

Karlsruhe Reports in Informatics 2015,5

Edited by Karlsruhe Institute of Technology,
Faculty of Informatics
ISSN 2190-4782

Ubiquitäre Systeme (Seminar) und Mobile Computing (Proseminar) WS 2014/15

Mobile und Verteilte Systeme
Ubiquitous Computing

Teil XII

Herausgeber:
Martin Alexander Neumann, Anja Bachmann,
Yong Ding, Till Riedel

2015



Fakultät für **Informatik**

Please note:

This Report has been published on the Internet under the following
Creative Commons License:

<http://creativecommons.org/licenses/by-nc-nd/3.0/de>.

**Ubiquitäre Systeme (Seminar)
und
Mobile Computing (Proseminar)
WS 2014/15**

Mobile und Verteilte Systeme
Ubiquitous Computing
Teil XII

Herausgeber

Martin Alexander Neumann

Anja Bachmann

Yong Ding

Till Riedel

**Karlsruhe Institute of Technology (KIT)
Fakultät für Informatik
Lehrstuhl für Pervasive Computing Systems (PCS) und TECO**

**Interner Bericht 2015-05
ISSN 2190-4782**

Vorwort

Die Seminarreihe Mobile Computing und Ubiquitäre Systeme existiert seit dem Wintersemester 2013/2014. Seit diesem Semester findet das Proseminar Mobile Computing am Lehrstuhl für Pervasive Computing System statt. Die Arbeiten des Proseminars werden seit dem mit den Arbeiten des zweiten Seminars des Lehrstuhls, dem Seminar Ubiquitäre Systeme, zusammengefasst und gemeinsam veröffentlicht.

Die Seminarreihe Ubiquitäre Systeme hat eine lange Tradition in der Forschungsgruppe TECO. Im Wintersemester 2010/2011 wurde die Gruppe Teil des Lehrstuhls für Pervasive Computing Systems. Seit dem findet das Seminar Ubiquitäre Systeme in jedem Semester statt. Ebenso wird das Proseminar Mobile Computing seit dem Wintersemester 2013/2014 in jedem Semester durchgeführt. Seit dem Wintersemester 2003/2004 werden die Seminararbeiten als KIT-Berichte veröffentlicht. Ziel der gemeinsamen Seminarreihe ist die Aufarbeitung und Diskussion aktueller Forschungsfragen in den Bereichen Mobile und Ubiquitous Computing.

Dieser Seminarband fasst die Arbeiten der Seminare des Wintersemesters 2014/15 zusammen. Die Themenvielfalt der hier zusammengetragenen Aufsätze umfasst Dynamische Software Updates für Java, menschliche Aktivitätserkennung, Affective Computing und Emotion Sensing, User Experience von Open-Source-Software und -Hardware, sowie die nicht-invasive Lastüberwachung von Geräten in Haushalten. Wir danken den Studierenden für ihren besonderen Einsatz, sowohl während des Seminars als auch bei der Fertigstellung dieses Bandes.

Karlsruhe, den 31. März 2015

Martin Alexander Neumann
Anja Bachmann
Yong Ding
Till Riedel

Inhaltsverzeichnis

Christopher Baumgärtner

Affective Computing, Emotion Sensing und Sentimental Mining:
Überblick und Stand der Technik 1

Thomas Kadow

Ablauf Dynamischer Software Updates in Java
mit dem Ansatz der Lazy-Zustandstransformation 17

Christian Herlemann

Possibilities of Activity Recognition on Apple's iOS 35

Josef Roth

Die Rolle von User Experience für Open Source Software 46

Mathias Kleiner

Human Behavior and Non-Intrusive Load Monitoring 55

Inna Belyantseva

Betrachtung des Einsatzes von Ontologien in der Aktivitätserkennung 68

Özgün Arziman

Design von Studien mithilfe der Experience Sampling Method
Grundlagen, Einsatzgebiete, Stärken und Schwächen 84

Affective Computing, Emotion Sensing und Sentimental Mining: Überblick und Stand der Technik

Christopher Baumgärtner*

Betreuer: Anja Bachmann[†]

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*christopher.baumgaertner@student.kit.edu

[†]bachmann@teco.edu

Zusammenfassung. In dieser Ausarbeitung wird ein Überblick über das Forschungsfeld des Affective Computing gegeben. Dabei wird sowohl auf grundlegende Begriffe wie Emotion, Stimmung und Affekt eingegangen, als auch auf die unterschiedlichen Bereiche im Affective Computing. Die Bereiche Emotion Sensing, Mood Sensing und Sentimental Mining werden charakterisiert und verglichen.

Es werden aktuelle Projekte wie StressSense [10] und MoodScope [9] vorgestellt, die den affektiven Zustand des Benutzers auf Basis von Sprache bzw. Smartphone-Nutzung erkennen. Dabei wird unter Anderem auf verwendete Sensoren, Features und Klassifikatoren eingegangen. Die Ergebnisse der Projekte werden vorgestellt.

Am Ende wird ein Ausblick zur aktuellen Forschung gegeben.

Schlüsselwörter: Affective computing, emotion sensing, sentimental mining, mood, mobile computing, wearables, smartphone

1 Einleitung

Im Laufe der letzten Jahre haben sich Smartphones zu leistungsfähigen tragbaren Computern mit einer Vielzahl von Sensoren entwickelt. Diese Sensoren können unter Anderem dafür verwendet werden, die emotionale Situation eines Benutzers zu erkennen. Affective Computing (AC) beschäftigt sich mit Systemen, die diese Emotionen aufnehmen und verarbeiten.

Im Zuge der Entwicklung solcher Systeme sind verschiedene Probleme zu lösen. Das erste Problem besteht darin, den emotionalen Zustand des Benutzers zu erfassen. Die Forschung hat in den vergangenen Jahren unterschiedliche Ansätze für dieses Problem hervorgebracht. Es wird unter Anderem durch die Verarbeitung von Gesichtsausdrücken [3], Stimmanalyse [10], sozialen Medien [8,11] oder Eingabegesten [2] versucht, die Emotionen zu erkennen und zuzuordnen.

Diese Ansätze werden immer häufiger auf mobilen Geräten eingesetzt. Daraus ergibt sich ein großer Vorteil: Die Benutzer tragen ihre Smartphones nahezu

den ganzen Tag am Körper. Damit bilden diese eine sehr gute Basis um den Affekt des Benutzers zu erfassen und daraus einen Mehrwert für diesen zu erzeugen. Beispielsweise könnte eine Musik-App die aktuelle Wiedergabe an den momentanen Gefühlszustand anpassen oder die Gefühle des Benutzers durch eine gezielte Auswahl an Songs beeinflussen.

Das folgende Kapitel dieser Seminararbeit beschäftigt sich mit den grundlegenden Begriffen, die im Forschungsfeld des AC verwendet werden. In Kapitel 3 werden drei Ansätze mit den verwendeten Methoden und Sensoren vorgestellt. Das letzte Kapitel dient der Zusammenfassung.

2 Hintergrund

In diesem Kapitel werden grundlegende Begriffe wie Emotion (emotion), Stimmung (mood) und Affekt (affect) erklärt. Des Weiteren wird ein Überblick über einige Unterkategorien des AC gegeben.

2.1 Affekt, Stimmung, Emotion

Affekt ist der allgemeinste Begriff aus der Reihe von Stimmung, Emotion und Affekt und kann als übergeordnete Kategorie betrachtet werden [5]. Er ist ein bewusster neurophysiologischer Zustand. Affekt kann als ein nicht weiter verarbeitetes Gefühl, das den Emotionen und Stimmungen zugrunde liegt, verstanden werden. Er hat keinen Bezug zu einem Ereignis, sondern ist ein primitiver und universeller Zustand. Trotz des durchgängigen Vorhandenseins von Affekt tritt er zeitweise in das Bewusstsein des Menschen oder ist im Hintergrund der bewussten Umwelt angesiedelt.

Russell [17] beschreibt Affekt als zweidimensionales Objekt, das sich in die Dimensionen Behagen und Erregung teilen lässt. Die erste Dimension erstreckt sich von Leid über einen neutralen, ausgeglichenen Zustand bis zur Ekstase. Die zweite Dimension beginnt bei Verschlafenheit und geht über verschiedene Zustände der Aufmerksamkeit bis zu fanatischer Aufregung. Verwendet man die erste Dimension als horizontale Achse und die Dimension der Erregung als vertikale Achse, spannen diese zwei Dimensionen einen Raum auf. In diesem affektiven Raum wird der aktuelle affektive Zustand als Punkt repräsentiert.

Betrachtet man Abbildung 1 sind die affektiven Zustände nahe dem Zentrum eher neutral, während die Randbereiche des Kreises die extremen Affekte beinhalten. Sowohl die Intensität als auch die Frequenz, mit der die Zustände wechseln, haben Einfluss darauf, wie sehr der aktuelle affektive Zustand in das Bewusstsein des Menschen rückt. Stabilisiert sich der Zustand, rückt der Affekt eher in den Hintergrund, als ein schnell wechselnder Affekt.

Stimmung (engl. mood) ist ein affektiver Zustand, der eher unbewusst ist. Er hat Ähnlichkeit zu einem anderen affektiven Zustand, der Emotion. Beide Begriffe können als Unterkategorie des Affekt betrachtet werden. Nowlis und Nowlis [12] haben als Erste eine Unterscheidung der beiden Begriffe vorgenommen und die Eigenschaften der Dauer und Intensität als maßgebende Kriterien

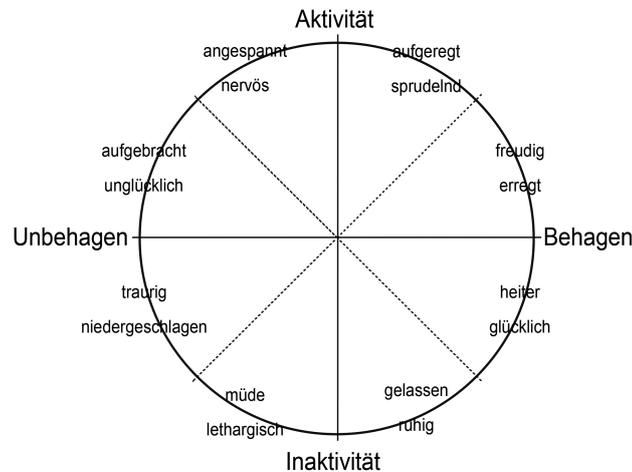


Abb. 1. Affekt als zweidimensionaler Raum (basiert auf [17])

identifiziert. Sie stellen fest, dass eine Stimmung ein anhaltender Zustand und in ihrer Intensität eher schwach ist. Eine **Emotion**, im Gegensatz dazu, ist ein kürzer andauernder affektiver Zustand der jedoch eine höhere Intensität hat. Emotionen werden durch spezielle Ereignisse ausgelöst. Dadurch können sie sich sehr schnell entwickeln und verändern. Sie sind dem Menschen durch ihre höhere Intensität eher bewusst. Eine Stimmung hingegen entwickelt sich aus einer Reihe von Ereignissen und festigt sich mit der Zeit. Sie hält über Stunden oder Tage an. Stimmungen sind grundlegende Affekte, die meistens unbewusst sind. Zur Beschreibung von Emotionen stellen Mehrabian und Russell 1977 das PAD Modell vor [18]. Das PAD Modell beschreibt mit drei Dimensionen alle emotionalen Zustände. Als Dimensionen wählen Mehrabian und Russell dazu Behagen (**P**leasure), Erregung (**A**rousal) und Dominanz (**D**ominance). Die Dominanz-Dimension repräsentiert dabei die kontrollierenden und dominierenden Eigenschaften einer Emotion. Wut und Angst sind beide unerfreuliche Emotionen. Wut ist jedoch eine sehr dominierende, Angst eine unterwürfige Emotion.

2.2 Affective Computing

Die Forschung im Bereich AC gründet auf Rosalind W. Picards Veröffentlichung "Affective Computing" aus dem Jahre 1995 [15]. Picard fasst unter diesem Begriff alle informationstechnischen Systeme zusammen, die menschliche Affekte verarbeiten können. Die Probleme, die dafür gelöst werden müssen, erstrecken sich von der Aufnahme, über die Interpretation bis hin zum Ausdrücken von Emotionen. Im Laufe der Zeit haben sich verschiedene Gebiete entwickelt, die unterschiedliche Ansätze für diese Probleme verwenden. Im Folgenden werden

drei Ansätze näher vorgestellt. Dabei wird sowohl auf die Gemeinsamkeiten, als auch auf die Unterschiede und Grenzen eingegangen.

2.3 Emotion Sensing

Der Bereich Emotion Sensing (ES) beschäftigt sich mit der Aufnahme und Verarbeitung der menschlichen Emotionen durch unterschiedliche Sensoren. Petrushin [14] verwendet in seinem Ansatz Mikrofone, die die Sprache der Personen aufnehmen. Einen anderen Ansatz verwenden Coutrix et.al. [2]. Sie leiten die Emotionen des Benutzers aus Gesten ab. Dabei verwenden sie die zweidimensionalen Wischgesten und dreidimensionalen Bewegungen des Smartphones als Eingabe. Im Allgemeinen werden aus den gewonnenen Daten, mit Hilfe von Machine Learning Verfahren, Modelle trainiert, die die Emotionen der Nutzer klassifizieren. Die Systeme im ES sind darauf ausgelegt, den affektiven Zustand des Benutzers in kurzen Abständen zu erfassen. Dadurch können auch kurzweilige Emotionen aufgenommen werden. Die Datenerfassung erfolgt oft mit zwei Methoden. Die erste Variante erfasst Emotionen situationsgebunden. Dadurch werden kurzweilige, intensive Emotionen erfasst, die durch ein bestimmtes Ereignis ausgelöst werden. Bei der zweiten Variante werden die Daten in regelmäßigen Abständen erhoben, um die weniger intensiven Emotionen festzuhalten. Die Daten werden oft in Studien gesammelt. Da Emotionen kurzweilige Affekte sind und mehrmals pro Tag erfasst werden, ist eine lange Studiendauer von einem Monat oder mehr nicht nötig. Häufig wird das PAD Modell zur Beschreibung der Emotionen verwendet. Es werden jedoch auch Modelle verwendet, die einen kategorisierenden Ansatz verwenden. Ekman stellte 1972 eine Liste mit 6 Basisemotionen vor: Wut, Ekel, Furcht, Fröhlichkeit, Traurigkeit und Überraschung. Diese Kategorien werden immer noch verwendet, jedoch teilweise erweitert.

2.4 Mood Sensing

Mood Sensing (MS) ist die Erfassung der Stimmung eines Benutzers durch Sensoren oder auf Basis von Text Mining Algorithmen. Die verwendeten Daten für die Text Mining Ansätze stammen aus sozialen Netzwerken, SMS, E-Mails oder Blogs [11]. LiKamWa et. al. [9] verwenden Informationen über die Nutzung des Smartphones eines Benutzers als Datenquelle. Sie beachten dabei die Häufigkeit und Länge von Anrufen, SMS, E-Mails, App Nutzung, Browsing und Standort. Die Inhalte der Textnachrichten werden dabei nicht beachtet. Auch im MS wird oft maschinelles Lernen genutzt, um die Modelle der Anwendungen zu trainieren. Im Gegensatz zum ES wird jedoch die grundlegende Stimmung des Benutzers erfasst. Die kurzweiligen, intensiven Emotionen werden meist gefiltert, so dass eine grundlegende Stimmung bestimmt werden kann. Die Erfassung der Daten erfolgt meistens 1-4 mal pro Tag. Durch die geringere Frequenz, mit der Daten erhoben werden, haben die Studien längere Laufzeiten als im ES. Die verwendeten Modelle sind den Modellen im ES ähnlich. Die Intensität der Werte ist jedoch geringer, da Stimmungen grundlegende Affekte mit geringer Intensität sind.

2.5 Sentiment Analysis/Opinion Mining

Sentiment Analysis (SA) beschäftigt sich mit dem Erfassen der allgemeinen Stimmung einer Personengruppe in Bezug auf ein bestimmtes Thema [20]. Für den Begriff Sentiment Analysis wird ebenso der Begriff Opinion Mining verwendet. Ein großes Anwendungsgebiet der SA liegt im Marketing. Sie liefert beispielsweise Informationen darüber, welche Produkte beliebt sind. Ebenso kann überprüft werden, welche Funktionen eines Produktes erwünscht sind und welche nicht. Die benötigten Daten werden hauptsächlich aus öffentlichen Blogs, Kommentaren, Erfahrungsberichten oder anderen Texten gewonnen, die im Internet verfasst werden.

SA teilt sich in drei Unterkategorien auf [20]:

- Sentiment Classification beschäftigt sich mit der Klassifikation von kompletten Dokumenten in Bezug zu bestimmten Produkten oder Themen.
- Feature-based Sentiment Classification geht auf die einzelnen Eigenschaften eines Produktes ein.
- Opinion Summarization ist ähnlich zu Text Summarization. Dabei wird aus einem vorhandenen Text ein neuer, zusammenfassender Text generiert. Diese Zusammenfassung soll die wichtigen Informationen in verkürzter Form enthalten. Opinion Summarization unterscheidet sich von Text Summarization dadurch, dass nur Informationen in Bezug zum Thema gesammelt werden. Andere Informationen werden nicht übernommen.

Im Gegensatz zu Emotion Sensing und Mood Sensing geht es nicht um die Emotionen oder Stimmung eines Einzelnen. Das Ziel ist es, ein allgemeines Meinungsbild einer Personengruppe zu erfassen.

3 Aktuelle Methoden und verwendete Sensorik

Im Folgenden werden drei aktuelle Ansätze im Bereich AC vorgestellt, die weit verbreitete Verfahren verwenden. Es wird auf die Anwendungsbereiche, die Durchführung und die verwendeten Methoden zum maschinellen Lernen eingegangen. Am Ende werden die Resultate diskutiert.

3.1 Stress Sensing anhand der Stimme

StressSense [10] ist ein Projekt zur Erkennung des Stress-Level eines Benutzers durch die **Analyse der Stimme** in unterschiedlichen akustischen Umgebungen. Es werden unter anderem Smartphones verwendet, die am Körper angebracht sind und die Gespräche der Personen sowie die Umgebungsgeräusche aufnehmen. Dafür wurde ein Prototyp der StressSense App entwickelt und auf den Smartphones installiert.

Um Daten zu erheben, wurde eine Studie mit mehreren Schritten durchgeführt.

Der erste Schritt ist die Datenerhebung unter Stress. Diese wurde in zwei Szenarien durchgeführt. Das erste Szenario ist ein Vorstellungsgespräch und das zweite

Szenario eine Marketingaktion im Freien. In einem weiteren Schritt werden Daten in einer neutralen Situation erhoben, in der die Probanden nicht unter Stress stehen. Es sind insgesamt 14 Probanden an der Studie beteiligt. Die Teilnehmer sind Studenten verschiedener Fachrichtungen und ein Doktorand.

Die Teilnehmer absolvieren zuerst das Vorstellungsgespräch, um sich für einen Platz in der, mit 200 CHF vergüteten, Marketingaktion zu qualifizieren. Im Vorstellungsgespräch werden den Teilnehmern Fragen gestellt, mit denen die Eignung für die nachfolgende Aufgabe ermittelt wird. Der Stress wird aufgebaut, indem den Probanden erklärt wird, dass die Teilnahme an der vergüteten Aufgabe von ihrer Leistung im Gespräch abhängt.

Um die Motivation für die Außenaufgabe zu erhöhen, wird den Teilnehmern erklärt, dass die Entlohnung einen festen und einen leistungsabhängigen Anteil hat. Die Marketingaufgabe wird in vier Abschnitten an unterschiedlichen Tagen durchgeführt.

Im letzten Schritt der Studie werden Referenzdaten erfasst. Dazu werden die Teilnehmer beim Lesen eines einfachen Textes in Innen- sowie Außenräumen aufgezeichnet. Dabei wird keine Leistung gefordert, um die Teilnehmer keinem Stress aus zu setzen. Die Teilnehmer tragen bei allen Aufgaben ein "Affectiva Armband", das die Hautleitfähigkeit (GSR) misst. Diese Daten werden als Ground Truth verwendet.

Die Audio Daten werden im Innenraum mit einem Array aus 7 Mikrofonen aufgenommen. Im Außenbereich hat jeder Proband jeweils ein Google Nexus One an Schulter und Hüfte befestigt.

Die Vorverarbeitung der Audiodaten wird in zwei Schritten durchgeführt. Die Daten werden zuerst in Teile, die Sprache oder keine Sprache enthalten, aufgeteilt. Im zweiten Schritt werden die individuellen Sprecher identifiziert. In Innenräumen wird die Identifizierung mit Hilfe der Informationen aus dem Mikrofonarray durchgeführt. Im Außenbereich wird die Entfernung der Sprecher zu den zwei Mikrofonen genutzt, um die Personen zu identifizieren. Die Stimme von Sprecher A, der die Mikrofone trägt, wird vom Mikrofon an der Schulter lauter aufgenommen als vom Mikrofon an der Hüfte. Sprecher B ist von beiden Mikrofonen ähnlich weit entfernt, wodurch seine Stimme von beiden Mikrofonen annähernd gleich aufgenommen wird. Daraus lässt sich die Stimme dem jeweiligen Sprecher zuordnen.

Die so gewonnenen Daten werden mit Hilfe von maschinellem Lernen klassifiziert. Dazu werden zuerst die benötigten Features ausgewählt. Die ersten beiden Features werden aus der Tonhöhe abgeleitet. Sie erhöht sich, sobald eine Person gestresst ist [19], während sich die Schwankungen in der Tonhöhe verringern [1]. Zwei weitere Features werden aus der Verteilung der Formanten im Spektrum der akustischen Energie abgeleitet. Ein Formant ist eine Anhäufung von akustischer Energie in einem bestimmten Bereich des Frequenzbandes. Sie verschieben sich in Richtung höherer Frequenzen, wenn eine Person unter Stress steht [1, 7]. Zusätzlich wird die Sprechgeschwindigkeit als weiteres Feature zur Klassifikation gewählt, die sich in gestressten Situationen ebenso erhöht.

Weitere Features werden aus dem 17-dimensionalen TEO-CB-AutoEnv [6] abge-

leitet. Ebenso werden 20-dimensionale MFCC (Mel-Frequenz-Cepstrum-Koeffizienten) [22] verwendet, die bereits in früheren Projekten zur Stresserkennung durch Sprache verwendet wurden [13,23]. Daraus ergeben sich die in Tabelle 1 aufgelisteten Features, die zur Klassifizierung verwendet werden.

Feature	Description
Pitch std	standard deviation of pitch
Pitch range	difference of max and min pitch
Pitch jitter	perturbation in pitch
Spectral centroid	centroid frequency of the spectrum
High frequency ratio	ratio of energy above 500Hz
Speaking rate	rate of speech
MFCCs	cepstral representation of the voice
TEO-CB-AutoEnv	Teager Energy Operator based nonlinear transformation

Tabelle 1. StressSense: Liste der verwendeten Features mit Beschreibung[10]

Das Framework zur **Klassifikation** der Sprache verwendet Gaußsche Mischverteilungen (GMM) mit einer diagonalen Kovarianzmatrix. Es wird eine GMM für jede der beiden Klassen, "Sprache gestresst" und "Sprache neutral", verwendet. Das Framework entscheidet anhand der Likelihood Funktion $p(X|\lambda)$ für jede Klasse mit gleicher Priorität; X steht dabei für einen Merkmalsvektor und λ für die GMM $\lambda(\omega, \mu, \Sigma)$ mit Gewicht ω , Mittelwert μ und Kovarianzmatrix Σ .

Es werden drei Trainingsvarianten untersucht:

Das **universelle Modell** verwendet einen allgemeinen Ansatz, bei dem ein Klassifikator für alle Personen trainiert wird.

Das **personalisierte Modell** verwendet einen benutzerbezogenen Ansatz. Es benötigt ein individuelles Training auf den Daten der jeweiligen Benutzer, wodurch ein individuelles Modell für jeden Nutzer erzeugt wird.

Das **adaptierte Modell** bildet einen Mittelweg zwischen den beiden genannten Modellen. Der Benutzer startet mit dem universellen Klassifikator, der jedoch mit individuellen Daten angepasst wird. Umso mehr Daten erfasst werden, umso besser wird der Klassifikator an den speziellen Benutzer angepasst. Dabei werden zwei Methoden umgesetzt. Bei der ersten Methode kennzeichnet der Benutzer die Daten selbst (überwacht). Bei der zweiten Variante wird ein selbst lernender Ansatz verfolgt, der auf dem Yarowsky Algorithmus [21] basiert. Zur Adaption der Parameter der GMM wird die Maximum-a-posteriori-Methode verwendet.

Die **Evaluation** wird in drei Schritten durchgeführt. Zuerst wird die Effektivität verschiedener Features für die unterschiedlichen Umfelder ausgewertet. Im zweiten Schritt werden die drei Modelle trainiert und jeweils innen und im Freien auf ihre Leistungsfähigkeit überprüft. Als Gütemaß werden dabei precision, recall, f-score und accuracy verwendet. Zuletzt wird die Klassifikation unter unterschiedlichen Szenarios im Innen- und Außenbereich zugleich evaluiert. Es sind jeweils ca. 3 Minuten Referenzdaten für beide Umfelder vorhanden. Für

die Klasse “Sprache gestresst“ sind im Innenraum-Szenario jeweils zwischen 4-8 Minuten pro Sprecher verfügbar, je nachdem wie viel der Proband in dem 10-minütigen Interview gesprochen hat. Für das Szenario im Freien wurden jeweils 4 Minuten Sprache verwendet.

Das universelle Modell wird mit der Leave-One-Out-Kreuzvalidierung getestet. Im adaptierenden Modell wird ein Teil der Daten zum Training, der andere Teil der Daten zum Testen verwendet. Mit den Daten, die zum Training verwendet werden, wird das zugrundeliegende universelle Modell adaptiert. Danach wird das Modell mit der zweiten Hälfte der Daten getestet. Im personalisierten Modell wird eine 5-fold-crossvalidation durchgeführt.

Um den Einfluss der verschiedenen Features auf die Klassifikation zu identifizieren, wurden diese in drei Gruppen aufgeteilt. Die erste Gruppe besteht aus den Features, die auf der Tonhöhe basieren, die zweite Gruppe aus dem TEO-CB-AutoEnv Satz und die letzte Gruppe aus den 20-dimensionalen MFCC.

Zuerst wird die Klassifizierung anhand der Features aus Gruppe A durchgeführt. Dabei wird für den Innenbereich eine accuracy zwischen 68,6% (universell) und 72,2% (personalisiert) erreicht. Die Werte im Außenbereich liegen zwischen 63,6% und 66,1%. Danach wird Gruppe B hinzugenommen, wodurch sich vor allem die Klassifizierung im Außenbereich stark verbessert. Hier wird eine Verbesserung der accuracy um bis zu 7,9% (personalisiert) erreicht. Die zusätzliche Verwendung der Features aus Gruppe C ergibt eine Erhöhung von 2%-4%. In den Ergebnissen zeigt sich, dass Gruppe A für beide Szenarien ähnlich gut funktioniert. Die MFCC sind eher für die Klassifizierung innen relevant, während TEO-CB-AutoEnv im Außenbereich deutliche Vorteile bringt.

Um die Modelle in allgemeinen Szenarios zu testen, wird das Indoor Modell auf die Daten aus dem Außenszenario angewendet und umgekehrt. Das universelle und personalisierte Modell werden direkt angewendet. Das adaptierende Modell wird auf dem universellen Modellen der ursprünglichen Daten mit den Daten des anderen Szenarios trainiert. Die Modelle, die innen trainiert wurden liefern auf den Daten aus dem Außenszenario nur eine schlechte Leistung. Einzig das adaptierende Modell mit überwachtem Lernen kann eine annehmbare accuracy von 74,8% erreichen. Die Modelle aus dem Außenbereich funktionieren in der ruhigeren Umgebung in Innenräumen verhältnismäßig besser.

3.2 Emotion Sensing anhand von Gesten

Ein Projekt im Bereich Emotion Sensing stellen Coutrix et. al. in [2] vor. Bei diesem Ansatz werden Touchscreen und Beschleunigungssensoren eines Smartphones als Sensoren verwendet. Die Emotionen werden aus den 2D (Touchscreen) und 3D Gesten abgeleitet, die der Benutzer ausführt.

Um geeignete Daten für die Untersuchung des Ansatzes zu gewinnen, wurde eine 15 tägige Studie mit insgesamt 12 Teilnehmern durchgeführt. Dabei waren 6 Personen männlich und 6 Personen weiblich. Die Teilnehmer waren aus unterschiedlichen Berufen und zwischen 25 und 47 Jahren alt (durchschnittlich 32,96 Jahre). 11 Teilnehmer verwenden ein iPhone, ein Teilnehmer verwendet einen iPod Touch.

Jeder Teilnehmer installierte eine App auf seinem Smartphone oder Medienplayer, die zur Erfassung der Daten benutzt wird. Dabei werden zwei Methoden zur Erfassung verwendet.

1. Die Nutzer sollen ihren affektiven Zustand in Situationen, in denen sie bewusst ein Gefühl verspüren, manuell eintragen. Dadurch wird sichergestellt, dass intensive Gefühle erfasst werden, die nur selten auftreten.
2. Die Teilnehmer werden einmal pro Tag, zu einem zufälligen Zeitpunkt, aufgefordert, ihre aktuelle Gefühlslage einzutragen. Dadurch werden eher neutrale und weniger intensive Gefühle erfasst.

Die Erfassung des Gefühlszustands erfolgt über die installierte App in drei Schritten (siehe Abbildung 2).

1. Die Teilnehmer führen eine frei wählbare Geste aus. Es können Eingaben auf dem Touchscreen oder Bewegungen des Geräts in der Luft durchgeführt werden.
2. Der emotionale Zustand wird über die vorgegebene Oberfläche eingegeben. Dabei wird die Emotion mit Hilfe des PAD Modells erfasst [18]. Die App bietet dazu eine Oberfläche mit einfachen Schiebereglern, die intuitiv zu bedienen sind.
3. Im letzten Schritt werden Kontextinformationen erfasst. Diese bestehen aus dem Auslöser der Emotion, dem Ort und ob die Person in Gesellschaft ist. Ebenso kann ein Text eingegeben werden, der Hilft sich an den Moment zu erinnern.

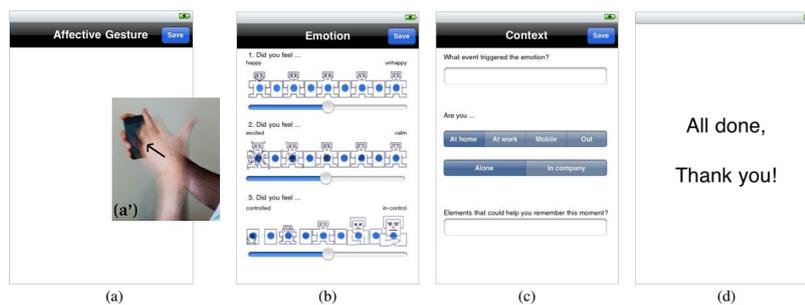


Abb. 2. Datenerfassung mit der installierten App: Geste ausführen (a)(a'), Emotion beschreiben (b), Kontext benennen (c) und Abschluss [2].

Während der 15 Tage wurden mit jedem Teilnehmer 3 Interviews durchgeführt. Das erste Interview fand vor Beginn der Studie statt, um den Ablauf der Studie zu erklären und die Teilnehmer mit der App vertraut zu machen. Das zweite Interview wurde in der Mitte der Studie durchgeführt, um die bisherigen Erfahrungen abzufragen. Ebenso wurden Informationen über die verwendeten Gesten gesammelt.

Es wurden 188 gültige Datensätze gesammelt, wobei es Teilnehmer gab, die an machen Tagen keine Daten erfasst haben. Andere Teilnehmer haben durchschnittlich 2 Datensätze pro Tag abgegeben. Der allgemeine Durchschnitt lag bei 1 Datensatz pro Tag. Wie in Abbildung 3 zu sehen, liegen die 3D Gesten als Messungen der Beschleunigung für alle 3 Achsen vor. Die Touchscreen Gesten liegen als Punkte bzw. Striche im zweidimensionalen Raum vor.

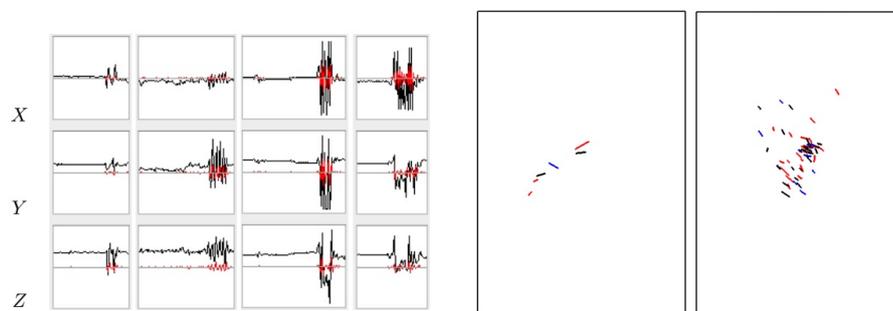


Abb. 3. Beispiele der Rohdaten von 3D(li) und 2D(re) Gesten [2].

Die Rohdaten der Gesten werden zuerst vorverarbeitet. Aus den Daten werden definierte Eigenschaften, wie die Dauer einer Geste, berechnet. Es werden für beide Typen von Gesten verschiedene Eigenschaften berechnet. Um die dreidimensionalen Gesten zu beschreiben, werden Beschleunigung, zeitliche Ableitung der Beschleunigung und Frequenzband der Beschleunigung betrachtet. Die Werte und Ableitungen werden dabei in Bezug auf die Achsen im dreidimensionalen Raum und die Länge der Vektoren berechnet. Ebenso werden Minimum, Maximum, Auslenkung, Median, Durchschnitt und Varianz berechnet. Die Eigenschaften werden auf Basis der Rohdaten, sowie auf gefilterten Daten (Hochpassfilter) berechnet.

Die Eigenschaften der 2D Gesten werden zum einen auf Ebene von Berührungen und zum Anderen auf der Ebene der gezogenen Striche berechnet. Zuerst werden die Position, Winkel und Geschwindigkeit einer Berührung errechnet. Winkel und Geschwindigkeit beziehen sich dabei auf zwei direkt aufeinanderfolgende Berührungen.

Danach werden Anzahl, Gesamtlänge, jeweilige Länge, Position, Geschwindigkeit

und Neigung der Striche berechnet. Wenn möglich werden alle Eigenschaften für die beiden Achsen, sowie die Vektorlänge berechnet.

Für die zweidimensionalen Gesten werden auch, falls möglich, Minimum, Maximum, Amplitude, Median, Durchschnitt und Varianz berechnet.

Bei der **Auswertung** der Daten stellen Coutrix et. al. fest, dass eine signifikante Korrelation zwischen den Gesten des Benutzers und der emotionalen Dimension der Erregung (nach PAD) existiert. Der stärkste Zusammenhang existiert mit der Projektion der Beschleunigung auf die x-Achse und der Dauer der 3D Gesten.

Für die restlichen zwei Dimensionen im emotionalen Raum existieren nur leichte Korrelationen. Das lässt sich dadurch erklären, dass die Erregung als Intensität der Emotion definiert wird. Umso höher die Erregung, desto stärker ist die Emotion.

Aufgrund der hohen Anzahl von insgesamt 249 Eigenschaften, die aus den Gesten abgeleitet werden, stellen Coutrix et. al. einen *affective gestures' space* (AGS) vor. Dieser besteht aus einer Teilmenge der Eigenschaften, die zusammen gut mit den affektiven Gesten eines Benutzers korrelieren. Dieser AGS wird mittels einer Hauptkomponentenanalyse aus den abgeleiteten Eigenschaften erzeugt. Es werden die drei signifikantesten Achsen des erzeugten Raumes gewählt, um den AGS zu bilden. Der AGS bietet Entwicklern die Möglichkeit, auf eine sinnvolle Auswahl der Eigenschaften zurückzugreifen, um ihre Systeme sensibel für den affektiven Zustand des Benutzers zu machen.

3.3 Mood Sensing durch Analyse der Smartphone-Nutzung

Im Bereich des Mood Sensing werden oft Inhalte aus Blogs oder sozialen Netzwerken zur Erkennung der Stimmung eines Nutzers verwendet. LiKamWa et. al [9] entwickeln jedoch ein System, das die Stimmung des Nutzers aus den Daten der Smartphone-Nutzung ableitet. Dabei verwenden sie keine Inhalte, sondern Informationen über die Kommunikation mit anderen Personen, das Browsingverhalten, verwendete Apps und den Standort des Benutzers. Ihr System "MoodScope" bietet Entwicklern ein Produkt, das wie ein Sensor für die Stimmung des Nutzers verwendet werden kann.

Die benötigten Daten wurden durch eine zweimonatige Feldstudie gesammelt. Es nahmen 25 Personen aus China und 7 Personen aus den USA teil. Das Alter lag zwischen 18 und 29 Jahren. 24 Teilnehmer waren Studenten. Die restlichen Teilnehmer waren in verschiedenen Bereichen tätig. Die Teilnahme wurde mit 75 USD und der Beteiligung an einer iPad 2 Verlosung vergütet. Zur Datenerfassung wurde sowohl eine App zur manuellen Erfassung der Stimmung, als auch ein Logger zur Speicherung der Daten auf den verwendeten iPhones installiert. Alle Teilnehmer sind im Alltag iPhone Nutzer.

Vor der Studie wurden die Teilnehmer aus China in zwei Gruppen befragt. Die erste Gruppe sollte sich dazu äußern, ob sich ihr Nutzungsverhalten abhängig von ihrer Stimmung ändert. Mit Ausnahme einer Person erklärten alle, dass sie je nach Stimmung mit anderen Leuten kommunizieren und andere Apps nutzen. Sie wünschen sich auch, dass sich ihr Smartphone an ihre aktuelle Stimmung

anpasst.

Die zweite Gruppe wird zum Teilen ihrer Stimmung befragt. Mit Ausnahme einer Person würden alle ihre Stimmung in Mikroblogs, Messengern oder anderen Anwendungen teilen. Dabei würden die Teilnehmer ihre Stimmung mit bestimmten Personenkreisen teilen, z.B. Freunden oder Arbeitskollegen. Alle Teilnehmer waren daran interessiert die Stimmung Anderer zu sehen.

Wie in Abbildung 4 zu sehen ist, können die Teilnehmer der Studie ihre aktuelle Stimmung über Schaltflächen eingeben. Es werden Fröhlichkeit (pleasure) und Erregung (activeness) in jeweils fünf Stufen erfasst. Die Stufen gehen dabei von sehr unzufrieden (P1) bzw. sehr inaktiv (A1) bis sehr zufrieden (P5) bzw. sehr aktiv (A5). Die zwei Dimensionen spiegeln die Dimensionen des verwendeten Circumplex Model [16] wieder. Die Nutzer werden gebeten, ihre Stimmung mindestens viermal pro Tag einzugeben. Wird die Stimmung mit einem Abstand von mehr als drei Stunden erfasst, erhält der Nutzer einen Stern. Die Anzahl der Sterne beeinflusst die Anzahl der Tickets bei der Verlosung am Ende der Studie. Dadurch werden die Teilnehmer motiviert, die Erfassung regelmäßig durchzuführen.

Der Logger, der im Hintergrund die Nutzungsdaten erfasst, benötigt keine Nutzerinteraktion.

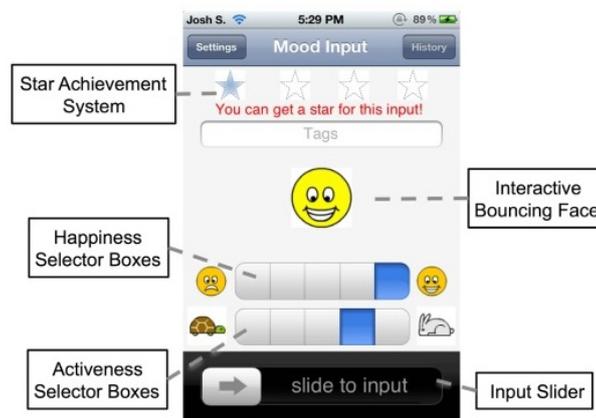


Abb. 4. MoodScope: Erfassung der Stimmung über die App [9].

Am Ende der Studie wurden die Teilnehmer ein weiteres Mal befragt. Diesmal sollten sie sich zur Bedienbarkeit und Funktionalität der App äußern. Die Teilnehmer fanden die App benutzerfreundlich und, mit Ausnahme einer Person, einfach zu bedienen. Alle bis auf drei Teilnehmer finden die Anzahl und Abstände der Erfassung gut. Einige Teilnehmer berichteten, dass die Anwendung die schnellen Änderungen ihres affektiven Zustandes nicht erfassen kann. Da LiKam-

Wa et. al. jedoch die länger andauernde Stimmung der Nutzer erfassen wollen, ist die Wahl des drei-stündigen Zeitfensters ihrer Meinung nach gut. Insgesamt wird die Anwendung als hoch effektiv bewertet, die Stimmung der Nutzer zu erfassen.

Die **Ergebnisse** der Feldstudie zeigen, dass sich die Stimmung der Teilnehmer im Allgemeinen über den Tag nur wenig ändert, jedoch an manchen Tagen große Veränderungen auftreten. Die durchschnittliche Standardabweichung pro Tag liegt bei 0,41. Die maximale Standardabweichung bei 1,73. Ein großer Teil der Datensätze sind neutrale Stimmungen. Die Zustände neutraler Fröhlichkeit (P3) und neutraler Erregung (A3) machen 33,2% bzw. 50,2% aus. Die Nutzer sind eher glücklich als unglücklich. Die Zustände P4 und P5 haben einen Anteil von 44,8% und 11,7%. Die neutrale Kombination (P3, A3) hat mit 25,6% den größten Anteil der 25 möglichen Kombinationen. Die beiden nächst höheren Werte liegen bei 18,5% (P4, A3) und 13,9% (P4, A4).

Es wurden insgesamt 15683 Anrufe, 36989 SMS, 13476 E-Mails, 142633 App Nutzungen, 27411 Webseitenaufrufe und 1146 unterschiedliche Orte erfasst.

Vor der Auswahl der Features für das maschinelle Lernen werden die Daten vorverarbeitet. Da die grundlegende Stimmung der Benutzer erfasst werden soll, werden für jeden Tag Durchschnittswerte berechnet. Dazu werden aus den eingegebenen Werten für Fröhlichkeit und Erregung, die ein Benutzer an einem Tag eingegeben hat, tägliche Durchschnittswerte berechnet. Die Berechnung erfolgt getrennt für die beiden Dimensionen, sodass sich 60 zweidimensionale Datensätze ergeben.

Die gesammelten Daten über die Smartphone-Nutzung werden in zwei Gruppen aufgeteilt. Die erste Gruppe enthält die sozialen Interaktionen. Es werden die Anzahl und Länge der Anrufe, SMS und E-Mails verwendet, um sechs 10-dimensionale Histogramme zu erstellen. Dabei werden nur die Daten der 10 am häufigsten kontaktierten Kontakte verwendet.

Die zweite Gruppe enthält das Browsingverhalten, die App-Nutzung und die Positionsdaten. Die Webseitenaufrufe werden anhand der URLs gruppiert, sodass sich die Besuche pro Seite berechnen lassen. Die Daten der App Nutzung werden nach Typen gruppiert. Dabei werden vorinstallierte Apps, Kommunikation, Unterhaltung, Finanzen, Spiele, Büro, Soziales, Reisen, Helfer, Wetter, Andere und "nicht kategorisierbar" als Typen verwendet. Die erfassten Positionsdaten werden mit Hilfe der DBSCAN Clusteranalyse [4] geclustert, um die besuchten Orte identifizieren zu können. Es werden Histogramme aus der Anzahl der Besuche pro Ort und der Frequenz, mit der Apps verwendet und Webseiten besucht werden, gebildet. Ebenso wird ein Histogramm aus der Dauer der App Nutzung gebildet. Daraus ergibt sich die Featuretabelle in Tabelle 2.

Um die Daten für das maschinelle Lernen vorzubereiten, wird eine multi-lineare Regression auf die Eingabedaten angewendet. Mit Hilfe einer Sequential Forward Selection wird eine Teilmenge der Features ausgewählt.

Danach werden die verschiedenen Modelle trainiert. Zuerst wird ein personalisiertes Modell trainiert. Dafür wird die multi-lineare Regression auf die Daten jedes Nutzers separat angewendet. Das personalisierte Modell erreicht eine

Data Type	Histogram by:	Dimensions
Email contacts	# Messages	10
	# Characters	10
SMS contact	# Messages	10
	# Characters	10
Phone call contacts	# Calls	10
	Call Duration	10
Website visits	# Visits	10
Location Clusters	# Visits	10
Apps	#App Launches	10
	App Duration	10
Categories of Apps	#App Launches	12
	App Duration	12
Previous 2 Pleasure and Activeness Averages	N/A	4

Tabelle 2. MoodScope: Feature Tabelle mit Histogrammen [9]

durchschnittliche mittlere quadratische Abweichung von 0,075 mit einer Standardabweichung von 0,05 in der Dimension der Fröhlichkeit. Für die Dimension der Erregung werden die Werte 0,085 bzw. 0,062 erreicht.

Danach wird ein universelles Modell trainiert. Die Ergebnisse sind für einige Nutzer gut, jedoch nicht konstant. Das universelle Modell erreicht in der Dimension Fröhlichkeit eine minimale mittlere quadratische Abweichung von 0,069. Die durchschnittliche Abweichung beträgt jedoch 0,269 mit einer Standardabweichung von 0,175. Die zweite Dimension verhält sich ähnlich.

Als drittes Modell wird ein Hybridmodell aus den beiden vorherigen trainiert.

4 Zusammenfassung

In dieser Arbeit wurden grundlegenden Begriffe der Emotionspsychologie angesprochen, sowie verschiedene Bereiche des AC vorgestellt. Dabei wurde der Schwerpunkt auf unauffälliges, mobiles AC gelegt. Emotion Sensing, Mood Sensing und Sentimental Mining wurde charakterisiert und die wichtigen Punkte hervorgehoben.

Im weiteren Verlauf der Arbeit wurden Projekte vorgestellt, die häufig verwendete Verfahren verwenden. Darunter sind StressSense, ein Projekt zur Erkennung von Stress durch Sprache, und MoodScope, eine Mood Sensing Anwendung, die auf Basis der Smartphone-Nutzung arbeitet. Es wurde auf Datenerhebung, Vorverarbeitung der Daten, Auswahl der Features, Klassifikation und Evaluation eingegangen.

Der Forschungsbereich des Affective Computing konzentriert sich immer stärker auf Smartphones und Wearables. Mit diesen Geräte können AC Systeme ohne zusätzliche Hardware entwickelt und unauffällig verwendet werden. Die starke Entwicklung im Bereich Wearables treibt diese Forschung weiter voran. Smartwatches und Datenbrillen stellen neue Sensoren bereit und bieten zusätzliche

Möglichkeiten, Daten zu erfassen. Durch die Integration von GSR-Sensoren in Smartwatches, kann eine einfache Möglichkeit geschaffen werden, Stress zu erfassen. Früher war dafür zusätzliche spezialisierte Hardware nötig.

Die zukünftige Forschung wird mit Sicherheit immer mehr massentaugliche Produkte wie MoodScope hervorbringen, das bereits von 60000 Personen genutzt wird (moodscope.com).

Literatur

1. Congleton, J., Jones, W., Shiflett, S., McSweeney, K., Huchingson, R.: An evaluation of voice stress analysis techniques in a simulated awacs environment. *International Journal of Speech Technology* 2(1) (1997), <http://dx.doi.org/10.1007/BF02539823>
2. Coutrix, C., Mandran, N.: Identifying emotions expressed by mobile users through 2d surface and 3d motion gestures. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. pp. 311–320. UbiComp '12, ACM, New York, NY, USA (2012), <http://doi.acm.org/10.1145/2370216.2370265>
3. Ekman, P.: Facial expression and emotion, vol. 48(4), pp. 384–392. *American Psychologist* (1993)
4. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *Kdd*. vol. 96, pp. 226–231 (1996)
5. Gross, J.J.: The emerging field of emotion regulation: An integrative review. *Review of General Psychology* (1998)
6. Hansen, J., Kim, W., Rahurkar, M., Ruzanski, E., Meyerhoff, J.: Robust emotional stressed speech detection using weighted frequency subbands. *EURASIP Journal on Advances in Signal Processing* 2011(1), 906789 (2011), <http://asp.eurasipjournals.com/content/2011/1/906789>
7. Lambert M. Surhone, Miriam T. Timpledon, S.F.M.: *Voice Stress Analysis*. beta-script publishing (2010)
8. Lee, H., Choi, Y.S., Lee, S., Park, I.: Towards unobtrusive emotion recognition for affective social communication. In: *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*. pp. 260–264 (Jan 2012)
9. LiKamWa, R., Liu, Y., Lane, N.D., Zhong, L.: Moodscope: Building a mood sensor from smartphone usage patterns. In: *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*. pp. 389–402. *MobiSys '13*, ACM, New York, NY, USA (2013), <http://doi.acm.org/10.1145/2462456.2464449>
10. Lu, H., Frauendorfer, D., Rabbi, M., Mast, M.S., Chittaranjan, G.T., Campbell, A.T., Gatica-Perez, D., Choudhury, T.: Stresssense: Detecting stress in unconstrained acoustic environments using smartphones. In: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. pp. 351–360. UbiComp '12, ACM, New York, NY, USA (2012), <http://doi.acm.org/10.1145/2370216.2370270>
11. Nguyen, T., Phung, D., Adams, B., Venkatesh, S.: Mood sensing from social media texts and its applications. *Knowledge and Information Systems* 39(3), 667–702 (2014), <http://dx.doi.org/10.1007/s10115-013-0628-8>
12. Nowlis, V., Nowlis, H.H.: The description and analysis of mood. *Annals of the New York Academy of Sciences* 65(4), 345–355 (1956)

13. Patil, S.A., Hansen, J.H.: Detection of speech under physical stress: Model development, sensor selection, and feature fusion (2008)
14. Petrushin, V.: Emotion in speech: Recognition and application to call centers. In: Proceedings of Artificial Neural Networks in Engineering. pp. 7–10. Citeseer (1999)
15. Picard, R.W.: Affective computing (1995)
16. Russell, J.A.: A circumplex model of affect. *Journal of personality and social psychology* 39(6), 1161 (1980)
17. Russell, J.A.: Core affect and the psychological construction of emotion. *Psychological review* 110(1), 145 (2003)
18. Russell, J.A., Mehrabian, A.: Evidence for a three-factor theory of emotions. *Journal of Research in Personality* 11(3), 273 – 294 (1977), <http://www.sciencedirect.com/science/article/pii/009265667790037X>
19. Scherer, K.: Voice, stress, and emotion. In: Appley, M., Trumbull, R. (eds.) *Dynamics of Stress*, pp. 157–179. The Plenum Series on Stress and Coping, Springer US (1986), http://dx.doi.org/10.1007/978-1-4684-5122-1_9
20. Vinodhini, G., Chandrasekaran, R.: Sentiment analysis and opinion mining: a survey. *International Journal* 2(6) (2012)
21. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics. pp. 189–196. ACL '95, Association for Computational Linguistics, Stroudsburg, PA, USA (1995), <http://dx.doi.org/10.3115/981658.981684>
22. Zhou, G., Hansen, J., Kaiser, J.: Nonlinear feature based classification of speech under stress. *Speech and Audio Processing, IEEE Transactions on* 9(3), 201–216 (Mar 2001)
23. Zhou, G., Hansen, J.H., Kaiser, J.F.: Nonlinear feature based classification of speech under stress. *Speech and Audio Processing, IEEE Transactions on* 9(3), 201–216 (2001)

Ablauf Dynamischer Software Updates in Java mit dem Ansatz der Lazy-Zustandstransformation

Thomas Kadow*

Betreuer: Martin Alexander Neumann[†]

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*thomas.kadow@student.kit.edu

[†]mneumann@teco.edu

Zusammenfassung. Dynamische Software Updates (DSU) ermöglichen die Aktualisierung eines Programmes zur Laufzeit. Obwohl dies in der Industrie bisher nur selten eingesetzt wird, gibt es bereits ein breites Spektrum verschiedenster Systeme und Techniken, welche DSU umsetzen. Dieser Artikel soll einen Überblick über den heutigen Stand der Technik im Bereich von DSU geben. Dabei wird der Blick auf die Systeme Gosh!, Javelus und Rubah gerichtet. Alle drei Systeme arbeiten mit der Programmiersprache Java, funktionieren mit einer gängigen Java Virtual Machine und verfolgen einen Lazy-Ansatz, um den Zustand eines Programmes zu aktualisieren. Zu Beginn des Artikels werden einige Teile der Java Virtual Machine, in welcher alle Java-Programme ausgeführt werden, beschrieben. Anschließend wird ein Einblick in die Funktionsweise der genannten Systeme gegeben. Daraufhin folgt eine Übersicht der Systeme in Bezug auf den Programmieraufwand des Anwendungsentwicklers beziehungsweise des Benutzers vor einem Update, sowie auf die Performance der Systeme während und nach einem Update.

Schlüsselwörter: Dynamische Software Updates, DSU, Java, JVM, Lazy-Transformation, Gosh!, Javelus, Rubah

1 Einführung

In der heutigen modernen Gesellschaft spielt Software eine bedeutende Rolle. Doch nicht jede ist von Anfang an perfekt sobald sie auf den Markt kommt. Je komplexer eine Software ist, desto mehr Fehler schleichen sich ein. Außerdem sollte sie auch an neue Bedürfnisse und Anforderungen anpassbar sein. In herkömmlichen Verfahren muss ein Programm für ein Update gestoppt, geupdated und wieder gestartet werden, um neue Funktionen zu ermöglichen oder Fehler zu beheben. Dies kann für einen Benutzer sehr lästig sein, da die Software in diesem Zeitraum nicht verfügbar ist. Viel gravierender ist es aber, für Anwendungen, welche von enormer Wichtigkeit sind und bei welchen eine hohe Verfügbarkeit von Bedeutung ist. Um diese Prozedur zu vermeiden, rückt ein alternatives Update-Verfahren in den Blickpunkt von Forschern und Entwicklern, das sogenannte dynamische Software Update (DSU). Ein DSU bietet

die Möglichkeit, eine Software zur Laufzeit zu aktualisieren, ohne diese herunterzufahren. Dadurch müssen wichtige Zustände eines Programmes während des Updates nicht mehr zur Zwischenspeicherung ausgelagert werden, Verbindungen zu anderen Systemen müssen nicht neu aufgebaut werden und Entwickler haben die Möglichkeit die Auswirkungen ihrer Codeänderungen On-the-Fly zu testen.

Es existieren bereits DSU-Systeme für verschiedene Programmiersprachen wie C [9,3,12,10], Python [11], Java [18,19,27,1,13] und weitere. Für Java gibt es dabei im Wesentlichen zwei verschiedene Ansätze der Umsetzung eines DSU. Ein Ansatz befasst sich damit, die Java Virtual Machine (JVM), in der Java-Applikationen ausgeführt werden, zu modifizieren [26,22,24]. Der andere Ansatz zielt darauf ab, komplett ohne Modifikationen der JVM auszukommen.

Dieser Artikel soll einen Überblick über die DSU-Systeme Gosh!, Javelus und Rubah geben. Jedes dieser Systeme arbeitet mit der Programmiersprache Java und verwendet ein Lazy-Transformationsverfahren (vgl. Kapitel 3). Außerdem funktionieren alle mit einer gängigen JVM (im Folgenden nur HotSpot JVM [15] betrachtet). Nachdem kurz auf verschiedene Teile der JVM eingegangen wird, sollen anschließend die Funktionsweisen der aufgeführten Systeme etwas detaillierter betrachtet werden. Im Hinblick auf den Programmieraufwand und die Performance zieht der darauffolgende Abschnitt einen Vergleich zwischen den einzelnen Systemen, bevor schlussendlich mit einer Zusammenfassung der Ausführungen des Artikels abgeschlossen wird.

2 Die Java Virtual Machine

Um besser verstehen zu können, wie die im darauffolgenden Kapitel beschriebenen Update-Systeme funktionieren, wird nun ein Einblick in die Funktionsweise der JVM [25] gegeben.

Nachdem die Klassen eines Java Programmes durch den Compiler in Java-Bytecode umgewandelt wurden, kann dieser in der Java-Laufzeitumgebung ausgeführt werden. Diese besteht aus der Java API, welche Standard-Klassen bereitstellt, und der JVM, welche den erzeugten Bytecode analysiert und ausführt. Für jede Java Applikation wird eine eigene Instanz der JVM gestartet. Die wesentlichen Komponenten der JVM, die für dieses Dokument entscheidend sind, werden im Weiteren beschrieben.

2.1 Classloader

Ein Classloader (vgl. Abbildung 1) lädt die Klassendaten einer Klasse in die Runtime Data Area (vgl. Kapitel 2.2) der JVM, verlinkt ihre Referenzen auf Konstanten und initialisiert Klassenvariablen mit ihren Startwerten. Dies geschieht in dem Moment, in welchem die Klasse das erste mal referenziert wird. Klassen die geladen wurden, können nicht ohne Weiteres gelöscht werden. Die einzige Möglichkeit sie zu entfernen ist, dass alle Klassen eines Classloaders dereferenziert sind, sodass der komplette Classloader gelöscht werden kann. Es gibt

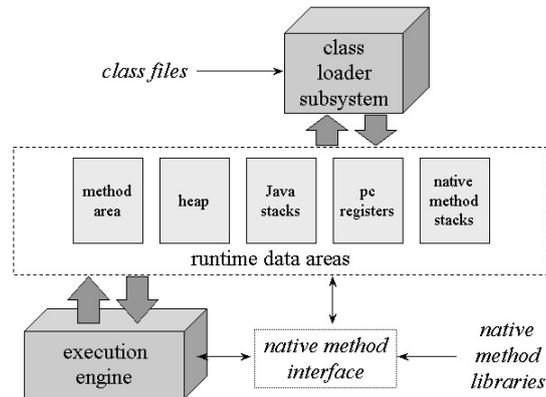


Abb. 1. Interne JVM Architektur [25]

verschiedene Arten von Classloadern, welche im Wesentlichen der Bootstrap-Classloader und der System-Classloader sind. Der Bootstrap-Classloader ist für das Laden der Klassen aus der Java Standard-Bibliothek zuständig, während ein System-Classloader Klassen aus dem Klassenpfad des Programmes lädt. Außerdem ist es möglich einen Custom-Classloader zu implementieren, um die Eigenschaften eines Classloaders anzupassen.

Jeder Classloader hat seinen eigenen Namensraum und ein spezifischer Name einer Klasse darf darin nur einmal vorkommen. Das heißt, dass jede Klasse höchstens einmal geladen werden kann. Eine Klasse identifiziert sich durch ihren Namen und den Classloader, durch den sie geladen wurde.

2.2 Runtime Data Area

Die Runtime Data Area (vgl. Abbildung 1) besteht aus verschiedenen Speicherbereichen, wie zum Beispiel der **Method Area**. Dieser Bereich wird von allen existierenden Threads geteilt. Wird ein Klassenobjekt geladen, so werden hier Klasseninformationen über die statischen Attribute, Methoden und Super-Klassen gespeichert, als auch von welchem Classloader sie geladen wurde. Soll beispielsweise eine Methode von einem Objekttyp ausgeführt werden, so wird der zugehörige Bytecode aus der Method Area geladen.

Desweiteren gibt es den **Heap**, welcher ebenfalls von allen Threads geteilt wird. Jedes erzeugte Objekt wird mit seinen Attributen im Heap gespeichert. Dabei referenziert jede Objektinstanz seine Klassendaten in der Method Area und eventuell andere Objekte mithilfe von Pointern. Sollte ein Objekt nicht selbst referenziert werden, so wird dieses mithilfe des Garbage Collectors [14] gelöscht und dessen Speicher wieder freigegeben.

Abschließend existieren mehrere **Stacks** in der Runtime Data Area. Im Gegensatz zu den anderen beiden Speichern gibt es für jeden Thread einen eigenen Stack. Kein anderer Thread hat Zugriff auf den Stack eines anderen. Während eine Methode ausgeführt wird, werden auf ihm alle zur Ausführung nötigen Werte wie Parameter, lokale Variablen und Sonstiges gespeichert.

2.3 Just-in-Time Compiler

Der Just-in-Time (JIT) Compiler ist Teil der Execution Engine (vgl. Abbildung 1), die dafür zuständig ist, den Java Bytecode jedes einzelnen Threads durch einen Interpreter auszuführen oder durch den JIT-Compiler in Maschinencode zu kompilieren und anschließend auszuführen. Das Besondere am JIT-Compiler ist, dass er während der Laufzeit häufig ausgeführte Programmteile identifizieren und optimieren kann. Ziel dieses Verfahrens ist es, die Ausführungsgeschwindigkeit zu verbessern, da das Interpretieren von oft verwendetem Bytecode insgesamt langsamer ist, als den vorher kompilierten Maschinencode auszuführen.

2.4 Java HotSwap Feature

Mittels Java HotSwap [4], einem Feature der HotSpot JVM, kann der Inhalt einer Methode zur Laufzeit geändert werden. Ermöglicht wird dies durch das Java Virtual Machine Tool Interface (JVMTI) [17]. Dazu wird eine neue Version der Klasse mit dem gleichen Klassenschema geladen und die Referenzen auf die alte Klasse werden auf die neue Version umgeleitet. Es ist damit jedoch weder möglich die Struktur einer Klasse zu verändern, noch Methoden hinzuzufügen, beziehungsweise zu entfernen.

2.5 Java Reflection API

Mit Hilfe der Java Reflection API [16] kann ein Programm das eigene Laufzeitverhalten einsehen, kontrollieren und gegebenenfalls modifizieren. Dadurch können Befehle ausgeführt werden, die ansonsten unmöglich oder unzulässig gewesen wären. Debugger und Test Tools verwenden diese API beispielsweise um private Attribute auszulesen. Der Nachteil von Reflection ist, dass Aufrufe der API nicht von der JVM optimiert werden können, da involvierte Klassentypen dynamisch aufgelöst werden.

3 Funktionsweise von Gosh!, Javelus und Rubah

In einem System, welches ein DSU unterstützt und bereitstellt, müssen im Wesentlichen drei Phasen beachtet werden. Die erste Phase zeichnet sich aus durch die Vorbereitung des Updates sowie des Auffindens von Update-Punkten. Für solche Punkte gibt es verschiedene Sicherheitseigenschaften (in diesem Artikel werden *Quiescence* und *Activeness-Safety* genannt). Durch diese Eigenschaften soll sichergestellt werden, dass nach einem Update kein neuer Programmcode auf

alten Zuständen ausgeführt wird. Nähere Erläuterungen sollen bei den jeweiligen Systemen, welche diese Eigenschaften überprüfen, stattfinden. Ein weiterer Teil der ersten Phase ist die Durchführung des Updates selbst.

In der zweiten Phase, der Zustandstransformation, wird eine Transformationsfunktion mit den dazu benötigten Daten in das Programm eingepflegt, welche den Programmzustand auf die neue Version aktualisiert. Diese Funktion kann auf zwei verschiedene Arten durchgeführt werden. Die eine ist die Eager-Transformation, bei welcher das komplette Programm direkt zum Updatezeitpunkt auf den neuen Stand gebracht wird. Dazu muss das Programm solange angehalten werden, bis das Update fertig ist. Das andere Verfahren ist die Lazy-Transformation. Hierbei kann das Programm, nachdem die Klassen aktualisiert wurden, weiterlaufen und die dazugehörigen Objekte werden erst bei einem Aufruf transformiert. Dadurch entsteht jedoch ein Overhead im Vergleich zur normalen Programmausführung. Da aber meist ein Großteil des Programmcodes nur selten aufgerufen wird, hat dieser Vorgang den Vorteil, dass er sich unnötige Transformationen erspart und deshalb bei großen Programmen insgesamt effizienter ist.

Die dritte Phase dient lediglich dazu, die einzelnen Threads des Programmes wieder zu starten und im ursprünglichen Kontrollfluss weiterlaufen zu lassen. Bei der Durchführung einer Lazy-Transformation findet diese Phase allerdings schon nach der ersten Phase, also direkt nach dem Update statt. Im weiteren Verlauf dieses Artikels wird nur noch die Lazy-Transformationen betrachtet.

Wie die einzelnen Phasen in den betrachteten Systemen umgesetzt werden wird im Folgenden beschrieben:

3.1 Gosh!

A. Gregersen, M. Rasmussen und B. Jorgensen beschreiben in ihren Papern (vgl. [5] und [6]) das von ihnen entwickelte DSU-System Gosh!. Gosh! ist dabei eine Weiterentwicklung eines vorherigen DSU-Projektes namens Javeleon. Dieses wurde von ZeroTurnAround gekauft, welche auch das DSU-System JRebel [27] entwickeln. Die Funktionsweise von Gosh! soll nun in den nächsten Abschnitten vorgestellt werden.

Architektur. Gosh! präsentiert ein Framework um ein DSU umzusetzen, welches in zwei abstrakte Ebenen unterteilt ist (vgl. Abb. 2). Die *Gosh! Agent*-Ebene ist dafür zuständig, die JVM und das zu updatende Programm auf ein DSU vorzubereiten. Dagegen implementiert die *Gosh! runtime*-Ebene das Update-Modell und damit die zum Update der Klassen wichtigen Konzepte der *In-Place-Proxification* und dem *Correspondence Mapping*, auf die später noch genauer eingegangen wird. Außerdem ist letztere Ebene für die Lazy-Zustandstransformation der Objekte und das korrekte Zusammenspiel von Objekten aus verschiedenen Versionen zur Laufzeit verantwortlich.

Die *Gosh! Agent*-Ebene lässt sich noch in vier einzelne Komponenten untergliedern. Der *bootstrap-class-transformer* und der *sub-process-spawner* modifizieren die Bootstrap-Klassen und starten einen neuen JVM-Prozess mit diesen,

sodass Gosh! für den Nutzer transparent ist. Das *class-loading plug-in* dient dazu, Gosh! mit verschiedenen Frameworks kompatibel zu machen. Die letzte Komponente, der *bytecode-transformer*, passt Klassen an, welche in die JVM geladen werden, sodass dynamische Updates durch Gosh! auf diesen Klassen möglich sind.

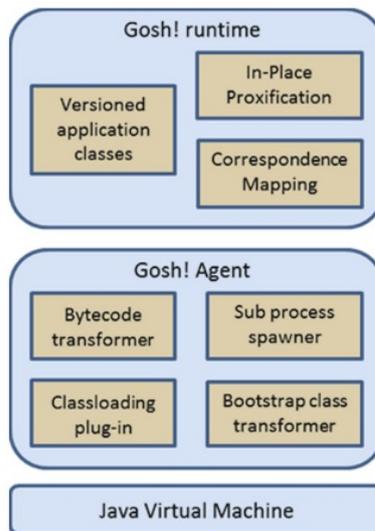


Abb. 2. Framework-Architektur von Gosh! [6]

Update durch In-Place-Proxification und Correspondence Mapping. In Gosh! wird für jede Version eines Programmes ein neuer Classloader (vgl. Kapitel 2.1) gestartet und somit existiert für jede Version ein unterschiedlicher Namensraum. Dadurch entsteht eine sogenannte Versionsbarriere [23] zwischen den verschiedenen Versionen aus Klassen und Objekten, welche so nicht mehr miteinander kompatibel sind. Aus diesem Grund schaffen sich die Entwickler von Gosh! die beiden Konzepte *In-Place-Proxification* und *Correspondence Mapping* um diese Barriere zu umgehen.

Die Aufgabe der *In-Place-Proxification* ist bei einem Update die Klassen und Objekte der alten Programmversion mit denen der neuen Version zu verbinden. Dafür wird Code generiert und in die alten Klassen eingefügt, um Aufrufe einer Methode auf deren neue Version umzuleiten. Es wird deshalb davon gesprochen, dass sich die alten Versionen wie Proxies verhalten. In Abbildung 3 wird dargestellt, wie ein Methodenaufruf von F auf D aus der alten Version auf die neue Version von D durch die *In-Place-Proxification* weitergeleitet wird. Um dabei die Versionsbarriere zu durchbrechen, wird die Java Reflection API verwendet (vgl. Kapitel 2.5). Diese Lösung funktioniert allerdings nur, wenn keine Typenverän-

derungen von Parameterobjekten oder Rückgabewert-Objekten in den Methoden der alten Version zur zugehörigen neuen Version der Methode stattgefunden haben. Andernfalls wird zusätzlich das *Correspondence Mapping* zur korrekten Übersetzung von einem Objekt mit dem Typ der alten Version zum Objekt mit dem Typ der neuen Version gebraucht.

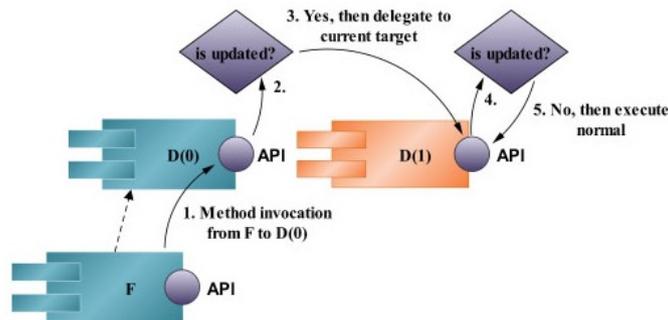


Abb. 3. Darstellung einer Weiterleitung durch die *In-Place-Proxification* [5]

Wie eben erwähnt wurde, gibt es beim Weiterleiten der Methodenaufrufe Fälle, in denen sich die Typen von Objekten verändert haben, welche in dem Aufruf benötigt werden, und somit Probleme beim korrekten weiterleiten auftreten. Um dies zu umgehen, führt Gosh! das Konzept des *Correspondence Mapping* ein. In solchen Maps werden paarweise Objekte aus verschiedenen Versionen gespeichert. Um diese Maps zu kontrollieren und zu benutzen wird ein *Dynamic Correspondence Controller* (DCC) verwendet. Findet ein Update statt, werden alle Klassen darin als Proxies markiert. Sobald ein solches Proxy verwendet wird, wird ein neues, nicht initialisiertes Objekt der neuen Klasse erzeugt. Dessen Attribute werden durch eine Zustandstransformation initialisiert und das Paar in insgesamt zwei Maps eingepflegt, einmal mit dem alten Objekt als Key und einmal andersherum. Somit können sowohl aktualisierte Objekte zu einem alten Objekt gefunden werden, als auch zu einem aktuellen Objekt das dazugehörige Objekt der nächst älteren Version.

Zustandstransformation. Gosh! verwendet, wie alle in diesem Artikel beschriebenen Systeme, einen Lazy-Ansatz um die Zustandstransformation durchzuführen. Sobald ein Programm das erste mal auf ein Attribut eines Objektes zugreift wird das Objekt, wenn dieses nicht bereits auf dem aktuellen Stand ist, auf die neue Version transformiert. Dafür werden in diesem DSU-System spezielle Getter-Methoden benutzt, um die Attribute eines Objekts auszulesen. Für jedes Attribut in einem Objekt wird dafür ein Attribut vom Typ Boolean angelegt, welches angibt, ob dessen Zustand aktuell oder veraltet ist (false, falls der Zu-

stand aktuell ist). Sollte eine Getter-Methode beim Zugriff auf ein Attribut merken, dass dessen Zustand veraltet ist, werden mit Hilfe des DCC alle vorherigen Versionen des Objektes untersucht, bis der zugehörige Boolean des Attributes in einem alten Objekt auf false gesetzt ist. Sobald dieses gefunden wurde und der Typ des Attributes ein nicht-updatefähiger Typ ist (Java-Systemklassen) wird die Referenz davon in das aktuelle Objekt kopiert. Anschließend muss das zugehörige Status-Attribut der alten Version auf true und das in der neuen Version auf false gesetzt werden. Falls das Attribut einen updatefähigen Typen hat (Benutzerdefinierte Klassen) wird wiederum der DCC genutzt, um die aktuellste Version des Typs zu finden und ein passendes Objekt zu erzeugen.

Zusätzlich zur Transformation von der alten zur neuen Version lassen sich in Gosh! auch neue Attribute in eine Klasse und somit auch in dessen Instanzen hinzufügen. Der Ablauf, eine ältere Instanz zu finden, ist analog zu dem eben erläuterten Verfahren. Ist der Algorithmus dabei jede frühere Version durchgegangen, ohne einen aktuellen Zustand in den alten Objekten zu finden, nimmt er an, das Attribut wurde neu eingeführt. Daraufhin wird der Zustand des neuen Attributes auf false gesetzt und das Attribut mit einem passenden Standardwert initialisiert. Um zu vermeiden, dass zum Beispiel *NullPointerException* auftreten, muss der Entwickler des Programmes dessen Wert bereits im Source-Code der Klasse zugewiesen haben.

3.2 Javelus

Grundlage für die folgenden Ausführungen, sind die Paper von Tianxiao Gu et al. (vgl. [7] und [8]), in welchem sie ihr DSU-System Javelus vorstellen. Wie solch ein DSU mit Javelus funktioniert und wie sie die Lazy-Transformation umgesetzt haben, soll nun beschrieben werden.

Offline-Preparation. Nachdem ein Entwickler eine neue Version eines Programmes erstellt hat, muss er bei Javelus zunächst ein Offline-Analyse-Tool verwenden. Durch dieses Tool werden die Klassen der beiden Versionen verglichen und ein dynamischer Patch generiert. Dieser enthält eine Liste aus Änderungen von Klassen und *Object Transformers*, mit dessen Hilfe später ein veraltetes Objekt in ein neues transformiert wird. Je nachdem wie sich eine Klasse verändert hat, unterscheidet Javelus zwischen verschiedenen Update-Typen. Für hinzugefügte Klassen, gelöschte Klassen und Klassen, in denen sich nur Methodenrumpfe geändert haben, müssen keine Klasseninstanzen angepasst werden. Nur für Klassen, bei welchen sich der eigene Typ, der Typ eines Attributes oder der Typ einer zugehörigen Super-Klasse verändert haben, müssen die Instanzen updatet und somit *Object Transformer* erzeugt werden.

Ein von Javelus generierter *Object Transformer* für die Klassen aus der alten und neuen Version in Abbildung 4(a) und (b)) könnte wie in Abbildung 4(c) aussehen. Gleiche Attribute würden während der Zustandstransformation einfach kopiert werden (Zeile 2-3), wohingegen neue und veränderte Attribute mit einem Standardwert passend zu ihrem Typ initialisiert werden (Zeile 4). Falls der Entwickler mit dem Standard-*Transformer* nicht zufrieden ist, kann er diesen

```

1 class User{
2   String userName;
3   String password;
4   String[] forwardAddresses;
5 }
    (a) v0

```

```

1 class User{
2   String userName;
3   String password;
4   EmailAddress[] forwardAddresses;
5 }
    (b) v1

```

```

1 void defaultTransformer(User_v0 v0, User v1){
2   v1.username = v0.username;
3   v1.password = v0.password;
4   v1.forwardAddresses = null;
5 }
    (c) a transformer template

```

```

1 class User{
2   String userName;
3   String password;
4   EmailAddress[] forwardAddresses;
5   public void updateObject(
6     @OldField(name="forwardAddresses", desc=" [String; ", clazz="User")
7     String[] p1){
8     final int length = p1.length;
9     forwardAddresses = new EmailAddress[length];
10    for(int i=0;i<length;i++){
11      forwardAddresses[i] = new EmailAddress(p1[i]);
12    }
13  }
14  public static void updateClass(){}
15 }
    (d) a transformer template

```

Abb. 4. Object Transformer von Javelus [8]

wie in Abbildung 4(d) selbst anpassen. Dadurch kann zum Beispiel, wie in Zeile 5 bis 13 implementiert, das neue Array mit den Werten aus dem Array des alten Objekts initialisiert werden. Im Wesentlichen beinhaltet ein solcher *Custom Transformer* alle Attribute der neuen Klasse, eine *updateObject()*-Methode, die ähnlich einem Konstruktor verschiedene Attribute initialisieren soll, und eine *updateClass()*-Methode um statische Variablen zu initialisieren. Dabei werden Attribute der alten Version als annotierte Parameter übergeben (vgl. Abb. 4(d) Zeile 6).

Wenn die Offline-Preparation schließlich beendet ist, kann der Patch über ein Interface an Javelus gesendet werden, wodurch das Online-Update eingeleitet wird.

Online-Update. Bevor das Update der Klassen eingespielt werden kann, muss ein Updatepunkt gefunden werden. Javelus untersucht dafür die Stacks, um herauszufinden, ob eine Methode gerade verwendet wird, welche durch das Update verändert werden würde. Ist dies nicht der Fall, spricht man von einem update-sicheren Zustand (in diesem Zusammenhang auch *Activeness-Safety* genannt) und das Update der Klassen kann durchgeführt werden. Andernfalls wird der Updateprozess unterbrochen und erst fortgeführt, wenn das Ausführen einer solchen Methode beendet wurde. Sollte es beim Start des Prozesses mehrere Fehlversuche geben, wird das Update möglicherweise abgebrochen.

Bei gelungenem Start, werden alle neuen Klassen aus dem Patch gelesen und topologisch beginnend bei der obersten Super-Klasse geladen. Dafür verwendet Javelus keinen Custom-Classloader, sondern benennt stattdessen die alten Klassen um, um dadurch Namenskonflikte zu vermeiden. Anschließend findet das Klassenupdate statt.

Bei Klassen, in welchen nur Methodenrümpfe verändert wurden, tauscht Javelus lediglich den Inhalt der alten Klasse durch den Inhalt der neuen Klasse aus. Die Objekte solcher Klassen müssen dabei nicht verändert werden und bleiben trotz neuer Version aktuell. Neu hinzugefügte Klassen werden lediglich in die Method Area eingefügt, sodass sie von Objekten und anderen Klassen verwendet werden können. Ihre statischen Attribute und Sonstiges werden nicht initialisiert bis dies bei ihrem ersten Zugriff durch ein *Transformer* passiert. Klassen, die in der neuen Version nicht mehr existieren, werden entfernt, sodass diese nicht mehr verwendet werden können. Instanzen dieser Klassen werden anschließend als Instanzen der Oberklasse interpretiert. Im Falle von Klassen mit Typenveränderungen werden die alten Versionen aus der Method Area entfernt und die neuen Klassendaten darin eingefügt. Dabei werden die unveränderten statischen Attribute aus dem alten Klassenobjekt kopiert. Die zugehörigen Klasseninstanzen, welche in so einem Fall geupdatet werden müssen, werden erst zur Laufzeit transformiert.

Bevor das Programm weiterlaufen kann und die Zustandstransformation bei Dereferenzierung eines veralteten Objektes stattfindet, durchläuft Javelus die Stacks aller Threads um Methoden darauf zu reparieren, welche noch Referenzen auf veraltete Klassen aufweisen.

Zustandstransformation. Zum Updatezeitpunkt wurde bereits jedes veraltete Objekt als ungültig markiert. Um veraltete beziehungsweise markierte Objekte zu identifizieren, werden von Javelus dynamische Checks eingefügt, sodass Zugriffe auf veraltete Objekte erkannt werden können. Dadurch kann nach dem Lazy-Prinzip zur Laufzeit überprüft werden, ob in einem Objekt eine Transformation vorgenommen werden muss oder nicht. Ist dies der Fall, wird der ausführende Thread blockiert und eine Objekt-Update-Routine eingeleitet.

Beim Aufruf der Update-Routine für ein Objekt wird ein Prototyp-Objekt erzeugt, in welches alle unveränderten Attribute kopiert werden. Anschließend erzeugt Javelus das neue Objekt, wodurch dieses vom Typ der neuen Klasse ist. Ist die Größe des Objektes dabei gewachsen, wird durch Javelus ein *Mixed Object Model* eingeführt.

Ein *Mixed Object* ist ein logisches Objekt, bestehend aus zwei physikalischen Objekten. Der erste Teil enthält so viele Attribute des neu erzeugten Objektes, wie in den Speicherbereich des alten Objektes hineinpassen (*In-Place Object*). Der zweite Teil wird an einem anderen Speicherplatz alloziert und enthält die Attribute, die nicht mehr in den ehemaligen Speicherplatz passen (*Phantom Object*). Das *Phantom Object* wird dabei nur vom zugehörigen *In-Place Object* referenziert und kann deshalb nur von dort aus abgerufen werden. Um Overhead durch indirekte Zugriffe und den Verlust der Speicherlokalität gering zu halten

setzt Javelus den Garbage Collector ein, um *Mixed Objects* in normale Objekte umzuwandeln. Dabei fügt er das *In-Place Object* mit dem *Phantom Object* zusammen, formt daraus ein normales Objekt an der Stelle des *Phantom Objects* und gibt danach den Speicher des *In-Place Objects* frei.

Als letzten Schritt der Transformation, werden die unveränderten Attribute aus dem erzeugten Prototyp-Objekt in das neue Objekt kopiert und die restlichen Attribute mithilfe der *Object Transformer* initialisiert. Danach ist die Zustandstransformation eines Objektes beendet und der Thread kann weiterlaufen.

3.3 Rubah

L. Pina, L. Veiga und M. Hicks stellen ihr DSU-System Rubah in den Papern [20] und [21] vor. Sie haben bereits das DSU-System Kitsune [9] für die Programmiersprache C entwickelt, aus welchem sie Ansätze übernommen haben. Eine Besonderheit von Rubah ist, dass sowohl ein Lazy- als auch ein Eager-Algorithmus zur Zustandstransformation entwickelt wurde. Nachfolgend wird nun die Funktionsweise eines DSU mit dem Lazy-Ansatz erläutert.

Vorbereitung. Um in Rubah Updates durchführen zu können wird schon vor dem ersten Programmstart ein *Versions-Deskriptor* der initialen Version durch Rubahs *Analyzer*-Tool erstellt, welcher unter anderem eine Liste aller Klassen, die geupdated werden können, bereitstellt. Es handelt sich hierbei um alle Klassen, außer die der Java API. Anschließend wird das Programm durch den Rubah *Driver* ausgeführt. Dieser ist mit einem Custom-Classloader ausgestattet, welcher beim Laden einer Klasse dessen Bytecode anpasst, sodass diese Klasse spätere Updates unterstützt.

Soll nun ein Update stattfinden, muss dieses vorbereitet werden, indem die neue Version und der alte Deskriptor dem *Analyzer* übergeben werden, woraufhin dieser einen Deskriptor der neuen Version und eine Update-Klasse durch Vergleich der beiden Versionen erzeugt. Diese Klasse enthält *Conversion Methods* für alle Klassen, in denen Attribute verändert wurden und initialisiert diese veränderten Attribute in den neuen Klassen standardmäßig mit 0, false oder null. Der Entwickler hat die Möglichkeit diese Methoden nach Bedarf anzupassen. Danach wird der Transformer und der neue Code an den Rubah *Updater* übergeben, welcher dem *Driver* ein Update signalisiert.

Bevor das Update stattfinden kann, müssen alle Threads des Programmes an einem Update-Punkt angehalten werden. Diese Punkte müssen vom Entwickler selbst in den Code eingefügt worden sein. Dazu stellt Rubah eine API bereit, mit deren Hilfe durch den Befehl `Rubah.update(String s)` ein solcher Zeitpunkt markiert werden kann, wobei das Argument der Methode ein Label ist, um verschiedene Update-Punkte auseinander zu halten. Geeignete Stellen befinden sich laut Rubahs Entwicklern vor längeren Schleifendurchgängen. Ist das Programm an einem solchen update-sicheren Ruhezustand angekommen (man spricht hierbei auch von *Quiescence*), kann das Update durchgeführt werden.

Update. Zunächst werden alle Klassen der neuen Version in das Programm geladen. Klassen, in denen nur Methodenrümpfe verändert wurden, passt Rubah durch Java HotSwap (vgl. Kapitel 2.4) an. Bei allen anderen Klassen muss der Bytecode umgeschrieben werden. Für Operationen in denen der Bytecode einer Klasse umgeschrieben werden muss, verwendet Rubah das ASM bytecode rewriting tool [2]. Anfangs werden die neuen Klassen umbenannt, um Namenskollisionen zu vermeiden, da Rubah nur einen einzigen Classloader verwendet. Dazu wird ein Klassenname *ClassName* in *ClassName__Versionsnummer* umbenannt. Um Fehler bei Aufrufen der Reflection API (vgl. Kapitel 2.5) zu vermeiden werden all diese Aufrufe auf eine API von Rubah umgeschrieben. Als nächstes werden alle updatefähigen Attributtypen durch *Java.lang.Object* ersetzt und Aufrufe dieser mit Typecasts ergänzt. Dadurch müssen Klassen, welche ein Attribut mit dem Typ einer nun umbenannten Klasse besitzen, weder aktualisiert noch umbenannt werden und eine Kettenreaktion von Aktualisierungen und Umbenennungen bei Folgeabhängigkeiten wird vermieden. Außerdem wird ein *hashCode* Attribut eingefügt und die *hashCode*-Methode so überschrieben, dass dieses Attribut zurückgegeben wird. Dadurch kann sichergestellt werden, dass die Objekte der neuen Klasse die gleiche Identität haben wie Objekte der alten Version, indem das Attribut bei der Transformation kopiert wird.

Kontrollfluss-Migration. Nachdem das Update der Klassen erfolgt ist werden alle angehaltenen Threads neu gestartet. Für den Fall, dass ein Thread initiale Befehle ausführt, welche nicht noch einmal ausgeführt werden sollen, kann ein Entwickler diesen Befehlsblock mittels `Rubah.isUpdating()` umgehen. Dieser Aufruf gibt zurück, ob gerade eine Kontrollfluss-Migration stattfindet oder nicht. Sobald alle Threads an den Update-Punkten mit gleichem Label wie zum Zeitpunkt des Updates angekommen sind, ist die Migration abgeschlossen und das Programm wird fortgeführt. Erst wenn ein Objekt benutzt wird, findet dessen Transformation statt.

Zustandstransformation. Um Lazy-Transformation umzusetzen, erzeugt Rubah für jede Klasse eine Proxy-Klasse, in der alle Methoden überschrieben werden, um somit zu Beginn jeder Methode die Ausführung eines Transformations-Algorithmus einzuleiten. Da jedes Objekt nur einen Pointer zu seiner typbestimmenden Klasse besitzen kann, zeigt dieser nun entweder auf seine eigentliche Klasse oder die korrespondierende Proxy-Klasse. Objekte, welche auf ihre Proxy-Klasse zeigen, sind Proxy-Objekte, die zur Umsetzung der Lazy-Transformation verwendet werden. Zusätzlich wird ein Attribut *forward* eingefügt, welches auf eine neue Instanz der neuen Klassenversion verweist. Um den Transformations-Algorithmus verständlicher darstellen zu können wird im Folgenden ein Proxy-Objekt durch eine *isProxy*-Flag kenntlich gemacht und das *forward*-Attribut durch eine globale *visited*-Map veranschaulicht, in der jedes veraltete Objekt auf seine neue Version abgebildet wird und zugleich die transformierten Objekte als besucht markiert werden können.

Während des Updates wurde der Algorithmus bereits auf den *Root*-Referenzen ausgeführt. Dadurch wurden alle von den *Root*-Referenzen referenzierten Instanzen als Proxy-Objekte markiert und sie können nach Fortsetzung des Programmes bei der Dereferenzierung nach dem gleichen Algorithmus transformiert werden.

```
1  Map visited;
2
3  LAZYmigrate (Object obj)
4    LAZYtraverse(obj);
5    obj.isProxy = false;
6    visited[obj] = obj;
7
8  LAZYtraverse (Object obj)
9    for (Field f : obj)
10     Object ref = obj.f;
11     if (ref.isProxy)
12       continue;
13     else if (visited[ref])
14       CAS(obj.f, ref, visited[ref]);
15       continue;
16     else
17       Class c = ref.getClass();
18       Class newC = Rubah.mapClass(c);
19       if (c != newC)
20         Object p = Rubah.new(newC);
21         Rubah.convert(ref, p);
22         p.isProxy = true;
23         p = CAS(visited[ref], 1, p);
24         CAS(obj.f, ref, p);
25     else
26       ref.isProxy = true;
27
28  Object method(Object ... args)
29    if (this.isProxy)
30      LAZYmigrate(this);
31    // Rest of original method
```

Abb. 5. Lazy-Transformation-Algorithmus von Rubah [21]

Der Ablauf des Algorithmus (vgl. Abb. 5) läuft folgendermaßen ab: Er beginnt bei Zeile 29, nachdem eine Methode innerhalb einer Klasse aufgerufen wurde. Rubah stellt sicher, dass Zugriffe auf alle Attribute nur über Methodenaufrufe stattfinden können, sodass der Algorithmus alle Referenzen über die Einschub-Methode *LAZYmigrate()* transformieren kann. Ist bei einem Objekt die *isProxy*-Flag gesetzt, so wird die Einschub-Methode aufgerufen.

Dabei werden alle Objekte, welche dieses Objekt in Attributen referenziert, durch die Methode *LAZYtraverse()* (Zeile 9-26) durchlaufen. Sollte ein solches Objekt bereits ein Proxy sein, wird dieses übersprungen (Zeile 11-12). Für den Fall, dass es kein Proxy-Objekt ist, es aber schon einmal besucht wurde, wird die Referenz auf das neue, aktualisierte Objekt gesetzt. Dieses aktualisierte Objekt wird aus der *visited*-Map beziehungsweise aus dem *forward*-Attribut des alten Objektes gelesen.

In allen anderen Fällen wird in Zeile 17-19 geprüft, ob ein Objekt transformiert werden muss. Falls nicht wird dieses Objekt einfach als Proxy-Objekt markiert (Zeile 25-26), damit bei dessen Dereferenzierung der Algorithmus auch wieder alle zugehörigen Attribute durchläuft. Falls doch, wird mit `Rubah.new()` ein Objekt der neuen Version der Klasse ohne Konstruktor erzeugt und der Zustand des alten Objektes auf das neue transferiert (Zeile 20-21). Dies geschieht mithilfe der *Conversion Methods* aus der Update-Klasse. Anschließend wird die *isProxy*-Flag des neuen Objektes gesetzt, das alte Objekt als besucht markiert und die Referenz des alten Objektes auf das neue Objekt getauscht (Zeile 22-24).

Nach dem *LAZYtraverse()* durchgeführt wurde, wird der Proxy-Status des Objektes, von dem aus der Algorithmus gestartet ist, entfernt und es wird als besucht markiert (Zeile 5-6). Zum Schluss wird dann mit der restlichen Ausführung der eigentlichen Methode fortgefahren (Zeile 31), womit die Transformation des Objektes abgeschlossen ist.

4 DSU-Systeme im Vergleich

Nachdem die Funktionsweise der einzelnen Systeme erläutert wurde, sollen in diesem Abschnitt die drei beschriebenen Systeme im Hinblick auf den Programmieraufwand eines Anwendungsentwicklers vor einem Update und die Performance nach einem Update untersucht und gegenübergestellt werden. Begonnen wird zunächst mit einem Vergleich des Programmieraufwandes um die Systeme benutzen zu können.

4.1 Programmieraufwand / Entwicklertransparenz

Die Entwicklertransparenz eines DSU-Systems beschreibt, wie viele Eingriffe eines Anwendungsentwicklers nötig sind, um ein Update erfolgreich durchführen zu können. Wird kein Eingriff benötigt und die Ausführung des Updates findet komplett eigenständig durch ein DSU-System statt, nennt man so ein System entwicklertransparent (vgl. [6]).

Gosh! ist das einzige der in diesem Artikel betrachteten DSU-Systeme, welches von der Funktionsweise her entwicklertransparent ist. Das Update findet also vollständig autonom statt. Somit muss der Entwickler den Update-Mechanismus von Gosh! nicht kennen und kann ein Programm weiterentwickeln ohne sich dabei um mögliche DSU kümmern zu müssen.

Im Vergleich dazu ist Rubah das System mit der geringsten Entwicklertransparenz. Um in Rubah Updates durchzuführen muss der Entwickler bereits Update-Punkte in seinen Code geschrieben haben, da Rubah geeignete Punkte nicht selbstständig finden kann. Gleichzeitig müssen fehlerbehandelnde Maßnahmen in Form von *catch*-Klauseln eingefügt worden sein, da die Rubah API im Falle von auftretenden Updates Exceptions auslösen kann. Außerdem muss die Umgehung von Neuinitialisierungen bei Durchführung der Kontrollfluss-Migration durch den Entwickler festgelegt worden sein. Zuletzt kann er die standardmäßig

generierten Update-Klassen (vgl. Kapitel 3.3) für eine spätere Zustandstransformation erweitern oder ergänzen. Alles in Allem benötigt der Entwickler bei diesem System sowohl ein tiefer gehendes Verständnis des Update-Mechanismus als auch das Wissen über die alten und neuen Klassen eines Programmes, um die Update-Klassen korrekt anpassen zu können.

Javelus, das letzte der betrachteten System, ist nahezu entwicklertransparent. Wie in Gosh! wird ein Update-Punkt selbständig durch das DSU-System gefunden und der Entwickler muss sich keine Gedanken um geeignete Punkte machen. Doch auch bei Javelus kann der Entwickler die *Object Transformer* (vgl. Kapitel 3.2) ähnlich, wie die Update-Klassen in Rubah, anpassen. Auch hier ist deshalb ein tiefes Verständnis des zu updatenden Programmes sowie dessen neuer Version zwingend, sodass eine korrekte Zustandstransformation stattfinden kann. Generell lässt sich sagen, je tiefgreifender und komplexer ein Update ist, desto schwieriger und fehleranfälliger wird die eigenständige Anpassung der Transformations-Funktionen, sowohl in Rubah als auch in Javelus.

4.2 Performance

Dieses Kapitel handelt von der Performance der einzelnen Systeme. Die Entwickler von Gosh!, Javelus und Rubah haben, um die Performance ihres Systemes zu testen, verschiedene Benchmark-Tools verwendet. Außerdem haben sie die Updates mit verschiedenen Programmen auf verschiedenen Endsystemen ausprobiert und dabei auch verschiedene Prioritäten gesetzt. Während Rubah seine Performance vor, während und nach einem Update mit der Steady-State Performance einer normalen Programmausführung vergleicht, vergleichen sich Gosh! und Javelus lediglich mit einem anderen DSU-System. Aufgrund dessen ist es nicht möglich, die Systeme direkt zu vergleichen, ohne eigene Tests durchzuführen. Deshalb werden im Folgenden die Performance-Tests aus den Veröffentlichungen der Systeme zusammengestellt.

Bei Gosh! wurde die Performance mit dem kommerziellen DSU-System JRebel verglichen, da beide Systeme ähnliche Ziele und Umsetzungen haben. In den meisten Benchmarks hat Gosh! dabei besser abgeschnitten. Der durchschnittliche Overhead gegenüber der Performance ohne DSU war dabei etwas um die 10%.

Die Entwickler von Rubah verglichen ihre Erkenntnisse mit der Steady-State Performance. Sie haben dafür unter anderem einige Updates auf einer H2-Datenbank durchgeführt. Im Durchschnitt wurde dabei ein Steady-State-Overhead von etwa 5-9% gemessen. Außerdem stellten sie fest, dass die Pause während des Klassenupdates durch ein DSU mit dem Lazy-Ansatz unabhängig von der Heap-Größe ist und im Gegensatz zu ihrem Eager-Ansatz zeitlich gesehen deutlich kürzer ausfällt. Dafür sinkt die Performance nach dem Update stärker ab als bei der Eager-Transformation. Dies liegt an den Entscheidungen des JIT-Compilers (vgl. Kapitel 2.3), die er bei Optimierung des Codes nach einem Update bei Existenz von Proxy-Klassen trifft.

Auch bei Javelus wurde ein DSU auf einer H2-Datenbank durchgeführt. Hier wurde zum Vergleich ein Eager-Ansatz in Javelus implementiert. Als Ergebnis

der Tests hielten sie fest, dass die Zugriffszeiten auf einzelne Objekte beim Lazy-Ansatz geringfügig länger sind, da die Existenz von *Mixed Objects* überprüft werden muss. Dieser Overhead soll aber in der Praxis kaum bemerkbar sein und über einen längeren Zeitraum durch das Mergen der *Mixed Objects* wieder geringer werden.

5 Zusammenfassung

Der vorliegende Artikel wurde mit dem Ziel verfasst, den Stand der Technik von Dynamischen Software Updates vorzustellen, durch welche Software ohne lästige Neustarts geupdated werden kann. Dabei wurden die drei Systeme Gosh!, Javelus und Rubah in den Fokus gesetzt. Diese zielen alle auf die Programmiersprache Java ab, funktionieren mit einer standardmäßigen JVM und setzen den Ansatz der Lazy-Zustandstransformation um. Deshalb lassen sich auch Gemeinsamkeiten finden, wie zum Beispiel die Proxifizierung von Objekten zur Erkennung veralteter Objekte in Rubah und Gosh! sowie die Ähnlichkeiten bei der Initialisierung neuer Objekte in Rubah und Javelus (Update-Klassen und *Object-Transformer*). Dennoch gibt es auch wesentliche Unterschiede in der Art und Weise der Umsetzung eines DSU. In Rubah müssen sichere Update-Punkte mit der Eigenschaft der *Quiescence* manuell gefunden werden wohingegen dies bei den anderen beiden Systemen automatisch durch Überprüfung von *Activeness-Safety* stattfindet. Außerdem lädt Gosh! alle Klassen einer Version in einen neuen Classloader, während Rubah und Javelus nur einen Classloader verwenden und ihre Klassen umbenennen um Namenskonflikte zu vermeiden. Zuletzt kann es nur bei Gosh! eine Koexistenz von Objekten verschiedener Versionen geben.

Aus der Gegenüberstellung der Entwicklertransparenz wird deutlich, dass Gosh! vollkommen transparent arbeitet und somit kein Wissen über dessen Architektur nötig ist, um ein Update erfolgreich durchzuführen. Bei Rubah hingegen ist ein großes Verständnis der Funktionsweise des Systems als auch des zu updatenden Programmes von Nöten, da hier der Programmieraufwand am größten ist. Javelus liegt bezüglich der Transparenz zwischen den beiden anderen Systemen.

Das System mit der besten Performance ausfindig zu machen ist schwierig, da alle Systeme verschiedene Techniken und Ziele unterstützen. Außerdem haben die Entwickler in ihren Veröffentlichungen verschiedenste Benchmarks auf unterschiedlichsten Programmen durchgeführt. Generell lässt sich aber sagen, dass durch die Lazy-Transformation immer Performance-Einbußen im Vergleich zur Steady-State Performance entstehen, da Objekte bei ihrer Dereferenzierung zunächst auf ihren Zustand überprüft und gegebenenfalls aktualisiert werden müssen.

Welches System letztendlich für einen Softwareentwickler am ehesten in Frage kommt hängt allein von dessen eigenen Prioritäten und vom spezifischem Programm ab, welches geupdated werden soll, und kann nicht verallgemeinert werden.

Alles in allem werden DSU-Systeme bisher eher selten in der Industrie eingesetzt. Dies liegt unter anderem am noch vorhandenen Fehlerpotenzial und den eingeschränkten Updatefähigkeiten mancher Systeme. DSU-Systeme werden aber auch noch nicht all zu lange erforscht und der Trend, diese zu entwickeln beziehungsweise weiterzuentwickeln, wurde erst in den letzten Jahren immer populärer. Es bleibt durchaus interessant zu beobachten, wie nahe die Performance während eines DSU an die Steady-State Performance herankommen kann und inwieweit sich die auftretenden Probleme während eines Updates oder einer Zustandstransformation einschränken lassen. Für größere und komplexere Veränderungen eines Programmes bewährt sich jedoch bisweilen die alte Stop-Update-Neustart Methodik.

Literatur

1. Bierman, G., Parkinson, M., Noble, J.: Upgradej: Incremental typechecking for class upgrades. In: ECOOP 2008–Object-Oriented Programming, pp. 235–259. Springer (2008)
2. Bruneton, E., Lenglet, R., Coupaye, T.: Asm: A code manipulation tool to implement adaptable systems. In: In Adaptable and extensible component systems (2002)
3. Chen, H., Yu, J., Chen, R., Zang, B., Yew, P.C.: Polus: A powerful live updating system. In: Proceedings of the 29th International Conference on Software Engineering. pp. 271–281. ICSE '07, IEEE Computer Society, Washington, DC, USA (2007), <http://dx.doi.org/10.1109/ICSE.2007.65>
4. Dmitriev, M.: Safe class and data evolution in large and long-lived java[tm] applications. Tech. rep., Mountain View, CA, USA (2001)
5. Gregersen, A.R., Jørgensen, B.N.: Dynamic update of java applications—balancing change flexibility vs programming transparency. *Journal of Software Maintenance and Evolution: Research and Practice* 21(2), 81–112 (2009)
6. Gregersen, A.R., Rasmussen, M., Jørgensen, B.N.: State of the art of dynamic software updating in java. In: *Software Technologies*, pp. 99–113. Springer (2014)
7. Gu, T., Cao, C., Xu, C., Ma, X., Zhang, L., Lu, J.: Javelus: A low disruptive approach to dynamic software updates. In: *Software Engineering Conference (APSEC), 2012 19th Asia-Pacific*. vol. 1, pp. 527–536. IEEE (2012)
8. Gu, T., Cao, C., Xu, C., Ma, X., Zhang, L., Lü, J.: Low-disruptive dynamic updating of java applications. *Information and Software Technology* 56(9), 1086–1098 (2014)
9. Hayden, C.M., Smith, E.K., Denchev, M., Hicks, M., Foster, J.S.: Kitsune: Efficient, general-purpose dynamic software updating for c. In: *ACM SIGPLAN Notices*. vol. 47, pp. 249–264. ACM (2012)
10. Makris, K., Bazzi, R.A.: Immediate multi-threaded dynamic software updates using stack reconstruction. In: *Proceedings of the 2009 Conference on USENIX Annual Technical Conference*. pp. 31–31. USENIX'09, USENIX Association, Berkeley, CA, USA (2009), <http://dl.acm.org/citation.cfm?id=1855807.1855838>
11. Martinez, S., Dagnat, F., Buisson, J.: Prototyping DSU techniques using Python. In: *USENIX (ed.) HotSWUp'13*. pp. <https://www.usenix.org/conference/hotswup13/prototyping-dsu-techniques-using-python>. San José, United States (Jun 2013), <https://hal.archives-ouvertes.fr/hal-00907744>

12. Neamtiu, I., Hicks, M., Stoye, G., Oriol, M.: Practical dynamic software updating for c. In: Proceedings of the 2006 ACM SIGPLAN Conference on Programming Language Design and Implementation. pp. 72–83. PLDI '06, ACM, New York, NY, USA (2006), <http://doi.acm.org/10.1145/1133981.1133991>
13. Nicoara, A., Alonso, G., Roscoe, T.: Controlled, systematic, and efficient code replacement for running java programs, vol. 42. ACM (2008)
14. Oracle: Garbage collector basics, <http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html>
15. Oracle: Java hotspot vm, <http://www.oracle.com/technetwork/java/whitepaper-135217.html>
16. Oracle: Java reflection api, <http://docs.oracle.com/javase/tutorial/reflect/>
17. Oracle: Java virtual machine tool interface, <http://www.docs.oracle.com/javase/7/docs/platform/jvmti/jvmti.html>
18. Orso, A., Rao, A., Harrold, M.J.: A technique for dynamic updating of java software. In: Software Maintenance, 2002. Proceedings. International Conference on. pp. 649–658. IEEE (2002)
19. Pina, L., Cachopo, J.: Atomic dynamic upgrades using software transactional memory. In: Hot Topics in Software Upgrades (HotSWUp), 2012 Fourth Workshop on. pp. 21–25. IEEE (2012)
20. Pina, L., Hicks, M.: Rubah: Efficient, general-purpose dynamic software updating for java. In: Proceedings of the Workshop on Hot Topics in Software Upgrades (HotSWUp) (2013)
21. Pina, L., Veiga, L., Hicks, M.: Rubah: Dsu for java on a stock jvm. In: Proceedings of the 2014 ACM International Conference on Object Oriented Programming Systems Languages & Applications. pp. 103–119. OOPSLA '14, ACM, New York, NY, USA (2014), <http://doi.acm.org/10.1145/2660193.2660220>
22. Ritzau, T., Andersson, J.: Dynamic deployment of java applications. In: Java for Embedded Systems Workshop. vol. 1, p. 21. Citeseer (2000)
23. Sato, Y., Chiba, S.: Loosely-separated “sister” namespaces in java. In: ECOOP 2005-Object-Oriented Programming, pp. 49–70. Springer (2005)
24. Subramanian, S., Hicks, M., McKinley, K.S.: Dynamic software updates: a VM-centric approach, vol. 44. ACM (2009)
25. Venners, B.: Inside the Java virtual machine. McGraw-Hill, Inc. (1996)
26. Würthinger, T., Wimmer, C., Stadler, L.: Dynamic code evolution for java. In: Proceedings of the 8th International Conference on the Principles and Practice of Programming in Java. pp. 10–19. ACM (2010)
27. ZeroTurnAround: Jrebel, <http://www.zereturnaround.com/jrebel/>

Possibilities of Activity Recognition on Apple's iOS

Christian Herlemann*

Advisor: Anja Bachmann[†]

Karlsruhe Institute of Technology (KIT)
Pervasive Computing Systems – TECO

* christian.herlemann@student.kit.edu

[†] bachmann@teco.edu

Abstract. In this paper I give an overview about the possibilities of activity recognition on Apple's iOS platform in particular on the iPhone 6. In the first part I will take a closer look on technical specifications of the iPhone. Which sensors does it include and how do they work. Then I will highlight some key features of the iOS 8 which can be used for activity recognition with a closer look on the Core Motion API where Apple already does some activity recognition for us. In the final part I will summarise the state of the art currently found in the App Store and ideas from Apple and developers to integrate activity recognition in normal apps.

Keywords: activity recognition, iPhone, iOS, mobile sensing

1 Introduction

In the past activity recognition relied on customised hardware which was expensive and didn't had a great range of distribution. During the last decade smartphones have become an essential companion throughout the day. The new generation of mobile devices with embedded sensors has created opportunities for exploring new ways of activity recognition. Some of the available sensors in smartphones are: GPS, a compass, accelerometer, three-axis gyro sensor and ambient light sensors. In addition to its sensor technologies smartphones offer a wide range of data which can be used to classify the user's activity and improve overall accuracy in activity recognition. With close to 800M sold iPhones [1] and nearly 800M iOS devices the Apple platform is an interesting area for researches for deploying apps to track activities and collect data. While existing activity tracking apps rely solely on accelerometers and location aware sensors like GPS the iOS platform offers considerably more data that can be used for activity recognition. In this paper I discuss the opportunities of the iPhone 6 by summarising its sensor powers and highlighting additional data that can be used. I will also show limitations of the iPhone below its technological capacities due to the strong regimentation by Apple to protect user's privacy.

2 State of the Art

Applications for activity recognition include areas such as healthcare , smart homes [20] and monitoring systems [6]. In health care activity recognition is used to monitor and diagnose remote patients using systems like MEDIC [19]. In smart homes the uncertainty of the environment and the various number of users can be troubling. Chiang [20] provides a first framework for activity recognition in smart homes. A key feature of activity recognition is whether trained or untrained data is used. Bao [2] used five biaxial accelerometer performing everyday activities by 20 subjects without supervision or observation. He concluded that while some activities are recognised well with subject-independent training data, others appear to require subject-specific training data. The limitations of the studies mentioned so far that they can't recognise new unseen activities. Cheng [4] propose a new method using a semantic attribute sequence model .

Recognition of higher-level activities such as shopping or doing housework is challenging. Huynh used a supervised approach in combination with accelerometer data to detect three different high-level activities with an accuracy of more than 90%. [8]

Activity recognition on the smartphone includes many application areas. Fundamental are studies about activity recognition using the accelerometer. Xing Su presents an application that uses motion data to identify the users activity. [17] Using only a smartphone and decision trees as the basic classification model he is able to classify the activity in realtime. Similar studies have been undertaken by Abdulah [3], Cruz-Silva [5] and Saponas [16]. A major issue with accelerometer based activity recognition is that the data differs depending on the device location and orientation. Henpraserttae [7] performed two experiments: one with a device fixed on the waist in sixteen different orientations and another with three different device locations in two different device orientations. Results show that the device orientation can be normalised using a projection-based technique and different models depending on the placement of the sensor.

3 iPhone 6 Hardware

Activity Recognition is always limited to its given hardware. While it is comparatively easy to design customised hardware to track human activities it is rather challenging to use an immutable system like a smartphone. Since its introduction in 2007 the capabilities of the iPhone have increased tremendously. In this chapter I highlight the abilities of the iPhone 6's hardware and its limits due to regulations by Apple.

Touch ID Touch ID is a fingerprint sensor build into the iPhone 6 which can only be used to authenticate a user. It is not possible to get any other information out of the Local Authentication Framework.

Cellular & Wifi Core Telephony Framework Reference provides information about current calls such as the calling state and a call id and the cellular network subscriber information. Call history can be accessed by reading the `call_history.db` file but won't be approved for the App Store. The only information you can get about wifi is the SSID. Signal strength and other information is restricted by apple.

Bluetooth & NFC There is no way to access the NFC chip with any public API as Apple restricted the use of the NFC chip for Apple Pay. Instead the iPhone supports Bluetooth Low Energy which covers the whole functionality of the NFC chip. CoreBluetooth (CB) provides a list of peripherals that are currently connected with the iPhone. You can scan for peripherals with a specific service and you get access to monitoring properties such as the state of the CB CentralManager and its delegate (Bluetooth Events). CB Peripheral provides detailed information about the peripheral such as its name, a delegate to receive events from this peripheral, its UUID, its service, its RSSI and its current state (Disconnected, Connecting, Connected).

GPS, Digital Compass & iBeacon CoreLocation (CL) LocationManager provides information about your current location, you can access it's delegate to get notifications on location changes. You can monitor for a specific region (entering or leaving) and get the CLHeading attribute. CLHeading provides information about the heading direction of the device in degrees relativ to the magnetic north or the true north. It also provides information about its accuracy of the given information and you can access the raw data (geomagnetic data for the x,y and z-axis measured in microtesla). CLLocation give you processed information of the location such as the coordinate (latitude and longitude), the altitude in meters, the logical floor of the building the iPhone is located and the horizontal and vertical accuracy. CLLocation can also measure the distance between two locations and provides speed (m/s) and course information (degrees starting at true north counting clockwise) CLVisit encapsulates information about interesting places that the user has been at such as the coordinates, the arrival and departure date. CLGeocoder provides services for converting coordinates to user-friendly representation of locations (CLPlacemark) including ISO country code, the full address and an array of relevant areas of interest associated with the placemark.

CLBeacon (refers to iBeacon indoor positioning system) provides information about proximity (unknown, immediate, near, far), accuracy and RSSI of the beacon.

Three-axis accelerometer CMMotionManager stores the raw data of the systemwide shared accelerometer object. The positive sections of the axes are indicated by the thumbs in Fig 1.

Three-axis gyroscope CMGyroData provides raw data about the device's rate of rotation around three axes (like in Fig. 1). UI Device provides DeviceOrientation: Unknown, portrait, portraitUpsideDown, landscapeLeft, landscapeRight, faceUp, faceDown.

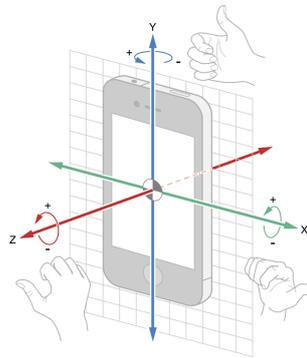


Fig. 1. Gyroscope measures rotation around the axes

Barometer CMAltimeter API gives you updates of the relative change in the current altitude, not the absolute altitude. It starts with the value 0 for the first event and then provides the changes relative to the previous event.

Lightning Connector UIDevice provides 4 battery power states: Unknown, unplugged, charging, full (only if device is plugged into power and the battery is 100% charged.) which is the only indication whether the lightning cable is plugged in or not.

Camera UIImagePickerControllerController gives you access to the iPhone's camera. Taking a picture usually requires user input you may avoid user input using a timer but this might cause that your app will be banned from the app store due to privacy concerns.

Proximity sensor UIDevice provides a Boolean value indicating whether the proximity sensor is close to the user (true) or not (false).

4 Core Motion Framework

In September 2013 Apple introduced the Core Motion Framework along with its M7 coprocessor in the iPhone 5S. Together they continuously measure data from the accelerometer, compass, gyroscope and the barometer (iPhone 6 only). The collected information of these sensors are processed by the iOS to measure your steps, the travelled distance and the elevation changes which will be available through the Core Motion Framework. Motion data will be saved for up to 7 days. The following raw data can be accessed: Accelerometer, Gyroscope, Magnetometer, Altimeter, Pedometer. The `CMMotionActivityManager` stores processed information about those sensors and therefore provides information whether the user is walking, in a vehicle or stationary for periods of time. To get this information the iOS provides two different techniques. The first one is a push request. You can register your delegate at the Activity Manager and you will get a push notification about the user's activity about every 2,5s. A notification occurs only if something has changed so you won't be notified that the user is walking over and over again. The second technique is a pull request. You can request data of a specific time period but only up to 7 days in the past. Key difference between the push and pull request is that the pull request classifies the current activity and pushes the information to the delegate while the push request uses additional info to smooth data. E.g. (e.g. no start/stop walking while waiting at the traffic lights). During a session at the 2014 WWDC Apple provided a view example scenarios of activity recognition by the CM Framework (Fig 2). Apple highlights that the only time the unknown state is true is immediately after the reboot. If the current activity is not recognised none of the flags is set true (like Zumba class example). Another important information is that the stationary flag will only be true if the device (not the user) is not moving at all like when the iPhone is lying on a table.

Device scenarios	stationary	walking	running	automotive	cycling	unknown
On table	true	false	false	false	false	false
On runner's upper arm	false	false	true	false	false	false
In dash of idling vehicle	true	false	false	true	false	false
In dash of moving vehicle	false	false	false	true	false	false
Passenger checking email	false	false	false	false	false	false
Immediately after reboot	false	false	false	false	false	true
In zumba class	false	false	false	false	false	false

Fig. 2. Example Scenarios for Activity Recognition by CM [1]

5 Available Data On iOS Devices

Smartphone like the Apple iPhone more and more become our personal assistance storing tones of personal data such as our friends and family in the addressbook.app, our dates in the calendar.app and biological information in the new health.app. This information could help developer to choose an appropriate classifier and therefore improve overall accuracy. In this section I will highlight the pre-installed apps that come on every iOS device and what information they might provide.

Health All information stored in the health.app can be accessed through the HealthKit Framework but should be used carefully due to privacy concerns which may result disapproval to the App Store.

Photos ALAssetsLibrary provides access to the photos.app but it requires user input to get a photo. There's no public API to get a list or array of all photos.

Phone & Messages Neither the information of the phone.app nor the information of the messenger.app can be accessed through public APIs.

Contacts ABAddressBook and ABPerson give access to all records in the addressbook. ABGroup gives you all records of a specified group.

Mail Can't be accessed through public API

Music MPMusicPlayerController can be used to access iPod.app. You can get a copy of the currently playing media item and it provides 3 types of notification: PlaybackStateDidChange, NowPlayingItemDidChange and VolumeDidChange.

Calendar & Reminder EKEventStore provides information such as the default calendar for events or reminder. You can either access single events by id or predicate or you can get an enumeration of events with matching predicates or matching start dates. Same for reminders.

Safari browsing history & bookmarks No public APIs to access browsing history or bookmarks.

HomeKit With HomeKit Apple introduced a standardised framework to connect to and control your smart accessory. HomeKit support may not widely supported yet due to its recent introduction but with growing support you will be able to scan the network for smart accessory and you will be notified if something is used.

Device monitoring Apple doesn't allow to monitor the device so you can't check whether the iPhone is in sleep mode / active, user input (unless your app is currently be used)

Always keep in mind that although those apps will be found on every iPhone it's up to the user to decide how much of his personal data he wants to share with an app. Therefore you have to ask for permission before you access personal information for the first time or your app might be banned from the app store.

6 Apple Watch

Early 2015 the Apple Watch will enter the market and therefore will extend the possibilities of activity recognition on the iOS platform. The sensor power of the first edition will be limited to an accelerometer, a microphone, a camera, Bluetooth LE and a NFC chip [11]. The last one will be used for Apple Pay only just like on the iPhone. The heart rate sensor sounds promising but according to recent studies it improves overall accuracy by only 2,1% [18]. There are still a lot of questions around the capabilities of the Apple Watch but the biggest disadvantage yet known might be the lag of multitasking [9]. This burden will make it hard to use the Apple Watch for online activity recognition but just as the iPhone the Watch is saving all sensor data for 7 days which will be provided through the Core Motion Framework.

Although we know only little about the Apple Watch it has great potential to improve activity recognition. John Guiry investigated the use of both smartphones and smartwatches for activity recognition and shows that a smartwatch can increase the number of recognised activities and the accuracy of recognition. He is able to differentiate between nine activities (walking, running, cycling, standing, sitting, elevator ascents, elevator descents, stair ascents and stair descents). Guiry used a Samsung Galaxy Nexus smartphone and the Motorola MotoActv smartwatch to obtain data from the accelerometer, magnetometer, gyroscope (all triaxial), GPS, light and pressure sensor. Data from his research was used to train five machine learning algorithms including Naïve Bayes, Multi-Layer Perceptrons and Support Vector Machines. Some of these models correctly classified up to 100% of all instances. Additionally Guiry was able differentiate between indoor and outdoor activities using the light and GPS sensors and the Naïve Bayes classifier.

Now that we have explored the possibilities of the iOS platform I'd like to give you an overview about existing apps which already use activity recognition. We will see that there are already numerous apps in the app store which only offer a small range of features.

7 Available Activity Tracking Apps

Activity recognition on smartphone is still in an early stage of development yet there are already many "Activity tracker" in the App Store. All of them aim to improve the users fitness and overall health by counting steps and measuring other physical activities like walking, driving, sitting and cycling. While they all have about the same capabilities you can divided them into two groups. In the first group are apps like UP (Jawbone), Nike Fuelband, Fitbit which are using external sensor worn on your wrist. The big disadvantage of those apps is that the additional hardware is usually expensive (Jawbone UP24 cost 122 € on Amazon) and is less accurate as they are solely based on accelerometer data. In the second group are apps like Moves, Breeze and Argus. These apps don't

require any additional hardware and are based on the internal sensors of the iPhone. Features include:

Moves Records walking, cycling, and running, views the distance, duration, steps, and calories burned for each activity [15]

Breeze Counts steps and shows your covered distance on a map.

Argus Counts steps and calorie counter, GPS mapping of Running, Cycling and Driving, Monitors your sleep cycles

These aren't the only apps available but all of them over about the same functionality as these three. While I couldn't find any information about how they work a close look on the app itself strongly suggests that they are just an interface for Apple's Core Motion Framework and do not classify the activities by themselves.

AcRe is a activity recognition app developed by Xing Su. [17] It is an approach that is able to recognise a wide rang of activities (e.g. walking upstairs, standing, sitting, jogging) with a high detection accuracy (up to 98.7%). Figure 3 shows the application workflow. First a basic activity recognition model is trained using the data recorded from different users inside a building. Then raw data is collected at a sampling rate of 64 Hz. Accelerometer data is used to extract features in both time and frequency domains. In the time domain they compute the basic statistics of each axis and in the frequency domain Fast Fourier Transformation is used to compute an energy/magnitude array. Decision trees are used as the basic classification method. AcRe includes an interface for user feedback that will refine the model by rewarding the corresponding learning parameter if positive feedback is given.

Contrary to the statement that the app will be available on the App Store AcRe doesn't seem to be available. (31.03.2015)

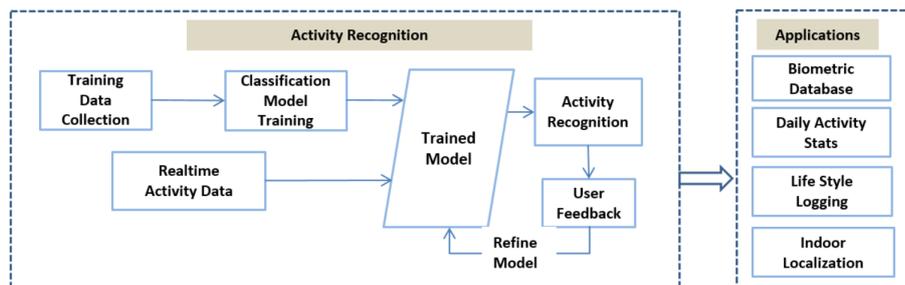


Fig. 3. Application Work Flow Chart [17]

8 Discussion: iOS vs. Android

In this section I will compare the most important aspects for researches between the Android and iOS platform.

Distribution Both Apple and Google devices have a great distribution with a market share of 19.7% for iOS [1] and 76.6% for Android [14]. While Androids market share is almost four times as high as the iOS market share there are still 800M iPhones / potential subjects.

Sensor Technology A benefit of the iOS platform is that every new generation of iPhones has the same or greater sensor power. Android on the other hand is used by many different manufacturers ranging from low-end to high-end smartphones. Every new model has its own specification making it difficult for developer as they have to check the availability of necessary sensors and adapt to their absence. The benefit of the Android is that it supports humidity and temperature sensors [12] which are not supported on the iOS platform at this point and Androids APIs offer a greater degree of utilisation. Especially virtual sensors like call history, text messages or device monitoring cannot be accessed on the iOS platform while being a good source of information on the Android platform.

OS Share iOS users quickly adapt the latest versions of Apple's iOS giving the iOS 8 a share of 77% of all Apple devices and 20% for iOS 7 [10]. The Android platform on the other hand is scattered. Old versions like Android "Gingerbread" (2.3) or "Ice Cream Sandwich" (4.0) still have shares of around 6% each, with Android "KitKat" (4.4) having the biggest share with 40,9%. The latest version Android "Lollipop" (5.0) only has a share of 3.3% [13]. The high adaption rate is a huge benefit of the iOS platform as developers need to support the two latest versions the reach 97% of all iOS users.

App Store iOS is a closed platform so Apple's app store is the only way of distribution. This is a big disadvantage for developers as there's no other option for distribution if Apple doesn't approve the app. In addition to the strict distribution channel you will need to get an Apple developer account for 99\$ per year before you can install your app on your own device. On Android apps can be downloaded from a number of different stores like Google Play or the Amazon Store.

Revenue In the fourth quarter 2014 Apple generate 10.8 billion dollar of revenue with the App Store [1] while Google only generated 3.1 billion dollar of revenue with his Google Play Store [14] making the iOS platform much more profitable than the Android platform.

Higher distribution, easier access to sensors and a less strict distribution channel makes the Android platform the more appealing platform for researchers. Less variations in sensor technologies and OS versions makes it easier for developer to create and test an app for the iOS platform. In addition to the easier development process the App Store generates three times the revenue as the Google Play Store making the iOS platform much more appealing to developers than the Android platform.

9 Conclusion

In this paper we provide a rough overview about the field of activity recognition and its latest research. We analysed the capabilities of the iOS platform by taking a close look on the available sensor technology and user data and how it can be accessed. We have seen that despite the great opportunities of the iPhone 6 we can use only a small part of it due to limitations of the iOS platform. Therefore, it is not surprising that activity recognition has not really found it's way to the App Store. All available activity tracking apps were based on the Core Motion Framework which we have analysed in section 4. Finally, we came to realise that the iOS platform may not be the best choice for researches. Missing API's and the App Store as only distribution option are serious disadvantage which you don't have to deal with on other platforms like Android. Also we summarised all available information about the upcoming Apple Watch in section 6 and evaluated its opportunities.

References

1. Apple world wide developer conference, June 2014.
2. Ling Bao and Stephen S Intille. Activity recognition from user-annotated acceleration data. In *Pervasive computing*, pages 1–17. Springer, 2004.
3. Mohd Fikri Azli bin Abdullah, Ali Fahmi Perwira Negara, Md Shohel Sayeed, Deok-Jai Choi, and Kalaiarasi Sonai Muthu. Classification algorithms in human activity recognition using smartphones. *World Academy of Science, Engineering and Technology*, 68:422–430, 2012.
4. Heng-Tze Cheng, Martin Griss, Paul Davis, Jianguo Li, and Di You. Towards zero-shot learning for human activity recognition using semantic attribute sequence model. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 355–358. ACM, 2013.
5. Nuno Cruz-Silva, João Mendes-Moreira, and Paulo Menezes. Features selection for human activity recognition with iphone inertial sensors. *EPIA2013. Springer*.
6. Gaëtanne Haché, Edward D Lemaire, and Natalie Baddour. Wearable mobility monitoring using a multimedia smartphone platform. *Instrumentation and Measurement, IEEE Transactions on*, 60(9):3153–3161, 2011.
7. Apiwat Henpraserttae, Surapa Thiemjarus, and Sanparith Marukatat. Accurate activity recognition using a mobile phone regardless of device orientation and location. In *Body Sensor Networks (BSN), 2011 International Conference on*, pages 41–46. IEEE, 2011.
8. Duy Tam Gilles Huynh. *Human activity recognition with wearable sensors*. PhD thesis, Technische Universität Darmstadt, 2008.
9. Apple Inc. <https://developer.apple.com/library/prerelease/ios/documentation/general/conceptual/watchkitprogrammingguide/>, January 2015.
10. Apple Inc. <https://developer.apple.com/support/appstore/>, March 2015.
11. Apple Inc. <http://www.apple.com/watch/>, January 2015.
12. Google Inc. <http://developer.android.com/reference/android/hardware/sensor.html>, March 2015.
13. Google Inc. <https://developer.android.com/about/dashboards/index.html>, March 2015.

14. Google Inc. <https://www.google.com/events/io>, March 2015.
15. ProtoGeo Oy. <https://www.moves-app.com/>, March 2015.
16. T Saponas, Jonathan Lester, Jon Froehlich, James Fogarty, and James Landay. ilearn on the iphone: Real-time human activity classification on commodity mobile phones. *University of Washington CSE Tech Report UW-CSE-08-04-02*, 2008.
17. Xing Su, Hanghang Tong, and Ping Ji. Accelerometer-based activity recognition on smartphone. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, CIKM '14*, pages 2021–2023, New York, NY, USA, 2014. ACM.
18. Emmanuel Munguia Tapia, Stephen S Intille, William Haskell, Kent Larson, Julie Wright, Abby King, and Robert Friedman. Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor. In *Wearable Computers, 2007 11th IEEE International Symposium on*, pages 37–40. IEEE, 2007.
19. Winston H Wu, Alex AT Bui, Maxim A Batalin, Lawrence K Au, Jonathan D Binney, and William J Kaiser. Medic: Medical embedded device for individualized care. *Artificial Intelligence in Medicine*, 42(2):137–152, 2008.
20. Jane Yung-jen Hsu Yi-Ting Chiang, Yi-Ting Tsao. A framework for activity recognition in a smart home.

Die Rolle von User Experience für Open Source Software

Josef Roth*

Betreuer: Till Riedel†

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*josef.roth@student.kit.edu

†riedel@teco.edu

Zusammenfassung. Diese Arbeit befasst sich mit der Frage inwiefern die User Experience eines Open Source Projektes seinen Einsatz und Erfolg beeinflusst. Die Arbeit untersucht die Motivationsgründe der Entwickler von Open Source Software und die Gründe Open Source Software einzusetzen. Aus diesen Gründen wird versucht mögliche Einflussfaktoren auf die Usability herauszuarbeiten.

Schlüsselwörter: oss, open source software, user experience, ux, usability

1 Einleitung

Die ursprüngliche Fragestellung dieser Seminararbeit war es zu untersuchen inwiefern die User Experience eines Open Source Projektes den Einsatz und den Erfolg dieses Projektes beeinflusst. Schwerpunkt der Arbeit sollte jedoch nicht die Usability von Software für Endanwender (wie zum Beispiel Firefox, Gimp oder Linux) sein, sondern die Usability von Open Source Libraries zur Entwicklung dieser Endanwendersoftware. Jedoch ergaben sich bei der Literaturrecherche Schwierigkeiten passende Literatur zu finden, woraus sich schließen lässt, dass diese Fragestellung noch nicht ausreichend erforscht ist. Jedoch fanden sich zwei verwandte Fragestellungen, aus denen sich Ansätze zur Erforschung der eigentlichen Fragestellung herleiten lassen.

Diese Fragestellungen sind zum einen die Frage was Entwickler dazu motiviert sich in einem Open Source Projekt zu engagieren und ihre Arbeit kostenlos zur Verfügung zu stellen. Auf der anderen Seite steht die Frage was Firmen dazu bewegt genau diese Software einzusetzen und welche Risiken und Vorteile dieser Einsatz mit sich bringt. Da diese Fragestellungen schon ausreichend erforscht sind, werden in dieser Seminararbeit der Forschungsstand dargestellt, um anschließend darauf aufbauend die eigentliche Fragestellung genauer zu spezifizieren und mögliche Ansätze zur Beantwortung dieser darzustellen.

2 Abgrenzungen

2.1 User Experience und Usability

User Experience und Usability sind zwei verwandte Begriffe, über deren Abgrenzung oft Unklarheit herrscht. Aus diesem Grund werden diese hier noch einmal klar voneinander abgegrenzt.

User Experience ist der umfassendere Begriff. Er bezieht sich auf das komplette Interaktionserlebnis des Benutzers. Sowohl die Erfahrungen vor, während und nach der Interaktion sind davon abgedeckt. Die User Experience umfasst dabei viele Merkmale: "Klare Botschaft, sinnvoller Inhalt, Vertrauen zur Marke, ansprechendes Design, einfache Lernbarkeit, schnelle Reaktionen auf Benutzereingaben, Spass an der Benutzung – diese Liste ließe sich noch eine Weile fortsetzen". [3]

Usability hingegen ist ein Teilbereich der User Experience. Sie bezeichnet nur die Benutzererfahrung während der Nutzung. Usability bezieht sich dabei vor allem auf das User Interface. Nielsen identifiziert dabei 5 Qualitätskomponenten: Lernbarkeit, Effizienz, Einprägsamkeit, Fehler, Befriedigung. [4]



Abb. 1. Unterscheidung zwischen User Experience und Usability[3]

2.2 Abgrenzung der betrachteten Open Source Software

In diesem Abschnitt soll verdeutlicht werden auf welche Art von Open Source Software sich diese Arbeit bezieht. Dazu unterscheiden wir Open Source Software nach ihrer Zielgruppe.

Die erste Gruppe ist Endanwender-Software. Sie wurde entwickelt, um vom Anwender verwendet zu werden, einen bestimmten Zweck direkt zu erfüllen.

Meistens, aber nicht notwendigerweise, hat diese Software eine grafische Benutzeroberfläche. Beispiele sind beispielsweise der Webbrowser Firefox oder das Betriebssystem Android.

Die zweite Gruppe sind Softwarebibliotheken, die sich eher an Entwickler richten. Die Software an sich stellt noch kein vom Endanwender nutzbares Produkt dar, sondern wird verwendet, um ein solches zu entwickeln. In diese Kategorie fallen vor allem Softwarebibliotheken oder Entwicklungstoolkits. Ein aktuelles Beispiel dafür ist das Android SDK zur Entwicklung von Apps.

Wie schon weiter oben erwähnt, wird in dieser Arbeit versucht Erkenntnisse über die User Experience von Softwarebibliotheken zu gewinnen.

3 Beweggründe der Entwickler

In diesem Abschnitt wird zuerst der Forschungsstand zu den Beweggründen von Entwicklern dargestellt. Zuerst werden dabei auf die Motivationsgründe für den Einstieg in die Mitarbeit in ein Open Source Projekt untersucht und anschließend die Motivationsgründe für den dauerhaften Verbleib in einem Open Source Projekt. Abschließend wird noch eine Arbeit vorgestellt, die spieltheoretisch den Einfluss der Architektur eines OSS-Projektes auf den Nutzen für die Entwickler untersucht.

3.1 Motivationsgründe für die Entwicklung von OSS

Von Krogh et al. [5] fassen in einer Übersichtsarbeit 2008 den Stand der Forschung zur Motivation der Entwickler gut zusammen. Sie identifizieren dabei drei große Gruppen von Motivationsgründen: intrinsische, verinnerlichte extrinsische und rein extrinsische Gründe.

Intrinsische Motivationsgründe sind Beweggründe, die als innerer Antrieb für die Arbeit an Open Source Projekten dienen. Dazu zählen zum Beispiel ideologische Überzeugung und Altruismus. Ein anderer wichtiger Grund ist aber auch die Freude an der Arbeit an den Problemstellungen des Projektes.

Verinnerlichte extrinsische Motivationsgründe bezeichnen die Gründe, die eigentlich extrinsisch motiviert sind, aber von Entwicklern so verinnerlicht werden, dass sie schon als innerer Antrieb dienen. Zu dieser Kategorie von Gründen gehören zum Beispiel das Erlangen von Ansehen, gegenseitiges Geben und Nehmen, das Erlernen neuer Fähigkeiten sowie der eigene Nutzen, den man aus den Open Source Projekt ziehen kann

Als rein extrinsische Gründe konnten von Grogh et al. zum einen die Hoffnung auf positive Effekte auf die persönliche Karriere identifizieren. Außerdem gibt es eine kleine Gruppe von Entwicklern, die für ihre Arbeit an Open Source Projekten bezahlt werden.

3.2 Motivationsgründe für den dauerhaften Verbleib in einem Open Source Projekt

Fang und Neufeld [2] beschäftigen sich in hingegen mit der Frage warum Entwickler auch langfristig bereit sind sich in einem Open Source Projekt zu engagieren. Zu diesem Zweck führen sie am Projekt phpMyAdmin eine Fallstudie durch. Ihre Haupthypothese baut dabei auf der Theorie der *Legitimate Peripheral Participation* auf, weswegen diese hier einführend dargestellt wird.

Die LPP-Theorie versucht zu erklären wie die drei Konzepte der *Beteiligung*, des *situierten Lernens* und der *Identitätsentwicklung* in einer *community of practice* zusammenhängen und sich gemeinsam entwickeln.

Unter Beteiligung wird in einer *community of practice* nicht nur das Engagement durch Mitarbeit verstanden, sondern auch, dass man ein aktiver Teilnehmer an den sozialen Prozessen der Community ist und dieser gegenüber eine Identität aufbaut. Die Mitwirkung ist dabei Grundlage für das situierte Lernen und die Identitätsentwicklung.

Das Konzept des *situierten Lernens* beschreibt die Aspekte des Lernens, die sich vor allem auf den sozialen und situationsbezogenen Kontext des Lernens beziehen. Es geht nicht um das theoretische Lernen neuer Fähigkeiten, sondern um das Lernen durch sachkundiges Handeln. Es geht dabei auch darum das eigene Handeln zu hinterfragen, Feedback einzuholen und Themen mit anderen zu diskutieren.

Der dritte Aspekt der LPP-Theorie ist die Identitätsentwicklung innerhalb der *community of practice*. Die Identitätsentwicklung bezieht sich auf den Prozess, dass man innerhalb der Community lernt welche Rolle man hat, was man zu dieser Community beitragen kann und auch wie man von der Community gesehen wird. Durch die Entwicklung einer positiven Identität wird man ein Mitglied der Community und erhält auch nach und nach mehr Privilegien (z.B. im Fall von OSS-Projekten Schreibrechte auf die Versionskontrollsysteme).

Fang und Neufeld stellen in ihrer Arbeit drei Hypothesen auf, die sie am Beispiel der Fallstudie untersuchen. Die erste Hypothese ist, dass dauerhafte Mitarbeit in einem Open Source Projekt mit der Entwicklung des *situierten Lernens* und der *Identität* korrelieren. Die zweite Hypothese besagt, dass die dauerhafte Mitarbeit mit den Einstiegsmotiven der Entwickler assoziiert sind. Die dritte Hypothese bezieht sich auf den ursprünglichen Status, den die Entwickler eingeräumt kriegen. Dieser drückt sich durch den eingeräumten Zugang zum Projekt aus. So wird ein eher peripherer Zugang, wie zum Beispiel Zugriff auf die Mailinglisten oder den Bugtracker, von einem privilegiierterem Zugang, z.B. in Form von Zugriff auf das verwendete VCS, unterschieden. Ein privilegiierterer Status soll nach der dritten Hypothese die Entwickler zu einer verstärkten und längerfristigen Mitarbeit motivieren.

An Hand der Fallstudie sehen sie die erste Hypothese bestätigt. Bei Entwicklern, die über einen langen Zeitraum hinweg im Projekt aktiv waren, konnten sie dauerhaft eine positive Identitätsentwicklung und dauerhafte Prozesse des *situierten Lernens* feststellen. Entwickler, die hingegen nur über einen begrenzten Zeitraum aktiv waren, machten in ihren Augen anfänglich ähnliche Erfahrungen,

jedoch stagnierten der Prozess des situierten Lernens und der Identitätsentwicklung nach einer Weile. Dies führte dazu, dass sich diese Entwickler wieder aus dem Projekt zurückzogen. Entwickler, die jedoch früh negative Erfahrungen in Bezug auf das situierte Lernen und die Identitätsentwicklung machten, zogen sich schnell wieder aus dem Projekt zurück.

Die beiden anderen Hypothesen, die sie vor allem als Ansätze in der bestehenden Literatur gefunden hatten, sehen sie durch die Fallstudie widerlegt. Sie konnten bei allen Entwicklern unabhängig von der späteren Beteiligung als Anfangsmotiv vor allem den eigenen Nutzwert ausmachen. Dennoch entwickelten sich nicht alle Entwickler zu dauerhaften Mitwirkenden. Auch die dritte These sehen sie widerlegt. Fast alle Entwickler (bis auf die Administratoren des Projektes) hatten ursprünglich nur peripheren Zugang. Erst durch konstante Mitarbeit wurden den dauerhaft mitwirkenden Entwicklern mehr Zugangsrechte eingeräumt. Daraus lässt sich also kein Motivationsgrund für die dauerhafte Mitarbeit herleiten.

3.3 Einfluss der Architektur

Wie im vorherigen Absatz schon geschrieben, ist der eigene Nutzwert eines Open Source Projektes ein wesentlicher Motivationsgrund für Entwickler, um sich an der Entwicklung dieser zu beteiligen. Baldwin und Clark [1] untersuchen in ihrer Arbeit durch ein spieltheoretisches Modell, ob sich die Architektur eines Open Source Software auf den Nutzwert für die Entwickler auswirkt.

Sie vergleichen dabei in mehreren Schritten eine Software mit einer monolithischen Architektur, die nur von einem einzelnen Entwickler geschrieben werden kann mit einer funktional gleichen Software mit einer modularen Struktur, die von mehreren Entwicklern geschrieben werden kann. Sie stellen fest, dass durch die modulare Architektur die Kosten für den einzelnen Entwickler sinken, der Nutzen jedoch gleich bleibt. Der Nutzwert (Nutzen - Entwicklungskosten) steigt jedoch. Es ist also für alle Entwickler attraktiver eine Software mit einer modulareren Struktur gemeinschaftlich zu entwickeln.

Auch durch einen höheren *option value* steigt der Nutzwert der Software. Damit ist gemeint, dass jede Designentscheidung in der Architektur auch wieder eine Reihe von Optionen schafft diese Software weiterzuentwickeln und einzusetzen. Eine Software mit einem hohen option value hat auch einen höheren Nutzwert, da es einfacher ist auf ihr aufbauend die eigentlich benötigte Software zu erstellen.

Gleichzeitig stellen sie fest, dass diese beiden Effekte auch das sogenannte *free-riding* vermindern. Dadurch, dass die Arbeit unter mehr Entwicklern aufgeteilt wird, beteiligen sich auch mehr Entwickler an der Arbeit. Es gibt also mehr aktive Mitarbeiter und weniger Free-rider, die ohne jeglichen Einsatz nur den Nutzen der Software abgreifen.

3.4 Bezug zur User Experience

Dieser Abschnitt hat zum Ziel aus den gefundenen Motivationsgründen der Entwickler den Bezug zur User Experience von Open Source Projekten herauszuarbeiten. Dazu werden die einzelnen Punkte aufgegriffen und versucht sie einem Einflussfaktor zur User Experience zuzuordnen.

Bei den intrinsischen Gründen kann man einen Einfluss auf die Faktoren Zufriedenheit, Glück und Freude vermuten. Die intrinsischen Gründe sind die Gründe, die den Entwickler aus seinem inneren Heraus motivieren. Eine erfolgreiche Arbeit an einem Projekt, dessen Motivation so begründet ist, sollte vermutlich zu einem Anstieg dieser Faktoren führen.

Bei den verinnerlichten extrinsischen Gründen dürfte vor allem der Nutzen des Projektes die User Experience beeinflussen. Jedoch auch das Vertrauen, dass durch das gegenseitige Geben und Nehmen ausgedrückt wird, dürfte einen positiven Einfluss auf die User Experience haben.

Die in der Arbeit von Fang und Neufeld als wesentlich erkannten Prozesse des situierten Lernens und der Identitätsentwicklung sind vor allem im Bereich der User Experience einzuordnen. Je nach Verlauf des Prozesses, kann man einen direkten Einfluss auf die Zufriedenheit oder persönliche Enttäuschung der Entwickler vermuten. Durch eine Unterstützung dieser beiden Prozesse könnte man also die User Experience der Entwickler verbessern und so mehr langfristig engagierte Entwickler gewinnen.

Der Einfluss der Architektur wird sich hingegen eher auf die Usability des Projektes auswirken. Eine modularer Aufbau mit mehr option value lässt vermuten, dass die Verwendung des fertigen Produktes und das Entwickeln eigener, darauf basierender Produkte, einfach fallen kann. Die Untersuchung der Messbarkeit dieses Zusammenhangs ist eine der durch diese Arbeit aufgeworfenen offenen Fragen.

4 Gründe für den Einsatz von Open Source Software

Für den Einsatz von Open Source Software sprechen zahlreiche Gründe aus einem breiten Spektrum von philosophischen beziehungsweise ideologischen, wirtschaftlichen und technischen Argumenten. Ein umfassender Überblick über dieses komplette Themenspektrum würde den Rahmen dieser Arbeit sprengen. In diesem Kapitel werden deswegen vor allem die Gründe betrachtet, aus denen man versuchen kann einen Einfluss auf die User Experience abzuleiten.

4.1 Allgemeine Gründe

In der Übersichtsarbeit von Wheeler[6] werden zahlreiche Gründe genannt, die für den Einsatz für den Open Source Software sprechen. Die in diesem Abschnitt dargestellten Gründe sind die Popularität, die Zuverlässigkeit, die Performance und die Sicherheit.

Popularität Wheeler führt zahlreiche bekannte Open Source Projekte an, die in ihrem Einsatzmarkt eine hohe Popularität genießen und teilweise auch eine marktführende Position inne haben. Dies führt zu einer hohen Markenbekanntheit und damit auch zu einem breiteren Einsatz dieser Projekte. Ein breiter Einsatz und eine marktführende Position schaffen nicht nur eine Marke, sondern erhöhen auch das Vertrauen in das verwendete Produkt. Aus einer starken Verbreitung lässt sich auch auf eine große Nützlichkeit für eine große Masse von Anwendern schließen.

Zuverlässigkeit Es werden auch zahlreiche Studien und Beispiele genannt, die zeigen, dass zahlreiche Open Source Produkte eine höhere Zuverlässigkeit bieten als vergleichbare proprietäre Produkte. Die genannten Programme liefen im Allgemeinen stabiler und mit weniger Fehlern als die proprietäre Konkurrenz. Diese Zuverlässigkeit schafft Vertrauen in die eingesetzte Software und hat damit auch einen positiven Einfluss auf die User Experience.

Performance Aber auch die Performance ist ein Einflussfaktor für die Usability. Wheeler zieht hier den Hauptvergleich zwischen Linux und Windows beziehungsweise zwischen vergleichbarer Software, die auf diesen beiden Plattformen läuft. Er erkennt dabei eine höhere Performance und bessere Benchmarkergebnisse der Linux-basierten Setups. Aus einer besseren Performance lässt sich auch auf eine bessere Effizienz der IT-Umgebung schließen, so wird diese zu einem Einflussfaktor für die User Experience.

Sicherheit Auch die Sicherheit von Software ist ein Faktor, der in das Vertrauen in die eingesetzte Software mit einfließt. Wheeler zeigt auf, dass es zahlreiche Studien gibt, die Open Source Software in vielen Fällen eine bessere Sicherheit als den vergleichbaren proprietären Lösungen bescheinigt. Durch die Verfügbarkeit des Quellcodes ist es außerdem möglich diesen auf Sicherheitslücken zu untersuchen und einen eigenen Audit durchzuführen. Die so insgesamt eher höher zu bewertende Sicherheit führt auch wieder zu einem stärkeren Vertrauen in die Open Source Projekte.

4.2 Kommerzieller Support und User Experience

In diesem Abschnitt möchte ich einen möglichen Faktor auf die User Experience aufzeigen. Dies soll nur als möglicher Ausgangspunkt für Forschungsarbeit dienen, die diese Idee versuchen könnte zu blegen. Es geht dabei um die Frage inwiefern kommerzieller Support die User Experience eines Open Source Projektes beeinflusst.

Firmen müssen bei der Auswahl von Software nicht nur auf technische Eigenschaften achten, sondern auch auf wirtschaftliche Gründe. Da bei Open Source Software keine Anschaffungs- und auch keine laufenden Lizenzkosten existieren,

ist hier natürlich ein Vorteil zu vermuten. Auch aus diesen wirtschaftlichen Gründen ist kommerzieller Support zu Software für Firmen ein wichtiges Argument bei der Auswahl von Software.

Fehlender Support kann schnell zu einem großen Kostenfaktor werden, da das notwendige Know How zur Auswahl und zum Betrieb der nötigen IT-Lösung ohne Support selbst aufgebaut werden muss. Auch bei technischen Problemen ist schnelle Abhilfe wichtig, ein Supportvertrag mit garantierter schneller Lösung von Problemen ist hier für Firmen wichtig.

Anders formuliert, kann man auch sagen, dass Firmen aus wirtschaftlichen Gründen durch Supportverträge eine möglichst hohe User Experience erwarten. Möglichst wenig eigener Aufwand, die geringeren Anforderungen an das notwendige Know How und ein möglichst reibungsloser Betrieb sind Faktoren, von denen man einen Einfluss auf die User Experience vermuten kann. Bei Softwarebibliotheken hingegen sind die Aufgaben des Supports weniger die Auswahl und der störungsfreie Betrieb der Lösung als viel mehr die Schulung der Entwickler, die die Softwarebibliothek einsetzen sollen. Denkbare Faktoren sind hier zum Beispiel die Existenz einführender und weiterführender Schulungen.

Auch für Open Source Produkte existiert häufig schon kommerzieller Support, der beispielsweise vom Hersteller (zum Beispiel im Umfeld von Linuxdistributionen wie Ubuntu) oder von kleineren, lokalen IT-Dienstleistern angeboten wird. Hier wären also die Fragen wichtig in welcher Qualität Support geboten wird, wie man gute Support-Dienstleister findet und auch ortsnahe Support bieten können.

5 Ergebnis und Ausblick

Wie in den vorherigen Kapiteln existieren einige Faktoren, die sich in Hinblick auf einen möglichen Einfluss auf die User Experience von Open Source Software interpretieren lassen. Allerdings sind diese Überlegungen bisher nur als Hypothese zu verstehen, die in weitergehenden wissenschaftlichen Untersuchungen belegt werden müssten.

Es gibt jedoch noch einige weitere denkbare Faktoren, die in dieser Arbeit mangels passender Literatur nicht untersucht wurden. Die Usability dürfte zum Beispiel durch die Existenz einer umfangreichen, aktuellen und gut verständlichen Dokumentation positiv beeinflusst werden. Aber auch eine gute begleitende Webseite, die einen schnellen Überblick über die technischen Fähigkeiten des Projektes gibt, dürfte die User Experience positiv beeinflussen. Jedoch sind auch dies nur Hypothesen, deren wirklicher Einfluss noch zu untersuchen ist.

Literatur

1. Baldwin, C.Y., Clark, K.B.: The architecture of participation: Does code architecture mitigate free riding in the open source development model? *Management Science* 52(7), 1116–1127 (2006)

2. Fang, Y., Neufeld, D.: Understanding sustained participation in open source software projects. *Journal of Management Information Systems* 25(4), 9–50 (2009)
3. Müller, J.: Feine unterschiede: User experience vs. usability. Web page (2013), <http://www.uxcite.de/forschung/user-experience-vs-usability/>, abgerufen: 16. März 2015
4. Nielsen, J.: Usability 101: Introduction to usability. Web page (2012), <http://www.nngroup.com/articles/usability-101-introduction-to-usability/>, abgerufen: 16. März 2015
5. Von Krogh, G., Haefliger, S., Spaeth, S., Wallin, M.: Open source software: What we know (and do not know) about motivations to contribute. The DRUID Conference: 17 June 2008; Copenhagen, Denmark (2008)
6. Wheeler, D.A.: Why open source software/free software (oss/fs, floss, or foss)? look at the numbers! Web page (2005), http://www.dwheeler.com/oss_fs_why.html, abgerufen: 22. März 2015

Human Behavior and Non-Intrusive Load Monitoring

Mathias Kleiner*

Betreuer: Yong Ding[†]

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*ugdgt@student.kit.edu

[†]ding@teco.edu

Zusammenfassung. Stromsparen bekommt in der Gesellschaft einen immer höheren Stellenwert, angesichts der Klimavorhersagen, Klimakonferenzen, etc. versucht man auch den privaten Verbrauchern ein Bewusstsein für kontrollierte Energienutzung im Haushalt zu bilden. Hier wird eine Methode nahegelegt die es kostengünstig schaffen soll den Verbrauchern so detailliert wie möglich ihre Verbrauchszahlen einzelner Geräte darzulegen. Der Non-Intrusive Load Monitoring soll in Zukunft einfach und effektiv diese Aufgaben für alle Haushalte übernehmen.

Schlüsselwörter: NIALM, George W. Hart, Eindringtiefe, Steady-State-Signatur, Stromverbrauch privater Haushalte

1 Einleitung

1.1 Motivation

Non-Intrusive Appliance Load Monitoring, im folgenden NALM genannt, ist eine Methode die sämtliche Änderungen im Stromverbrauch eines Haushaltes messen und analysieren soll und Rückschlüsse darüber schließt welche Geräte zu welchem Zeitpunkt wie lange an waren. Dabei wird die gesamte elektrische Energie über einen bestimmten Zeitraum mithilfe spezieller Algorithmen durchsucht nach so genannten Signaturen die Hinweise auf einzelne Geräte zulassen. Die Methode braucht im Gegensatz zu herkömmlichen Methoden keine Stromzähler oder andere Sensoren in den einzelnen Geräten, man misst nur an einem einzelnen Punkt, meist dort wo der gesamte Strom eingespeist wird, dadurch wird die benötigte Hardware auf ein Minimum beschränkt. Um auf herkömmliche Weise diese Auflistung der einzelnen Geräteaktivphasen zu realisieren bräuchte es in jedem Gerät ein Sensor, ein Aufwand der sich kostentechnisch nicht lohnt. Die Idee wurde hauptsächlich entwickelt um Verbrauchern eine genauere Einsicht in ihren Stromverbrauch zu geben, detailliert aufgelistet welche Geräte sie wie oft einsetzen und wie viel Strom sie damit verbrauchen. Damit soll das Bewusstsein der Verbraucher für Energieverschwendung und Energieeinsparung geschärft werden. Studien haben gezeigt dass dieses Bewusstsein durch größeres Feedback

gesteigert werden kann. Gleichzeitig, wie durch "Non-Instrusive", also Nicht eindringen vermittelt wird, ist diese Methode so wenig beeinflussend wie möglich und wird von den Verbrauchern kaum wahrgenommen. Ein positiver Neben-aspekt der Forschung ist das es auch für die Dienstanbieter wie Stromkonzerne interessant ist, da sie viele Haushalte(bis zu 1000) mit einer NALM-Methode günstig überwachen können.

In dieser Arbeit gehe ich auf die Funktionsweise des NIALM ein, beispielhaft erklärt durch den ersten entwickelten Algorithmus. Vorher werde ich dafür einige Komponente des Algorithmus spezifischer erklären. Danach erkläre ich die geschichtliche Entwicklung des Algorithmus bis zur heutigen Forschung. Zum Schluss werfe ich einen kurzen Blick auf die eigentliche Implementierung der Software und schließe mit einem persönlichen Fazit.

2 NALM-Algorithmus

Die Idee zu dem Algorithmus hatte George W. Hart vom MIT, 1982 nahm er an einem Projekt teil in dem er im 5s-Zeitraster Messungen am Einspeisepunkt in privaten Haushalten durchführte. Er erkannte, dass ein Algorithmus möglich sein müsste um Schaltereignisse aus seinen Messungen zu filtern. Sein daraufhin entstandener Algorithmus ist bis heute der meist rezitierte, auf ihn Bauen die nachfolgenden Verbesserungen auf. Deshalb werde ich diesen Prototypen im folgenden vorstellen. Dazu benötigt man zum Verständnis einige Grundlagen und Ideen auf die der Algorithmus aufbaut.

2.1 Verschiedene Ansätze

Es gibt zwei grundsätzliche Ansätze für die NALM-Entwicklung, sie unterscheiden sich hinsichtlich des Grades ihrer Eindringtiefe in die Haushalte:

1. Manual Setup(MS-NALM): Der MS-NALM benötigt ein einmaliges Eindringen in den Haushalt um sämtliche Geräte zu identifizieren, anhand ihrer typischen Signaturen. Dabei werden sämtliche Geräte einmalig manuell ein/ausgeschaltet, anhand der Änderung im Gesamtstrom wird die Signatur festgelegt.
2. Automatic Setup(AS-NALM): AS-NALM dagegen vergibt allen erfassten Signaturen mit einer a-priori Liste in einer Initialisierungsphase den am logischsten Erscheinenden Namen. Der Automatic Setup ist das erwünschte totale Uneindringende System, es benötigt dazu allerdings die Datensätze des MS-NALM.

AS-NALM ist das Erklärte Ziel für private Haushalte durch seine komplette Nichteindringung in den Haushalt, für andere Bereiche eignen sich auch heute noch der Manual Setup mehr, wenn der Grad der Nichteindringung irrelevant ist bietet dieses System immer noch kaum Hardware und dadurch für z.B. Firmen geringe Kosten bei höherer Genauigkeit und Akkurarität.

2.2 Modellmöglichkeiten

Die Verbrauchsgegenstände im Haushalt bestehen aus unterschiedlich vielen verschiedenen Stufen. Diese Stufen müssen klar erkennbar sein um Signaturen den einzelnen Geräten zuordnen zu können. Man unterscheidet hauptsächlich zwischen 4 verschiedenen Verbrauchstypen.

1. On/Off-Geräte, bestehend aus genau zwei Zuständen, z.B. für Toaster oder Glühbirne
2. Finite State Machines(FSM): Geräte die zwischen mehreren Zuständen hin und herspringen, dabei aber täglich oder wöchentlich ähnliche Muster verfolgen, z. B. eine Waschmaschine.
3. kontinuierliche variable Geräte: Stromverbrauch folgt keinen periodischen Mustern, z.B. Bohrmaschine.
4. stetige 24h-laufende Geräte wie z.B. Uhren.

Die Schwierigkeit Signaturen der Geräte richtig zu deuten steigt von 1 zu 4 an, stetig laufende Geräte sind kaum oder nur extrem schwer und aufwendig zu entdecken, aufgrund der geringen Stromverbräuche dieser Geräte liegt darauf allerdings auch kaum ein Fokus.

Jedes Gerät kann als Zustandsmodell dargestellt werden, in Abbildung 1 sieht man ein Gerät mit zwei Zuständen, Anschalten und Ausschalten und die zugrundeliegende Veränderung in der Wirkleistung(P) und der Blindleistung(Q). Die meisten Geräte im Haushalte sind in zwei Zuständen modellierbar, Komplexere Systeme wie Spülmaschine sind zerlegbar in mehrere On/Off-Zustandsmodelle.

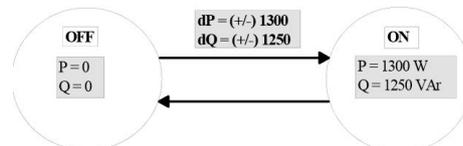


Abb. 1. Eine Wasserpumpe mit zwei Zuständen

2.3 Signaturen

Das Messen und Deuten von Gerätesignaturen ist die Essenz aller NALM-Algorithmen. Eine Signatur eines Geräts ist die persönliche individuelle elektrische Charakteristik, man definiert sie häufig über einen messbaren Parameter der Gesamtlast, welcher Informationen über den aktuellen Zustand eines Gerätes liefert. Es gibt sowohl eindringende wie auch nicht-eindringende Signaturen, für den

NALM benötigen wir logischerweise die nicht-eindringenden. Diese sind passiv messbar durch das Beobachten der Gesamtlast und dem Analysieren von Änderungen(Step Changes).

Man unterscheidet zwei Arten von Signaturen während einer Zustandsänderung eines Geräts, einmal die Steady-State-Signatur, dem gegenüber die transiente Signatur. Die Transiente Signatur ist nur während der Zustandsänderung kurz messbar, ein Aufklackern in der Last während die Steady-State-Signatur dauerhaft messbar ist.

- Steady-State-Signaturen: Die Signatur basiert auf dem Erkennen der Steady-State-Wechsel eines Gerätes, im folgenden Events genannt, berechnet wird dabei die Differenz zwischen Zustand Off \rightarrow On, sowohl in Wirk/ als auch in Blindleistung(dP , dQ in Abbildung 1). Anderst ausgedrückt ein Event ist wenn eine hereinkommene Messung von einem Wert auf einen anderen Springt. In Abbildung 2 sieht man mehrere Events. Event Signaturen treten in Grundfrequenz und harmonischen Oberwellen auf. Zum Beispiel Motoren haben eine dreispitzige Stromwellenform mit mehreren Oberwellen. Um insgesamt alle elektrischen Geräte von Haushalten zu erfassen sind die ersten 11 Oberwellen ausreichend.

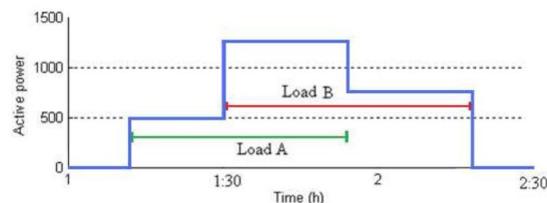


Abb. 2. statistische Features des Fraunhofer-NALM

- Transiente Signaturen: Wenn ein Gerät eingeschaltet wird kommt es wie erwähnt zu einem Wechsel der Steady-State-Zustände. Während diesem Vorgang kommt es im elektrischen Signal zu kurzen Schwankungen, die transienten Signaturen. Diese Signaturen sind schwieriger zu entdecken und geben weniger Informationen über den Gerätetyp, sind aber dennoch häufig sehr wertvoll. Es kann vorkommen dass Event Signaturen von zwei Geräten sehr ähnlich sind, während die transienten Signaturen gravierende Unterschiede haben, dadurch können die Geräte unterschieden werden.

Der Algorithmus von Hart berücksichtigt nur Event Signaturen, die harmonischen Oberwellen werden ebenfalls nicht von ihm erfasst, auf Harts Muster aufbauende Algorithmen dagegen verwenden zum Teil diese Signaturen wie später gezeigt wird.

2.4 Der Algorithmus

Die vorgestellten Komponenten zusammengesetzt ergeben nur den Algorithmus von Hart. Hier vorgestellt wird die AS-Konfiguration.

1. Stromspannung messen

Am Hausanschlusspunktes eines Haushaltes messen wird die Spannung und der Strom aller 3 Phasen gemessen. Die Wirk- und Blindleistung jeder Phase wird im Zeitraster von einer Sekunde zugrunde gelegt.

2. Normalisieren

Die Schwankung der Versorgungsspannung würde denn Algorithmus schwer irritieren, da die Teils heftigen Schwankungen als Event betrachtet werden können, deshalb muss sie normalisiert werden.

3. Kantenerkennung

Der normalisierte Strom wird in einen Kantenerkennungsalgorithmus eingespeist, dieser sucht nach allen Ereignissen und listet sie mit Größe und Zeitpunkt auf. Der Detektor zerlegt den Gesamtstrom in Segmente in denen der Strom statisch ist und in welche in denen er Änderungen zeigt. Die statistischen Segmente haben eine Toleranzschwelle von 15 W oder VAR, sie werden gemittelt um eine möglichst geringe Störung darin zu haben. Der Unterschied zwischen den einzelnen gemittelten Perioden ist dann die gemessene Stromänderung. Die Zeit des ersten Messwertes in einer Änderungsphase setzt einen Zeitstempel, diese werden zu einem p-Vektor zusammengefasst, dieser Vektor ist die Ausgabe an den nächsten Schritt. Allerdings werden Ausgaben die unter einer bestimmten Größe sind verworfen um Speicher zu sparen und wir sehr kleine Geräte ignorieren.

4. Clusteranalyse

Wir definieren die Stromänderungen im p-Raum als Punktwolken und gruppieren diese in Cluster. Ideal wäre dabei ein Cluster für jede Änderung einer Applikation. Ein für NALM entwickelter Algorithmus dafür durchläuft einmal die kompletten Daten und bestimmt die Anzahl an Clustern. Dabei hat er die Möglichkeit Cluster zu teilen und zu verbinden.

5. Modellierung der Applikationen

Mithilfe der Cluster bilden wir jetzt für jedes Gerät ein Zustandsmodell. Um ein On/Off-Modell zu konstruieren nimmt man immer die Cluster die einander gegensätzlich zum Nullpunkt stehen, also der Schwerpunkt eines Clusters ist die negative Umkehrung des anderen. Die beiden Schwerpunkte bilden dann einen Kreisbogen mit zwei Zuständen, dadurch ist der Satz ZL-SC erfüllt.

6. Spurverhalten

Die nun vorhandenen Zustandsmodellen werden mithilfe des oben genannten

Viterbi-Algorithmus dekodiert, damit erhält man nun die genauen Zeiten in denen eine Applikation an und aus war.

7. Statistisches Erfassen

Es liegt jetzt für jede Zustandsänderung das Energieniveau und die Zeit vor, dies wird tabellarisch geordnet für alle Geräte.

8. Gerätebenennung

Zuletzt müssen die Geräte bestimmt werden, beim MS-NALM ist das einfach da die Verbräuche darauf hindeuten, beim AS-NALM muss man allerdings anhand der gegebenen Daten Entscheidungen treffen.

Die Schritte 4, 5 und 8 sind nur für den Automatic Setup nötig, beim MS-Algorithmus wird dies vorab bestimmt.

3 Fortschritt und Erweiterungen

Ca 10 Jahre dauerte es bis nach unzähligen Tests ein AS-NALM, basierend auf dem Algorithmus von Hart erstmals kommerziell vermarktet werden sollte. 1996 übernahm die Firma Telog Instruments die Vermarktung. Die Kommerzielle Version für Haushalte besteht aus Messgeräten und einem Hostcomputer. Über eine Telefonleitung werden periodisch die Daten über die Stromänderungen von Messgerät zu PC geladen. Der Algorithmus ist im Host installiert, die erzeugten Cluster werden mit einer Softwarebibliothek voller Gerätesignaturen verglichen, bei einem Treffer kommt es zur Identifikation. Mit bis zu 300 Messgeräte sollte ein Hostcomputer damals schon klarkommen. Gleichzeitig wurde ein System für Firmen entwickelt, es misst die Leistung an allen 3 Phasen, nicht nur an zwei wie das System für private Haushalte. Außerdem bezieht es neben der Grundfrequenz auch harmonische Oberwellensignaturen mit ein. Zeitgleich erforschte Dr. Steven Leep am MIT die Einbeziehung von transienten Signaturen in den NALM-Algorithmus.

Insgesamt unterscheiden sich die auf Hart folgenden Ansätze in der Hardware zum Messen,

3.1 Ansatz von H. Pihala

Ein anderer Ansatz kam 1998 von Hannu Pihala heraus, dieser orientierte sich stark an Harts Algorithmus und verbesserte dessen MS-NALM durch zusätzliche Identifikation von Finite State Machines, also Zustandsmodelle für mehr als zwei Zustände. Dem zugrunde lag ein neuer kWh-Meter der bis zu 12 elektrische Zustandsvariablen pro Phase und pro Sekunde messen konnte. Die ermittelten Daten werden auf einem normalen Standard-PC geladen, eine entwickelte Software führt den Algorithmus aus. An den PC selbst werden nur die Rms-Daten geliefert, die Software berechnet daraus Versorgungsspannung, Wirk- und Blindleistung jeder Phase.

3.2 Ansatz von Albicki und Cole

1998 verbesserten Cole und Albicki am MIT den Algorithmus von Hart durch das Hinzufügen von zwei Features, Edges und Slopes die zusätzlich zu Steady-State-Signaturen erkannt wurden. Dadurch wurden Geräte mit sehr ähnlichen Lastgängen besser erkannt.

3.3 Ansatz von Baranski und Voss

2003 entwickelten Baranski und Voss einen sehr günstigen optischen Sensor zur Messung der Wirkleistung.

3.4 Ansatz von S. Leep und K. Norford

Ebenfalls 2003 verbesserte Norford und Leep den Hartschen Algorithmus durch Miteinbeziehung zusätzlicher Stromoberwellen. Damit wurde das Differenzieren elektrischer Geräte deutlich besser, ihr Augenmerk galt damals der Industrie in der es sehr häufig zu Schaltereignissen kommt. Norford verwendete eine Abtastrate von teils mehr als 8 kHz und benutzte außerdem eine Spektralanalyse. Damit konnte er die spektrale Hüllkurve berechnen wodurch Amplituden der einzelnen harmonischen Wellen aufgedeckt werden konnten.

3.5 Ansatz von Srinivasan

2006 benutzte Srinivasan Harmonische Oberwellen um sowohl transiente als auch Steady-State-Signaturen zu entdecken. Für jegliche Kombination

3.6 SPEED

Das meist verbreitetste NIALM-Produkt ist SPEED, entwickelt von der Firma Enetics aus den USA. 1996 kam das System auf den Markt, es ist sehr einfach zu installieren (unter 15 Minuten), ist flächendeckend aber ein viel zu teures Unterfangen.

3.7 2011: Fraunhofer Institut

Am Fraunhofer Institut wird in den letzten Jahren sehr viel mit NIALM gearbeitet, zwei Ansätze haben sich seit 2011 herauskristallisiert an denen das Institut bis heute forscht und testet.

1. Das erste Forschungsgebiet beschäftigt sich damit, möglichst billig trotzdem eine sehr hohe Akkuratheit zu erhalten. Hier arbeitet das Institut mit einem sehr billigen, einfach zu installierendem Sensor der die komplette elektrische Leistung in einem stationärem Gerät mit einer Abtastrate von 1 Hz misst.

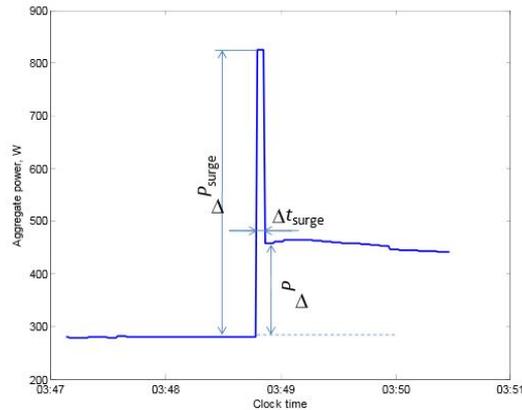


Abb. 3. statistische Features des Fraunhofer-NALM

Ein möglicher Sensor der verwendet werden kann ist der Energy Detective (TED), verbunden mit einem Google Power Meter. Die Genauigkeit dieser Methode liegt allerdings nur bei ca. 80%, hauptsächlich verursacht durch Überlappungen verschiedener Geräte im Energiefluss. Dieser Algorithmus hier soll dieses Problem lösen können, 3 Neuerungen sorgen dafür. Erstens benutzt man neben den Änderungen im Energiefluss auch eine statistische Verteilung von Stromstößen mit Dauer des Stoßes und Amplitude. Das führt zu einer besseren Segmentierung. Zweitens werden weitere Statistiken festgehalten wie Nutzungszeit, damit sind Geräte auseinander zu halten. Als letztes bezieht der Algorithmus bei Zustandswechseln von Geräte diese vorher erwähnten statistischen Daten mit ein bei vermeintlichen Treffern von Signatur und passendem Gerät. Abbildung 3 zeigt die erwähnten energierelevanten Features. ΔP_{surge} zeigt den Stromstoß, ΔP denn Steady-State-Wechsel und Δt_{surge} die Zeitdauer des Stromstoßes.

Ein Cluster wird durch diese statistischen Hilfen verteilt sollte es mehr wie eine Applikation enthalten, verstreute Cluster zusammengefügt die eindeutig derselben Applikation zugeschrieben werden können. Die Analyse die dies entscheidet wird laufend auf den neusten Stand gebracht durch die vorher verwendeten Daten. Der Hauptalgorithmus der denn wahrscheinlichen Pfad der Zustände der Geräte vorgibt wird durch einen Viterbi-Algorithmus realisiert. Abbildung 4 zeigt nochmal denn gesamten Algorithmus.

Die Zielsetzung des Algorithmus war also zusammengefasst eine möglichst billige Hardware mit hoher Genauigkeit zu verbinden, hauptsächlich durch das Unterscheiden von Geräten mit selben Zustandswechseln. Tests haben gezeigt das der Algorithmus dafür sehr gut funktioniert.

2. Dem zweiten Algorithmus liegt eine sehr hoher Abtastrate zugrunde, 500 kHz. Diese hohe Rate erfordert ein hohes Speichervolumen, pro Sekunde

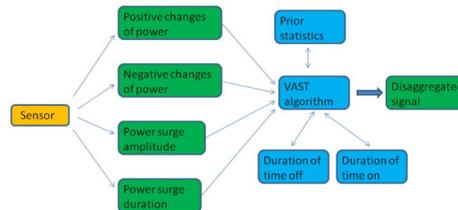


Abb. 4. Blockscheema des NIALM

kommt es zu mehreren MB an Daten. Der Algorithmus basiert auf dem Erkennen von Transienten Signaturen in der Wellenform.

Der erste Algorithmus weist eine Genauigkeit von 97% auf, der zweite 100%, wenn auch mit begrenztem Testbereich bisher.

4 Aktuelle Forschung

4.1 Fraunhofer Institut

Die beiden Algorithmen von Fraunhofer sind noch nicht abgeschlossen, der Augenmerk auf dem ersten Algorithmus liegt in Zukunft darauf die NIALM-Methode in vielen Haushalten zu installieren. Dies fördert auch schneller das Erfassen der statistischen Daten. Ebenfalls ein Fokus liegt in der Kombination der beiden Algorithmen, damit kann man Robustheit des Algorithmus garantieren.

Weiterhin arbeitet man an der Erkennung variabler und stetiger Geräte, mögliche schon erforschte Methoden können dem Algorithmus recht einfach zugeführt werden.

4.2 LoReMa Projekt

Das Ziel des Lorema Projekt ist es den Energieverbrauch eines Haushaltes in Echtzeit zu erfassen und dem Verbraucher zu visualisieren. Die Hardwarekomponente hierbei wurde miniaturisiert, der LoReMa Messknoten kann in eine Steckdose eingebaut werden. Dieser intelligente Messknoten soll neben dem reinen Messen auch automatische Funktionen anbieten wie das automatische Umschalten ganzer Lastgruppen wie Zimmerbeleuchtung. Die einzelnen Messknoten in den Steckdosen sollen über WLAN miteinander kommunizieren, dabei wird der Kommunikationsstandard ZigBee verwendet. Übertragen werden sollen detaillierte Energiemesswerte, Wirkleistung, Blindleistung, Stromgrundwellen und die ersten 11 Stromoberwellen. Diese 11 sind für das Erkennen sämtlicher Haushaltsgeräte laut Studien ausreichend. Die Daten gehen über einen Koordinator an die LoReMa-Zentrale. In der zentrale selbst findet die Lasterkennung statt

mittels des NIALMS-Algorithmus. Der Unterschied zu den meisten anderen Ansätzen wie das Fraunhofer Institut zum Beispiel ist das dezentrale Messen an mehreren Steckdosen. Aufgrund des Low-Power und Low-Cost Ansatzes ist eine hohe Abtastrate ebenfalls nicht möglich. 2011 wurde der 1. LoReMa Prototyp in einem Labor getestet, er erfreute sich großer Beliebtheit bei Besuchern. neben einer Visualisierung auf dem Monitor wurde es auch auf Smartphones angezeigt, siehe Abbildung 5.



Abb. 5. Visualisierung von Energieverbräuchen auf einem Smartphone

4.3 Smart Grid

In Zukunft eine bedeutende Rolle wird der NIALM zukünftig in Smart Grid einnehmen. Realisiert wird dies durch eine Kombination aus dem NIALM-Algorithmus und einem Webinterface für die User. Smart Meters werden vor allem in Europa auch von der Politik wahrgenommen, laut Europäische Union sollen bis 2020 80% aller Haushalte Smart Meters aufweisen. Die European Alliance for Innovation arbeitet deshalb an einem Konzept für Smart Grid in Verbindung mit der NIALM-Technologie. Die Anforderungen an Smart Meter sind Bezahlbarkeit, das Erkennen und Verhindern von Stromausfällen, das Verhindern von Energieverschwendung und das Bilden einer Datengrundlage für Smart Grid. Was sind Smart Meter?

Digitaler Service der Energienutzdaten sammelt und sie weiterleitet, sowie andere Nutzdaten erhält. Smart Meter zeichnen sich aus durch **Intelligenz**, da das Gerät entscheidet welche Daten nützlich und wichtig sind momentan, außerdem sind sie **Effizient** in der Hinsicht das Privatsphäre und Sicherheitskonzepte berücksichtigen.

Abbildung 5 zeigt einen Überblick über ein System, es besteht wie zu sehen ist aus 3 Subsystemen:

1. Local Monitoring Subsystem: Besteht aus einem Sensor in den Haushalten und misst dort den gesamte Energieverbrauch. Ebenfalls benötigt wird hier

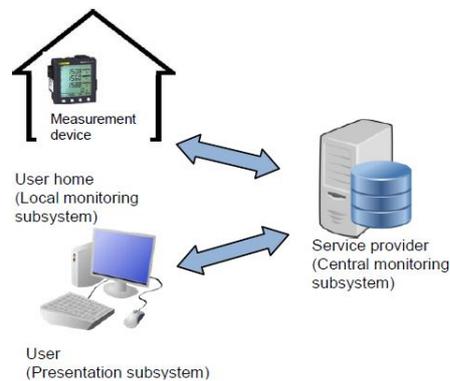


Abb. 6. Aufbau eines Smart Grid Systems

ein Netzwerksystem das mit dem Zentralrechner kommuniziert und die Daten überträgt.

2. Central Monitoring Subsystem: Hier ist der Kern des Systems, sämtliche Prozesse laufen zentral auf diesem Subsystem. Somit sind auch sämtliche relevanten Softwarekomponenten wie eben der NIALM-Algorithmus hier installiert.
3. User Präsentation Dies ist eine webbasierte Anwendung für denn Endkunden, hier werden denn Verbrauchern graphische Datenpräsentationen vorge-tragen.

Der hier verwendete NIALM-Algorithmus wird in MATLAB geschrieben, er verwendet ausschließlich Wirk- / und Blindleistung.

Das SEEE-Projekt von der Universität Catania in Italien beschäftigt sich mit der stärkeren Einbeziehung der User in das Smart Grid Projekt. Die Idee dabei ist soviel wie möglich den Verbraucher machen zu lassen, die Schnittstelle zu dem System ist eine Website auf der Verbraucher elektrische Charakteristiken zu ihren Geräten hinterlegen können. Für Fehlererkennung, die entweder durch Verbraucher oder NIALM Algorithmus entstehen, kann der Verbraucher auf der Website zwei Arten von Feedback geben.

- **Check Status:** Verbraucher können hier den Status ihrer Geräte vergleichen, dabei wird dem Verbraucher nur vorgelegt ob ein Gerät an oder aus sein sollte. Der NIALM Algorithmus ist in der Lage bei Feedback Fehler zu korrigieren.
- **Verify signature:** Hier wird der Verbraucher aufgefordert ein bestimmtes Gerät manuell an und wieder auszuschalten um die Signatur zu bestätigen.

Die User Präsentation folgt einem Model-View-Controller Paradigma. Der Controller wird mittels Java Servlet implementiert, die Views und Models werden mit HTML-Seiten implementiert.

Wichtig für dieses Projekt ist ein Verbraucherfreundliche Webinterface, ohne Verbraucher fehlen wichtige Bezugspunkte da der Algorithmus eben nur Wirk- und Blindleistung verwendet. Für Verbraucher die sich das bisschen Zeit nehmen wollen um Strom zhu sparen ist diese Methode aber eine sehr gute Zukunftsvision.

5 Fazit

Non-intrusive Load Monitoring ist eine Methode die in viele verschiedene Richtungen entwickelt wurde aber immer im Konzept eines größeren Ganzen stand. Der Durchbruch das jeder Haushalt über einen NIALM-Algorithmus seine Energieverbräuche aufgelistet bekommt kam so nie zustande. NIALM wurde stets in Verbindung mit anderen Sachen benutzt und weiterentwickelt. Von dem Stand der Technik gibt es heute sehr gute Algorithmen die sämtliche Gerätetypen eines Haushaltes analysieren können, die Vermarktung für eine breite Maße bleibt aber nach wie vor aus. Verändert werden könnte das in naher Zukunft durch denn Smart Meter, insgesamt schärft sich das Bewusstsein für Energie und Strom einsparung immer weiter und es sollte nicht mehr lange dauern das in denn Industrieländern dieser Erde der NIALM Einzug in viele Haushalte führt. Für massentaugliche Systeme muss aber endlich eine Sammlung von Standarts eingeführt werden. Sollte es soweit sein muss aber auch der Punkt der Privatsphäre und der Sicherheit näher erleuchtet werden, gerade beim Smart Grid über das Interface kann der Verbraucher unbewusst sehr viele Daten über sich preisgeben durch seine Eingabe wann er welche Geräte anschaltet. Ein ebenfalls sehr interessantes Projekt ist das LoReMa-Projekt aus Zürich das ein sehr kleines und massentaugliches System zur Visualisierung von Energieverbräuchen anbietet.

Literatur

1. Non-intrusive-Appliance-Load-Monitoring.pdf
2. Michael Baranski: Energie-Monitoring im privaten Haushalt
3. Pascal Walther, Stephan Tomek, Alexander Klapproth: Load Recognizing Meter and Actuator , www.bfe.admin.ch
4. <http://georgehart.com/research/Hart1985.pdf>
5. <http://www.vtt.fi/inf/pdf/publications/1998/P356.pdf>
6. Zeifman and K. Roth. Nonintrusive appliance load monitoring: Review and outlook.
7. K. Norford and S. B. Leeb. Non-intrusive electrical load monitoring in commercial buildings based on steady-state and transient load-detection algorithms.

8. Leeb S.B., Shaw S.R., Kirtley J.L. 1995. Transient event detection in spectral envelope estimates for non-intrusive load monitoring
9. "Nonintrusive appliance load monitoring (NIALM) for energy control in residential buildings," Energy Efficiency in Domestic Appliances and Lighting Conference, Copenhagen, 2011
10. Web interactive non intrusive load disaggregation system for active demand in smart grids

Betrachtung des Einsatzes von Ontologien in der Aktivitätserkennung

Inna Belyantseva*

Betreuer: Anja Bachmann†

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*uncxu@student.kit.edu

†bachmann@teco.edu

Zusammenfassung. Die Seminararbeit beschäftigt sich mit dem Thema Aktivitätserkennung und dem Einsatz von Ontologien zur Unterstützung dieser. Es werden unterschiedliche Einsatzgebiete der Aktivitätserkennung vorgestellt und aufgezeigt, welche Vorteile sie für die Menschheit bringt. Der Vorgang beim Erkennen einer Aktivität wird erklärt und es wird auf Zusammenhänge, die auf Aktivitäten hinweisen, eingegangen. Im Zuge dessen wird ebenfalls die benötigte Sensorik vorgestellt. Anschließend werden die unterschiedlichen Ansätze zum Verfahren der Aktivitätserkennung erläutert. Der mögliche Einsatz von Ontologien in der Aktivitätserkennung wird besprochen und die Vorzüge herausgearbeitet. An einem Beispiel wird erläutert, wie der Vorgang der Aktivitätserkennung in einem auf einer Ontologie basierenden System funktioniert. Bereits vorhandene Arbeiten zu diesem Thema werden anschließend verglichen. Zum Abschluss wird ein Vorschlag für ein aktivitätserkennendes System geliefert, das Ontologien mit maschinellem Lernen kombiniert, um die Vorteile beider Ansätze ausnutzen zu können.

Schlüsselwörter: Aktivitätserkennung, Ontologien, ADL

1 Einleitung

In der heutigen Zeit nimmt die Technologie einen wichtigen Platz im Leben des Menschen ein. In einer Gesellschaft, in der immer mehr Menschen durch ihr Alter auf Hilfe angewiesen sind [6], wäre es von Vorteil, wenn die Technologie nicht nur der Unterhaltung dienen würde, sondern den Menschen unterstützen könnte [2].

Zukunftsversionen präsentieren oft eine Technik, die sich automatisch dem Nutzer anpasst, immer die relevanten Informationen auf ein Display projiziert, den gewünschten Gegenstand schon parat hat, noch bevor danach gegriffen werden kann, und viele Aufgaben selbständig erfüllt. Derzeit jedoch muss sich der Mensch der Technik anpassen, er muss Tastenbefehle eingeben, mit der Maus navigieren und auf Knöpfe drücken, um die Hilfe, die er braucht, zu erhalten [9].

Daher stellt sich die Frage, was noch fehlt, um die Vision einer intelligenten Maschine in Realität umsetzen zu können.

Um eigenständig auf die Wünsche des Menschen reagieren zu können, muss die Technik erst die Absichten des Menschen verstehen. Dabei hilft Aktivitätserkennung, denn wenn die Absichten des Nutzers bekannt sind, lässt sich die Situation besser erfassen und die Maschine kann flexibel und angemessen auf die Bedürfnisse des Menschen eingehen [2,9]. Deshalb beschäftigt sich die Forschung mit dem Thema Aktivitätserkennung.

Während früher vor allem visueller Input, wie z.B. Kameras und Bilder benutzt wurden, ist man heute dank günstiger Sensoren in der Lage, viele unterschiedliche Informationen zu erlangen [2].

Auch in der Verarbeitung der Daten zeichnet sich inzwischen ein Wandel ab. Die früher beliebten, datengetriebenen Verfahren werden zunehmend durch wissensbasierte Verfahren abgelöst [3]. Um diese Methoden zu vereinfachen bieten sich Ontologien, die eine Struktur vorgeben, die Aktivitätserkennung transparent gestalten und ein übertragbares System ermöglichen, an. Deshalb wird in dieser Seminararbeit die Rolle von Ontologien in der Aktivitätserkennung untersucht.

2 Aktivitätserkennung

2.1 Einsatzgebiete der Aktivitätserkennung

Überall, wo Mensch und Maschine aufeinandertreffen, ist eine Aktivitätserkennung im Rahmen von Context Awareness, dem Bewusstsein der Maschine über die Umgebung des Nutzers, von Vorteil. Durch den intelligenten Umgang mit Mensch und Umwelt wird aus dem Spielzeug und Werkzeug ein nützlicher Helfer, der angemessen reagieren und sich der aktuellen Situation anpassen kann [8].

Das Smart Home bietet durch seine Ausstattung den perfekten Anwendungsbereich für die Aktivitätserkennung. Hier gibt es nicht nur zahlreiche Sensoren die Daten über den Nutzer erfassen können, sondern es wird der Technologie die Möglichkeit geboten, physisch in die Welt des Menschen einzugreifen [3].

Die am häufigsten auftretenden Aktivitäten, die es zu erkennen gilt, sind Activities of Daily Living (ADL), Aktivitäten die ständig im alltäglichen Leben ausgeübt werden. Sie bieten ein breites Spektrum an unterschiedlichen Handlungen, sei es die Zubereitung von Essen und Getränken, Körperpflege oder Entspannung [2,3]. Wenn sie erkannt werden können, wird es möglich, den Menschen in seinem Alltag zu unterstützen. Es wäre denkbar, dass der Computer das Öffnen einer Tür übernimmt, den nächsten Einkauf plant oder das Wasser zum Kochen aufsetzt, sobald der Nutzer einen Tee zubereiten will [9].

In einer älter werdenden Gesellschaft sind immer mehr Menschen auf Unterstützung angewiesen, deswegen bietet es sich an Aktivitätserkennung zur Altenpflege oder Krankenpflege zu nutzen. Dadurch kann den Patienten nicht nur bei simplen ADL assistiert, sondern auch vor Gefahren gewarnt werden. Sie können z.B. medizinisch, physiologisch und psychologisch überwacht werden [2], sodass in einer Notfallsituation automatisch ein Pfleger kontaktiert wird. Zusätzlich können Übungen, wie Krankengymnastik, unter medizinischer Aufsicht

des Computers durchgeführt werden [8]. Dies würde sogar chronisch kranken Patienten, wie z.B. Alzheimer Erkrankten, die Möglichkeit der Autonomie zurückgeben und das derzeit überforderte Krankenpflegesystem entlasten.

Die Technik der Aktivitätserkennung kann auch zu Zwecken der Sicherheit genutzt werden [1], um Einbrecher frühzeitig zu registrieren und einen Alarm auszusenden oder sogar eine Abriegelung des Hauses zu veranlassen. Zusätzlich können andere Gefahren verhindert werden, wie z.B. durch das Ausschalten des Herdes, sobald der Bewohner das Haus verlassen hat.

Aber nicht nur innerhalb des Zuhauses ist die Aktivitätserkennung hilfreich. In mobilen Geräten verbaut, kann sie diesen helfen besser auf das Verhalten des Nutzers einzugehen. Durch das Wissen um das Verhalten des Nutzers und seine derzeitige Aktivität, wird es möglich Anrufe oder Nachrichten in unpassenden Situationen zu vermeiden. Zudem kann der Nutzer bei sportlichen Outdoor Aktivitäten [8] unterstützt werden oder eine, seinen Präferenzen angepasste, Wegbeschreibung erhalten. Zum Beispiel können Treppen bei Beeinträchtigung der Beweglichkeit vermieden werden.

Die intelligente Umschaltung zwischen Sprach- und Gestensteuerung kann sogar in der Arbeitswelt behilflich werden [2,9]. Denn bei vielen Aufgaben hat der Nutzer keine Hand frei, um Befehle manuell eingeben zu können und wenn diese Aktivitäten erkannt werden würden, könnte der Computer auf Sprachsteuerung umschalten und eine zur Arbeit simultane Befehlseingabe ermöglichen. Somit würde die Produktivität eines Angestellten gesteigert.

2.2 Ablauf während der Aktivitätserkennung

Aktivitätserkennung folgt zumeist einem ähnlichen Aufbau, wie in Abbildung 1 dargestellt ist. Eine Aktivitätserkennung kann entweder in bestimmten vorher festgelegten Zeitintervallen erfolgen [3] oder durch ein Sensorereignis ausgelöst werden.

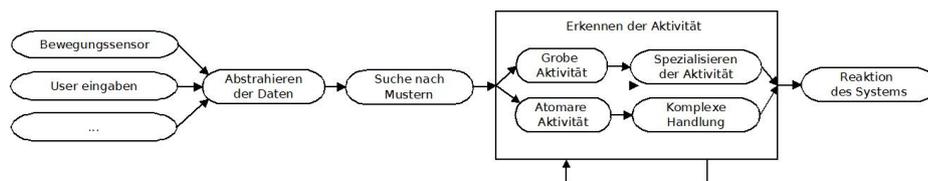


Abb. 1. Ablauf der Aktivitätserkennung

Zunächst müssen Informationen über die Umgebung gesammelt werden. Dies kann durch Sensoren, Kameras oder auch Eingaben des Nutzers geschehen. Doch diese Informationen in ihrer rohen Form sind heterogen, da sie von unterschiedlichen Quellen stammen [3] und liefern noch keine konkrete Auskunft über die ausgeführte Aktivität. Deshalb müssen sie zuerst abstrahiert und verarbeitet

werden [2,9], damit sie vom Computer interpretiert werden können. Die so erhaltenen Daten können mit einer Datenbank abgeglichen und auf Muster untersucht werden [9]. Wenn eine Übereinstimmung gefunden erfolgt, wird die entsprechende Aktivität erkannt [2,9].

Zumeist wird bei der Aktivitätserkennung mit atomaren Aktivitäten begonnen, grundlegenden Gesten, die nicht weiter zerlegt werden können, wie z.B. das Benutzen eines Messers. Deutet eine Abfolge von atomaren Aktivitäten auf eine komplexere Handlung, wie das Zubereiten einer Mahlzeit, hin, so kann diese aus Kombinationen der atomaren Aktivitäten erschlossen werden [9]. Dieser Ansatz ermöglicht die Erkennung von Aktivitäten auf unterschiedlichen Granularitätsstufen. Aber auch ein Ansatz von oben ist möglich, bei dem zunächst grobe Aktivitäten erkannt werden und durch passenden Input verfeinert und spezialisiert werden können [3]. Es kann z.B. zunächst erkannt werden, dass es sich um eine Aktivität der Entspannung handelt, wenn der Nutzer das Sofa benutzt. Falls noch registriert wird, ob er ein Buch zur Hand hat oder dass der Fernseher eingeschaltet wird, kann auf Lesen oder Fernsehen spezialisiert werden.

Sobald die Aktivität erkannt worden ist, kann eine assistierende Aktion vom Computer in Gang gesetzt oder zumindest eine Rückmeldung gegeben werden. Es können z.B. die so erschlossenen Informationen auf einem Display für Pflegepersonal präsentiert werden.

2.3 Kontext der Aktivitätserkennung und verfügbare Sensorik

Eine Aktivität kann nicht direkt erkannt werden, sondern muss aus dem Kontext erschlossen werden [8,3,6]. Es gibt verschiedene Zusammenhänge, die auf eine Aktivität hinweisen können, wie in Abbildung 2 dargestellt. Um die Veränderungen dieser erkennen zu können, sind Sensoren essenziell.

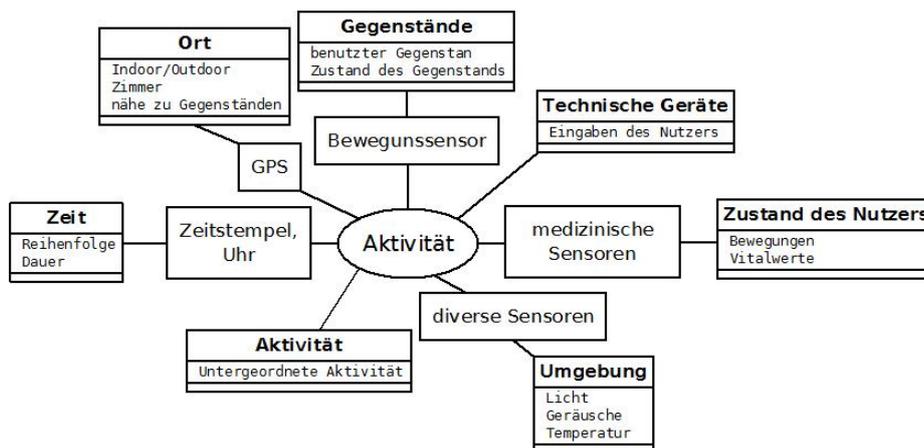


Abb. 2. Kontext einer Aktivität

Deshalb haben die Verbesserungen im Bereich der Sensorik auch das Forschungsgebiet der Aktivitätserkennung vorangetrieben [3]. Derzeit sind Sensoren günstig, energiesparend und klein [2]. Dadurch wird es möglich sehr viele Sensoren in die Umgebung einzubetten und so möglichst viele Informationen zu extrahieren.

Früher wurden zur Beobachtung der Umwelt vor allem Bild und Videoaufnahmen verwendet. Dadurch konnte man bereits Menschen erkennen und Bewegungen verfolgen, allerdings bringen Bild und Videoaufnahmen einige Nachteile mit sich. Denn visuelle Daten sind groß, schwer zu verarbeiten und es ist aufwändig, sie für andere Aufgaben zu skalieren [2]. Da Sensoren sich zumeist nur auf ihren eigenen kleinen Bereich fokussieren, lassen sich ihre Werte einfacher auswerten.

Die Zeit kann über die Absichten des Nutzers Auskunft geben [2,9], deshalb benötigen die Sensordaten einen Zeitstempel. Früh am Morgen folgen die meisten Menschen dem gleichen Ritual, während sie nachts zumeist am Schlafen sind. Aber auch die Dauer einer Beschäftigung hilft diese näher zu spezialisieren [3], wenn z.B. die Dusche nur kurz läuft, könnte dies auf die Reinigung dieser hinweisen, bei längerem Gebrauch deutet es darauf hin, dass geduscht wird. Die Reihenfolge bietet ebenfalls Anhaltspunkte [3]. Wenn zuerst ein Buch aufgehoben wird und danach das Sofa benutzt wird, deutet es auf die Aktivität des Lesens hin, bei umgekehrter Reihenfolge dagegen könnte es auf Aufräumen hinweisen.

Weitere Auskunft gibt der Ort an dem die Handlungen vollzogen werden [6,3], der mit Hilfe von GPS Sensoren ermittelt werden kann. Zum einen ist die Unterscheidung zwischen In- und Outdoor Aktivitäten gegeben. Des weiteren lassen sich innerhalb des Hauses noch Schlussfolgerungen aus dem Aufenthalt in einem bestimmten Zimmer treffen. Denn die meisten Räume erfüllen spezielle Aufgaben. Während die Küche zum Kochen benutzt wird, wendet der Mensch sich im Badezimmer meist der Körperpflege zu, schläft im Schlafzimmer und entspannt sich im Wohnzimmer. Zudem kann auch die Nähe zu bestimmten Geräten, die auf die Benutzung dieser hinweisen könnte, betrachtet werden.

Es ist auch möglich direkt die Gegenstände zu betrachten, die benutzt werden, da sie oft nur für bestimmte Handlungen gebraucht werden [2,3]. Ein Messer wird zum Schneiden benutzt, ein Kamm zur Körperpflege. Auch hier können Sensoren helfen [2,3]. Beschleunigungssensoren können die Bewegung eines Gegenstandes erfassen, die auf die Nutzung hinweisen [3]. Durch Drucksensoren ließe sich z.B. die Menge des Getränks in einer Tasse oder mit einem Temperatursensor seine Wärme ermitteln. Sollte es sich dabei um technische Geräte handeln, können weitere Daten z.B. aus den Eingaben des Nutzers extrahiert werden.

Nicht zu vernachlässigen ist der Mensch selbst. Hier kann man nicht nur durch das Erkennen des derzeitig beobachteten Nutzers Schlüsse durch seine Gewohnheiten ziehen [3], sondern auch seine Bewegungen und Körperwerte überwachen, um daraus weiter Informationen zu beziehen [8,6]. Besonders bei der Altenpflege sind die Vitalwerte sehr wichtig, da sie Auskunft über das Wohlbefinden des Pa-

tienten geben. Dazu können Sensoren verwendet werden, die direkt am Körper getragen werden (Wearable Sensors) [2,3]. Während Beschleunigungssensoren über Bewegungen Aufschluss geben, können medizinische Sensoren Blutdruck, Herzschlag, Temperatur und andere Vitalwerte überwachen [6].

Weitere Sensoren können direkt im Raum angebracht werden und liefern somit Informationen über die Umwelt des Nutzers [3]. Ein Lichtsensor erkennt die Lichtsituation, die zwischen Schlafen oder Ausruhen entscheiden kann, Audio-Sensoren geben Hinweise über die Geräuschkulisse, Thermosensoren messen die Temperatur im Raum.

Aber auch die Aktivitäten selbst liefern Hinweise auf andere komplexere Handlungen [3]. In der heutigen Zeit müssen nicht einmal unbedingt neue Sensoren angebracht werden, denn viele technische Geräte beinhalten bereits eigene Sensorik, die zur Aktivitätserkennung genutzt werden kann. Ein Smartphone ist zum Beispiel mit vielen Sensoren, wie Beschleunigungssensor, GPS und Kamera, ausgestattet, ebenso wie ein modernes Auto, welches zumeist über GPS verfügt. All diese Informationen helfen die Aktivität des Menschen besser nachvollziehen zu können, da oft ein Zusammenspiel unterschiedlicher Faktoren in die Beweggründe des Nutzers einfließt.

Allerdings ist beim Umgang mit Sensoren auch Vorsicht geboten, denn viele Sensoren produzieren große Datenmengen, die in unterschiedlichen Formaten auftreten [3]. Zudem sind Sensoren leider fehleranfällig und es kann zu Fehlauslösungen, verfälschten Daten oder Inkonsistenzen zwischen unterschiedlichen Quellen kommen [9,3]. Nicht zu vergessen ist ebenfalls, dass Sensoren viele Informationen über den Menschen aufzeichnen und somit in seine Privatsphäre eingreifen.

2.4 Datengetriebener und wissensbasierter Ansatz

Bei der Erkennung von Aktivitäten gibt es zwei grundlegende Ansätze, die in der Umsetzung variieren können.

Datengetriebener Ansatz: Bei diesem Ansatz werden maschinelle Lernverfahren eingesetzt, um durch die von Sensoren gesammelten Daten ein Muster der ausgeführten Aktivität zu erschließen [3]. Dieser Ansatz kann noch in zwei Unterkategorien unterteilt werden: überwachtes und unüberwachtes maschinelles Lernen.

Beim überwachten maschinellen Lernen wird das System zunächst in kontrollierter Umgebung trainiert. Das heißt, es werden vorgegebene Aktivitäten von Probanden ausgeführt und dem Computer gezeigt, so dass dieser die Verbindung zu den Sensorwerten herstellen kann. Alternativ dazu kann die Aktivität auch nachträglich im entsprechenden Zeitabschnitt eingetragen werden. Dabei werden die Inhalte mit Labels versehen. Zur Entscheidungsfindung wird hier meist mit probabilistischen Methoden, wie Hidden Markov Model oder Bayesian Network, gearbeitet, um Fehlmessungen der Sensoren und den Variationen im Verhalten der Nutzer entgegenzuwirken [2]. Einerseits wird es dadurch möglich, das System auf einen bestimmten Nutzer zu trainieren, allerdings wird es schwieriger

das System auf andere Nutzer umzustellen, da diese unterschiedliche Sensorwerte produzieren könnten. Zudem lassen sich die Werte kaum skalieren oder auf ein anderes Einsatzgebiet übertragen. Zusätzlich nimmt das Trainieren des überwachten maschinellen Lernsystems sehr viel Zeit in Anspruch, was sich mit wachsender Komplexität steigert [3].

Alternativ dazu gibt es das unüberwachte maschinelle Lernen. Diese Methode ist nicht auf ein zeitintensives aktives Training im Voraus angewiesen, denn es wird auf das Labeling verzichtet. Stattdessen wird hier versucht, in der Datenmenge Muster zu erkennen und diese, durch Bildung von Clustern, Aktivitäten zuzuordnen [3]. Hier werden zur Entscheidung ebenfalls meistens probabilistische Methoden verwendet. Obwohl dieses Verfahren anfangs schneller erscheinen mag, da es auf eine aktive Trainingsphase verzichtet, dauert es auch hier, bis genug Daten gesammelt wurden, um tatsächlich die korrekte Aktivität zu erkennen [3]. Da der Computer selbständig und ohne Überwachung lernt, kann es zu Fehlern kommen, wenn Cluster falsch interpretiert werden. Diese Fehler wirken sich auch langfristig auf die Funktionalität des Systems aus. Allerdings kann unüberwachtes maschinelles Lernen auch während des Betriebs weitere Aktivitäten erlernen und sich unterschiedlichen Situationen anpassen [3].

Bei beiden datengetriebenen Verfahren besteht ein weiteres Problem in der großen Datenmenge. Um diese zu verarbeiten, werden ein leistungsstarker Computer und schnelle Algorithmen benötigt.

Wissensbasierter Ansatz: Es bietet sich auch an das Wissen über den Menschen und seine Aktivitäten zu untersuchen und daraus Axiome abzuleiten. Dieser wissensbasierte Ansatz nutzt alltägliches Wissen, um Regeln zu erstellen, die für einen Computer verständlich sind [2,3]. Zum Beispiel kann festgelegt werden, dass zur Aktivität "Tee zubereiten" eine Tasse, heißes Wasser und Teebeutel benutzt werden müssen. Da diese Prämissen dem menschlichen Grundwissen entsprechen, können sie problemlos aufgestellt werden. Zu einer Aktivität können Voraussetzungen, die auf Sensorwerten beruhen, definiert werden, so dass die Aktivität erst erkannt wird, wenn diese Voraussetzungen erfüllt sind. Allerdings ist zu bedenken, dass sich Menschen in ihrem Verhalten stark unterscheiden können. Nicht jeder Nutzer verwendet Teebeutel zum Aufbrühen von Tee und auch die Reihenfolge der Verwendungen der benötigten Objekte kann variieren. Deshalb sollten die Regeln nicht zu streng formuliert sein und Platz für persönliche Freiheiten lassen. Der Vorteil einer so erstellten Wissensdatenbank besteht darin, dass sie wiederverwendbar ist.

Für den wissensbasierten Ansatz bietet sich eine Entscheidungsfindung durch Reasoner-Systeme an. Diese sind oft frei verfügbar [9] und können durch Logik, basierend auf den aufgestellten Voraussetzungen, entscheiden welche Aktivität gerade vorliegt [2]. Ein solches Verfahren lässt sich leicht nachvollziehen und ermöglicht eine verständliche Präsentation des Lösungsweges. Dadurch, dass es auf Logik basiert, ist es besonders gut für Computer geeignet und auf Grund der kontrollierbaren Datenmenge auch sehr elegant. Allerdings kann es problematisch werden wenn es zu Ungenauigkeiten in der Messung kommt oder wenn sich die

Profile zweier Aktivitäten zu stark überschneiden. Wenn das Reasoner-System keine klaren Daten bekommt oder die Daten die Wissensbasis überschreiten, kann es keine Entscheidung treffen [9]. Zudem ist es nicht darauf ausgelegt, selbstständig weiter zu lernen oder sich auf einen bestimmten Nutzer anzupassen.

Natürlich können die unterschiedlichen Ansätze auch miteinander kombiniert werden um ihre Vorteile ausnutzen zu können [8], wie in Kapitel 4 gezeigt wird.

3 Ontologien in der Aktivitätserkennung

3.1 Rolle von Ontologien im Bezug auf Aktivitätserkennung

Nun stellt sich die Frage, wie Ontologien bei der Aktivitätserkennung unterstützend eingesetzt werden können.

Ontologien sind eine Art der Darstellung von Zusammenhängen zwischen Sachverhalten. Es wird ein Graph erstellt, in dem alle Verbindungen dargestellt sind [2]. Durch Ontologien wird es möglich, nicht nur die Existenz einer Relation darzustellen, sondern auch aufzuzeigen, welcher Art diese ist [2], da die Relationen in der Ontologie über eigene Attribute verfügen.

Besonders beim wissensgetriebenen Ansatz lässt sich diese Eigenschaft gut nutzen. Denn das dem Alltag entnommene Wissen muss entsprechend gespeichert werden. So wird eine Vorlage geliefert, die die Gegebenheiten entsprechend darstellen kann. Dadurch können Kernkonzepte und ihre Anforderungen formuliert werden [3] und die Zusammenhänge zwischen Zeit, Ort, Objekt und Person ausgedrückt werden.

Zudem folgen Ontologien bestimmten Konventionen und es existiert sogar eine eigene Sprache, die *Ontology Web Language (OWL)*, in der Ontologien verfasst werden können [3]. Ontologien sind logisch aufgebaut, da sie als Graph formuliert sind, und eignen sich somit gut für den Computer [2,3]. Für OWL existieren bereits fertige und frei verfügbare Reasoner und andere Software zur Verwaltung, so dass diese nicht erst neu programmiert und getestet werden müssen [3].

Zusätzlich machen Ontologien es auch einfacher, das gesammelte Wissen transparent zu halten und austauschen zu können [2,6]. Somit können Forschungsgruppen sich eine Datenbank teilen und an ihren eigenen Teilgebieten weiterentwickeln, ohne das Gesamtgefüge zu zerstören [2]. Auch zwischen den Systemkomponenten, wie Sensoren, Datenbank und Reasoner, wird die Kommunikation beträchtlich erleichtert, wenn alle die gleiche Sprache verwenden [9]. Durch die festgelegte Struktur lassen sich Redundanzen leichter vermeiden, was zum einfacheren Debuggen führt [9]. Zudem ermöglichen es mehrgliedrige Zusammenhangs-Ketten unterschiedliche Granualitäten auszudrücken [2]. Sie sind hierarchisch aufgebaut, wobei jede Klasse ihre eigenen Eigenschaften besitzt, die an die Unterklassen vererbt werden [2,3]. Wenn die Ontologie wie ein Baum angesehen wird, gibt es für die Blätter mehr Voraussetzungen, während die Beschreibungen nahe der Wurzel recht einfach zu erfüllen sind.

Durch die offene Gestaltung müssen die Sachverhalte auch nicht bis ins letzte Detail beschrieben werden, sondern können locker formuliert sein, damit es

noch für eine zufriedenstellende Lösung ausreicht [9], sie allerdings nicht durch Diskrepanzen im Verhalten des Nutzers ruiniert werden. Später können sie problemlos erweitert und verfeinert werden [3]. Ontologien eignen sich somit sehr gut für die Initialisierung eines Systems [6].

Ontologien haben auch den Vorteil, dass sie ohne große Transformation dem Nutzer anschaulich präsentieren können, welchen Lösungsweg der Computer genommen hat. Dies ermöglicht eine gute Kommunikation mit dem Nutzer [6].

Durch die Realisierbarkeit eines Profils für einen Nutzer, kann das System sich einerseits an die Gewohnheiten eines bestimmten Nutzers anpassen, während er für andere Nutzer diese zusätzlichen Einschränkungen ignoriert.

Der Nachteil von Ontologien ist allerdings, dass sich durch die feste Struktur auch Einschränkungen bilden. Es ist zum Beispiel mit Standardsoftware nur schwer möglich Ontologien, mit einem probabilistischen Ansatz zu kombinieren, da OWL keine Bewertungen der Verbindungen vorsieht [9].

Dennoch ist es eine gute Möglichkeit, die Sachverhalte der Aktivitätserkennung darzustellen und als Ausgangspunkt für diese zu fungieren [3].

3.2 Beispiel einer Aktivitätserkennung mit Hilfe von Ontologien

Nachdem die Grundlagen besprochen wurden, wird nun ein Beispiel die Funktionsweise der, auf Ontologien basierenden, Aktivitätserkennung verdeutlichen.

Der gesamte Vorgang beginnt damit, dass Person A in die Küche geht. Ihre Bewegung wird vom Bewegungssensor aufgezeichnet und an das System geschickt. Dort wird es, mit zusätzlichen Daten aus dem GPS Sensor in einer Ontologie der Lokalität (siehe Abbildung 3) verarbeitet.

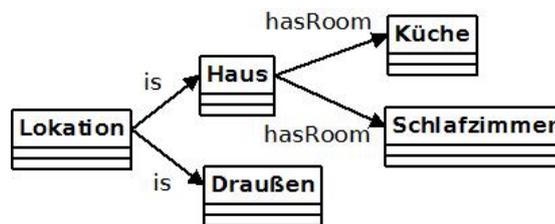


Abb. 3. Beispielhafte Ontologie der Lokalität

Nun versucht das System herauszufinden, welche Absichten der Nutzer verfolgt. Dazu läuft der Reasoner die Ontologie der Aktivität (siehe Abbildung 4) ab, und prüft welche Voraussetzungen zutreffen. Am Anfang muss sich der Reasoner zwischen einer Indoor- oder einer Outdoor-Aktivität entscheiden. Durch die Ontologie der Lokalität weiß er, dass sich der Nutzer am Ort *Haus* befindet, die Voraussetzung *hasLocation Haus* ist somit erfüllt. Folglich wird eine Indoor-Aktivität gewählt.

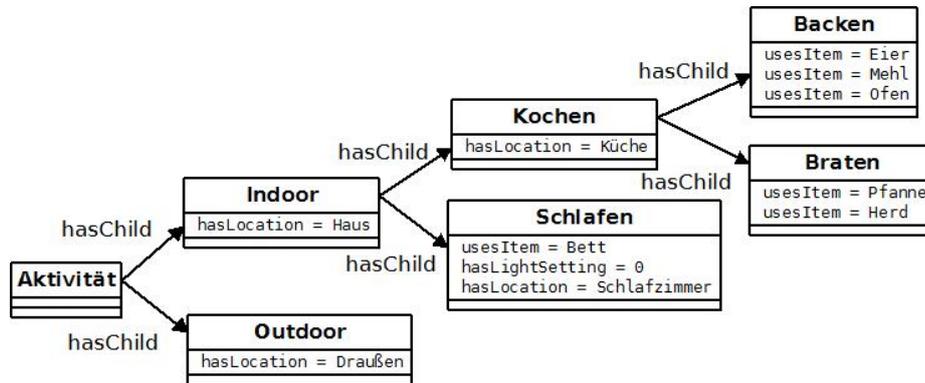


Abb. 4. Beispielhafte Ontologie der Aktivitäten

Dies kann noch weiter spezialisiert werden, da sich der Nutzer gerade in der *Küche* aufhält und die Voraussetzung *hasLocation Küche* ebenfalls zutrifft. Jetzt weiß das System, dass es sich um eine Aktivität des *Kochens* handelt. Weitere Aussagen kann es vorerst nicht treffen.

Nun öffnet Person A den Kühlschrank und entnimmt zwei Eier. Durch Sensoren im Kühlschrank, wird dem System mitgeteilt, dass *usesItem Eier* erfüllt ist. Der Reasoner prüft, ob für eine der Unterklassen von *Kochen* alle Voraussetzungen erfüllt sind, dies ist allerdings noch nicht der Fall.

Das ändert sich, als Person A Mehl aus dem Regal entnimmt, was dem System ebenfalls durch einen Bewegungssensor am Mehlbehälter mitgeteilt wird. Mittlerweile sind bereits zwei der drei Voraussetzungen der Aktivität *Backen* erfüllt.

Zu ihr gehört allerdings auch *usesItem Ofen*, das Bedienen des Ofens, das derzeit von Person A noch nicht ausgeführt wurde. Falls diese Voraussetzung, als nicht notwendig vermerkt ist, kann der Reasoner auf die Aktivität *Backen* schließen. Da das System jetzt die Absicht des Nutzers erkannt hat, kann es ihm helfen. Der Ofen wird automatisch eingeschaltet, um diesen vorheizen zu können. So wird Nutzer A die Arbeit erleichtert, da er nicht selbst daran denken muss, den Ofen einzuschalten.

3.3 Übersicht der verfügbaren Arbeiten zum Thema Aktivitätserkennung mit Ontologien

Es existieren bereits zahlreiche Arbeiten, die versuchen Ontologien in die Aktivitätserkennung einzubinden. Einige davon werden in Tabelle 1 verglichen.

Wie in der Tabelle zu sehen ist, wird zumeist ein wissensgetriebener Ansatz verwendet, obwohl auch Hybride, die wissensbasierte und datengetriebene Ansätze kombinieren, als auch ein rein Daten getriebener Ansatz vorhanden sind. Dadurch, dass beim wissensgetriebenen Ansatz eine Struktur zur Speicherung

Tabelle 1. Ausgewählte Arbeiten zum Thema Aktivitätserkennung mit Ontologien

Verfasser	Ansatz	Einsatzgebiet	Sensoren
Chen, Liming and Nugent, Chris [2]	Wissen	Assistenz, Smarthome	Objekt
Springer, Thomas and Turhan, Anni-Yasmin [9]	Wissen	Smarthome	Objekt
Latfi, Fatiha and Lefebvre, Bernard and Descheneaux, Céline [6]	Hybrid	Smarthome, Altenpflege	Körper, Technick
Chen, Liming and Nugent, Chris D and Wang, Hui [3]	Wissen	Alltagsleben, Smarthome, Altenpflege	Körper, Umgebung
Riboni, Daniele and Bettini, Claudio [8]	Daten	Krankenpflege	Körper, Umgebung
Helaoui, Rim and Riboni, Daniele and Stuckenschmidt, Heiner [5]	Hybrid	Ubiquitäres Computing	Körper, Umgebung
Ni, Hongbo and Abdulrazak, Bessam and Zhang, Daqing and Wu, Shu [7]	Hybrid	Altenpflege	Umgebung
Guzzoni, Didier and Baur, Charles and Cheyer, Adam [4]	Wissen	Assistenz	Technik, Umgebung
Akdemir, Umut and Turaga, Pavan and Chellappa, Rama [1]	Wissen	Sicherheit, Überwachung	Video
Yamada, Naoharu and Sakamoto, Kenji and Kunito, Goro and Isoda, Yoshinori and Yamazaki, Kenichi and Tanaka, Satoshi [10]	Daten	Ubiquitäres Computing	Objekt, Lokation

des Wissens benötigt wird, ist hier eine Verbindung mit Ontologien, die diese bieten, sehr häufig.

Die Einsatzgebiete variieren. Wobei sie sich meistens dennoch auf eine Assistenz im Alltag konzentrieren. Oft wird Aktivitätserkennung im Smarthome eingesetzt. Aber auch Ubiquitäres Computing, das Prinzip vom allgegenwärtigen Computer, ist beliebt. Manchmal wird das System speziell für alte oder kranke Menschen ausgelegt. Es ist allerdings auch eine Arbeit, die sich mit der Sicherheit in öffentlichen Gebäuden befasst, vorhanden.

Zumeist werden unterschiedliche Sensoren kombiniert. Sensoren, die die Umgebung erfassen sind sehr beliebt. Hierbei werden alle Sensoren verwendet, die Auskunft über die Situation, in der sich der Mensch derzeit befindet, bieten. Neben Licht, Sound und zahlreichen anderen Sensoren gehören auch an Objekten angebrachte Sensoren dazu. Teilweise wird sich hierbei nur auf die Objekt-Sensoren beschränkt, die manchmal mit der Lokation des Nutzers kombiniert werden, um eine Aktivität zu erkennen. Auch Informationen, die man aus der modernen Technik durch Eingaben des Nutzers, extrahieren kann werden gerne unterstützend eingesetzt. Manche Arbeiten nehmen neben den Umgebungssensoren auch am Körper getragene Sensoren zu Hilfe.

4 Vorstellung der Idee eines neuen Systems

Was zeichnet nun eine gute Aktivitätserkennung aus? Sie sollte ihr Einsatzgebiet gut abdecken und für möglichen viele Aktivitäten funktionieren. Dennoch sollte sie nahe der Echtzeit arbeiten, um unnötige Wartezeiten zu vermeiden. Auch wenn eine hohe Trefferquote, das häufige Erkennen der Aktivitäten, wichtig ist, sollte diesbezüglich die Präzision, die Vermeidung von Fehlerkennungen, nicht vernachlässigt werden.

Wie könnte ein System aussehen, das diesen Ansprüchen genügt? In dem nachfolgenden Text wird eine Idee für ein solches System präsentiert und in Abbildung 5 graphisch dargestellt.

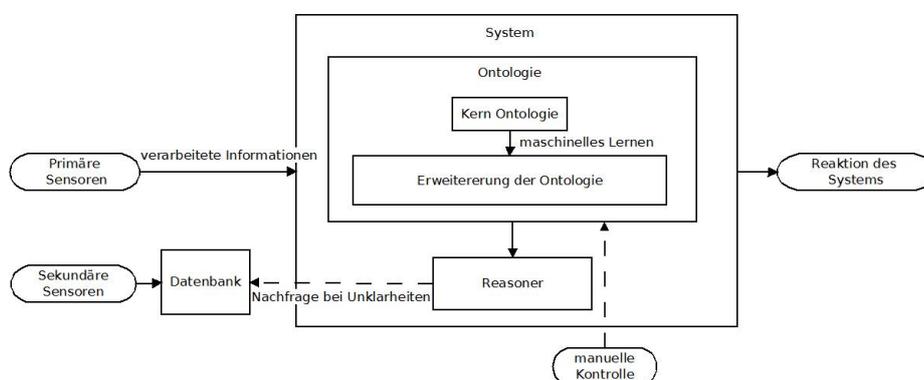


Abb. 5. Entwurf eines Systems zur Aktivitätserkennung

Natürlich ist eine große Menge von Sensoren hilfreich um möglichst viele Informationen über die Umwelt zu extrahieren, allerdings könnte die große Datenmenge die Echtzeitfunktion beeinflussen. Deshalb wäre es denkbar, sich primär auf wenige Sensoren zu fokussieren. Hierbei wären vor allem an Objekten angebrachte Sensoren vorteilhaft, da sie meist einfach auszuwerten sind. Sensoren, die am Körper getragen werden, sollten je nach Einsatzgebiet zunächst nur auf Lokalisation des Nutzers beschränkt werden. Dadurch wird die eingehende Datenmenge verringert, da weniger Sensoren vorhanden sind und die leicht zu verarbeitenden Sensoren bevorzugt werden, und kann somit schneller verarbeitet werden, wodurch die Echtzeitfunktion unterstützt wird.

Dennoch wäre es empfehlenswert möglichst viele Sensoren unterschiedlicher Art anzubringen und ihre Daten kurzfristig in einer Datenbank zu speichern, auch wenn sie nicht gleich auf dem Hauptsystem ausgewertet werden sollen, so dass diese bei Zweifeln während der Entscheidung oder um die Aktivität zu präzisieren immer noch abgerufen werden können. Dadurch wird es einfacher die Fehlerrate zu verringern, da die Entscheidungen bei knappem Ausgang durch zusätzliche Informationen klarer werden. Zusätzlich ist es dadurch möglich eine

feine Granularität aufzubauen für Aktivitäten, die diese benötigen, ohne dass es sich negativ auf Aktivitäten auswirkt, bei denen eine grobe Granularität ausreicht.

Vorteilhaft ist es auch Sensoren zu nutzen, die bereits in der Umgebung integriert sind. Besonders während der Mensch unterwegs ist, kann das Smartphone, das viele bereits täglich bei sich haben, viele wichtige Informationen liefern.

Als Kern der Aktivitätserkennung wäre ein wissensgetriebener Ansatz wünschenswert, mit einer Ontologie als Kernstruktur. Dies erleichtert die Entwicklung und macht das System übersichtlich. Zusätzlich kann am Anfang eine bereits entwickelte Ontologie als Ausgangspunkt verwendet werden und auch Reasoner sind bereits vorhanden. Dadurch kann ein Arbeitsoverhead vermieden werden und Geld und Zeit eingespart werden, was die Lösung auch für den Verbraucher günstiger macht.

Die Voraussetzungen für die oberen Ebenen der hierarchisch aufgebauten Ontologie, sollten möglichst gering gehalten werden und sich erst nach unten hin vermehren. Dadurch kann eine hohe Trefferquote erzielt werden, da selbst falls keine fein granulいた Aktivität erkannt werden kann, immer noch die übergeordnete Aktivität erkannt wird (siehe Abbildung 6).

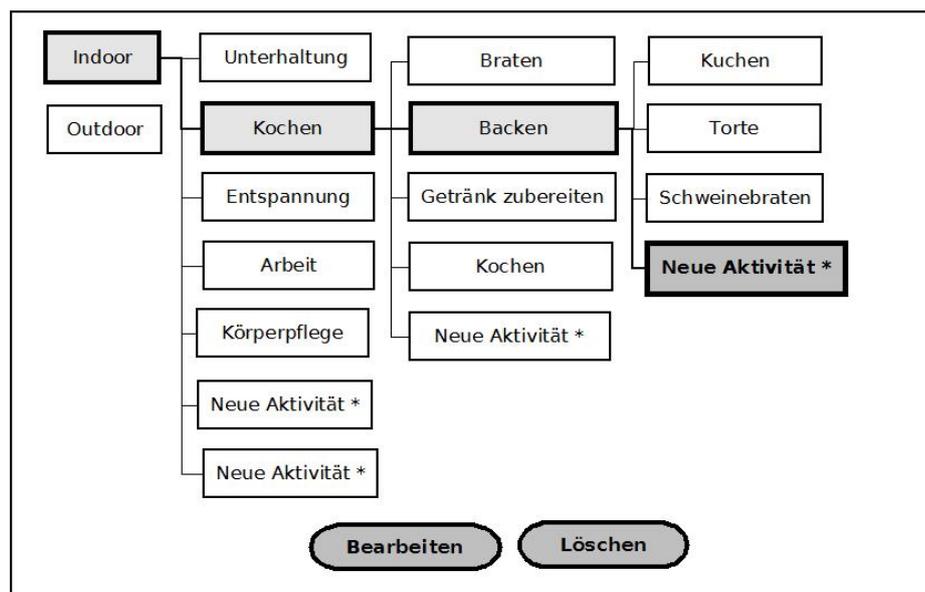


Abb. 6. Skizze eines Userinterface und grober Überblick über die Hierarchie der Ontologie

Das größte Problem von einem rein wissensgetriebenen Ansatz ist, dass dieser nicht lernfähig ist. Dies kann vermieden werden, wenn er mit einem maschinellen Lernen kombiniert wird. Es wäre denkbar den Kern als Ontologie zu

gestalten und durch Maschinelles Lernen während des Betriebes diese Ontologie strukturert zu erweitern. Beispielsweise könnte bei der Aktivität *Backen* festgestellt werden, dass regelmäßig bestimmte Zutaten verwendet werden, die bisher allerdings von den anderen, dem *Backen* entspringenden Aktivitäten, nicht beschrieben wurden. Wenn sich diese Konstellation oft genug wiederholt, kann davon ausgegangen werden, dass es sich um eigene Aktivitäten handeln. Die so erkannte Aktivität kann daraufhin in der Ontologie als Unteraktivität von *Backen* eingegliedert werden (siehe Abbildung 6, *Neue Aktivität** unter *Backen*). So bliebe die Übersichtlichkeit und die Vorteile einer Ontologie erhalten, während das System sich selbstständig weiter entwickelt und so mit der Zeit immer mehr Aktivitäten erkennen kann. Dies erhöht die Trefferquote.

Durch regelmäßige Wartungen oder ein entsprechendes Userinterface (siehe Abbildung 6) könnten auch durch Maschinelles Lernen entstandene Fehler korrigiert werden. Zusätzlich könnte das Interface dem Nutzer die Möglichkeit geben, die Ontologie selbstständig nach eigenen Wünschen zu erweitern (siehe Abbildung 7).

Bearbeiten

Name:

Übergeordnete Aktivität:

Aktion:

-
-
-

Voraussetzungen:

-
-
-
-
-

Abb. 7. Skizze für die Oberfläche zum Bearbeiten einer Aktivität

Bei der, im vorherigen Beispiel, erkannten Unteraktivität des *Backens*, könnte es sich bei den verwendeten Zutaten, die zu den Voraussetzungen der Aktivität gehören, beispielsweise um Tomatensoße, Mehl, Salz und Käse handeln. Der Nutzer kann nun, wenn er die neu erkannte *Neue Aktivität** ansieht, anhand der Zutaten erkennen, dass es sich hierbei um Pizza backen handelt. Nun kann er nicht nur die neue Aktivität mit einem für ihn verständliche Titel kennzeichnen,

sondern kann zudem noch Aktionen des Computers einstellen, die ihn bei dieser Aktivität unterstützen sollen. Somit wäre das System möglichst vollständig und es würden dennoch Fehler vermieden werden. Zudem könnte sich das System dem Nutzer sehr gut anpassen. Dieser Ansatz entspräche Semi-Überwachtem maschinellem Lernen und würde hohe Präzision und Trefferquote unterstützen.

Nicht zuletzt muss bei dem Thema Aktivitätserkennung sehr auf Sicherheit geachtet werden. Da Aktivitätserkennung nun mal die Überwachung eines Menschen erfordert, muss sehr umsichtig mit den sensiblen vertraulichen Daten umgegangen werden. Deshalb sollten diese verschlüsselt und nur autorisierten Personen zugänglich sein.

5 Fazit

Aktivitätserkennung ist heutzutage ein wichtiges Thema, das zukunftsweisend ist. Es kann in vielen Gebieten genutzt werden und ist besonders in der alternden Gesellschaft sehr vorteilhaft.

Dank der Fortschritten, die in der Sensorforschung erreicht wurden, kann sich auch die Aktivitätserkennung viel besser weiterentwickeln, da nun die Zusammenhänge, die zum Erkennen einer Aktivität benötigt werden, besser erkannt und beobachtet werden können.

Es existieren unterschiedliche Methoden und Ansätze, um Aktivitätserkennung realisieren zu können und möglichst elegant zu gestalten. Ein wissensbasierter Ansatz, der dem Alltag entnommenes Wissen nutzt, um Voraussetzungen zum Erkennen einer Aktivität festzulegen und zwei datengetriebene Verfahren, die mit maschinellem Lernen arbeiten, um, durch Mustererkennung und das Erkennen von Clustern, aus den Sensordaten auf Aktivitäten schließen zu können. Während die datengetriebenen Verfahren zwar mit ihrer autonomen Lernfunktion glänzen, allerdings einen sehr langsamen Start und einen Mangel an Übersichtlichkeit aufweisen, bietet das wissensbasierte Verfahren zwar nicht die autonome Lernfunktion, kann dafür allerdings ohne Vorlaufzeit starten und ist in seinem Aufbau übersichtlich.

Zusätzlich wurde die Funktion von Ontologien als Hilfsmittel zur Aktivitätserkennung untersucht. Diese können eine einheitliche und übersichtliche Struktur in die Verfahren bringen und somit die Arbeit an dieser Thematik erleichtern. Zudem erleichtern sie auch die Kommunikation, sowohl zwischen den Entwicklern als auch zwischen den Komponenten des Systems.

Bereits entwickelte Verfahren wurden gegenüber gestellt und diskutiert. Es wurde dabei auf ihre Ansätze, Einsatzgebiete und die von ihnen bevorzugt verwendeten Sensoren eingegangen.

Abschließend wurde ein erster Vorschlag für ein System, welches Ontologien zur Aktivitätserkennung nutzt, formuliert. Der Systementwurf trennt hierbei primäre und sekundäre Sensoren anhand ihrer Aussagekraft. Der Kern des Systems besteht aus einer Ontologie, die den wissensbasierten Ansatz fördert. Der wissensbasierte Ansatz wird zusätzlich durch maschinelles Lernen erweitert und mit dem datenbasierten Ansatz kombiniert. Das so entstandene hybride System

kann sich selbständig erweitern und anpassen. Der Systementwurf soll Trefferquote, Genauigkeit, Fehlerquote, Echtzeitfunktion und Vollständigkeit verbessern.

Literatur

1. Akdemir, U., Turaga, P., Chellappa, R.: An ontology based approach for activity recognition from video. In: Proceedings of the 16th ACM international conference on Multimedia. pp. 709–712. ACM (2008)
2. Chen, L., Nugent, C.: Ontology-based activity recognition in intelligent pervasive environments. *International Journal of Web Information Systems* 5(4), 410–430 (2009)
3. Chen, L., Nugent, C.D., Wang, H.: A knowledge-driven approach to activity recognition in smart homes. *Knowledge and Data Engineering, IEEE Transactions on* 24(6), 961–974 (2012)
4. Guzzoni, D., Baur, C., Cheyer, A.: Modeling human-agent interaction with active ontologies. In: *AAAI Spring Symposium: Interaction Challenges for Intelligent Assistants*. pp. 52–59 (2007)
5. Helaoui, R., Riboni, D., Stuckenschmidt, H.: A probabilistic ontological framework for the recognition of multilevel human activities. In: *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. pp. 345–354. ACM (2013)
6. Latfi, F., Lefebvre, B., Descheneaux, C.: Ontology-based management of the telehealth smart home, dedicated to elderly in loss of cognitive autonomy. In: *OWLED*. vol. 258 (2007)
7. Ni, H., Abdulrazak, B., Zhang, D., Wu, S.: Cdtom: A context-driven task-oriented middleware for pervasive homecare environment. *arXiv preprint arXiv:1102.1152* (2011)
8. Riboni, D., Bettini, C.: Context-aware activity recognition through a combination of ontological and statistical reasoning. In: *Ubiquitous Intelligence and Computing*, pp. 39–53. Springer (2009)
9. Springer, T., Turhan, A.Y.: Employing description logics in ambient intelligence for modeling and reasoning about complex situations. *Journal of Ambient Intelligence and Smart Environments* 1(3), 235–259 (2009)
10. Yamada, N., Sakamoto, K., Kunito, G., Isoda, Y., Yamazaki, K., Tanaka, S.: Applying ontology and probabilistic model to human activity recognition from surrounding things. *Information and Media Technologies* 2(4), 1286–1297 (2007)

Design von Studien mithilfe der Experience Sampling Method

Grundlagen, Einsatzgebiete, Stärken und Schwächen

Özgün Arziman*

Betreuer: Anja Bachmann†

Karlsruher Institut für Technologie (KIT)
Pervasive Computing Systems – TECO

*ubeft@student.kit.edu

†bachmann@teco.edu

Zusammenfassung. Die Experience Sampling Method (ESM) ist eine Methode, qualitative und quantitative Daten in Echtzeit zu messen. Sie wird vor allem in den Bereichen eingesetzt, in denen das Sammeln von relevanten Informationen sehr schwer zu erreichen ist, wie zum Beispiel bei der Messung von Gedanken, Gefühlen und Stimmungen. Früher wurde die ESM hauptsächlich im psychologischen Bereich eingesetzt, doch mit dem technologischen Fortschritt und dem Einsatz von diversen ESM-Tools, wird der ESM auch in anderen Forschungsgebieten immer mehr Beachtung geschenkt. In meiner Arbeit werde ich die ESM vorstellen, auf ihre verschiedenen Einsatzgebiete eingehen, indem ich einige Arbeiten zusammenfasse, einige ESM Tools vorstellen und die Stärken und Schwächen der Methode aufzeigen. Da zum Sammeln der Daten immer mehr Mobiltelefone eingesetzt werden, werde ich vor allem auf den Einsatz der ESM über elektronische Tools für Smartphones eingehen.

Schlüsselwörter: Experience Sampling Method, ESM Tools, Einsatzgebiete, Smartphones

1 Einleitung

Stimmungen, Gefühle und Gedanken werden von Menschen im Alltag nicht bewusst wahrgenommen, weswegen sie sehr schnell in Vergessenheit geraten können. Das ist ein großes Problem für Forscher, die es interessiert, wie sich ein Proband in einer bestimmten Situation oder im Umgang mit einer neuen Erfindung fühlt. Die Experience Sampling Method (ESM) nach Csikszentmihalyi und Larson [8] ist eine Methode, die es Forschern ermöglicht Stimmungen und Gedanken in der erlebten Situation abzufragen und somit der Vergessenheit in Situ entgegen zu wirken. Die Methode hat ihren Ursprung in der Psychologie

und wird in der Gegenwart auch immer öfter in anderen Wissenschaften eingesetzt. Für den Bereich der Mensch-Maschine-Interaktion wird die ESM immer interessanter [28] und wird beispielsweise von Consolvo et al. [4] vermehrt aufgegriffen und diskutiert. Um den Bedürfnissen der Mensch-Maschine-Interaktion gerecht zu werden, werden verschiedene ESM-Tools entwickelt und angeboten. Durch ihren Einsatz, in Form von Applikationen für Smartphones, wird das Sammeln von Daten für Forscher erleichtert, da sie die eingegebenen Daten von den Probanden direkt in elektronischer Form geliefert bekommen. Da Smartphones einen immer größer werdenden Stellenwert in unserer Gesellschaft haben und mindestens jeder zweite über 13 in Deutschland ein Smartphone besitzt¹ muss den Teilnehmern von Studien weder ein Endgerät ausgehändigt werden noch ein Einführungskurs in die Nutzung von Smartphones angeboten werden. Die Tools, die für Smartphones angeboten werden, werden für die Wissenschaftler auch immer benutzerfreundlicher, da sie keine Programmierkenntnisse brauchen. Außerdem ist es nach Hofmann und Patel [17] eine kostengünstige Methode. Aus diesen Gründen wird in folgender Arbeit vermehrt auf die Durchführung von ESM Studien eingegangen, die das Smartphone als Hilfsmittel heranziehen.

Die Forscher werden mit verschiedenen Fragestellungen konfrontiert, denen sie sich beim Arbeiten mit der ESM zu stellen haben, beispielsweise "Wie oft kann man einem Nutzer Fragen stellen ohne ihn zu stören oder zu überfordern?", "Wie kann man valide Informationen erhalten?" oder "Wann ist der perfekte Zeitpunkt einen Nutzer nach seinen Gefühlen zu fragen?". Auf diese Fragen werde ich im folgenden versuchen Antworten zu liefern.

Zunächst werde ich in meiner Arbeit die Experience Sampling Method definieren und die Möglichkeiten für ihren Einsatz beschreiben, anschließend werde ich auf Studien aus verschiedenen Einsatzgebieten eingehen und verschiedene Tools für Smartphones vorstellen. Zum Schluss werde ich auf die Best-Practises eingehen und über die Stärken und Schwächen beim Einsatz der ESM in der Forschung diskutieren.

2 Grundlagen

2.1 Definition "Experience Sampling Method"

Die Experience Sampling Method (ESM) wird nach Schallberger et al. [30] im deutschen als "Befindensstagebuch" oder als "Erlebens-Stichproben-Methode" bezeichnet. Sie ist eine Befragungstechnik, die darauf abzielt Stimmungen, Gefühle und Gedanken in Echtzeit aufzufangen [8]. Dabei werden Daten direkt in der Situation, in der sie erlebt werden gesammelt, die den Alltag verschiedener Untersuchungspersonen in einer bestimmten Untersuchungsperiode ausmachen [30]. Es bietet die Möglichkeit, Veränderungen in der Datenreihe des Wohlbefindens zu analysieren und baut eine Verbindung zwischen dem externen Umfeld eines Probanden und dessen Gefühlen auf [15]. Nach Barrett und Barrett [3]

¹ <http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-der-smartphonenuutzer-in-deutschland-seit-2010/>

wurden die Befragungen früher mit Papier und Stift durchgeführt. Dafür musste der Proband den ganzen Tag Notizbücher mit vorformulierten Fragebögen bei sich tragen und zu bestimmten Zeitpunkten ausfüllen. Heutzutage braucht der Teilnehmer nur noch ein mobiles Endgerät, das einen Signal ausgibt, wenn der Teilnehmer einen Fragebogen ausfüllen soll. Dieser Fragebogen wird elektronisch visualisiert und beantwortet und somit gehören das Mitnehmen des Papiers und des Stiftes der Vergangenheit an [31].

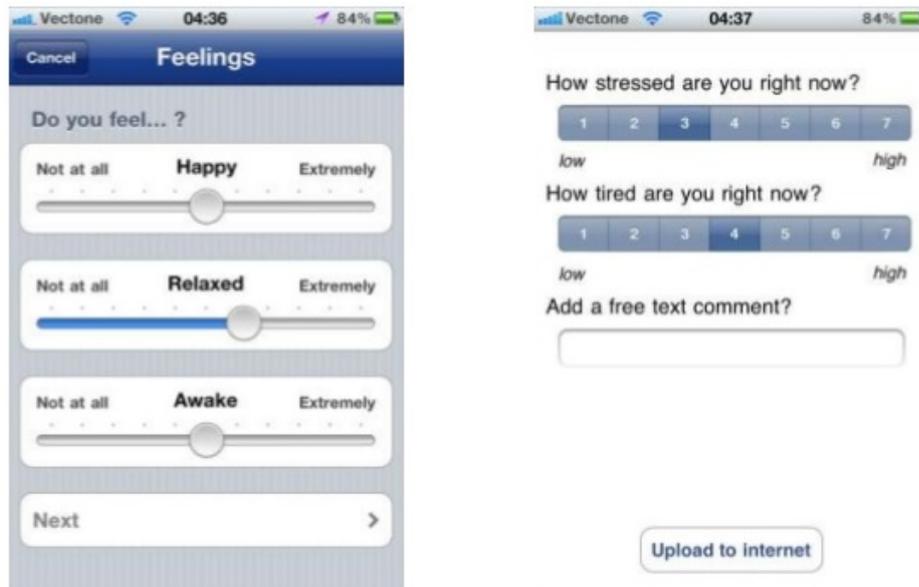


Abb. 1. Beispiel für ein User Interface einer ESM Studie. [39]

2.2 Andere Bezeichnungen

In der Literatur findet man auch andere Bezeichnungen für die Experience Sampling Method, u.a. Ambulatory Assessment, ecological momentary assessment - EMA, Day Reconstruction Method - DRM, Beeper Study. Der wesentliche Unterschied der ESM von den anderen Ansätzen liegt nach Papastefanou [27] in der Untersuchungsperiode.

Ambulatory Assessment Beim Ambulatory Assessment geht es darum, die "Aspekte menschlichen Verhaltens" an Ort und Zeit zu sammeln, wenn sie erlebt werden [27]. Der Fortschritt des "Wearable Computing" macht Messungen von gesundheitlichen Merkmalen in valider Form möglich. Wearable Computing

umfasst Techniken, bei denen das messende Engerät nahe am Körper des Probanden getragen wird. Dies kann in Form von Armbändern, Brillen oder auch Kleidungsstücken sein.

EMA Die Ecological Momentary Assessment Methode ist auf Stone und Shiffman [34] zurückzuführen, wobei kein wesentlicher Unterschied zum Ambulatory Assessment zu beobachten ist [3].

DRM Bei der Day Reconstruction Method erhalten die Probanden zuerst Fragen zum Vortag, so dass Erinnerungen hervorgerufen werden und die erlebten Situationen vom Vortag in Tagebuchform rekonstruiert werden. Für jede Sequenz müssen sie dann, wie bei der ESM, Fragen zur Situation und über ihre Gefühle beantworten [20].

Beeper Study Die Beeper Study ² [19] ist eine Technik, bei der die Probanden einen Beeper und eine Broschüre erhalten. Immer wenn der Beeper ein Signal gibt, soll der Proband einen Fragebogen ausfüllen.

2.3 ESM Studien

Es gibt zwei verschiedene Typen an Experimenten: das Laborexperiment und das Feldexperiment. Beim Laborexperiment findet die Studie in einer künstlichen Umgebung statt, in der die äußeren Einflussfaktoren kontrollierbar sind und somit die interne Validität gesteigert werden kann. Die externe Validität ist relativ niedrig, da die ermittelten Ergebnisse schwer auf die Realität übertragbar sind. Beim Feldexperiment ist es genau das Gegenteil, da keine künstliche Umgebung erzeugt wird, sondern die Studie im realen Umfeld der Probanden durchgeführt wird. Folglich ist die externe Validität sehr hoch aber die interne Validität eher niedrig, da hier äußere Bedingungen nicht kontrollierbar und Störgrößen nicht abschaltbar sind. ³

ESM Studien kommen nach Khan [21] und nach Consolvo und Walker [6] vorwiegend im Feld zum Einsatz, um an Daten zu gelangen, die man nicht unmittelbar beobachten kann. Bevor eine Feldstudie mit der Experience Sampling Methode durchgeführt wird, müssen die Forscher viele Entscheidungen bezüglich der Form treffen. Zunächst müssen sie wählen wie lange die Studie durchgeführt werden soll, wie oft die Teilnehmer pro Tag einen Fragebogen ausfüllen sollen, wie viele Fragen solch ein Fragebogen enthalten soll und zu welchen Zeitpunkten die Teilnehmer antworten sollen. Dann müssen sie entscheiden, ob sie das Tool, das sie einsetzen möchten selbst entwickeln oder ob sie eine fertige Plattform benutzen. Außerdem müssen sie sich bei der Auswahl des Endgerätes festlegen.

² <http://www.christinecostello.com/projects/beeper.html>

³ <http://i-literacy.e-learning.imb-uni-augsburg.de/node/1250>

Die unterschiedlichen Möglichkeiten wurden von Consolvo und Walker [6] wie folgt zusammengefasst. Um auf einen Fragebogen aufmerksam zu machen, werden Alarmsignale verwendet. Nach Reis und Gable [29] gibt es drei verschiedene Alarmtypen: die zufällige, die geplante und die ereignisgesteuerte Alarmierung. Bei der zufälligen Alarmierung wird pro Tag eine bestimmte Anzahl an Signalen gesendet, wobei die Zeitpunkte nicht definiert sind. Im Falle einer geplanten Alarmierung, wird im vornherein festgelegt zu welchen Uhrzeiten das Endgerät Signale abgibt. Bei einer ereignisgesteuerten Alarmierung hingegen, wird immer dann ein Signal abgegeben, wenn der Teilnehmer ein bestimmtes Ereignis ausführt. Zum Beispiel wenn er in seinem Smartphone eine bestimmte Applikation öffnen möchte oder wenn er sie öffnet und eine bestimmte Funktion ausführen will, wird er gebeten einen Fragebogen auszufüllen. Es gibt auch Studien, bei denen beispielsweise per GPS überprüft wird, ob der Teilnehmer sich bewegt und immer wenn er eine bestimmte Zeit irgendwo stehen bleibt, wird er gefragt wo er denn gerade ist und wie er sich an diesem Ort fühlt. Solche Fragen sind im Bereich des Tourismus sehr interessant, welcher im dritten Kapitel näher beleuchtet wird. Die ereignisbasierte Alarmierung wird nach Lathia et al. [24] auch als sensorbasierte Alarmierung bezeichnet.

Bei der Auswahl des Softwaretools muss beachtet werden, ob ein Zeitfenster einstellbar ist, ob die Anzahl der Alarme pro Tag und die Anzahl der Alarme für die Untersuchungsperiode variierbar sind. Zeitfenster können notwendig sein, um beispielsweise nächtliche Störungen zu vermeiden. Es kann aber auch sein, dass Zeitfenster bei den Teilnehmern unterschiedlich sind. In diesem Fall müsste man eine flexible Plattform wählen. Eine entscheidende Frage ist auch, wie viele Alarme pro Tag gesendet werden sollen, so dass die Teilnehmer nicht gestört oder belästigt werden und trotzdem genug Daten gesammelt werden können. Die Anzahl der Alarme in der Untersuchungsperiode sollte auch festgesetzt werden, damit die Untersuchungspersonen nicht nach Durchführung der Studie noch Alarme empfangen. Dies betrifft vor allem Studien, bei denen elektronische Endgeräte an die Teilnehmer verteilt werden.

Darüber hinaus muss entschieden werden, ob das Endgerät einen Ton ausgibt, vibriert oder ob eine LED aufleuchtet, wenn die Untersuchungsperson einen Fragebogen ausfüllen soll. Diese Frage kann in Situationen, wie beispielsweise in einem Kino oder auf der Arbeit von Bedeutung sein. Hier wäre eine flexible Lösung von Vorteil, so dass der Nutzer das Endgerät auch auf stumm schalten kann. Fragen und Antworten in Textform sind leicht zu übermitteln und auszuwerten, da man sie größtenteils direkt in statistische Auswertungsprogramme einspielen kann. Eine Alternative zu ausgeschriebenen Fragen sind aufgenommene Fragen, die der Teilnehmer anhört anstatt zu lesen und auf die er einfach antwortet und nicht schreiben oder tippen muss. Dies kann auch in Form von einem Telefonat erfolgen. Bei Audioaufnahmen ist zu beachten, dass sie fehleranfälliger als ausgeschriebene Daten sind.

Eine weitere Entscheidung muss getroffen werden, wenn es um die Reihenfolge der Fragen im Fragebogen geht. Soll die Reihenfolge zufällig gewählt oder vorher bestimmt werden? Sollten Wahrscheinlichkeiten festgelegt werden um be-

stimmte Fragen häufiger als andere zu stellen? Soll es bestimmte Fragen auf bestimmte Antworten der Nutzer geben? All diese Fragen sollte sich der Forscher vorher überlegen.

Zur Durchführung der Befragungen stehen zahlreiche Endgeräte zur Verfügung: Traditionelle Mobiltelefone und Smartphones, Pager, Uhren, PDAs, Audiorecorder, Computer, speziell angefertigte Endgeräte oder ganz einfach Papier und Stift. Es wird immer mehr auf die Nutzung von Smartphones zurückgegriffen, da wie bereits erwähnt, die Anzahl der genutzten Smartphones in den letzten Jahren sehr angestiegen ist und mögliche Studienteilnehmer leichter zu finden sind.

2.4 Akquise von Probanden

Es gibt unterschiedliche Möglichkeiten um Teilnehmer für Studien zu rekrutieren. Abdesslem et al. [1] haben beispielsweise Poster aufgehängt, E-Mails an Studenten verschickt, deren E-Mail-Adresse auf sogenannten Mailing-Listen waren und haben zusätzlich eine Anzeige direkt auf Facebook platziert. Sie haben mehr Rücklauf bekommen als dass sie freie Plätze für Probanden haben und haben die tatsächlichen Teilnehmer dann nach bestimmten Kriterien ausgewählt.

Consolvo und Walker [6] haben Poster in Einkaufsläden und Restaurants aufgehängt, Anzeigen im Internet platziert und über Mundpropaganda Probanden geworben. Sie haben vorausgesetzt, dass die Kandidaten zwischen 18 und 75 Jahre alt sind und regelmäßig ein Mobiltelefon verwenden.

Auf ähnliche Weise haben auch Fröhlich et al. [12] ihre Teilnehmer gefunden. Sie haben Poster in Restaurants, Cafes und an Gebäude in Seattle gehängt. Außerdem haben sie Anzeigen auf Craigslist.org platziert. Die Probanden wurden anhand von Merkmalen wie ihre Erfahrung im Umgang mit einem Mobiltelefon, ihre Angaben über die Zeiten, in denen sie nicht zu Hause sind und über ihre Internetverfügbarkeit zu Hause und auf der Arbeit, ausgewählt.

Lathia et al. [24] haben direkt nach Probanden gesucht, die ihr eigenes Endgerät in der Studie benutzen würden und haben ihnen keinerlei finanzielle Anreize angeboten. Sie haben Anzeigen auf Twitter und Facebook aufgegeben, auf der Seite der selbst entwickelten Applikation gesucht und über die Mailing-Listen von Universitäten die Studenten angeschrieben.

3 Einsatzgebiete

Ihren Ursprung hat die Experience Sampling Method in der psychologischen Forschung. Inzwischen ist sie sowohl in der Kommunikations- und Medienwissenschaft, in der Unterrichtsforschung als auch in der Glücksforschung weit verbreitet [28].

3.1 Gesundheitssektor

Hofmann et al. [16] führten eine ESM Studie im Bereich "Diät und Selbstkontrolle beim Essen im alltäglichen Umfeld" durch. Dazu statteten sie 208 Teilnehmer

beim Einführungsgespräch mit *Blackberry Smartphones* aus, die nicht fähig waren, Anrufe zu tätigen oder zu erhalten. Die Befragung fand über eine Zeitspanne von *einer Woche* statt, in der die Probanden *pro Tag sieben Signale*, in einem *Zeitfenster von 14 Stunden*, empfangen haben. Innerhalb des Zeitfensters, sollte in jedem zwei Stunden Block ein Fragebogen ausgefüllt werden, dessen Alarmierung *zufallsgesteuert* war. Zuerst wurde ihnen die Frage gestellt, ob sie Gelüste haben oder innerhalb der letzten halben Stunde hatten. Falls nicht, ist die Befragung an dieser Stelle beendet. Falls sie Gelüste haben oder hatten, sollten die Probanden diese genauer kategorisieren und noch weitere Fragen beantworten. Nach der Durchführung der ESM Studie sollten die Teilnehmer noch einen weiteren Fragebogen ausfüllen. Bei der Abgabe erhielten sie *20 Dollar* und für den Fall, dass sie mehr als 80 Prozent der Befragungen beantwortet haben noch *weitere Geschenke* wie z.B. Kinokarten. Deshalb hatten sie eine *Rücklaufquote von 92,2 Prozent*. Sie fanden heraus, dass die Teilnehmer öfter Lust auf ungesundes Essen hatten als auf gesundes. Es gab keine Unterschiede in der Frequenz zwischen den Gelüsten zwischen Leuten, die ihre Diät ernster nahmen und denen, die die Diät nicht so ernst nahmen. Sie heben hervor, dass im Bereich des Essverhaltens in der Psychologie noch Handlungsbedarf besteht. Und dass dieser mit einem konzeptuellen Rahmen und verschiedenen Tools gelöst werden kann.

3.2 Unterrichtsforschung

In Lehr-Lern-Arrangements ist es nicht gerade unüblich, dass *Langeweile* vorkommt. Götz et al. [14] führten eine ESM Studie durch, um heraus zu finden, was Schülerinnen und Schüler tun, wenn sie Langeweile im Unterricht empfinden und wie sie diese regulieren. Insgesamt diskutieren sie die Ergebnisse von *vier Studien*, die sie mit Schülerinnen und Schülern aus beruflichen Schulen des Gesundheitswesens, aus Hauptschulen, aus Realschulen und Gymnasiasten und Studierenden einer Universität durchführten. Zwei Studien dauerten *sechs bzw. sieben Tage* und die Teilnehmer sollten *pro Tag fünf bzw. sechs Fragebögen* ausfüllen, an die sie per einen *Handheld-Computer* erinnert wurden. Eine Studie fand in Form eines *30-minütigen Interviews* statt, der mit einem *Diktiergerät* aufgezeichnet wurde und bei der anderen sollten die Probanden einen *qualitativen Fragebogen* ausfüllen. Sie haben festgestellt, dass Langeweile im Unterricht eher aufkommt, wenn die Schülerinnen und Schüler das Fach als unwichtig empfinden und unerwartet viele Schülerinnen und Schüler diese einfach hinnehmen und nicht bewusst regulieren. Falls sie versuchen die Langeweile zu regulieren, ist die Handlung nicht lern- und leistungsförderlich.

Auch Venetz [35] möchte eine ESM Studie im Unterricht durchführen, um Daten über das *Wohlbefinden von Schulkindern* zu sammeln. Das Wohlbefinden ist jedoch schwierig von außen zu messen. Deshalb soll die ESM angewendet werden, so dass die Kinder ihr Wohlbefinden und die Situation, in der sie sich gerade befinden, selbst beschreiben können. Es sollen *45 Schulklassen* der sechsten Klasse über *eine Schulwoche* befragt werden. Geplant sind *14 Fragebögen innerhalb des Untersuchungszeitraumes*.

Nach Venetz und Zurbriggen [36] beinhaltete die erwähnte Studie *8600 Befragungen von mehr als 700 Schülerinnen und Schülern*. Zudem wurden *600 Fragebögen von unterrichtenden Lehrern* ausgefüllt. Die Forscher stellten fest, dass Schulkinder mit Leistungsschwächen, wie ihre Mitschülerinnen und Mitschüler, aktiv am Unterricht teilnehmen und nicht überfordert sind, so wie es viele erwarten würden.

Wertli und Audeoud [37] führten ebenfalls eine Datenerhebung im ESM-Design mit hörgeschädigten Kindern in der Schule durch. Sie waren am *habituellen und gegenwärtigen Befinden* interessiert. Jedes Kind bekam einen *elektronischen Piepser* und wurde insgesamt *35 mal innerhalb von einer Woche* befragt. Die Kinder haben Kopien eines kurzen Fragebogens ausgehändigt bekommen, die *immer die gleichen Fragen* beinhalteten. Die Lehrer wurden auch befragt. Hörgeschädigte Kinder sind im jungen Alter zufriedener mit ihrem Leben als wenn sie älter sind. Das Wohlbefinden in Lehr-Lern-Situationen ist bei ihnen höher als bei Kindern ohne Hörprobleme, da sie dem Unterricht konzentrierter folgen müssen. Das positive Feedback der hörgeschädigten Kinder ist anders als man erwartet hätte und somit war es gut, die Kinder in der Unterrichtssituation über ihr Wohlbefinden zu befragen. Hier ist die ESM eine gute Möglichkeit, um an aussagekräftige Informationen zu gelangen.

3.3 Glücksforschung

Um *persönliches Glück* zu messen, führten Csikszentmihalyi und Hunter [7] eine Datenerhebung mit der ESM durch. Sie statteten *828 Studenten* mit *Armbanduhren* aus, die Signale ausgegeben haben, wenn ein Fragebogen ausgefüllt werden sollte. *Eine Woche* lang ertönten *pro Tag acht Signale* und ein Fragebogen mit offenen Fragen sollte beantwortet werden. Die Forscher fanden heraus, dass die Studenten sich glücklicher fühlten, umso näher das Wochenende heranrückte. Montags, dienstags und mittwochs waren sie weniger glücklich. Die soziale Struktur des Tagesablaufs beeinflusst das persönliche Glücksgefühl sehr. Wenn man die einzelnen Stunden des Tages betrachtet ist das empfundene Glück abends höher, wobei die Mittagszeit, in der die meisten Menschen ihr Mittagessen zu sich nehmen, eine Ausnahme darstellt. Studenten fühlen sich besser, wenn sie sich mit ihren Freunden unterhalten als wenn sie Aufgaben für die Universität lösen. Die Wissenschaftler betonen, dass die ESM es möglich macht, Daten in diesem Kontext zu messen. Außerdem wird nicht nur eine Befragung durchgeführt, sondern es besteht die Möglichkeit eine Untersuchung über einen langen Zeitraum durchzuführen. In diesem Bereich könnten Forscher mithilfe der ESM an Erkenntnisse gelangen, die langfristig das Glück steigern und die Qualität des Lebens erhöhen.

”A wandering mind is an unhappy mind” [22] behaupten philosophische und religiöse Traditionen. Die Forscher entwickelten ein *Web-Portal für das iPhone* um festzustellen, ob an dieser Aussage etwas wahres dran ist. Mithilfe des Portals sollen Gefühle, Gedanken und aktuelle Handlungen der Probanden in Situ aufgenommen werden. Die Datenbank sammelte über *1 Mio. Befragungen* von

mehr als 5000 Probanden aus 83 unterschiedlichen Ländern. Sie stellten folgende Fragen:

- "Wie fühlst du dich gerade?",
- "Was machst du gerade?" und
- "Denkst du gerade über etwas nach, dass nicht mit deiner Handlung in Verbindung steht?"

Sie fanden heraus, dass die Gedanken der Probanden im Alltag oft woanders sind und vor allem öfter wandern als in Laborstudien. Zudem stellten sie fest, dass Menschen, die weniger an andere Dinge denken, glücklicher sind. An diese Erkenntnisse zu gelangen, ohne ein Echtzeit-Tool ist schwierig beziehungsweise fast unmöglich.

3.4 Musikwissenschaft

Nach Krause et al. [23] gibt es seit der Digitalisierung immer mehr Geräte, die angeboten werden, auf denen man *Musik hören* kann. Die Musikwissenschaftler konnten bisher nur raten, welche Geräte von den Nutzern verwendet werden und wie diese deren tatsächlichen Umgang mit Musik beeinflusst haben. Digitale Endgeräte bieten so viele Möglichkeiten an, um unterwegs Musik zu hören. Die ESM soll dabei unterstützen, herauszufinden welche Geräte Menschen in ihrem Alltag benutzen und ob die Wahl des Endgerätes etwas mit dem Alter, das Geschlecht, o.ä. zu tun hat. Werden zu unterschiedlichen Tageszeiten oder an unterschiedlichen Tagen andere Geräte verwendet? Außerdem soll geprüft werden ob die aktuelle Stimmung des Probanden oder die Möglichkeit zur Auswahl des Titels in Verbindung mit dem Gerät stehen. Die Probanden haben vor der Durchführung der ESM einen *Fragebogen mit deren demographischen Daten und deren Umgang mit der Musik* ausgefüllt. Die ESM Studie ging über eine *Zeitspanne von sieben Tagen*, an denen an die *177 Probanden* eine Nachricht vor 15.30 Uhr und eine nach 15.30 Uhr gesendet wurde, dass sie möglichst bald einen *Online-Fragebogen*, über deren Kontakt mit Musik innerhalb der letzten zwei Stunden, ausfüllen sollen.

Sloboda et al. [33] haben eine ähnliche Studie durchgeführt und haben den Probanden Fragen über ihren aktuellen Aufenthaltsort, über deren Aktivität dort, über die laufende Musik, über deren Stimmung in dieser Situation und über die Kontrollmöglichkeit über den Musiktitel gestellt. Es wurden *acht Nutzer* über einen *Pager* über den *Zeitraum von einer Woche* befragt. Das Ergebnis der Studie ist, dass Musik nicht die zentrale Rolle in der erlebten Situation spielt und dass die Stimmung der Teilnehmer besser ist, wenn sie die Musik selber auswählen können.

Die ESM ist nach Greasley und Lamont [13] ein guter Ansatz, Daten über musikalische Erfahrungen im Alltag zu sammeln. Außerdem sagen sie, dass es normalerweise schwierig ist, im musikalischen Bereich an nützliche Daten zu kommen. Sie haben *25 junge Menschen eine Woche lang* befragt und anschließend *ein Interview* durchgeführt. Die Probanden wurden dann in zwei Cluster

eingeteilt: Diejenigen, die wenig Musik über den Zeitraum von einer Woche hören und diejenigen, die viel Musik hören und selbst entscheiden möchten, was sie hören. Die Studie soll Aufschluss über die Rolle der Musik im täglichen Leben geben, mit einem Blick auf die Cluster und die Effekte, die die Musik mit sich bringt.

Bisher war es nur möglich, Forschungen über die Wahrnehmung von Musik in Form von Laborstudien durchzuführen [2]. Die ESM ist jedoch ein wahrer Fortschritt in diesem Bereich und ermöglicht neue Einsichten in die Wahrnehmung von Menschen in ihrem natürlichen Umfeld. Bailes hat *elf Musikstudenten eine Woche lang* über das *Mobiltelefon* beauftragt *Fragebögen in Papierform* auszufüllen. Immer wenn das Mobiltelefon einen Signal ausgegeben hat, sollten sie einen zweiseitigen Fragebogen beantworten, der sich an dem von Sloboda et al. [33] orientiert. Die Probanden konnten den Weckruf zweimal verschieben. Zuvor wurde ein zehnminütiges Training durchgeführt, an dem sie 42 Papierbögen erhalten haben, also *sechs für einen Tag*. Eine Woche nach der Durchführung der ESM Studie wurde ein *abschließendes Interview* über die Erfahrungen der Teilnehmer geführt. Bailes kam auf das Ergebnis, dass Musikstudenten in hohem Maße passiv Musik wahrnehmen und aktiv Musik hören.

3.5 Tourismus

Ob eine Beziehung bei Besuchen von verschiedenen Standorten einer Person eine Vorliebe besteht, interessiert Fröhlich et al. [12]. Die Autoren untersuchten, ob die *Präferenz einen bestimmten Ort zu besuchen in Verbindung mit der Häufigkeit des Besuches oder der Entfernung des Ortes steht*. Hierfür führten sie eine Studie mit *16 Teilnehmern über einen Zeitraum von vier Wochen* durch, indem sie die Probanden über ein *sensorbasiertes Tool*, das sie für Mobiltelefone entwickelt haben, befragten. Das verwendete Tool heißt *My Experience* und wird in Kapitel 4 näher erläutert. In diesem Zeitraum wurden *3458 Befragungen an 1981 Orten* realisiert. Mit den *16 Probanden* wurden zusätzlich zur Datenerhebung in ESM Form, *Kennenlern- und Abschlussgespräche* geführt. Die Probanden wurden mit einem *Pocket-PC* ausgestattet, das sie *bis zu elf mal am Tag* dazu aufforderte, einen Fragebogen zu beantworten. Um qualitativ hochwertigere Daten zu sammeln, sollten die Teilnehmer zusätzlich *Web Tagebücher* ausfüllen. Ungefähr in der Mitte des Befragungszeitraumes wurde ein Gespräch unter vier Augen gehalten, um sich über die bisherige Erfahrungen auszutauschen. Die ESM Studie wurde *sensorbasiert* ausgeführt, in diesem Fall über GSM Signale. Wenn der Proband sich bewegte, und dann für zehn Minuten stehen blieb, erhielt er ein Signal, dass ihn aufforderte eine Befragung durchzuführen. Eine Befragung kann *bis zu 23 Fragen* beinhalten. Der kürzeste Fall ist, wenn er seinen Standort nicht gewechselt hat, also vom letzten bis zum aktuellen Fragebogen am gleichen Ort ist. Dann wird er nur gefragt, ob er immer noch am Ort xyz ist und falls er mit "Ja" antwortet ist die Befragung beendet. Die Forscher fanden heraus, dass die Präferenz mit der Entfernung eines Ortes und der Häufigkeit des Besuches korreliert ist, aber nicht in erwartetem Maße. Sie machen darauf aufmerksam, dass in diesem Bereich in Zukunft mehr geforscht werden muss. Beispielsweise

könnte man prüfen ob die Aufenthaltsdauer an einem Ort, zeitliche Faktoren oder die Art des Transportmittels eine besondere Rolle bei der Auswahl einen Ort zu besuchen, spielen.

3.6 Kommunikationswissenschaft

Die Anzahl der Nutzer in sozialen Netzwerken, die ihre Standorte mit anderen Nutzern teilen, steigt. Eine Studie, die in die Privatsphäre greift durchzuführen ist schwierig, aber dennoch haben Abdesslem et al. [1] dies mithilfe der ESM realisiert. *Zwei Wochen lang* wurde das Verhalten von *40 Studenten* beim Teilen von Inhalten auf Facebook, einem beliebten sozialen Netzwerk, aufgenommen und ihnen wurden dafür *50 Dollar* ausbezahlt. Die Teilnehmer wurden mit einem *Mobiltelefon* ausgestattet, das deren Standorte alle 10 Sekunden über GPS lokalisiert und speichert. Wenn die Probanden möchten, können sie die aufgenommenen Standorte auf Facebook teilen. Außerdem macht das Mobiltelefon auf die ESM Fragebögen aufmerksam und stellt über die *SMS Funktion Fragen*, die direkt über das Endgerät beantwortet werden können. Alle 5 Minuten werden sämtliche Daten auf einen internen Server transferiert. Es wurden zwei verschiedene Alarmtypen verwendet: die *zufällige und die ereignisgesteuerte* Alarmierung. *Zehn zufällige Befragungen pro Tag* wurden durchgeführt, bei denen danach gefragt wurde, wie der Proband sich fühlen würde, wenn der aktuelle Standort auf Facebook geteilt werden würde. Außerdem soll der Teilnehmer ein Foto vom derzeitigen Standort aufnehmen. Zusätzlich sollten *bis zu zehn Fragebögen* ausgefüllt werden, wenn sich die Probanden an bestimmten Orten länger aufhielten. In diesem Fall wurde gefragt ob er seinen Standort veröffentlichen möchte. Wenn nicht, wurde ihm die Frage gestellt ob er seinen Aufenthaltsort an eine bestimmte Freundschaftsliste freigeben möchte. Manchmal wurde gefragt, ob er sich an einem bestimmten Ort aufhält und ob er diesen teilen möchte und manchmal wurde der Standort genannt und gesagt, dass dieser jetzt auf Facebook veröffentlicht wird. Im Anschluss wurde die Frage gestellt, wie sich der Teilnehmer gegenwärtig fühlt. Jeder Teilnehmer hat also *pro Tag zwischen 10 und 20 Signale* erhalten. Die Ergebnisse wurden in drei Kategorien unterteilt: *privat* - der Standort wurde nicht geteilt, *geteilt* - der Standort wurde mit einer bestimmten Gruppe geteilt und *veröffentlicht* - der Standort wurde mit allen Freunden oder mit der Öffentlichkeit geteilt. Die Studenten teilten eher die Standorte von akademischen Gebäuden oder von Freizeitaktivitäten als die von zu Hause oder von der Bibliothek.

Consolvo et al. [5] führten eine ähnliche Studie durch, bei der sie *16 nicht technisch veranlagte Teilnehmer* über eine Zeitspanne von *zwei Wochen* befragten. Zu erwähnen ist, dass die Befragungen sowohl im *Labor* als auch auf dem *Feld* in Form von *hypothetischen Anfragen von bekannten Leuten, einer nächtlichen Voicemail Studie, Aufgaben, Fragebögen, einer Befragung, die die Privatsphäre klassifiziert und Interviews* stattgefunden haben. Die Teilnehmer erhielten *pro Tag zehn zufällige Signale* und jeder Fragebogen dauerte zwischen zwei und drei Minuten. Sie stellten Fragen wie

- "Wo bist du?",
- "Was machst du?",
- "Mit wem bist du unterwegs?" und
- "Möchtest du deine Standortinformationen mit Person xyz teilen?".

Die Person xyz ist jemand aus einer vorher definierten Freundesliste. Insgesamt wurden *2015 Fragebögen* ausgefüllt. Sie fanden heraus, welche Standorte die Teilnehmer teilen würden, ob die Beziehung zwischen Person xyz zum Probanden eine Rolle spielt, ob der Standort, die aktuelle Handlung oder die Stimmung in Verbindung zum Zeitpunkt der Anfrage steht und weitere, ähnliche Informationen. Außerdem warnen Consolvo et al. [5] in ihrer Einleitung davor, dass die Technologie in diesem Bereich, es immer einfacher für andere macht, unseren Standort zu bestimmen. Der Leser soll sich einmal vorstellen, dass er in naher Zukunft, an einem Samstag Abend, eine Anfrage von seinem Chef erhält, dass er seinen Standort mit ihm teilen soll.

4 Experience Sampling Tools

Mit unserer immer größer werdenden Mediengesellschaft, steigt auch die Anzahl an Experience Sampling Tools, die angeboten werden, um die Befragungen mit möglichst geringem Aufwand durchzuführen. In folgendem Kapitel werden einzelne Tools, die für Smartphones angeboten werden, näher beleuchtet.

4.1 Beispieltools

	myexperience	movisensXS	SurveySignal	BeepMe	METRICwire
Open Source	✓	begrenzt	✗	✓	begrenzt
Plattform	Windows	Android	alle	Android	alle
Endgerät	Computer PDA Smartphones	Smartphones Tablets	Smartphones	Smartphones	Smartphones Computer
Funktion	online offline	online offline	online	online offline	online offline
Sensoren	✓	✓	✗	✓	✓
Anhang	✓	✓	✗	✓	✓

Abb. 2. Die ESM-Tools im Überblick.

MyExperience "MyExperience is a context-aware data collection platform for capturing objective and subjective data as it's experienced." [42] Es ist eine **frei zugängliche, kostenfreie Plattform**, die für **mobile Endgeräte von Windows** angeboten wird, wobei PDAs und Handys einbezogen werden, um kontextbezogene, objektive und subjektive Daten im Feld zu sammeln. Es hat die Möglichkeit die Daten sofort zum Forscherteam zu transferieren, wenn das Endgerät mit dem Internet verbunden ist. Außerdem kann der Benutzer **verschiedene Daten anhängen**, beispielsweise Bilder, Videos, Audioaufnahmen und Nutzerbefragungen. Die Beta Version bietet zahlreiche Möglichkeiten an, **über 50 eingebettete Sensoren** Daten zu sammeln, via GPS oder GSM und das Endgerät gibt nutzerbezogene Daten weiter, wie z.B. Tastendruck- und Akkumutzungsinformationen. Wenn man andere Sensoren benötigt, ist es sehr einfach diese über eine Plug-in Architektur zu verbinden.

Link: <http://myexperience.sourceforge.net/>

movisensXS "movisensXS is the next generation research tool for ambulatory assessment." [43] MovisensXS ist eine Plattform, die für **Android** angeboten wird und aus einer **Internetplattform und einer App für Smartphones** und Tablets besteht. Die Onlineplattform dient beim Überwachen der Studien, beim Erstellen von Fragebögen und beim Kontrollieren der Ergebnisse. Ein großer Vorteil dieses Tools besteht darin, dass keinerlei Programmierkenntnisse erwartet werden. Es ist sofort einsatzbereit und einfach zu bedienen. Die Plattform wird in einer **kleinen Ausführung kostenfrei** und in größerem Umfang mit vielen Features kostenpflichtig angeboten. Dem Forscher stehen **verschiedene Features** zur Verfügung: Sie können beispielsweise mit den Probanden über eine Texting-Funktion in Verbindung bleiben, die Fragebögen in vielen verschiedenen Sprachen gestalten, beobachten wie lange ein Proband gebraucht hat, um eine Frage zu beantworten oder auch nach Medien wie Audioaufzeichnungen, Videos oder Fotos in den Studien fragen. Es sind noch weitere Sensoren geplant und stehen auch bald zur Verfügung.

Link: <https://xs.movisens.com/>

SurveySignal SurveySignal ist eine Plattform, die für **alle gängigen Smartphones** angeboten wird. Sie bietet eine benutzerfreundliche Oberfläche an und ist aus diesem Grund sehr einfach zu bedienen. Außerdem werden keine Programmierkenntnisse erwartet. SurveySignal ist eine **kostenpflichtige Anwendung**, bei der die Forscher für jedes an den Befragten gesendete Signal 10 Cent bezahlen müssen. Die Benutzer erhalten eine **SMS Nachricht als Signal** für einen bevorstehenden Fragebogen, in der ein **Link** mitgeschickt wird, über den sie auf eine **Internetplattform** weitergeleitet werden. Das bedeutet, dass die Befragten **keine lokale App** auf ihr Smartphone installieren müssen. Die Akzeptanz ist deshalb sehr hoch, da keine persönlichen Daten im Hintergrund abgefangen

werden. Jedoch ist es bei diesem Tool **nicht möglich Dateien** mit dem Smartphone anzuhängen oder auch die unterschiedlichen Sensoren der Smartphones zu nutzen.

Link: <http://surveysignal.com/Default.aspx>

BeepMe BeepMe ist eine **kostenfreie** Experience Sampling Plattform, die für **Android Smartphones** entwickelt wurde. Nach jedem gesendeten Signal, das sich wie ein Wecker verhält kann der Teilnehmer entweder sofort oder zu einem späteren Zeitpunkt Fragen beantworten oder Notizen machen. Es besteht die Möglichkeit **Fotos und Audiodateien anzuhängen** oder per GPS Standortinformationen abzuspeichern. Der Export der Daten verläuft nicht automatisiert, sondern muss vom Teilnehmer durchgeführt werden. Die Dateien werden als CSV-Dateien geliefert und können dann in folgende Statistikprogramme Calc, R, PSPP, Excel oder SPSS eingespielt werden.

Link: <http://beepme.youexp.at/de/>

METRICWire METRICWire ist eine Plattform, auf der man **mobile oder online Studien** innerhalb von wenigen Minuten kreieren kann, ohne einen Code programmieren zu müssen. Es werden verschiedene Fragetypen, eine freundliche Benutzeroberfläche, Geo-Standpunkte, die offline Nutzung und die vollständige Kontrollmöglichkeit angeboten. Die **App** muss heruntergeladen werden, wobei dies für die **offline Nutzung** ein wesentlicher Vorteil ist, da keine ständige Verbindung mit dem Internet aufgebaut sein muss. Sobald das Smartphone eine Verbindung zum World Wide Web aufbaut, werden die gesammelten Daten synchronisiert. Die Applikation ermöglicht es dem Forscher Daten zu sammeln, zu analysieren und zu visualisieren, wenn der Forscher mit dem Webportal für Forscher verknüpft ist. "Mobile research is no longer the exception, it's the expectation." [46]

Link: <https://metricwire.com/>

4.2 Beispiel zum zeitlichen Ablauf einer Studie

Abbildung 3 zeigt, wie die einzelnen Phasen einer ESM Studie mithilfe von SurveySignal aussehen. Die Probanden werden auf die Seite der Anmeldung weitergeleitet, nachdem sie die nötigen Informationen über die Studie erhalten haben und in einem Auswahlverfahren nicht aussortiert wurden. Nachdem sie sich erfolgreich registriert haben, werden sie dazu aufgefordert an einer Einführungsstudie teilzunehmen. Einige Tage später nehmen die Probanden an der mobilen Phase der eigentlichen Studie teil, in der sie die SMS Nachrichten auf ihre Smartphones erhalten, die den Link zur Online-Studie enthalten. Im Anschluss an die Datenerhebung, erhalten die Teilnehmer eine Einladung per E-Mail, und können an einer weiterführenden Studie teilnehmen.⁴

⁴ <http://surveysignal.com/examplepage.aspx>

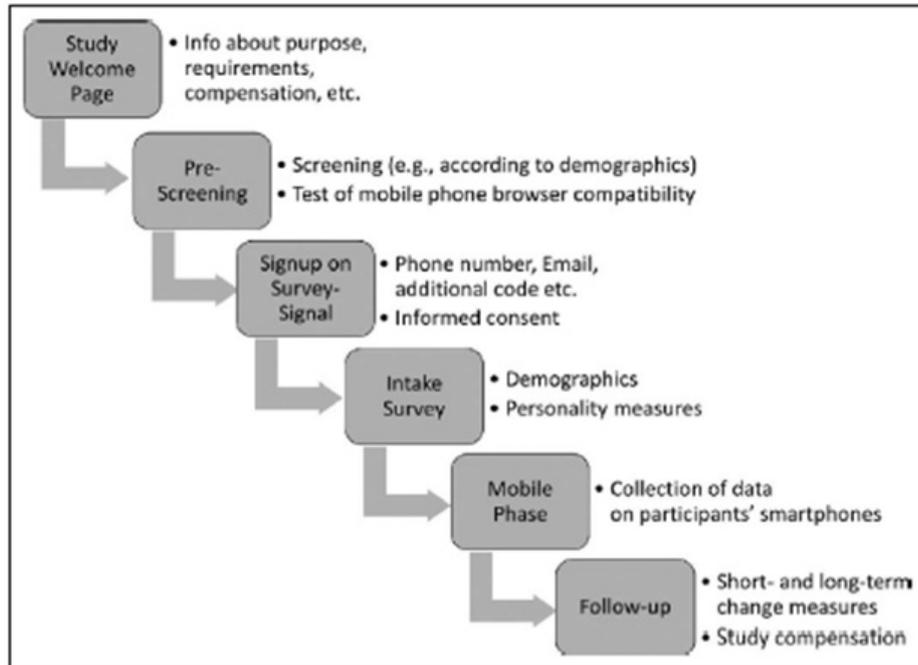


Abb. 3. Die zeitlichen Phasen bei der Durchführung einer Experience Sampling Studie mit SurveySignal. [47]

5 Diskussion

5.1 Best Practises

Schallberger et al. [30] empfehlen, dass eine Befragung nicht länger als zwei bis drei Minuten dauern sollte. Ansonsten sei die Belastung der Probanden zu hoch. Der Fragebogen sollte sich auf eine kleine Anzahl wesentlicher Merkmale beschränken. Zudem sollten die gestellten Fragen verallgemeinerbar sein, so dass sie in jeder Situation verwendet werden können. Forscher sollten die Fragebögen vorher selber ausfüllen und in ihren Alltag einbetten, so dass sie mögliche Schwierigkeiten im Vorfeld identifizieren können. Nach Reis und Gable [29] sollte eine Studie im Idealfall ein bis zwei Wochen andauern und pro Befragungstag zwei bis zwölf Befragungen an den Probanden senden.

Eine Möglichkeit, um viele vollständige Antworten auf Fragebögen zu erhalten, bietet der Vorschlag von Hsieh et al. [18]. Sie fanden in ihrer Arbeit heraus,

dass die Zufriedenheit der Probanden während der Durchführung der Studie höher ist, wenn sie Feedback erhalten.

Wie bereits erwähnt, gibt es drei verschiedene Alarmtypen. Und für jeden Alarmtyp schlagen Fisher und To [11] eine Empfehlung für den richtigen Einsatz vor:

1. zeitgesteuerte Alarmierung:
 - wenn der Alarm als Weckruf genutzt wird, ist die Zustimmung zu diesem Alarmtyp besser
 - wird meistens eingesetzt, wenn Erfahrungen über eine Zeitspanne abgefragt werden und es nicht direkt auf den gegenwärtigen Augenblick ankommt
 - wenn eine Datenerhebung länger andauert, wird zu bestimmten Zeiten die Erfahrung vom letzten Signal bis zum aktuellen Signal erfragt
2. zufällige Alarmierung:
 - wird verwendet, wenn eine Zufallsauswahl an Erfahrungsberichten gebraucht wird
 - wird verwendet wenn Daten über die Stimmung, Mühe, etc. gesammelt werden, die sich in einer Zeitreihe immer wieder verändern
 - misst auch die Zeit in einer Zeitreihe, wenn sich beispielsweise die Stimmung ändert, so dass interpretiert werden kann, weshalb sich die gemessene Variable verändert
3. ereignisgesteuerte Alarmierung:
 - wenn bestimmte Handlungen oder Aktivitäten dem Nutzer widerfahren, wird ein Fragebogen ausgefüllt
 - falls nicht sensorbasiert, wird dem Nutzer beigebracht, in welcher Situation Daten gesammelt werden

Außerdem geben Fisher und To [11] Empfehlungen für den idealen Gebrauch, an denen sich Forscher orientieren können:

- Den Ablauf der Datenerhebung genau planen (Zeitraum, Anzahl der Signale pro Tag) und Alarmtyp passend zum Untersuchungsgegenstand wählen (zeitgesteuerte, zufällige, ereignisgesteuerte Alarmierung).
- Zu allen relevanten Phasen sich Mühe geben, um die erwartete Anzahl an Rückläufen zu erhalten.
- Den Zeitraum der erforderlichen Daten genau erklären (aktueller Zeitpunkt, seit letztem Signal,...), so dass die Messungen eine Hohe Reliabilität und Validität aufweisen.
- Um Fehlmessungen zu vermeiden, sollte eine Pilotstudie durchgeführt werden.
- Die Technologie der Alarmierungen und der Datensammlung sollten in einer Pilotstudie auf dem Feld getestet werden.
- Den Teilnehmern sollten genaue und deutliche Instruktionen gegeben werden und was in besonderen Fällen zu tun ist.

- Eine gute Beziehung zu den Probanden ist sehr wichtig und sollte stabil bleiben. Die Teilnehmer sollten motiviert werden.
- Um die Ergebnisse richtig zu analysieren sollte der Forscher ein Training absolvieren oder Hilfe von Experten holen.
- Die Ergebnisse sollten sauber dokumentiert werden.

Zu viele offene Fragen führen zu wenig Lust bei den Teilnehmern die Fragen zu beantworten, deshalb sollten im Falle von offenen Fragen weniger Signale pro Tag ausgegeben werden oder die Fragebögen sollten in kürzester Zeit zu beantworten sein.

5.2 Stärken und Schwächen

Wie bereits erwähnt, wird durch das immer größer werdende Angebot an ESM Tools, den Forschern die Arbeit beim Sammeln der Daten erleichtert. Die Tools werden den aktuellen Bedürfnissen der Forscher und der Nutzer angepasst. Nach Scollon et al. [31] haben die Probanden früher, als die ESM Fragebögen per Hand ausgefüllt wurden, es leichter gehabt, die Antworten zu späteren Zeitpunkten auszufüllen oder abzuändern, wenn sie einmal keine Lust hatten oder vergessen haben zu antworten. Mit dem technologischen Fortschritt und dem vermehrten Einsatz an technischen Endgeräten kann man dem entgegenwirken, da die Kontrollmöglichkeit gegeben ist. Scollon et al. [31] betonen, dass es fünf Hauptpunkte in der Experiment Sampling Method gibt, die einen Vorteil zu herkömmlichen Methoden darstellen:

- Zum einen können Notizen über zufällige Ereignisse gemacht werden. Dadurch können Schlüsse über die Verbindung zwischen einem Individuum und der gegebenen Situation gezogen werden.
- Die externe Validität ist sehr hoch, da die Ergebnisse aus der Feldforschung stammen und so auch leicht auf die Realität übertragbar sind.
- Mit der ESM wird versucht Erkenntnisse über in der Person liegende Daten zu erheben, wie Gefühle und Gedanken. Solche Daten zu sammeln ist sehr schwer und die ESM ist eine mögliche Lösung die Forschung in diesem Bereich voran zu treiben.
- Die Erinnerung an Gefühle und Gedanken in bestimmten Situationen ist zu einem späteren Zeitpunkt sehr schwach. In der ESM wird der Proband immer wieder dazu aufgefordert, Fragen zu beantworten und somit wird versucht, die Zeitspanne zwischen einer Handlung und dem Ausfüllen eines Fragebogens, gering zu halten. So werden die Antworten aussagekräftig gehalten und nicht verfälscht.
- Die ESM bietet dem Forscher die Möglichkeit an, unterschiedliche Befragungsformen miteinander zu vereinen. Die kurzen Fragebögen können durch Tagebücher, Interviews und längere Fragebögen ergänzt werden. Die Befragten können so Notizen zu den erlebten Handlungen machen, die nicht in den Fragebögen abgefragt wurden und somit die Aussagekraft der Studie erhöhen.

Fahrenberg et al. [10] heben hervor, dass die computergestützten Methoden zahlreiche neue Möglichkeiten bieten, um gegenwärtige Situationen zu erforschen. Unter den folgenden Möglichkeiten wird unterschieden:

- dauerhafte Überwachung
- zeitbasierte oder ereignisgesteuerte Kontrolle
- kontrolliertes Monitoring
- nicht stationäre, psychometrische Untersuchung
- Feldforschung
- interaktive Beobachtung
- und Symptom-Kontrolle und Selbstmanagement.

Es werden einige Vorteile von technologisch unterstützenden Methoden gegenüber herkömmlichen Methoden aufgelistet. Beispielsweise sagen sie, dass die unterschiedlichen Alarmtypen einen hohen Nutzen bringen. Außerdem können die genauen Zeitpunkte der Beantwortung der Fragebögen verfolgt werden. Die Tools sind sehr flexibel und die Echtzeitübertragung der Daten spielt eine wesentliche Rolle. Zudem nennen sie weitere Punkte, die in diesem Bereich von Bedeutung sind.

Auf der anderen Seite verbergen sich auch einige Schwächen hinter der ESM.

- Bei manchen Applikationen muss das Endgerät ständig mit dem Internet verbunden sein, so dass die Signale für einen anstehenden Fragebogen gesendet werden. Die Forscher sollten bedenken, dass nicht jeder Proband einen Datentarif auf dem Mobiltelefon hat und sie dies eventuell gewährleisten müssten.
- Ein weiterer Nachteil ist, dass die Teilnehmer die Endgeräte immer mitnehmen müssen, so dass sie die Fragebögen beantworten können.
- Das Aufladen des Akkus spielt ebenfalls eine große Rolle, da sichergestellt werden muss, dass er den ganzen Tag geladen ist. Bei der Nutzung von privaten Endgeräten der Nutzer kann dieser Punkt eher vernachlässigt werden, da eine gewisse Abhängigkeit zu Smartphones in unserer Gesellschaft zu beobachten ist [17] und es schon zur Gewohnheit gehört, den Akku zu laden. Eine Idee von Abdesslem et al. [1] ist, dass die Teilnehmer kontaktiert werden, wenn der Akku einen bestimmten Stand erreicht, so dass er nicht leer geht. Außerdem könnte man sich auch mit ihm in Verbindung setzen, falls er das Endgerät auf stumm stellt.
- Im Falle einer Verteilung von anderen Endgeräten müssen Forscher bedenken, dass diese Endgeräte geklaut oder kaputt gemacht werden können und sich eventuell einen möglichen Anreiz überlegen, dass dieser Fall nicht eintritt. Für die Sicherstellung der Rückgabe der Endgeräte kann entweder ein Bonusbetrag ausbezahlt werden oder zu Beginn eine Kaution durch die Probanden entrichtet werden, die sie bei Rückgabe wieder bekommen. Anreize in Form von Geld können auch die Rücklaufquote sehr beeinflussen, da die Probanden an mehr Befragungen teilnehmen, wenn sie dafür einen kleinen Betrag erhalten [6]. Dies würde auch die Anzahl an fehlenden Antworten

- minimieren, wenn der Informant keine Lust hat zu antworten. Abdesslem et al. [1] warnen davor, dass wenn man den Teilnehmern zu hohe Anreize in Geldform gibt, dass sie nur aus diesem Grund teilnehmen und dies Auswirkungen auf die Qualität der Datenerhebung hat. Sie schlagen vor, dass man die Höhe des Geldes ein wenig senkt und so höher motivierte Teilnehmer anspricht, die bei der Verwirklichung der Studie mitwirken.
- Ein anderes Risiko bei der Nutzung von Smartphones für ESM Studien besteht darin, dass hier eine falsche Antwort schnell durch zu schnelles Klicken auftreten kann. Für diesen Fall sollten sich die Forscher auch eine Lösung überlegen, vielleicht in Form von einer Verbesserungsfunktion innerhalb weniger Minuten, so dass kein nachträgliches Ändern zu späteren Zeitpunkten mehr möglich ist.

5.3 Benutzerfreundlichkeit von Smartphones

Smartphones sind sowohl für den Forscher als auch für den Probanden ein benutzerfreundliches Tool zur Messung von Daten in Situ. Nach Raento et al. [28] gibt es zwei Hauptgründe, weshalb Forscher Smartphones als Erhebungsinstrument nutzen. Zum einen sind sie sehr flexibel, da sie so vorprogrammiert werden, dass sie mit dem Probanden interagieren und sie können eine Menge an Daten über Kommunikations- und Bewegungsmerkmale speichern. Außerdem besteht die Möglichkeit den Inhalt der Erhebung sofort zum Forscher zu übertragen. Zum anderen ist die Einbeziehung von Smartphones kostengünstig, da wie bereits erwähnt viele Menschen ein eigenes Smartphone besitzen und bereit sind dieses zu Forschungszwecken zu benutzen. Bei längeren Beobachtungsspannen nehmen sie den Forschern viel Arbeit ab, da diese nicht beim Probanden sein müssen um die Daten zu erheben. Raento et al. [28] betonen auch, dass die Validität der Studie durch zwei wesentliche Merkmale von Smartphones, auf der Seite der Probanden, gewährleistet werden. Die Geräte sind schon in das soziale Leben der Probanden eingebettet und die Datenerhebung findet ohne die Anwesenheit des Forschers statt, so dass die externe Validität gesteigert wird. Zudem kann das Gerät über diverse Sensoren Informationen aufnehmen, ohne in direkter Interaktion mit dem Besitzer zu stehen.

5.4 Fehlende Antworten auf Signale

Nach Silvia et al. [32] gibt es drei verschiedene Einflussfaktoren, weshalb Probanden nicht auf die Signale von ESM-Tools antworten. Zum einen liegen die Gründe in der Person selbst, das bedeutet, die Probanden antworten nicht, da sie an Psychose oder Hypomanie leiden oder da sie depressiv sind. Es hat sich ergeben, dass männliche Teilnehmer öfter mal auf das Ausfüllen eines Fragebogens verzichten als weibliche Probanden [25]. Eine andere Ursache kann der Tag sein, an dem gefragt wird. Zu Beginn der Studie wurden mehr Fragebögen pro Tag beantwortet als gegen Ende der Studie. Zum anderen kann man das Nichtantworten der Probanden auf die ausgegebenen Signale zurückführen.

Vormittags und Abends werden mehr Fragebögen ausgefüllt als zu den Nachmittagsstunden. Wenn der Proband sich beim letzten Signal enthusiastisch gefühlt hat, war er dazu geneigt, den folgenden Fragebogen nicht auszufüllen. Silvia et al. [32] betonen, dass die Hauptgründe von fehlenden Antworten innerhalb der Person liegen. Wenn die Gründe sichtbar sind, haben Forscher die Möglichkeit zu handeln und die Anzahl an fehlenden Antworten zu minimieren. Hofmann und Patel [17] schlagen vor, dass eine Erinnerungsfunktion in das Tool einprogrammiert werden soll, da so die Anzahl an vollständigen Fragebögen erhöht wird. Außerdem sagen sie, dass die Rücklaufquote erhöht wird wenn den Teilnehmern Anreize geboten werden. Forscher sollten bedenken, dass die Probanden durch die Signale aus ihrer Alltagssituation gerissen werden, auch wenn diese gerade mit jemandem sprechen oder arbeiten [31].

6 Zusammenfassung

6.1 Fazit

Morris und Aguilera [26] sagen, dass die Nutzung von Mobiltelefonen für ESM Studien zunehmend an Bedeutung gewinnen. Sie bieten so viele Möglichkeiten an, qualitative und quantitative Daten zu sammeln. Es gibt zahlreiche Methoden, die die Nutzung von mobilen, sozialen und tragbaren Technologien erlauben. Über diverse Sensoren wird unser Leben beeinflusst. Zum Beispiel wenn wir mit dem Auto an einen uns unbekanntem Ort fahren, schalten wir die Navigationsfunktion unseres Smartphones an oder erledigen mit dem Smartphone unsere Einkäufe in Online-Shops. Wir benutzen immer mehr Applikationen in unserem Alltag, die uns bei unseren Aufgaben behilflich sind.

Im psychologischen und gesundheitlichen Bereich finden unterschiedliche Applikationen, die es uns beispielsweise erlauben, über Selbstberichte unsere mentale Gesundheit, unseren erlebten Stress oder unsere Schlafgewohnheiten zu monitoren, immer mehr Beachtung. Heutzutage gibt es auch die Möglichkeit, den Puls oder die gelaufenen Schritte über das Smartphone zu messen⁵. Man muss nur sein Smartphone bei sich tragen und weiß dann am Ende des Tages, wie viele Schritte man an dem Tag gelaufen ist. Die Applikation "S Health" ermöglicht es, sein eigenes Fitnessprotokoll zu erstellen, das die folgenden vier Kategorien beinhaltet: Herzfrequenz, Ernährung, Training und Schlaf. Samsung wirbt damit, dass der Nutzer seine Lebensqualität erhöhen kann.

In naher Zukunft wird das Smartphone immer mehr technische Fortschritte machen und es ermöglichen, dass man immer einfacher Daten von Nutzern sammeln kann. Viele Informationen müssen nicht erfragt werden, da sie mithilfe von maschinellem Lernen dem Smartphone beigebracht werden. Die eingebetteten Sensoren nehmen diese Informationen im Hintergrund auf und leiten sie weiter. Beispielsweise können Aussagen darüber gemacht werden, was der Benutzer gerade macht, wenn sein Smartphone in einer bestimmten Position ist oder ob er sein Smartphone gerade in der Hand hält oder in der Tasche aufbewahrt.

⁵ <http://content.samsung.com/de/contents/aboutn/sHealthIntro.do>

Über die Geschwindigkeit der Tastendrucke auf dem Touchscreen, werden die aktuellen Gefühle des Nutzers gedeutet. Es gibt immer mehr Möglichkeiten Daten automatisch aufzunehmen und deshalb werden immer mehr technische Tools eingesetzt, was ebenfalls in den aktuellen Studien zu beobachten ist.

Die Fragen aus der Einleitung sind leider nicht allgemein beantwortbar, da die Forscher prüfen müssen, welche Form für sie die geeignetste ist. Im Kapitel 5.1 habe ich viele Möglichkeiten beschrieben, eine ESM Studie zu gestalten und durchzuführen und denke, dass man sich an ihnen gut orientieren kann.

Die Experience Sampling Method ist vielleicht nicht die beste Lösung um Daten zu erheben, aber in den passenden Situationen und mit richtigem Einsatz ist sie sehr mächtig und kann gute Ergebnisse liefern.

6.2 Ausblick

Den Probanden werden manchmal, wie beschrieben, höhere Anreize für die Vollständigkeit der Befragungen angeboten, um die Rücklaufquoten zu erhöhen. Aber kann mithilfe der Anreize auch die Validität der Antworten gewährleistet werden? Oder sind die Teilnehmer bereit falsche Antworten zu liefern, nur um die finanziellen Belohnungen abzukassieren? Auf diese Fragen bietet die aktuelle Forschung keine Antworten an. Um solche Fragen zu klären, sollten in diesem Bereich weitere Erhebungen durchgeführt werden.

Literatur

Literatur

1. Abdesslem, F.B., Parris, I. and Henderson, T.N.H. (2010). Mobile experience sampling: Reaching the parts of Facebook other methods cannot reach.
2. Bailes, F.A. (2006). The use of experience-sampling methods to monitor musical imagery in everyday life. *Musicae Scientiae*, 10(2), 173-190.
3. Barrett, L.F. and Barrett, D.J. (2001). An Introduction to Computerized Experience Sampling in Psychology. *Social Science Computer Review*, 19(2), 175-185.
4. Consolvo, S., Harrison, B., Smith, I., Chen, M.Y., Everitt, K., Froehlich, J. and Landay, J.A. (2007). Conducting in situ evaluations for and with ubiquitous computing technologies. *International Journal of Human-Computer Interaction*, 22(1-2), 103-118.
5. Consolvo, S., Smith, I.E., Matthews, T., LaMarca, A., Tabert, J. and Powledge, P. (2005). Location Disclosure to Social Relations: Why, When, and What People Want to Share. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 81-90.
6. Consolvo, S. and Walker, M. (2003). Using the Experience Sampling Method to Evaluate Ubicomp Applications. *PERVASIVEcomputing*, 24-31.
7. Csikszentmihalyi, M. and Hunter, J. (2003). Happiness in everyday life: The uses of experience sampling. *Journal of Happiness Studies*, 4(2), 185-199.
8. Csikszentmihalyi, M. and Larson, R. (1987). Validity and Reliability of the Experience-Sampling Method. *Journal of Nervous and Mental Disease*, 175, 526-536.

9. Csikszentmihalyi, M., Larson, R. and Kubey, R. (1996). Experience Sampling Method Applications to Communication Research Questions. *Journal of Communication*, 46(2), 99-120.
10. Fahrenberg, J., Myrtek, M., Pawlik, K. and Perrez, M. (2007). Ambulatory assessment—Monitoring behavior in daily life settings: A behavioral-scientific challenge for psychology. *European Journal of Psychological Assessment*, 23(4), 206-213.
11. Fisher, C.D. and To, M.L. (2012). Using experience sampling methodology in organizational behavior. *Journal of Organizational Behavior*, 33(7), 865-877.
12. Froehlich, J., Chen, M.Y., Smith, I.E. and Potter, F. (2006). Voting with your feet: An investigative study of the relationship between place visit behavior and preference. *UbiComp 2006: Ubiquitous Computing*, 333-350.
13. Greasley, A.E. and Lamont, A. (2011). Exploring engagement with music in everyday life using experience sampling methodology. *Musicae Scientiae*, 15(1), 45-71.
14. Götz, T., Frenzel, A.C. und Pekrun, R. (2007). Regulation von Langeweile im Unterricht: Was Schülerinnen und Schüler bei der "Windstille der Seele" (nicht) tun. *Unterrichtswissenschaft*, 35(4), 312-333.
15. Hektner, J.M., Schmidt, J.A. and Csikszentmihalyi, M. (2007). Experience Sampling Method, Measuring the quality of everyday life. *Sage*.
16. Hofmann, W., Adriaanse, M., Vohs, K.D. and Baumeister, R.F. (2014). Dieting and the self control of eating in everyday environments: An experience sampling study. *British journal of health psychology*, 19(3), 523-539.
17. Hofmann, W. and Patel, P.V. (2014). SurveySignal A Convenient Solution for Experience Sampling Research Using Participants' Own Smartphones. *Social Science Computer Review*.
18. Hsieh, G., Li, I., Dey, A., Forlizzi, J. and Hudson, S.E. (2008). Using visualizations to increase compliance in experience sampling. *Hsieh, Gary, et al. Using visualizations to increase compliance in experience sampling. "Proceedings of the 10th international conference on Ubiquitous computing*, ACM, 164-167.
19. Hudson, S., Fogarty, J., Atkeson, C., Avrahami, D., Forlizzi, J., Kiesler, S., ... and Yang, J. (2003). Predicting human interruptibility with sensors: a Wizard of Oz feasibility study. *Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM, 257-264.
20. Kahneman, D., Krueger, A.B., Schkade, D.A., Schwarz, N. and Stone, A.A. (2004). A survey method for characterizing daily life experience: The day reconstruction method. *Science*, 306(5702), 1776-1780.
21. Khan, V.J., Markopoulos, P., Eggen, B. and Ijsselstein, W. (2008). Reconexp: A way to reduce the data loss of the Experience Sampling Method. *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, ACM, 471-476.
22. Killingsworth, M.A. and Gilbert, D.T. (2010). A wandering mind is an unhappy mind. *Science*, 330(6006), 932.
23. Krause, A.E., North, A.C. and Hewitt, L.Y. (2013). Music-listening in everyday life: Devices and choice. *Psychology of music*.
24. Lathia, N., Rachuri, K.K., Mascolo, C. and Rentfrow, P.J. (2013). Contextual dissonance: Design bias in sensor-based experience sampling methods. *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, ACM, 183-192.

25. Messiah, A., Grondin, O. and Encrenaz, G. (2011). Factors associated with missing data in an experience sampling investigation of substance use determinants. *Drug and alcohol dependence*, 114(2), 153-158.
26. Morris, M.E. and Aguilera, A. (2012). Mobile, Social, and Wearable Computing and the Evolution of Psychological Practice. *Professional Psychology: Research and Practice*, 43(6), 622-626.
27. Papastefanou, G. (2008). Ambulatorisches Assessment und Empirische Sozialforschung. *soFid Methoden und Instrumente der Sozialwissenschaften*, 2, 11-20.
28. Raento, M., Oulasvirta, A. and Eagle, N. (2009). Smartphones an emerging tool for social scientists. *Sociological methods & research*, 37(3), 426-454.
29. Reis, H.T. and Gable, S.L. (2000). Event-sampling and other methods for studying everyday experience. *Handbook of research methods in social and personality psychology*, 190-222.
30. Schallberger, U., Pfister, R. und Venetz, M. (1999). Qualität des Erlebens in Arbeit und Freizeit. Theoretische Rahmenüberlegungen zum Erlebens-Stichproben-Fragebogen (ESF) und zu den Operationalisierungen. *Arbeitsberichte aus dem Projekt Qualität des Erlebens in Arbeit und Freizeit*.
31. Scollon, C.N., Kim-Prieto, C. and Diener, E. (2003). Experience Sampling: Promises and Pitfalls, Strengths and Weaknesses. *J. Happiness Studies*, 4(1), 5-34.
32. Silvia, P.J., Kwapil, T.R., Eddington, K.M. and Brown, L.H. (2013). Missed Beeps and Missing Data: Dispositional and Situational Predictors of Nonresponse in Experience Sampling Research. *Social Science Computer Review*.
33. Sloboda, J.A., O'Neill, S.A. and Ivaldi, A. (2001). Functions of music in everyday life: An exploratory study using the experience sampling method. *Musicae scientiae*, 5(1), 9-32.
34. Stone, A.A. and Shiffman, S. (1994). Ecological momentary assessment (EMA) in behavioral medicine. *Annals of Behavioral Medicine*, 16(3), 199-202.
35. Venetz, M. (2009). Das momentane (Wohl-) Befinden von Kindern im Schulunterricht-ein in der Heilpädagogik (bis anhin) wenig erforschtes Untersuchungsgebiet. *KSH Mitteilungsblatt*, 22, 17-20.
36. Venetz, M. und Zurbriggen, C. (2011). Wie Schülerinnen und Schüler in integrativen Klassen den Unterrichtsalltag erleben. *Schweizerische Zeitschrift für Heilpädagogik*, Jg. 17, 5-11.
37. Wertli, E. und Audeoud, M. (2010). "Mir geht's gut" Und was heisst das? Forschungsprojekt "Befindensqualität hörgeschädigter Kinder in der Regelschule." *tagungsberichteforum*, 34-43.
38. <http://de.statista.com/statistik/daten/studie/198959/umfrage/anzahl-dersmartphonennutzer-in-deutschland-seit-2010/>. Eingesehen am 30.03.2015.
39. <http://de.slideshare.net/quantifiedself/1-ulrich-qs-experience-sampling-v2>. Eingesehen am 30.03.2015.
40. <http://www.christinecostello.com/projects/beeper.html>. Eingesehen am 30.03.2015.
41. <http://i-literacy.e-learning.imb-uni-augsburg.de/node/1250>. Eingesehen am 30.03.2015.
42. <http://myexperience.sourceforge.net/>. Eingesehen am 30.03.2015.
43. <https://xs.movisens.com/>. Eingesehen am 30.03.2015.
44. <http://surveysignal.com/Default.aspx>. Eingesehen am 30.03.2015.
45. <http://beepme.youexp.at/de/>. Eingesehen am 30.03.2015.
46. <https://metricwire.com/>. Eingesehen am 30.03.2015.

47. <http://surveysignal.com/examplepage.aspx>. Eingesehen am 30.03.2015.
48. <http://content.samsung.com/de/contents/aboutn/sHealthIntro.do>.
Eingesehen am 30.03.2015.