalampos ; MAGLOGIANNIS, Ilias: Enabling human status awareness in assistive environments based on advanced sound and motion data classification. In: *PETRA '08: Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments*. New York, NY, USA : ACM, 2008, S. 1–8. – ISBN 978-1-60558-067-8
- [Dreschke 2009] DRESCHKE, Oliver: Computational Furniture / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2009-aw2/dreschke/bericht.pdf>, 2009. – Forschungsbericht. [Vortrag vom 25.06.2008, AW2]
- [Dreschke 2011] DRESCHKE, Oliver: Entwicklung kontextsensitiver Möbel für intelligente Wohnumgebungen / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/dreschke.pdf>, 2011. – Forschungsbericht. [Masterarbeit, 2011]
- [Ellenberg 2011] ELLENBERG, Jens: Entwicklung des Weckers 2.0 / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/ellenberg.pdf>, 2011. – Forschungsbericht. [Projektbericht zum Master-Projekt 2, Wintersemester 2010/2011]

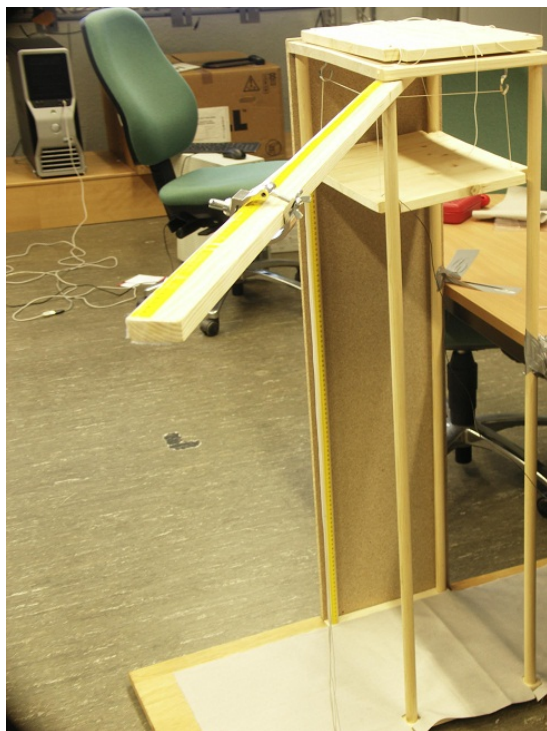
- [Erman und Lesser 1980] ERMAN, Lee D. ; LESSER, Victor R.: The HEARSAY-II speech understanding system: Integrating knowledge to resolve uncertainty. In: *Computing Surveys* 12 (1980), S. 213–253
- [Fraden 2004] FRADEN, Jacob: *Handbook of modern sensors : physics, designs, and applications*. Auflage 3. Springer Verlag, 2004. – ISBN 0-387-00750-4
- [Hardenack 2010a] HARDENACK, Frank: Bodymonitoring in Smart Homes / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-proj1/hardenack.pdf>, 2010. – Forschungsbericht. [Projektbericht zum Master-Projekt 1, Sommersemester 2010]
- [Hardenack 2010b] HARDENACK, Frank: Das intelligente Bett - Interpretation von Schlafphasen als Beispiel für Bodymonitoring im Living Place Hamburg / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/Hardenack/bericht.pdf>, 2010. – Forschungsbericht. [Vortrag vom 04.11.2009, AW1]
- [Hardenack 2010c] HARDENACK, Frank: Das intelligente Bett - Semantische Interpretation von Bodymonitoring Rohdaten / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw2/Hardenack/bericht.pdf>, 2010. – Forschungsbericht. [Vortrag vom 05.05.2010, AW2]
- [Hardenack 2011a] HARDENACK, Frank: Das intelligente Bett - Semantische Interpretation auf Basis kapazitiver Sensoren / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-seminar/hardenack/bericht.pdf>, 2011. – Forschungsbericht. [Vortrag vom 11.11.2010, Seminar]
- [Hardenack 2011b] HARDENACK, Frank: Kapazitive Body-Sense Funktionen in der Praxis / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/hardenack.pdf>, 2011. – Forschungsbericht. [Projektbericht zum Master-Projekt 2, Wintersemester 2010/2011]
- [infactory innovations & trade GmbH 2009] INFACOTRY INNOVATIONS & TRADE GMBH: *aXbo Schlafphasenwecker*. 2009. – URL <http://www.axbo.com>. – [Online; Stand 11. Oktober 2009]
- [Innovative Sleep Solutions 2009] INNOVATIVE SLEEP SOLUTIONS: *SLEEPTRACKER Schlafphasenwecker*. 2009. – URL <http://www.sleeptracker.de>. – [Online; Stand 11. Oktober 2009]
- [Langereis u. a. 2007] LANGEREIS, G. ; VOOGD-CLAESSEN, L. de ; SPAEPEN, A. ; SIPLIA, A. ; ROTSCH, C. ; LINZ, T.: ConText: Contactless Sensors For Body Monitoring Incorporated In Textiles. In: *Portable Information Devices, 2007. PORTABLE07. IEEE International Conference on*, may 2007, S. 1 –5

- [LPHH Homepage 2011] LPHH HOMEPAGE: *Living Place Hamburg*. 2011. – URL <http://livingplace.informatik.haw-hamburg.de/blog/>. – [Online; Stand 23. Mai 2011]
- [von Luck u. a. 2010] LUCK, Prof. Dr. K. von ; KLEMKE, Prof. Dr. G. ; GREGOR, Sebastian ; RAHIMI, Mohammad A. ; VOGT, Matthias: *Living Place Hamburg – A place for concepts of IT based modern living / Hamburg University of Applied Sciences*. URL http://livingplace.informatik.haw-hamburg.de/content/LivingPlaceHamburg_en.pdf, Mai 2010. – Forschungsbericht
- [Lund und Clarenbach 1995] LUND, Reimer ; CLARENBACH, Peter: *Das klinische Schlaflabor*. Arcis Verlag, 1995. – ISBN 3-89075-089-3
- [Nimmala u. a. 2008] NIMMALA, Venkatarama S. ; PENDERS, Julien ; VAN HYFTE, Dirk ; BRANDS, Michael ; GYSELINCKX, Bert: *Semantic Wireless Body Area Networks*. In: *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, aug. 2008, S. 3409 –3412. – ISSN 1557-170X
- [Schwanecke, Prof. Dr. Ulrich 2007] SCHWANECKE, PROF. DR. ULRICH: *Das diskrete Kalman-Filter*. 03 2007. – URL <http://www.mi.hs-rm.de/~schwan/Projects/CG/P4/Dokumentation/node16.html>. – [Online; Stand 21.10.2011; Hochschule RheinMain]
- [Singer 2011] SINGER, Emily: *A Nightshirt to Monitor Sleep*. 2011. – URL <http://www.technologyreview.com/biomedicine/37606/>. – [Online; Stand 14. Juni 2011]
- [Snyder u. a. 2011] SNYDER, Bruce ; BOSANAC, Dejan ; DAVIES, Rob: *ActiveMQ in Action*. Manning, 2011. – ISBN 1933988940
- [Speckmann 2008] SPECKMANN, Erwin-Josef: *Physiologie*. Ausgabe 5. Elsevier, Urban & Fischer Verlag, 2008. – ISBN 978-3437413186
- [Stegelmeier u. a. 2009] STEGELMEIER, Sven ; WENDT, Piotr ; VON LUCK, Kai: *iFlat - Eine dienstorientierte Architektur für intelligente Räume / HAW Hamburg*. 2009. – Forschungsbericht. Published in Proc. of the VDE2. Ambient Assisted Living Kongress mit Ausstellung, 27.-28.01.2009, Berlin
- [Taelman u. a. 2006] TAEELMAN, J. ; ADRIAENSEN, T. ; SPAEPEN, A. ; LANGEREIS, G. ; GOURMELON, L. ; VAN HUFFEL, S.: *Contactless EMG Sensors for continuous monitoring of muscle activity to prevent musculoskeletal disorders*. In: *Belgian Day on Biomedical Engineering, IEEE Benelux EMBS Symposium*, URL ftp://ftp.esat.kuleuven.ac.be/pub/SISTA/jtaelman/reports/paper_IEEE_EMBS_JoachimTaelman.pdf, 2006
- [Teske 2011a] TESKE, Philipp: *Human Fall Detection / HAW Hamburg*. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/teske.pdf>, 2011. – Forschungsbericht. [Projektbericht zum Master-Projekt 2, Wintersemester 2010/2011]

- [Teske 2011b] TESKE, Philipp: Human Fall Detection / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-seminar/teske/bericht.pdf>, 2011. – Forschungsbericht. [Vortrag vom 20.01.2011, Seminar]
- [Universität Ulm 2010] UNIVERSITÄT ULM: *SFB 62 Transregio*. 06 2010. – URL <http://www.uni-ulm.de/index.php?id=16563>. – [Online; Zugriff 17.10.2011; Stand 17.06.2010]
- [Voskuhl und Otto 2010] VOSKUHL, Sören ; OTTO, Kjell: Entwicklung einer Architektur für den Living Place Hamburg / HAW Hamburg. URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-proj1/otto_voskuhl.pdf, 2010. – Forschungsbericht. [Projektbericht zum Master-Projekt 1, Sommersemester 2010]
- [Voskuhl und Otto 2011] VOSKUHL, Sören ; OTTO, Kjell: Weiterentwicklung der Architektur des Living Place Hamburg / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master10-11-proj2/otto-voskuhl.pdf>, 2011. – Forschungsbericht. [Projektbericht zum Master-Projekt 2, Wintersemester 2010/2011]
- [Weiser 1991] WEISER, Mark: The Computer for the Twenty-First Century. In: *Scientific American* 265 (1991), Nr. 3, S. 94–104
- [Weiss 2011] WEISS, Ann-Kathrin: Smart Textiles: Entwicklung textiler Sensoren für intelligente Umgebungen am Beispiel eines Sofas / HAW Hamburg. URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/bachelor/weiss.pdf>, 2011. – Forschungsbericht. [Bachelorarbeit, 2011]
- [Wimmer u. a. 2007] WIMMER, Raphael ; KRANZ, Matthias ; BORING, Sebastian ; SCHMIDT, Albrecht: A Capacitive Sensing Toolkit for Pervasive Activity Detection and Recognition. In: *Proceedings of the Fifth Annual IEEE Conference on Pervasive Computing and Communications (PerCom)*, März 2007

A Messreihen

In diesem Abschnitt werden Messreihen der in Abschnitt 4.3.2 evaluierten Versuchsaufbauten zur Findung eines geeigneten Sensors dargestellt. Dabei wird unterschieden zwischen Aufbauten, die eine Annäherung detektieren (Aufbau 1-3) und Aufbauten, die eine Druckbelastung bzw. Durchbiegung detektieren (Aufbau 4-7). Bei der Detektion einer Annäherung wurde eine geerdete Metallplatte (Maße L x B x H: 310mm x 210mm x 1,5mm) in einer für (Weiss, 2011) eigens angefertigten Konstruktion (siehe Abbildung 39) langsam bis zur ersten deutlichen Detektion angenähert, bei einer Druckbelastung die Sensorfläche mit genau 500 Gramm Gewicht belastet.



39.1: Draufsicht



39.2: Ansicht der Metallplatte von unten

Abbildung 39: Konstruktion zur Bestimmung der Distanz der ersten Detektion einer Annäherung (Quelle: (Weiss, 2011))

A.1 Aufbau 1

Bei der originalen Elektrode von Rode Melder (Maße L x B: $\approx 500\text{mm} \times 500\text{mm}$) ist eine Detektion der Annäherung in der Messkurve bereits ab einer Entfernung von 75 Zentimetern deutlich zu erkennen. Der Messwert fällt nach Entfernung der Metallplatte wieder fast auf den Ausgangswert zurück.

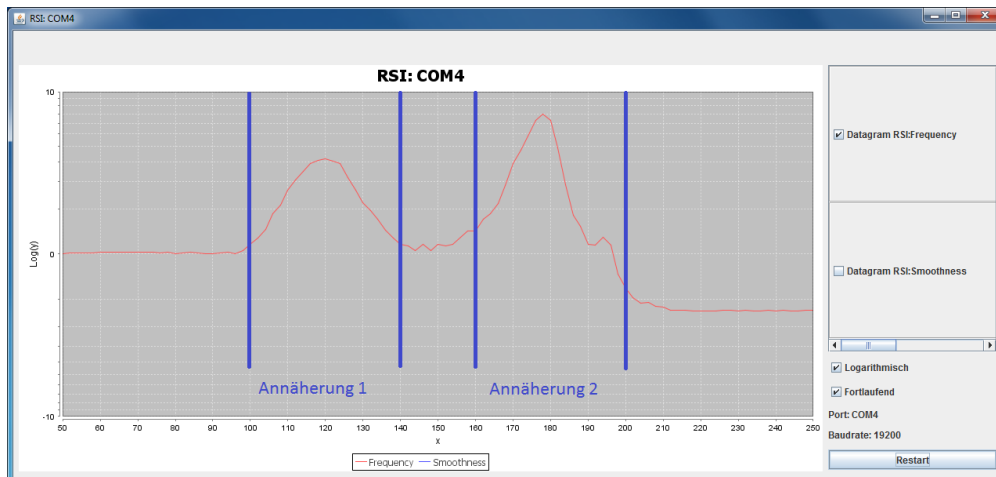


Abbildung 40: Messreihe Aufbau 1

A.2 Aufbau 2

Bei der Verwendung eines 200mm x 200mm großen Stücks Estrichmatte als Sensorfläche in Aufbau 2 war eine erste Detektion der Annäherung ab einer Höhe von 46 Zentimetern deutlich zu erkennen, auch in diesem Fall kehrte der Messwert nach Entfernung der angenäherten Metallplatte fast zum Ausgangswert zurück.

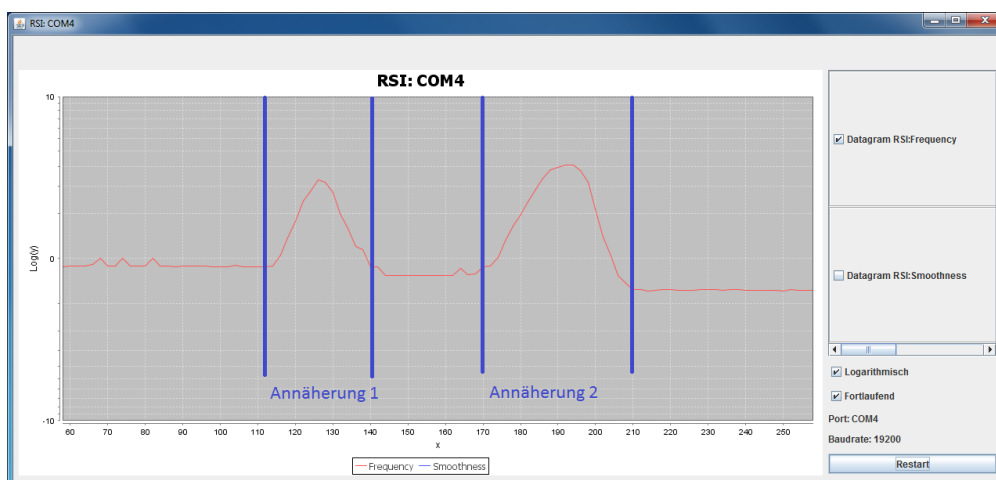


Abbildung 41: Messreihe Aufbau 2

A.3 Aufbau 3

Die Annäherung an das aktive Segment der Abschirmmatrix wird ab einer Höhe von 15 Zentimetern detektiert, es sind deutlich die Sprünge in der Messkurve zu sehen, wenn das aktive Segment abgeschaltet (geerdet) und eine anderes Segment dafür hinzugeschaltet wird.

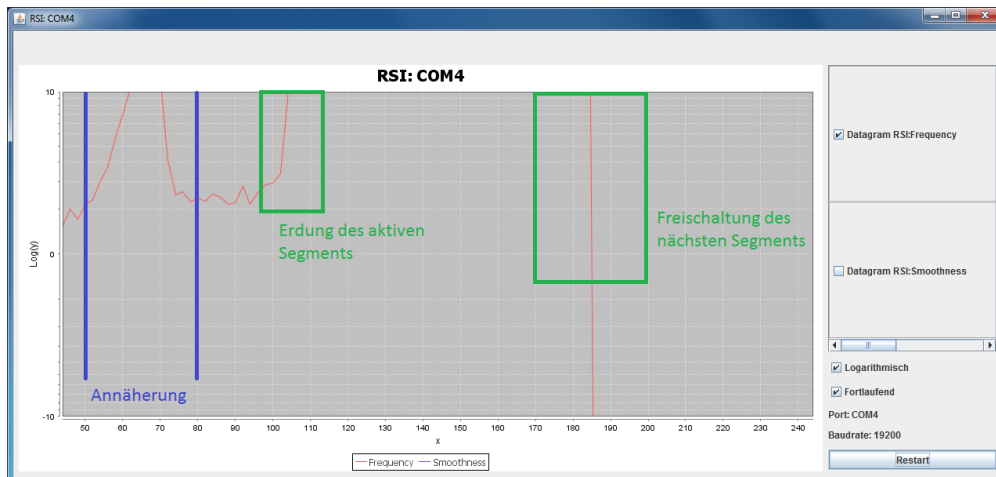


Abbildung 42: Messreihe Aufbau 3

A.4 Aufbau 4

Die Messreihe zu Aufbau vier zeigt deutlich, wie sensibel der Aufbau auf Druck reagiert. Das Versuchsgewicht von 500 Gramm führt zu einem sehr starken Ausschlag auf der logarithmischen Skala der Visualisierungssoftware, der Messwert normalisiert sich im Anschluß auch nach Entfernung des Versuchsgewichts nicht mehr.

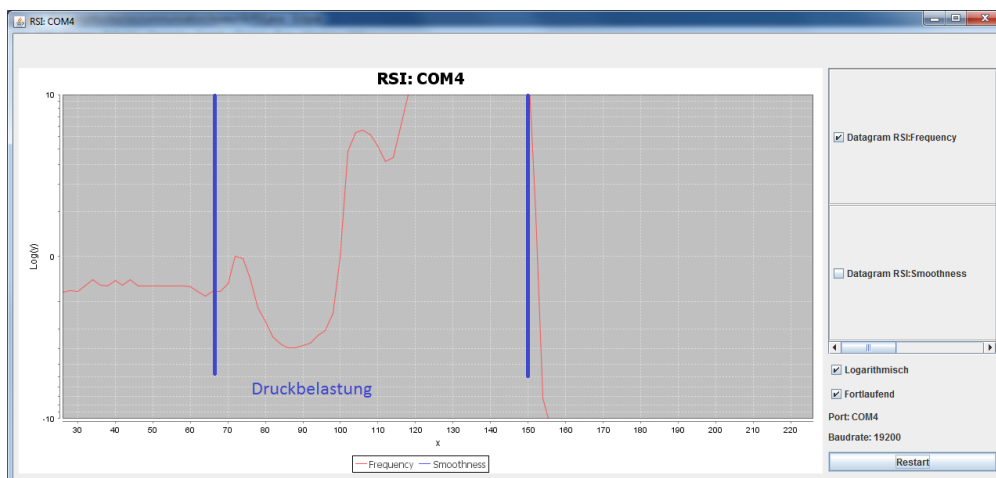


Abbildung 43: Messreihe Aufbau 4

A.5 Aufbau 5

Die stabilere Version des Sensors in Sandwich-Bauweise aus Aufbau 5 ist bei Weitem nicht mehr so empfindlich, wie die vorangegangene Version aus Aufbau 4. Bei dem Versuchsgewicht ist nach wie vor eine deutliche, wenn auch nicht mehr ganz so starke Auslenkung der Messkurve zu erkennen. Auch in diesem Aufbau normalisiert sich der Messwert bei Entfernung des Versuchsgewichts nicht vollständig.

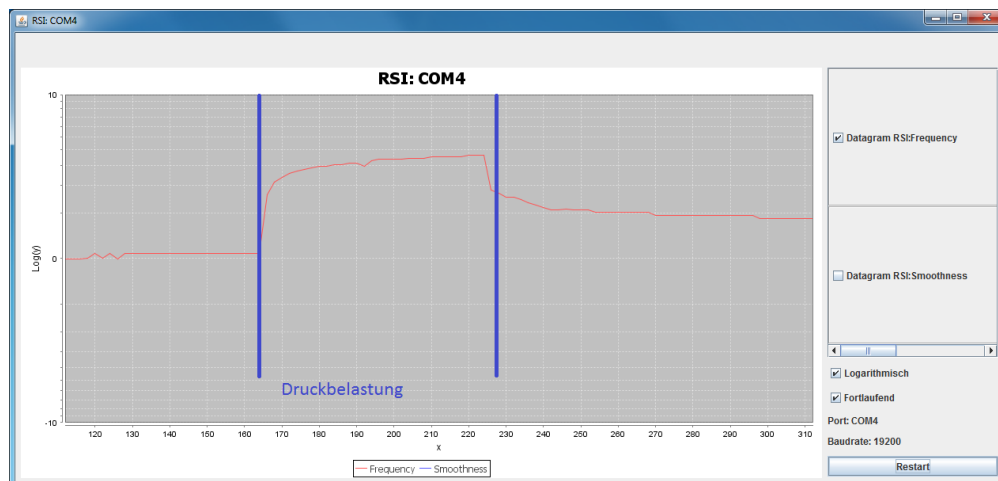


Abbildung 44: Messreihe Aufbau 5

A.6 Aufbau 6

Die modifizierte Latte eines Lattenrosts zeigt in ihrer ersten Version aus Aufbau 6 gute Ergebnisse bei der Belastung mit dem Versuchsgewicht, die Schirmung aus leitfähigem Gewebe verschiebt sich aber bei Belastung bis hin zum Kontakt mit der Sensorfläche, was zu wilden Sprüngen in den Messwerten führt.

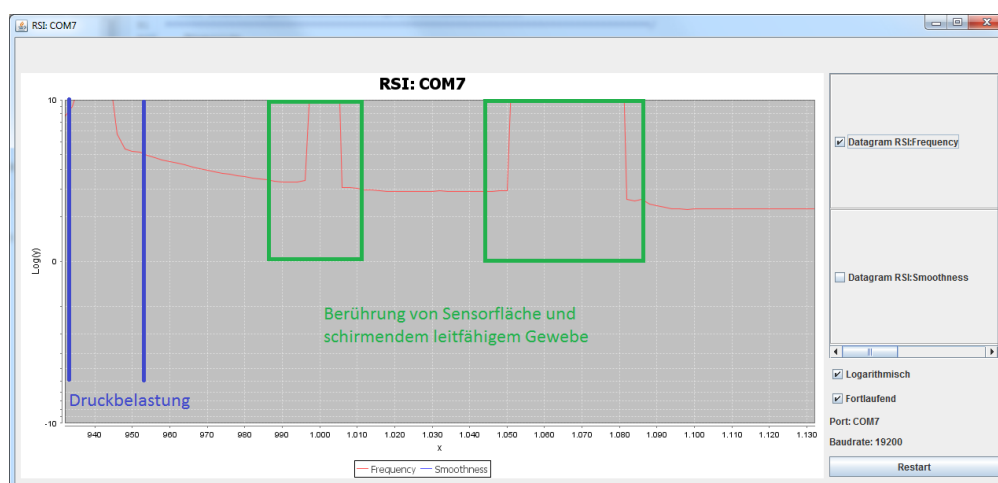


Abbildung 45: Messreihe Aufbau 6

A.7 Aufbau 7

In der verbesserten Version der modifizierten Latte des Lattenrosts in Aufbau 7 sind klare Auslenkungen bei der Belastung mit dem Versuchsgewicht zu erkennen. Auch eine Normalisierung nach Entfernung des Gewichts ist zu sehen und eine Verfälschung der Messwerte durch eine nicht ausreichend befestigte Schirmung tritt nicht auf.

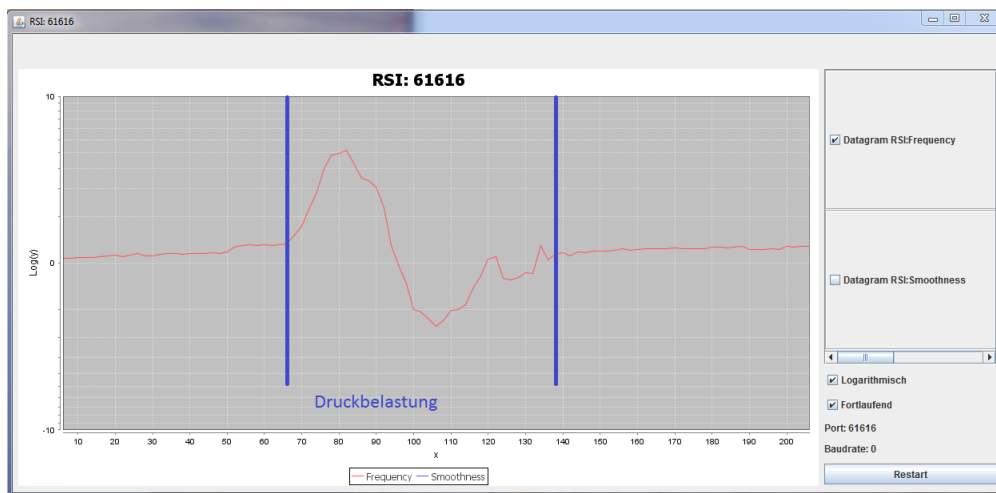


Abbildung 46: Messreihe Aufbau 7

B Messwerte einer ganzen Nacht

In diesem Abschnitt werden beispielhaft die erfassten Daten einer gesamten Nacht in einem intelligenten Bett dargestellt. Charakteristische Ereignisse sind durch die grünen Markierungen einmalig kenntlich gemacht worden. Die Aufzeichnung und Verarbeitung der Daten erfolgte unter Einsatz verschiedener Programme:

- Mit der Visualisierungssoftware von Oliver Dreschke(Abschnitt 5.4) wurden die an den drei aufgebrauchten Sensoren befestigten RSI Schwingkreiserzeuger ausgelesen und deren Messwerte auf einen lokalen ActiveMQ abgelegt.
- Neben dem Ablegen der Daten auf dem ActiveMQ wurden diese ebenfalls durch den Living Place Wrapper in einer MongoDB persistiert, um die Daten mehrfach wieder einspielen zu können (zum Beispiel mit dem MongoDB-Player von Jens Ellenberg).
- Die auf dem ActiveMQ abgelegten Daten wurden von der SmartBed-Software verarbeitet und mit den in Abschnitt 6.3 beschriebenen Verfahren vorbereitet. Somit ist aus der Datenmenge von ca. 1 Million Datensätzen für eine Nacht eine überschaubare Anzahl von ca. 64.000 Messwerten je Sensor entstanden.
- Die durch die SmartBed-Software entstandenen Daten wurden als CSV-Datei exportiert und dann in ein Tabellenkalkulationsprogramm importiert und als Diagramm dargestellt.

Zu der aufgezeichneten Nacht und den dargestellten Kurven sind noch folgende Eigenschaften zu erwähnen:

Beginn der Aufzeichnung	01:00 Uhr
Ende der Aufzeichnung	10:00 Uhr
Gesamtschlafdauer	9:00 Std
Sensor auf Schulterhöhe	Rote Kurve (Com14)
Sensor auf Hüfthöhe	Gelbe Kurve (Com15)
Sensor auf Kniehöhe	Blaue Kurve (Com12)

Tabelle 2: Eigenschaften einer beispielhaft aufgezeichneten Nacht

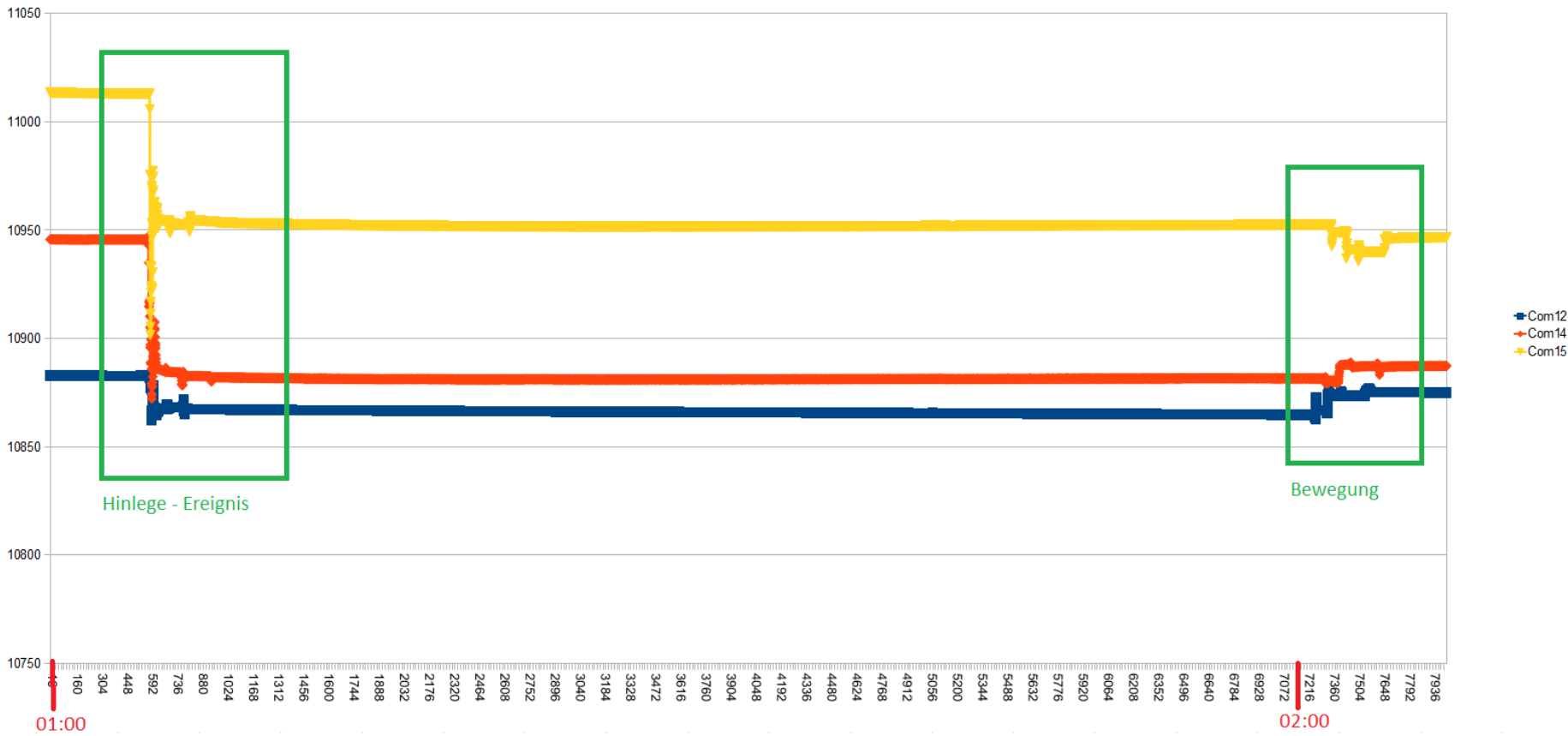


Abbildung 47: Messwerte einer ganzen Nacht (Messwert 0-8000)

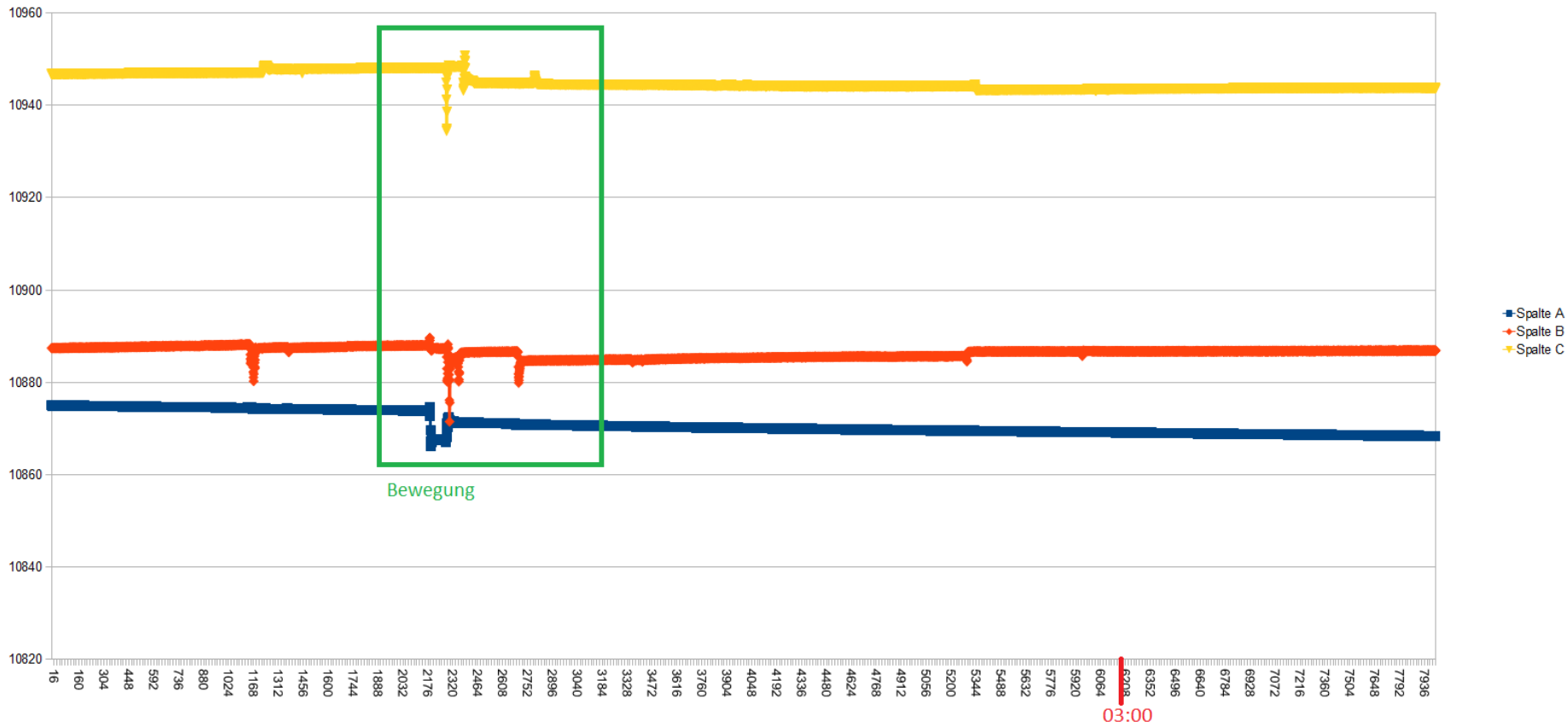


Abbildung 48: Messwerte einer ganzen Nacht (Messwert 8001-16000)

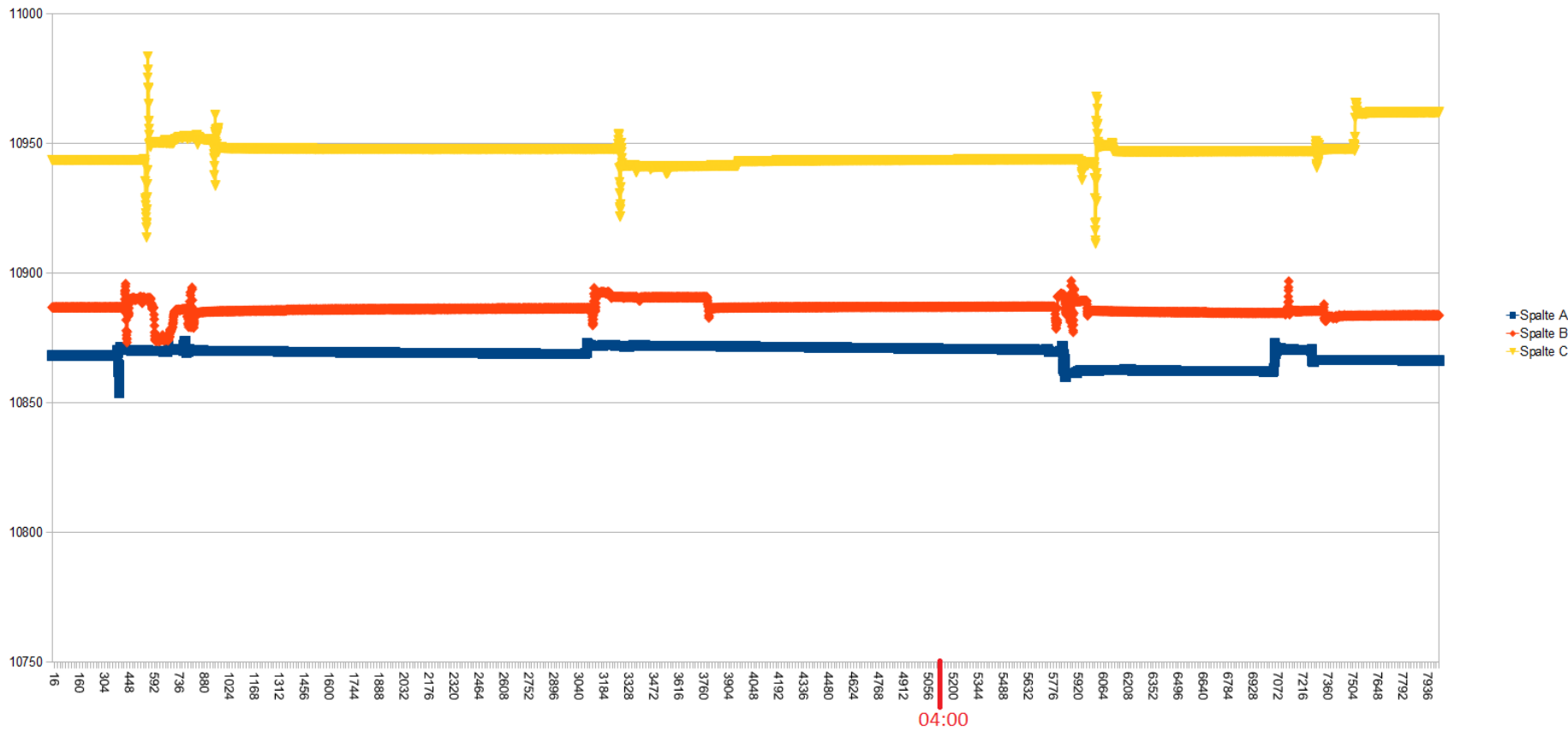


Abbildung 49: Messwerte einer ganzen Nacht (Messwert 16001-24000)

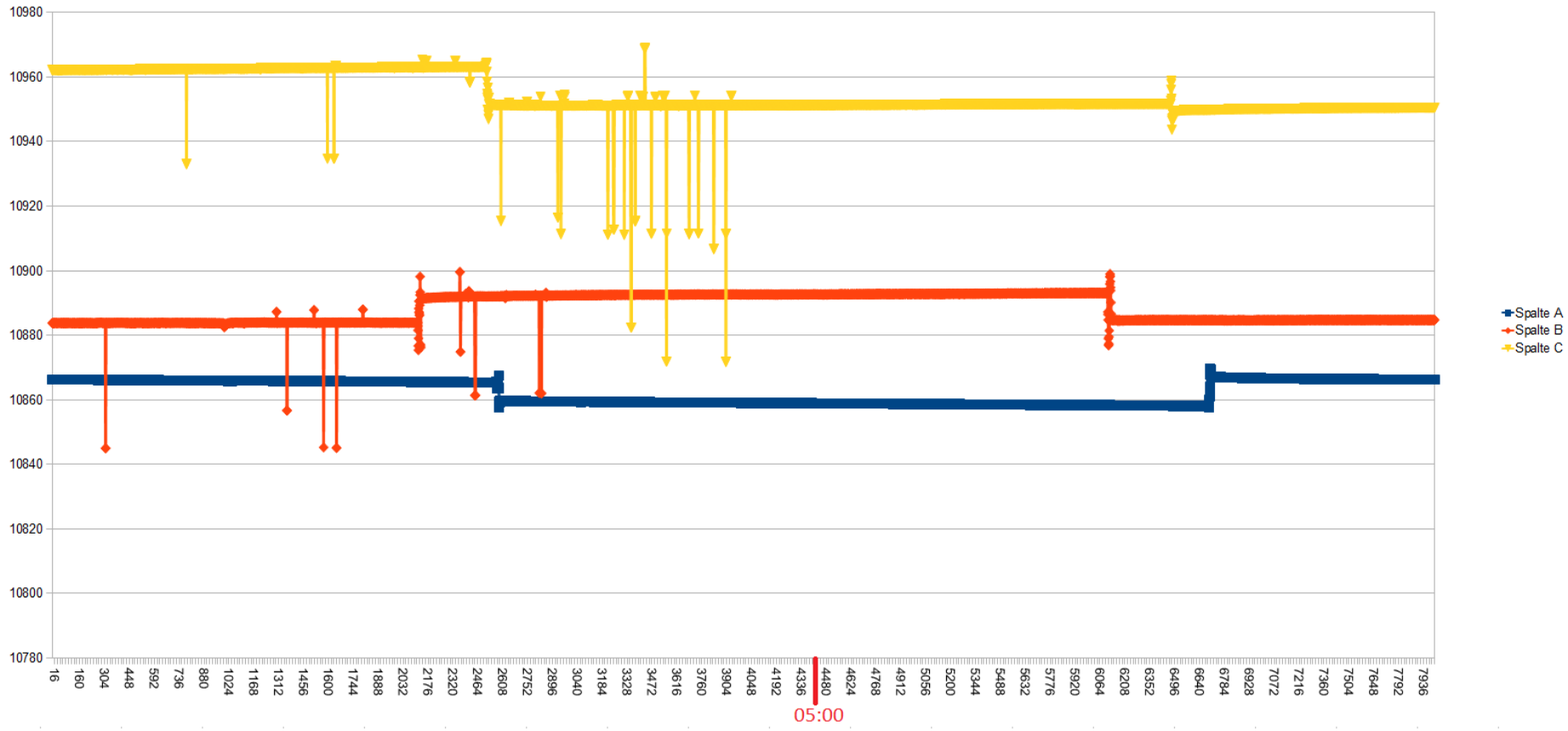


Abbildung 50: Messwerte einer ganzen Nacht (Messwert 24001-32000)

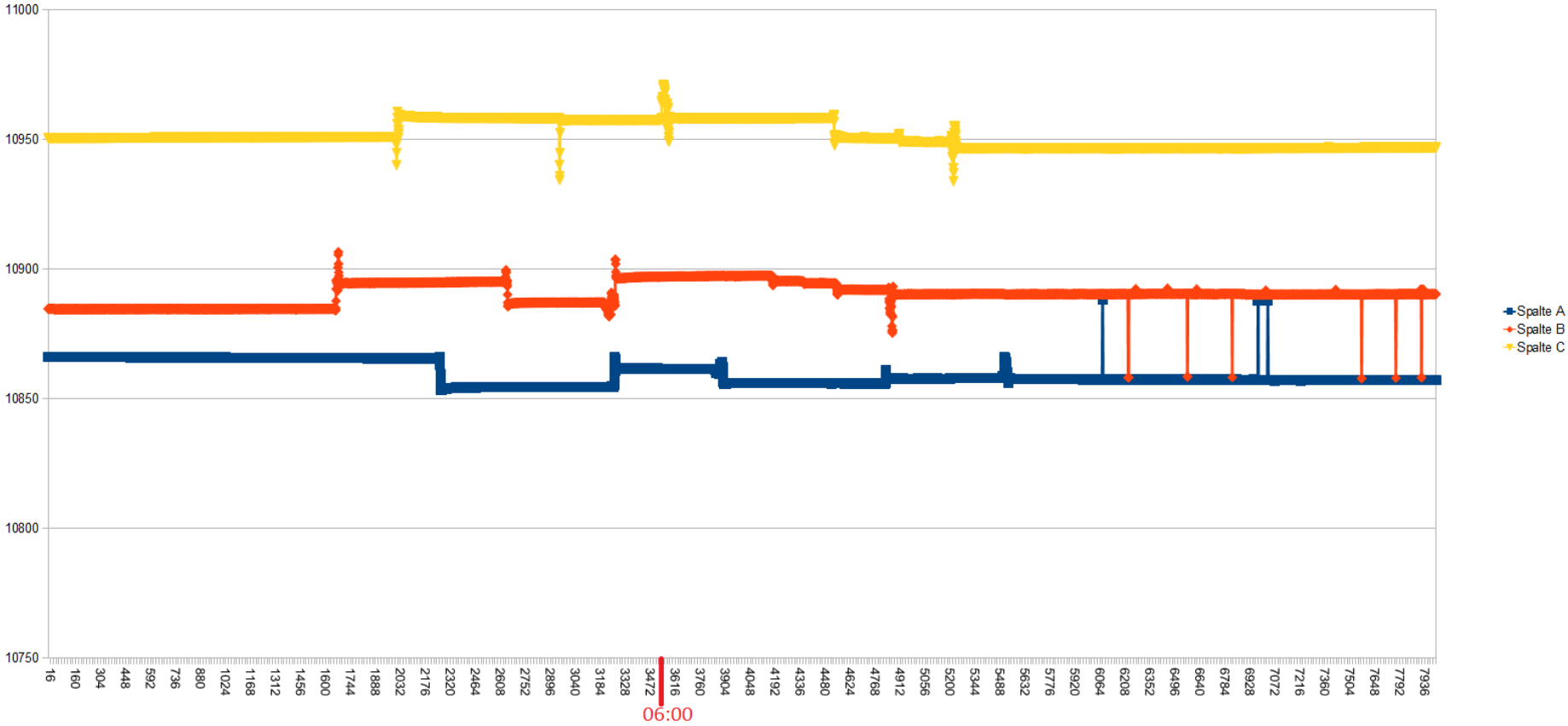


Abbildung 51: Messwerte einer ganzen Nacht (Messwert 3201-4000)

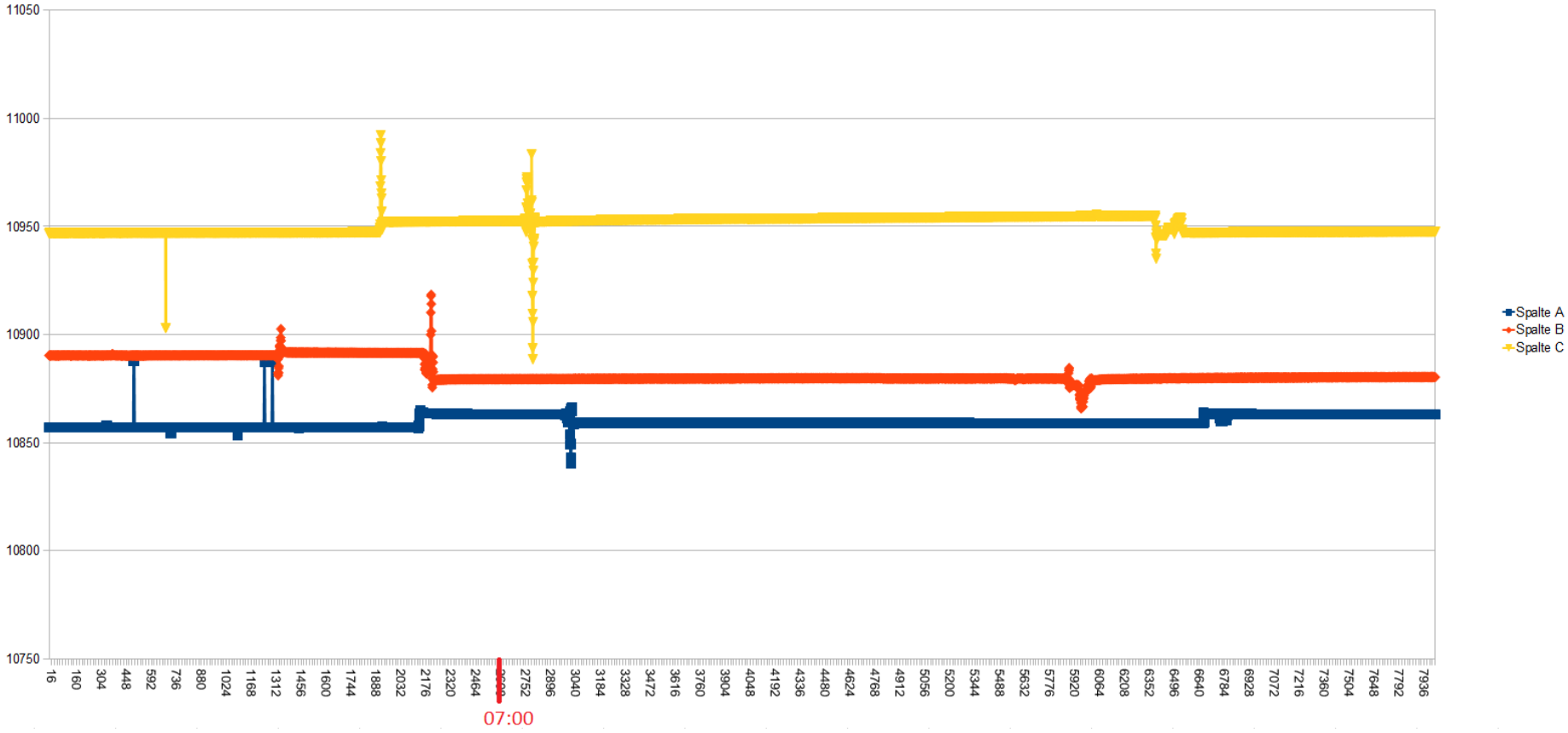


Abbildung 52: Messwerte einer ganzen Nacht (Messwert 40001-48000)

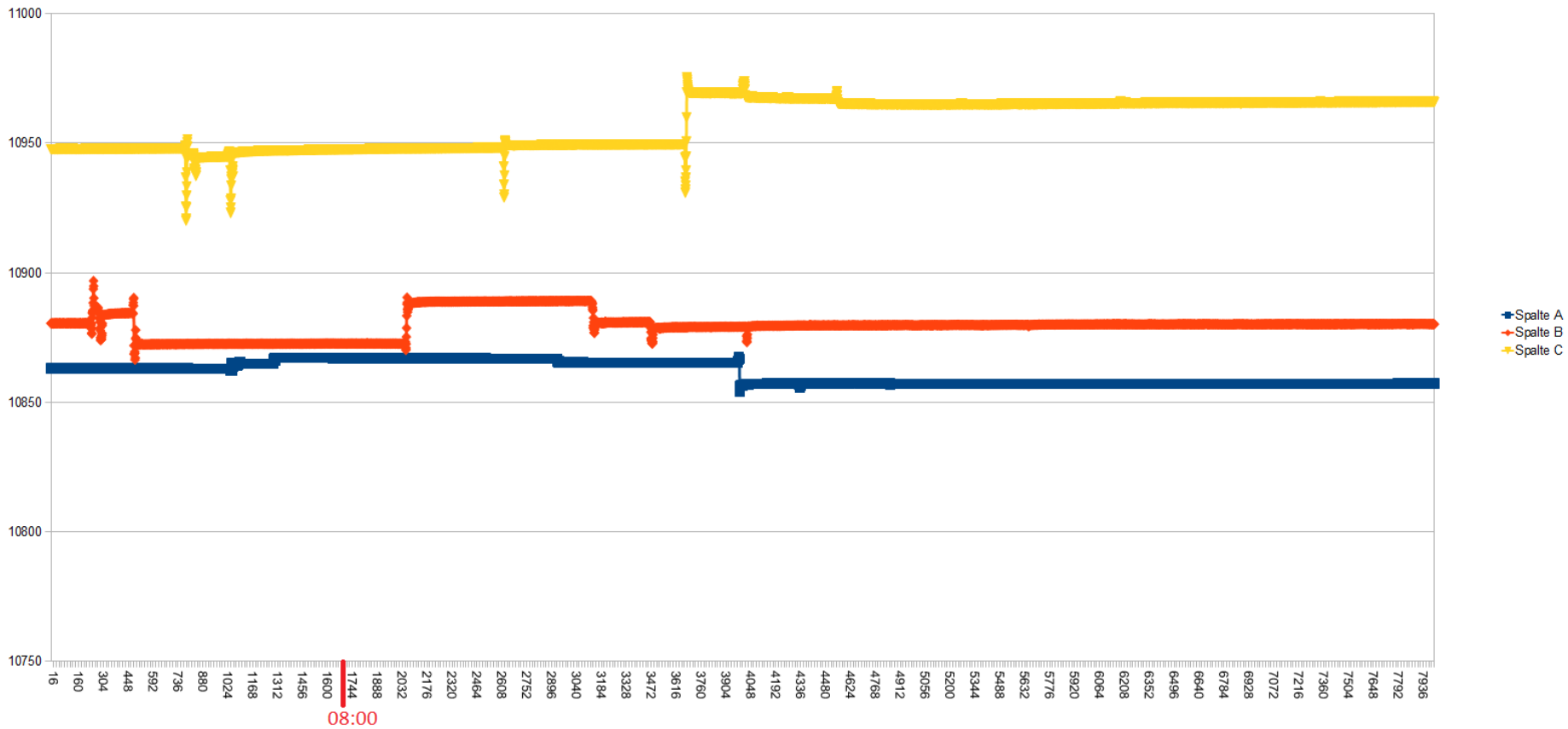


Abbildung 53: Messwerte einer ganzen Nacht (Messwert 48001-56000)

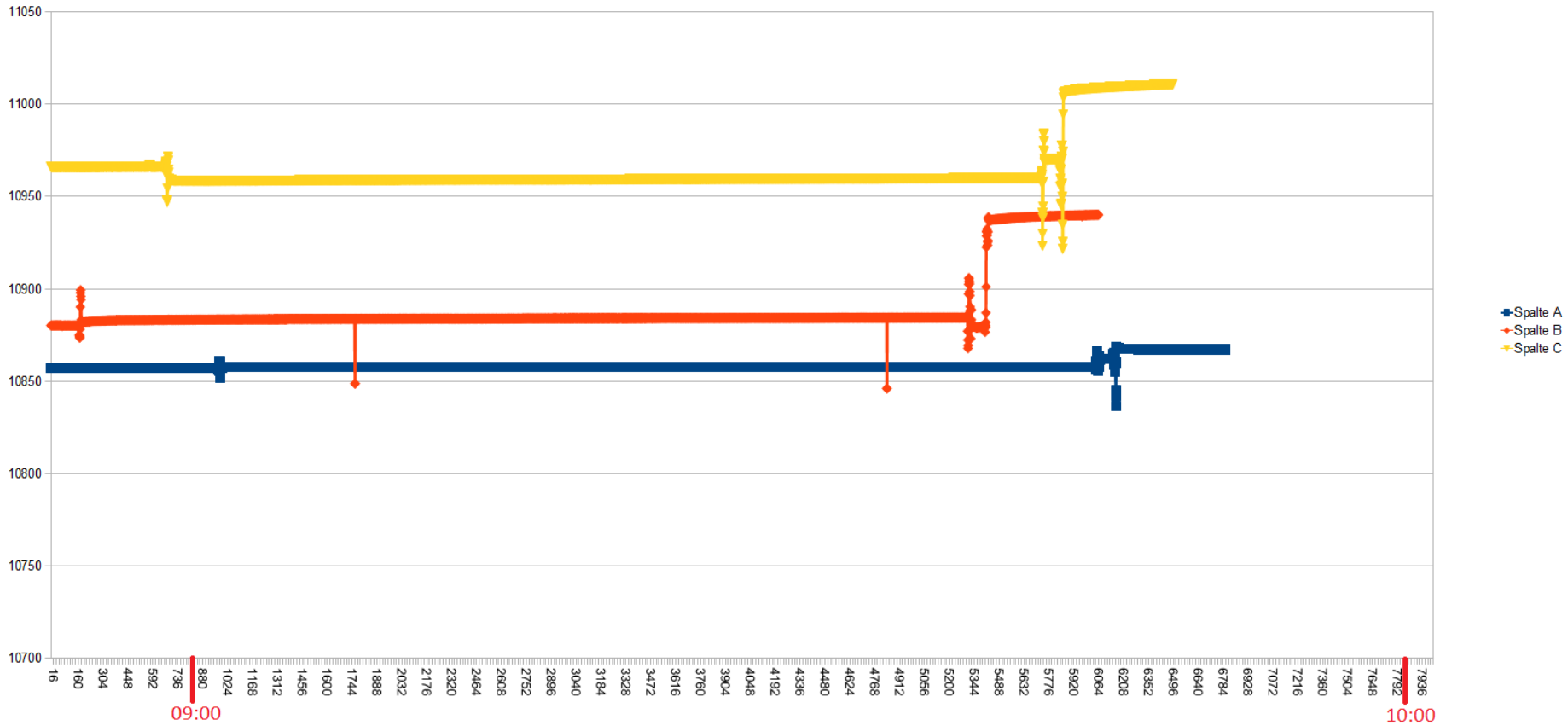


Abbildung 54: Messwerte einer ganzen Nacht (Messwert 56001-64000)

C Konfiguration der Software

In diesem Abschnitt wird die Konfiguration der Software beschrieben, um eine spätere Anpassung an neue Gegebenheiten (z.B.: Änderung der Adresse des ActiveMQ) zu ermöglichen. Als standardisiertes Eingabeformat werden .csv-Dateien verwendet.

Konfiguration der Consumer und Producer

activemqinputconfig.csv		
Eintrag	Datentyp	Dokumentation
<i>activeMQAddress</i>	String	Die Adresse, unter der der zu verwendende ActiveMQ zu erreichen ist.
<i>queueOrTopicName</i>	String	Ein String zur Festlegung des Queue- bzw. Topicnamens.
<i>topic</i>	boolean	Ein Boolean-Flag, das eine Aussage darüber trifft, ob der jeweilige Eintrag ein Topic (true) oder eine Queue (false) sein soll.
<i>buffer</i>	boolean	Ein Boolean-Flag, das eine Aussage darüber trifft, ob ein Ringpuffer verwendet werden soll. Voreinstellung: true
<i>buffersize</i>	int	Gibt an, wie groß der verwendete Ringpuffer sein soll. Voreinstellung: 10

Tabelle 3: Konfigurationsoptionen für die Consumer

activemqoutputconfig.csv		
Eintrag	Datentyp	Dokumentation
<i>activeMQAddress</i>	String	Die Adresse, unter der der zu verwendende ActiveMQ zu erreichen ist.
<i>queueOrTopicName</i>	String	Ein String zur Festlegung des Queue- bzw. Topicnamens.
<i>topic</i>	boolean	Ein Boolean-Flag, das eine Aussage darüber trifft, ob der jeweilige Eintrag ein Topic (true) oder eine Queue (false) sein soll.
<i>layer</i>	int	Layer, welches in das Topic / die Queue publizieren soll (nur ein Eintrag je Layer möglich).

Tabelle 4: Konfigurationsoptionen für die Producer

Konfiguration der Interpretationsebenen

Die einzelnen Interpretationsebenen verfügen über einige „Stellschrauben“, die nachfolgend erläutert werden. Für die Konfiguration der Ebenen gibt es je Ebene eine .csv-Datei.

layer0config.csv		
Eintrag	Datentyp	Dokumentation
<i>calculationMethod</i>	String	<p>Berechnungsmethode für die Messwerte. Es stehen folgende Berechnungsverfahren zur Verfügung:</p> <ul style="list-style-type: none"> • Standardabweichung („<i>standarddeviation</i>“) • Mittelwert („<i>average</i>“) • Varianz („<i>variance</i>“) • Streuung („<i>scatter</i>“) <p>Wird die Methode nicht erkannt, so wird „<i>standarddeviation</i>“ gewählt. Voreinstellung: „<i>standarddeviation</i>“</p>
<i>leaveOutMeasurements</i>	int	<p>Intervall, in dem die Berechnung mit der gewählten Berechnungsmethode durchgeführt wird. Dies ist sinnvoll, da nicht alle 100ms eine Berechnung durchgeführt werden muss. Voreinstellung: 5</p>

Tabelle 5: Konfigurationsoptionen für die Interpretationsebene 0

layer1config.csv		
Eintrag	Datentyp	Dokumentation
<i>numberOfMeasurements</i>	int	Anzahl der Messwerte, die zur Kalibrierung (Findung des initialen Mittelwerts) der Sensoren verwendet werden. Voreinstellung: 5
<i>lowerLaydownBorder</i>	double	Untere Schranke zur Festlegung, ob ein Messwert nahe dem initialen Mittelwert ist (also in dem Intervall ((initialer Mittelwert - lowerLaydownBorder),(initialer Mittelwert + upperLaydownBorder)) liegt) oder nicht. Voreinstellung: 15.0
<i>upperLaydownBorder</i>	double	Obere Schranke zur Festlegung, ob ein Messwert nahe dem initialen Mittelwert ist (also in dem Intervall ((initialer Mittelwert - lowerLaydownBorder),(initialer Mittelwert + upperLaydownBorder)) liegt) oder nicht. Voreinstellung: 15.0
<i>measurementTrigger</i>	double	Auslöseschwelle, ab der eine Statusveränderung angenommen wird. Der Wert bezieht sich auf das Ergebnis des Berechnungsverfahrens aus Ebene 0. Voreinstellung: 2.0
<i>assumeEmptyCount</i>	int	Anzahl der Messwerte, die jeder Sensor eines Lattenrosts in Ruheposition (initialer Mittelwert des Sensors) verbraucht haben muss, damit bei einem nicht korrekt erkannten Aufstehereignis das Bett anschließend als leer angenommen wird. Voreinstellung: 8

Tabelle 6: Konfigurationsoptionen für die Interpretationsebene 1

layer2config.csv		
Eintrag	Datentyp	Dokumentation
<i>noMovementCount</i>	int	Anzahl der Messergebnisse aus Ebene 1, bei denen es, je Sensor, keine erkennbare Bewegung geben darf, damit eine Bewegungsphase als abgeschlossen gilt. Voreinstellung: 4
<i>movementWaitingCount</i>	int	Anzahl der Messwerte, die maximal zwischen der Registrierung einer Bewegung durch den ersten und einen weiteren Sensor liegen dürfen, damit Bewegungsphase angenommen wird. Wird dieser Wert überschritten, so liegt wahrscheinlich ein fehlerhafter Messwert eines Sensors und somit keine Bewegungsphase vor. Voreinstellung: 4

Tabelle 7: Konfigurationsoptionen für die Interpretationsebene 2

layer3config.csv		
Eintrag	Datentyp	Dokumentation
<i>timePerSegment</i>	int	Länge des zu betrachtenden Intervalls in Sekunden. Voreinstellung: 3600

Tabelle 8: Konfigurationsoptionen für die Interpretationsebene 3

layer4config.csv		
Eintrag	Datentyp	Dokumentation
<i>minimumSleepLength</i>	int	Minimale Länge des Nachtschlafs in Minuten, damit der Schlaf als erholsamer interpretiert wird. Voreinstellung: 360
<i>averageSleepLength</i>	int	Durchschnittliche Länge des Nachtschlafs in Minuten, damit der Schlaf als erholsamer interpretiert wird. Voreinstellung: 480
<i>maximumNoOfMovements</i>	int	Maximale Anzahl der Bewegungsphasen in einer Nacht, damit der Schlaf als erholsamer interpretiert wird. Voreinstellung: 5
<i>timeBetweenMovements</i>	int	Minimale Zeit zwischen Bewegungsphasen in Minuten, die dann als erholsamer Tiefschlaf interpretiert wird und sich auf die Bewertung der Schlafqualität auswirkt. Voreinstellung: 90

Tabelle 9: Konfigurationsoptionen für die Interpretationsebene 4

Starten und Beenden der Software

Um die Software korrekt starten und beenden zu können ist es wichtig, dass eine bestimmte Reihenfolge eingehalten wird und die gestarteten Komponenten wieder korrekt beendet werden.

Starten der Anwendung:

- ActiveMQ starten (*Desktop/MA/starte_ActiveMQ_5_4_2.bat*)
- MongoDB starten (*Desktop/MA/starte_MongoDB.bat*)
- FrequenzMonitor starten (*Eclipse-Projekt, MainClass: SensorDeviceMonitor*) und mit der Konfiguration
 - Boxtype: RSI
 - GUI-Style: lan Server.ActiveMQ

unter einzelner Auswahl des COM-Ports jeden Sensor starten.

- Die Smartbed-Software starten (*Eclipse-Projekt, MainClass: SmartbedUI*) und die Sensorpositionen konfigurieren (Abbildung 36.1).

Beenden der Anwendung:

- FrequenzMonitor und Smartbed schließen.
- Das CMD-Fenster des ActiveMQ **NICHT** direkt schließen, sondern über *Strg + C* und anschließende Bestätigung den Batch-Prozess beenden.
- Das CMD-Fenster der MongoDB **NICHT** direkt schließen, sondern über *Strg + C* und anschließende Bestätigung den Batch-Prozess beenden.

Anmerkungen:

- Sollten bei der FrequenzMonitor-Software nicht alle Sensoren (COM-Ports) erkannt und zur Kommunikation mit dem ActiveMQ gestartet werden (öffnet sich für einen COM-Port kein eigenes Popup), so kann dieser Sensor durch die Auswahl des jeweiligen COM-Ports und erneutes Drücken auf „Start“ für den gleichen Modus nachgestartet werden.
- Die Smartbed-Software muss sich kalibrieren, weshalb sich für den Start und die ersten 30 Sekunden die Sensoren in unbelasteter Ruhelage befinden müssen.
- Sollen die Daten der MongoDB aufgehoben werden (zum Beispiel für den MongoDB-Player), so müssen diese vor dem nächsten Start der MongoDB aus dem Verzeichnis *C:\data\db* kopiert werden, da beim Start der MongoDB alle vorhandenen Dateien in diesem Verzeichnis gelöscht werden.
- Auf dem MacMini: Benutzerkonto: smartBed, Passwort: smartBed

D Publizierung der Interpretationsergebnisse

Die einzelnen Ebenen stellen über den ActiveMQ des Living Place ihre Ergebnisse zur Verfügung. Die bereitgestellten Ergebnisse werden in ein Objekt der Klasse SmartbedPublishingResult (siehe Listing 6) verpackt und mit Gson serialisiert, bevor sie auf den ActiveMQ abgelegt werden.

Da die Ebenen drei und vier mit zeitlich einzuordnenden Intervallen arbeiten, wurde für diese eine Linked List implementiert (siehe Listing 7).

Listing 6: Die Klasse SmartbedPublishingResult

```
1 package de.fh.ma.interpreter ;
2
3 import java.util.HashMap;
4 import java.util.Map;
5
6 public class SmartbedPublishingResult<T> {
7
8     private boolean left;
9     private Map<String , T> result = new HashMap<String , T>();
10
11     //default constructor to satisfy Gson
12     public SmartbedPublishingResult() {
13     }
14
15     public SmartbedPublishingResult(boolean left) {
16         this.left = left;
17     }
18
19     public T getResult(String key) {
20         return result.get(key);
21     }
22
23     public void addResult(String key, T result){
24         this.result.put(key, result);
25     }
26
27     public boolean isLeft() {
28         return left;
29     }
30
31     public void setLeft(boolean left) {
32         this.left = left;
33     }
34
35 }
```

Listing 7: Die Klasse SmartbedLinkedList

```
1 package de.fh.ma.interpreter;
2
3 public class SmartbedLinkedList {
4
5     private SmartbedLinkedList child = null;
6
7     private long starttime = 0;
8     private long endtime = 0;
9
10    private boolean movementPhase = false;
11    private boolean phaseCompleted = false;
12
13    public SmartbedLinkedList getChild() {
14        return child;
15    }
16
17    public void setChild(SmartbedLinkedList child) {
18        this.child = child;
19    }
20
21    public boolean isMovementPhase() {
22        return movementPhase;
23    }
24
25    public void setMovementPhase(boolean movementPhase) {
26        this.movementPhase = movementPhase;
27    }
28
29    public boolean isPhaseCompleted() {
30        return phaseCompleted;
31    }
32
33    public void setPhaseCompleted(boolean phaseCompleted) {
34        this.phaseCompleted = phaseCompleted;
35    }
36
37    public long getStarttime() {
38        return starttime;
39    }
40
41    public void setStarttime(long starttime) {
42        this.starttime = starttime;
43    }
44
45    public long getEndtime() {
46        return endtime;
47    }
48
```



```

49 public void setEndtime(long endtime) {
50     this.endtime = endtime;
51 }
52
53 public SmartbedLinkedList addChild() {
54     child = new SmartbedLinkedList();
55     child.setStarttime(System.currentTimeMillis());
56     return child;
57 }
58
59 public long getDurationInMillis() {
60     return endtime - starttime;
61 }
62
63 public boolean hasChild() {
64     return child != null;
65 }
66 }

```

Ergebnisse der einzelnen Ebenen

In diesem Abschnitt werden die Ergebnisse der einzelnen Ebenen genauer dargestellt. Bei den Ergebnissen handelt es sich um Einträge in einem Objekt der Klasse *SmartbedPublishingResult*.

Eintrag	Dokumentation
<i>left</i>	Ein Flag, das kennzeichnet, ob es sich um den linken (true) oder rechten (false) Lattenrost handelt. (<i>Boolean</i>)
<i>position</i>	Ein String zur Angabe der Sensorposition. Diese kann sein: <ul style="list-style-type: none"> • „<i>shoulder</i>“ • „<i>hip</i>“ • „<i>knee</i>“ (<i>String</i>)
Einträge in der Result-Map	
Key	Value
<i>measurement</i>	Der durch einen Mittelwert über ein floating window bereinigte Messwert des Sensors.
<i>calcResult</i>	Das Ergebnis des vorkonfigurierten (Siehe Tabelle 5) Berechnungsverfahrens.

Tabelle 10: Ergebnisse der Interpretationsebene 0

Eintrag	Dokumentation
<i>left</i>	Ein Flag, das kennzeichnet, ob es sich um den linken (true) oder rechten (false) Lattenrost handelt. (<i>Boolean</i>)
<i>position</i>	Ein String zur Angabe der Sensorposition. Diese kann sein: <ul style="list-style-type: none"> • „<i>shoulder</i>“ • „<i>hip</i>“ • „<i>knee</i>“ (<i>String</i>)
Einträge in der Result-Map	
Key	Value
<i>result</i>	Ein String mit der Angabe der vorliegenden Aktion. Es können folgende Aktionen auftreten: <ul style="list-style-type: none"> • „<i>SITDOWN</i>“ • „<i>LAYDOWN</i>“ • „<i>MOVEMENT</i>“ • „<i>NO MOVEMENT</i>“ • „<i>STANDUP</i>“ (<i>String</i>)

Tabelle 11: Ergebnisse der Interpretationsebene 1

Einträge in der Result-Map	
Key	Value
<i>result</i>	Ein String mit der Angabe über die vorliegende Aktion. Es können folgende Aktionen auftreten: <ul style="list-style-type: none"> • „<i>LAYDOWN</i>“ • „<i>MOVEMENT START</i>“ • „<i>MOVEMENT END</i>“ • „<i>STANDUP</i>“ (<i>String</i>)

Tabelle 12: Ergebnisse der Interpretationsebene 2

Einträge in der Result-Map	
Key	Value
<i>result</i>	Eine <i>SmartbedLinkedList</i> , entweder über die gesamte Länge eines vorkonfigurierten Intervalls (siehe Tabelle 8) oder bis zum Auftreten eines Aufstehereignisses. (<i>SmartbedLinkedList</i>)

Tabelle 13: Ergebnisse der Interpretationsebene 3

Einträge in der Result-Map	
Key	Value
<i>result</i>	<p>Ein <i>String</i>, der auf Basis der vorgenommenen Konfiguration für die Ebene 4 aussagt, wie die Schlafqualität angenommen wird und ob es sich um den linken oder rechten Lattenrost handelt. Es können dabei auftreten:</p> <ul style="list-style-type: none"> • „Good“ • „Medium“ • „Bad“ <p>Der angenommenen Schlafqualität ist ein „Left“ oder „Right“, getrennt durch einen Bindestrich, nachgestellt (Beispiel „Good-Left“). (<i>String</i>)</p>

Tabelle 14: Ergebnisse der Interpretationsebene 4

Auslesen der Daten vom ActiveMQ

In Listing 8 wird beispielhaft gezeigt, wie man das serialisierte Objekt vom ActiveMQ liest und wieder in ein `SmartbedPublishingResult` umwandelt. Das Beispiel zeigt ein `SmartbedPublishingResult` mit `String` als generischem Typ.

Listing 8: Beispielhaftes Lesen vom ActiveMQ

```
1 import java.lang.reflect.Type
2 import com.google.gson.Gson
3 import com.google.gson.reflect.TypeToken
4 import javax.jms.TextMessage
5
6 // ...
7 TextMessage textmessage = getTextMessageFromAnywhere();
8 Gson gson = new Gson();
9
10 //Example for having a generic SmartbedPublishingResult with generic
11 //type "String"
12 Type resultType = new TypeToken<SmartbedPublishingResult<String>>()
13     {}.getType();
14 SmartbedPublishingResult<String> result = gson.fromJson(textmessage.
15     getText(), resultType);
16 // ...
```

Tabellenverzeichnis

1	Bewertung evaluierter Sensoren	30
2	Eigenschaften einer beispielhaft aufgezeichneten Nacht	VI
3	Konfigurationsoptionen für die Consumer	XV
4	Konfigurationsoptionen für die Producer	XV
5	Konfigurationsoptionen für die Interpretationsebene 0	XVI
6	Konfigurationsoptionen für die Interpretationsebene 1	XVII
7	Konfigurationsoptionen für die Interpretationsebene 2	XVIII
8	Konfigurationsoptionen für die Interpretationsebene 3	XVIII
9	Konfigurationsoptionen für die Interpretationsebene 4	XVIII
10	Ergebnisse der Interpretationsebene 0	XXII
11	Ergebnisse der Interpretationsebene 1	XXIII
12	Ergebnisse der Interpretationsebene 2	XXIII
13	Ergebnisse der Interpretationsebene 3	XXIV
14	Ergebnisse der Interpretationsebene 4	XXIV

Abbildungsverzeichnis

1	Hypnogramm	5
2	3D Modell des Living Place	7
3	Impressionen aus dem Living Place	7
	3.1 Küche	7
	3.2 Wohnbereich	7
	3.3 Essbereich	7
	3.4 Schlafbereich	7
4	Sleeptracker	8
5	aXbo-Wecker	9
6	Logo der Sleep-Cycle App	9
7	Elektroden zur Schlafanalyse im Schlaflabor	10
8	ConText EMG-Sensor	11
9	Weste des ConText-Projekts	12
	9.1 Beispiel: Fliesenleger	12
	9.2 Beispiel: Informatiker	12
10	sWBAN Sensor	13
11	sWBAN Lookup-Tabelle	13
12	Übersicht der Funktionsweise des sWBAN	15
13	Sensor zur Sturzerkennung	16
14	Übersicht der Funktionsweise des Human Status Awareness Projekts	17
15	Somnus	17
16	Informationsgehalt einer Drehbewegung anhand der Rohdaten	19
17	Cx-1 von RSI	21
18	EMC von Rode Melder	22
19	CapSense Board	22

20	Elektrode von Rode Melder	23
	20.1 Plattenelektrode von Rode Melder (Aufbau 1)	23
	20.2 Modifizierte Elektrode (Aufbau 2)	23
21	Elektrode von Rode Melder (Aufbau 3)	24
	21.1 Rode Melder Plattenelektrode (klein)	24
	21.2 Abschirmmatrix	24
22	Sensor in Sandwich-Bauweise & Aufbau (Aufbau 4)	25
	22.1 Gesamtansicht	25
	22.2 Aufbau	25
23	Sensor in Sandwich-Bauweise 2 & Aufbau (Aufbau 5)	26
	23.1 Gesamtansicht	26
	23.2 Gesamtansicht (offen)	26
	23.3 Übersicht Aufbau	26
	23.4 Aufbau	26
24	Modifizierter Lattenrost & Aufbau (Aufbau 6)	27
	24.1 Draufsicht	27
	24.2 Übersicht Aufbau	27
25	Entwickelter Lattenrost-Sensor (Aufbau 7)	28
	25.1 Lattenrost	28
	25.2 Tragschicht	28
	25.3 Schirmung	28
	25.4 Trennschicht	28
	25.5 Kabelverbindung	28
	25.6 Sensorschicht	28
26	Prototyp im Projektraum	30
27	Liegepositionen	31
28	Bett im Living Place	31
29	Verarbeitung der Sensordaten	33
30	Darstellung der Sensordaten	34
31	Beispiel für die verschiedenen Ebenen der Interpretation	34
32	Rohdaten einer Umdrehbewegung (Seiten- in Rückenlage, logarithmisch aufgetragen)	40
	32.1 Schulter	40
	32.2 Hüfte	40
	32.3 Knie	40
33	Over- und Underfitting	41
34	Visualisierungssoftware	41
	34.1 Log-Fenster	41
	34.2 Chart-Fenster	41
35	Bett mit Sensoren	42
36	Die Smartbed-Software	44
	36.1 Auswahl der Sensoren	44
	36.2 Das Log-Fenster	44
	36.3 Live-Darstellung der Ergebnisse	44
	36.4 Konfiguration der Interpretationsebenen	44

37	Informationsfluss zwischen den einzelnen Interpretationsebenen	45
38	Impressionen vom intelligenten Bett	55
	38.1 Das intelligente Bett	55
	38.2 Person im Bett, Licht aus	55
	38.3 Person aufgestanden, Licht an	55
39	Konstruktion zur Bestimmung der Distanz der ersten Detektion einer An- näherung	I
	39.1 Draufsicht	I
	39.2 Ansicht der Metallplatte von unten	I
40	Messreihe Aufbau 1	II
41	Messreihe Aufbau 2	II
42	Messreihe Aufbau 3	III
43	Messreihe Aufbau 4	III
44	Messreihe Aufbau 5	IV
45	Messreihe Aufbau 6	IV
46	Messreihe Aufbau 7	V
47	Messwerte einer ganzen Nacht (Messwert 0-8000)	VII
48	Messwerte einer ganzen Nacht (Messwert 8001-16000)	VIII
49	Messwerte einer ganzen Nacht (Messwert 16001-24000)	IX
50	Messwerte einer ganzen Nacht (Messwert 24001-32000)	X
51	Messwerte einer ganzen Nacht (Messwert 32001-40000)	XI
52	Messwerte einer ganzen Nacht (Messwert 40001-48000)	XII
53	Messwerte einer ganzen Nacht (Messwert 48001-56000)	XIII
54	Messwerte einer ganzen Nacht (Messwert 56001-64000)	XIV

Listings

1	Algorithmus der Ebene 0 (Pseudocode)	46
2	Algorithmus der Ebene 1 (Pseudocode)	47
3	Algorithmus der Ebene 2 (Pseudocode)	49
4	Algorithmus der Ebene 3 (Pseudocode)	50
5	Algorithmus der Ebene 4 (Pseudocode)	52
6	Die Klasse SmartbedPublishingResult	XX
7	Die Klasse SmartbedLinkedList	XXI
8	Beispielhaftes Lesen vom ActiveMQ	XXV

Index

- AAL, 1, 15, 57
- ActiveMQ, VI, XV, 6, 39, 53
- aXbo, 9

- CapSense, 22
- Circadianer Rhythmus, 4
- Computational Furniture, 6
- Consumer, XV, 53
- ConText, 11
- Context Aware Computing, 6
- Context Awareness, 1
- CSV, VI, XV

- Feature, 32
- FTDI, 42

- Gson, XX, 53

- JESS, 37, 54
- JMS, 6

- Kontext, 6, 18
 - Provider, 18

- Lattenrost, 25
- Linked List, XX
- Living Place, 7
 - Wrapper, VI, 6, 39, 43
- Logging, 43

- MacMini, 42
- Mittelwert, 32
- MongoDB, VI, XIX, 39, 43
 - Player, VI, 43
- Muskeltonus, 4

- Observable, 46
- Observer, 45

- Producer, XV, 53

- Reset, 43
- Rode Melder, 22
- RSI, VI, 21, 39, 42, 56
- Rule Engine, 37, 54

- Schlaf
 - labor, 9
 - phase, 1, 4, 18
 - phasenwecker, 8
- Schwingkreis, 5
 - erzeuger, 21, 42, 54
- Sleep-Cycle, 9
- Sleeptracker, 8
- Smart
 - Environment, 1, 54
 - Furniture, 6
 - Home, 1, 54
- Somnus, 17
- Standardabweichung, 32
- Streuung, 32
- Sturzerkennung, 16
- SuperCSV, 45
- SVM, 15, 37
 - Overfitting, 38
 - Underfitting, 38
- sWBAN, 12

- Thread, 43
 - verwaltung, 43

- Ubiquitous Computing, 1, 7
- USB, 42

- Varianz, 32

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der APSO-TI-BM (Stand 16.11.2006) nach §16(5) ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 31. Oktober 2011

 Frank Hardenack