



Studienarbeit

Selbstauskunfts-fähigkeit von Alltags- gegenständen

Verfasser: Joschka Sondhof

Prüfer: Prof. Dr. Thomas Lehmann

Betreuer: Prof. Dr. Kai von Luck
Dr. Susanne Draheim

Eingereicht am Department Informatik der Fakultät Technik und Informatik der Hochschule für Angewandte Wissenschaften Hamburg.

Abgabe am 19.05.2017

Versicherung der Selbstständigkeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, den _____

Inhalt

1. Einführung	3
1.1. Motivation	3
1.2. Problembeschreibung	4
1.3. Zielsetzung und Abgrenzung	5
1.4. Vorgehensweise	5
2. Theoretischer Hintergrund	6
2.1. Ubiquitous Computing	6
2.2. Selbstauskunftsfähigkeit und Kontext	8
2.3. Projektumfeld und vorangegangene Studien	9
3. Anforderungsanalyse	11
4. Konzeption	15
4.1. Vorüberlegungen zur Hardwareauswahl	16
4.2. Systemabbildung in der Software	19
5. Realisation	22
5.1. Erfahrungen aus dem Entwicklungsprozess der Prototypen	22
5.2. Beschreibung und Visualisierung des abschließenden Projektstands	24
5.3. Testen	26
6. Auswertung und Erkenntnisse	27
7. Ausblick	29
8. Quellen	31
Appendix A	34
Appendix B	38
Appendix C	40

Abbildungs- und Tabellenverzeichnis

Abbildung 1: Forschungsumfeld mit Fokus auf Augmented Everyday Objects	10
Abbildung 2: Abstraktionsebenen des Systems und Zusammenhänge zum Objektmodell	20
Abbildung 3: Klassendiagramm des Gesamtsystems (mateMugSys)	21
Abbildung 4: Querschnitt durch den Tassenfuß	25
Abbildung 5: Hülle des konstruierten Tassenfußes	25
Abbildung 6: Komponentenschichten	25
Abbildung 7: Beispielcode zur Nutzung der MateMug Bibliothek	26
Abbildung 8: Mindmap zur Aufdeckung von Anwendungsszenarien	34
Abbildung 9: Komponentenauswahl anhand von Szenarien und Kriterien (Sensoren)	35
Abbildung 10: Komponentenauswahl anhand von Szenarien und Kriterien (Konnektoren)	36

Abbildung 11: Komponentenauswahl anhand von Szenarien und Kriterien (Aktoren).....	37
Abbildung 12: Verdrahtungs- und Lageschema	38
Abbildung 13: Pinbelegung, Komponentenbezeichnungen, Kostenübersicht.....	39
Tabelle 1: Anforderungen auf der Projekt-Ebene	14
Tabelle 2: Anforderungen auf der Hardware-Ebene.....	14
Tabelle 3: Anforderungen auf der Software-Ebene	15

Kurzfassung

In Zukunft wird der Mensch zunehmend von versteckten Computern umgeben sein, die ihn und sein Umfeld erfassen. Diese These liegt Forschungsfeldern wie dem *Ubiquitous Computing* oder der Auseinandersetzung mit *Augmented Everyday Objects* zugrunde. Dabei können die für die Computerisierung verwendeten Gegenstände einerseits als aktiv genutzte Begleiter dienen, andererseits eine digitale Repräsentation des Menschen erzeugen. Die technologische Machbarkeit solcher Anwendungen kann bereits in vielen Fällen bewiesen werden – doch die Reaktion des Menschen auf das veränderte Lebensumfeld bleibt ungeklärt. Daher widmet sich diese Studienarbeit der Fragestellung, wie die Wirkung der versteckten Computerisierung beobachtet werden kann. Zu diesem Zweck wird ein selbstauskunftsfähiges Alltagsobjekt, eine kontext-sensitive Tasse, geschaffen. In den theoretischen Vorüberlegungen wird untersucht, was unter der Erfassung des Kontexts und der Situation verstanden werden kann. Es resultieren sowohl abstrakte als auch konkrete Anwendungsszenarien, aus denen Anforderungen an einen Prototyp abgeleitet werden. In der anschließenden Konzeptionsphase wird ein System umrissen, welches einerseits nach kurzer Zeit für erste Erkenntnisse genutzt werden kann und das andererseits Chancen für zukünftige Weiterentwicklungen bietet. Das Resultat der iterativen Umsetzung von drei Prototypen ist ein funktionsfähiges System auf Arduino Basis mit Sensoren für Bewegung, Temperatur, Druck, Gewicht, Licht und Geräusche, mit zwei Aktoren sowie mit der Unterstützung für Verbindungen über WLAN, Bluetooth und RFID. Damit liefert diese Arbeit die Grundlage für die Untersuchung der sozialen und gesellschaftlichen Folgen der Digitalisierung, der Auswirkungen auf die Privatsphäre oder der Sicherheitsaspekte aus informationstechnischer Sicht. Daneben sind vielfältige Weiterentwicklungen wie die Einbindung in die Virtuelle oder Erweiterte Realität oder die Datenfusion mit anderen vernetzten Gegenständen zur Kontexterfassung denkbar.

Stichworte

Ubiquitous Computing, Augmented Everyday Objects, IoT, Selbstauskunftsfähigkeit

1. Einführung

Diese Studienarbeit ist in enger Zusammenarbeit mit dem Creative Space For Technical Innovations (CSTI) der HAW Hamburg entstanden, einer Plattform, die nach neuen Ansätzen und Möglichkeiten in den Bereichen Mensch-Maschine-Interaktion sowie *Smart Systems* sucht. Die thematische Nähe zu derzeit populären Themen wie *Smart Home* zeigt sowohl die Orientierung am Zeitgeist als auch das Bestreben auf, gesellschaftlich und nicht zuletzt wirtschaftlich hoch relevanten Fragestellungen wissenschaftlich zu begegnen. Wie leben Menschen in Umgebungen, die zunehmend von unsichtbaren, digitalen Objekten durchdrungen und in eine virtuelle Parallelwelt überführt werden? Ein solcher Gedankengang liegt der vorliegenden Arbeit zu Grunde, in der zur Beantwortung dieser und ähnlicher Fragen ein Prototyp eines digital angereicherten Objektes realisiert wird.

1.1. Motivation

Mit der fortschreitenden Entwicklung leistungsfähiger Mikrocontroller, miniaturisierter Sensoren und kostengünstiger Herstellungsverfahren verschwinden Computer allein wegen ihrer abnehmenden Größe aus der bewussten Wahrnehmung des Menschen. Neue Anwendungen auf Basis vernetzter Gegenstände (*Internet of Things*, IoT) sollen zu einer Vereinfachung der Mensch-Maschine-Interaktion sowie Verbesserungen alltäglicher Lebensabläufe führen. Neben Entwicklungen in den Bereichen Infrastruktur (z.B. Gebäudeautomation), Energiemanagement (z.B. intelligente Heizungen) oder Medien (z.B. drahtlose Entertainment Systeme) kommen auch Gegenstände in den Fokus der Forschung, die den Menschen über den Großteil des Tages begleiten, obwohl sie keine allzu bedeutsame Aufgabe in der realen Welt erfüllen. Kommerzielle Produkte wie Smartwatches oder Aktivitätstracker greifen einen Teil dieser Überlegungen auf, wenn auch hier eher die bewusste Nutzung des Gegenstandes als Werkzeug im Vordergrund steht. Mit dem Begriff *Ubiquitous Computing* (kurz *UbiComp*) wird dagegen der Ansatz bezeichnet, dass der Mensch gar nicht selbst mit dem Gegenstand interagiert. Stattdessen entsteht durch die Kommunikation zwischen vielen, unterschiedlichen Objekten und deren Fähigkeit, den eigenen Kontext wahrzunehmen, ein neues Lebensumfeld [WEISER-91]. Solche Gegenstände können zutreffend als *Augmented Everyday Objects* bezeichnet werden. Obwohl in den letzten Jahren neben den zuvor genannten kommerziellen Produkten auch vermehrt wissenschaftliche Auseinandersetzungen in diesem Themenumfeld veröffentlicht wurden, verbleiben noch viele Fragestellungen in Bezug auf die Wirkung der nachweislich umsetzbaren Technologien offen.

Mit dieser Studienarbeit soll ein physischer Gegenstand geschaffen werden, der das Sammeln von Erfahrungen in vernetzten Umgebungen vereinfacht, und dadurch den Grundstein zukünftiger Studien legt. Der Fokus liegt dabei weniger auf den Möglichkeiten, die die heutigen Technologien schaffen, sondern vielmehr darauf, welche Auswirkungen auf den Menschen resultieren. Nichtsdestotrotz beeinflussen natürlich die Anwendungen deren Auswirkungen – aus diesem Grund erfolgt insbesondere im analytischen Teil eine grundlegende Auseinandersetzung mit denkbaren Nutzungsszenarien vernetzter Gegenstände.

1.2. Problembeschreibung

Nicht zuletzt aufgrund der hohen Geschwindigkeit, mit der neue Technologien entwickelt und daraus hervorgehende Produkte am Markt platziert werden, ist eine fundierte Aussage darüber, welche Veränderungen diese Entwicklungen mit sich bringen, äußerst relevant. Die reine technische Machbarkeit droht an vielen Stellen die tatsächliche Notwendigkeit der Produkte in Frage zu stellen oder gänzlich obsolet erscheinen zu lassen. Diese Diskrepanz wird allerdings die fortschreitende technologische Entwicklung kaum bremsen, sodass letztlich nicht fraglich ist, ob etwas in Zukunft möglich sein wird, sondern nur wann. Um nachzuvollziehen, wie sich der Mensch in das resultierende Lebensumfeld integriert, können in Institutionen wie dem CSTI schon heute entsprechende Erfahrungswelten erschaffen werden. Dabei stellt sich zunächst die Frage, welches Objekt im Sinne der zuvor genannten Kontexterfassung für die Suche nach Erkenntnissen geeignet ist. Inspiriert durch ein bereits einige Jahre zurückliegendes Projekt der Forschungsgruppe des *Telecooperation Office (TecO)* am Karlsruher Institut für Technologie, erscheint eine Tasse als täglicher Begleiter des Menschen insbesondere im Arbeitsumfeld reizvoll. Der vom *TecO* entwickelte *MediaCup* war mit Sensoren, Kommunikationsschnittstellen und einem integrierten Mikrocomputer ausgerüstet und schuf eine Grundlage zur Beobachtung verschiedener Szenarien vernetzter Objekte [BEIGL-01]. Die seit der Veröffentlichung des *MediaCups* stark gewandelten technologischen Bedingungen und Arbeits- bzw. Lebensumgebungen wie z.B. die wachsende Infrastruktur für Vernetzung und Auswertung großer Datenmengen erfordern allerdings eine Neukonzeption, in der aktuelle Hardware genutzt und die Tasse in ein bestehendes System eingebunden wird. Die zentrale Anforderung an die Tasse ist die Selbstauskunft, also die zuvor erwähnte Fähigkeit, den eigenen Kontext wahrzunehmen und mitzuteilen.

Daraus entstehen folgende Forschungsfragen:

- Wie ist die konkrete Ausgestaltung der Selbstauskunftsfähigkeit? In welchem Zusammenhang steht diese zu den Begriffen Kontext und Situation?
- Welche Anwendungsszenarien sollen berücksichtigt werden?
- Welcher Aufbau und welche Funktionalitäten der Tasse ergeben sich hieraus?
- Welche modernen Technologien können eingesetzt werden?

Dabei liegt der Fokus auf der Entwicklung des physikalischen Prototyps, der eine Interaktion zwischen Tasse(n) und Mensch, sowie die Integration in die Infrastruktur des CSTI ermöglicht.

1.3. Zielsetzung und Abgrenzung

Aus der zuvor erläuterten Notwendigkeit eines *Augmented Everyday Objects* für einen Erkenntnisgewinn in Bezug auf die zukünftigen Auswirkungen heutiger Technologien geht das Hauptziel dieser Studienarbeit hervor: ein solches Objekt bereitzustellen. Dabei ist es im Sinne des *UbiComp* wichtig, die gewohnte Form und Verwendung des Objektes, in diesem Fall einer Tasse, nicht übermäßig dahingehend zu verändern, dass die Digitalisierung in die bewusste Wahrnehmung des Menschen rückt. Ebenfalls ist es zielführend, die Nutzung der Tasse für den Forscher möglichst einfach zu gestalten, nicht zuletzt um die Barrieren für aufbauende Studien zu verringern. Im Hinblick auf den zeitlichen und fachlichen Rahmen dieser Arbeit muss auf die tatsächliche Anwendung der Tasse zur Abbildung komplexerer Szenarien sowie auf umfangreiche Verifizierung der aufgenommenen Daten verzichtet werden. Dies kann Gegenstand aufbauender Studien sein.

1.4. Vorgehensweise

Vorbereitend zur eigentlichen Analyse und Konzeption des Systems müssen einige grundlegende Fragen wie etwa die theoretischen Hintergründe von *UbiComp* geklärt werden. Zur Verdeutlichung, wie die unterschiedlichen Anwendungen derselben Technologie in Bezug auf den Nutzen für den Menschen und die bewusste oder unbewusste Wahrnehmung untergliedert werden können, soll in diesem Zuge auch eine Gegenüberstellung des Ansatzes der *Digital Persona* und jenem eines aktiven Begleiters vollzogen werden. Ebenfalls erscheint die Erläuterung des Zusammenhangs bzw. der Unterschiede zwischen Kontext und Situation für die späteren

Schlussfolgerungen notwendig. Diese Grundlagen werden in Kapitel 2 erörtert. Im anschließenden dritten Kapitel können die daraus abzuleitenden technischen Anforderungen an das in der Zielstellung umrissene System geklärt werden. Hilfreich dabei ist die Bewertung denkbarer Szenarien mit Hilfe von Kriterien, sodass daraus die zu erfassenden Daten ersichtlich werden. Um den Aufwand der in Kapitel 4 beschriebenen Realisation zu minimieren, sollen iterativ mehrere Prototypen mit jeweils zunehmendem Funktionsumfang entwickelt werden – auf diese Weise können kritische Faktoren bei der Umsetzung frühzeitig identifiziert und gelöst werden. Die wichtigsten Erkenntnisse der einzelnen Entwicklungsschritte sind zusammen mit der Beschreibung der Prototypen in Kapitel 5 zu finden. Während der Projektlaufzeit konnten insgesamt drei Prototypen bzw. Entwicklungsstufen umgesetzt werden, wobei der letzte Stand die geforderten Merkmale in hohem Maße (dazu Kapitel 6) erfüllt. Das abschließende, siebte Kapitel bietet einen Ausblick auf die offen gebliebenen Fragen, ausbleibende Erweiterungen und mögliche Weiterentwicklungen.

2. Theoretischer Hintergrund

Um neue Kenntnisse über die Wirkung versteckter Computer auf den Menschen gewinnen zu können, ist ein Rückblick auf vorangegangene Überlegungen sowie eine grobe Übersicht daraus resultierender Umsetzungen unabdingbar. So können unterschiedliche Perspektiven auf die Problemstellung gefunden und hilfreiche Erkenntnisse in die Beantwortung der Eingangsfragen eingebunden werden. Der zuvor erwähnte Ansatz des *Ubiquitous Computing* spielt dabei eine zentrale Rolle, weil daraus sowohl technische Anforderungen als auch ein wissenschaftlicher Fokus gewonnen werden können. Demzufolge beschäftigt sich der nächste Abschnitt mit den Kernaussagen des *UbiComps* und verwandten Forschungsideen.

2.1. Ubiquitous Computing

Zunächst ist wichtig festzustellen, dass neben dem hier präferierte Begriff *Ubiquitous Computing* in den Veröffentlichungen zum zugrundeliegenden Thema viele weitere Bezeichnungen für dieselben Aussagen existieren. So ist in der Englischsprachigen Literatur parallel häufig von *Ambient Intelligence*, *Pervasive Computing* oder *Intelligent Environments* die Rede; ebenso werden als Synonym für *Augmented Everyday Objects* die Begrifflichkeiten *Everyware* oder IoT verwendet.

Daher sollte herausgestellt werden, welche Ausprägungen des breiten Themenfeldes in der vorliegenden Arbeit gemeint sind. Die erste Nennung von *UbiComp* im Zusammenhang mit computerisierten Gegenständen geht auf Mark Weiser und dessen Forschungen am Xerox Palo Alto Research Center zurück. In der viel zitierten Ausarbeitung *The Computer for the 21st century* wird der wesentliche Unterschied – auch bezüglich der Forschungsmotivation – zu einer Weiterentwicklung herkömmlicher Personal Computer verdeutlicht: Indem der Computer in den Hintergrund rückt, entstehen anstatt von neuen Anforderungen an den Menschen neue Möglichkeiten, ihn (nicht zuletzt kognitiv) zu entlasten. Schon 1991 wird von einer wachsenden Informationsflut und der Überforderung gesprochen, sich mit der unnatürlich erscheinenden Welt der Computer auseinander setzen zu müssen. Als Gegenentwurf solle sich der Computer auf eine natürliche Weise in das Lebensumfeld des Menschen eingliedern, mit dem Ziel, wieder die direkte Interaktion zwischen Menschen ohne vermittelnde Ein- und Ausgabegeräte in den Fokus zu heben [WEISER-91, S. 104]. Heute haben Smartphones und das mobile Internet den Zugriff auf Daten so stark vereinfacht, dass kaum noch eine Trennung zwischen der realen und digitalen Welt möglich ist. Mehr denn je erscheint der Mensch auf die Hilfe angewiesen, die *UbiComp* beim Umgang mit der digitalen Dimension in Aussicht stellt. Zugleich wird an einigen Stellen auf die Gefahren hingewiesen, die mit einer unbedachten Übergabe von Verantwortung an die Technologie anstelle einer wohl überlegten Umverteilung einhergehen können. Bei offensichtlicheren technischen Anwendungen wie etwa der voranschreitenden Automatisierung von Produktionsprozessen in der Massenfertigung liegen solche Gedanken z.B. in Bezug auf den Verbleib der Beschäftigten noch Nahe; doch auch die versteckte Digitalisierung könnte zu ähnlichen Problemen führen, indem der Mensch scheinbar an Bedeutung verliert und keine neuen Aufgaben erhält. [KUNIAVSKY-03] hebt in diesem Zusammenhang die Verantwortung des Gestalters neuer Anwendungen hervor, dass diese Menschen zu Neuem befähigen anstatt ihnen zu schaden. Letztlich gilt dies sowohl für Anwendungen, die in die bewusste Wahrnehmung des Menschen treten, als auch jene nach Ansätzen des *UbiComp* entwickelten Objekte, die zu einer unbewussten Überführung des Menschen ins Digitale dienen. Der von [CLARKE-94] verwendete Begriff der *Digital Persona* fasst die Funktion, eine digitale Repräsentanz des Menschen zu erzeugen, passend zusammen. Das Besondere an der *Digital Persona* ist ihr Streben nach Vollständigkeit: Um Interaktion zwischen Menschen etwa in virtueller Realität glaubhaft zu machen, ist die möglichst lückenlose Erfassung des Menschen und seiner Umgebung maßgeblich. Daher unterscheidet sich der Ansatz, versteckte Computer in den Alltag des Menschen einzubinden, wesentlich von der Erschaffung eines digitalen Begleiters hinsichtlich seiner Auswirkungen. Ein Begleiter reichert das Lebensumfeld des Menschen in der realen Welt an, die *Digital Persona* digitalisiert den Menschen an sich. Deswegen sollen die hier entwickelten Prototypen insbesondere

zur Untersuchung der Reaktion des Menschen und möglicher negativer Folgen dienen. Diese Erkenntnisse in Erinnerung behaltend, muss als Nächstes geklärt werden, welchen Umfang die *Digital Persona* bzw. der für ihre Erschaffung zu erfassende Kontext aufweist.

2.2. Selbstauskunfts-fähigkeit und Kontext

Die vorangegangene Darlegung hat die Chancen aufgezeigt, die sich mit einer digitalen Repräsentation des Menschen ohne die Notwendigkeit von dessen Interaktion ergeben. Die zu diesem Zweck genutzten Objekte müssen in der Lage sein, zumindest einen Teil ihrer Umgebung wahrzunehmen und proaktiv der Infrastruktur zur Verfügung zu stellen – sie müssen selbstauskunfts-fähig sein. Die Forderung eines eigenständigen Kommunikationsverhaltens zeigt einen wichtigen Aspekt auf: Wenn die Objekte selbst in der Verantwortung sind, nur die jeweils relevanten Informationen preiszugeben, kann ein Teil der Rechenleistung in das Objekt verlagert werden. Folglich ist es durchaus möglich, dass die kommunizierten Informationen dynamisch angepasst werden, nicht zuletzt, um das Datenaufkommen in der vernetzten Umgebung zu reduzieren. Dem gegenüber stehen allerdings Limitationen der Hardware oder Herausforderungen bei der Konzeption dynamischer Schnittstellen. In jedem Fall können (in der Theorie) aus der Verknüpfung der bereitgestellten Daten aller Objekte die Gesamtsituation bestimmt und angemessene Handlungen initiiert werden. Auf die konkrete Problemstellung bezogen ist allerdings zu klären, welchen Teil des Umfelds eines Menschen eine Tasse überhaupt wahrnehmen kann bzw. was unter dem Tassenkontext zu verstehen ist. Natürlich kann ausgehend von verfügbaren Sensoren, von physikalisch messbaren Größen argumentiert und der Kontext als Summe der Messergebnisse aufgefasst werden. Doch wie zuvor angedeutet ist Verfügbarkeit nicht selbstverständlich gleichbedeutend mit Relevanz. Generell lässt sich Kontext zum einen als ein Repräsentationsproblem darstellen, bei dem es um eine korrekte Erfassung von Ort, Zeit, Identität, Umgebung und den Zuständen anderer Objekte oder Menschen geht. Dies impliziert sowohl den Informationscharakter von Kontext, als auch seine Definierbarkeit und Stabilität. Anders formuliert kann demnach anhand von Anforderungen im Vorhinein festgelegt werden, was im Betrieb zu erfassen ist. Andererseits ist eine auf die Interaktion zwischen Objekten und Menschen fokussierte Perspektive auf Kontext denkbar. Erst eine solche Interaktion erzeugt eine bestimmte Situation – und die jeweils relevanten Informationen sind an sie gebunden. Die Unvorhersehbarkeit von Interaktionen verhindert eine starre Definition von Kontext *ex ante* [DOURISH-04, S. 4ff]. Stattdessen sollte eine fortlaufende Relevanzbewertung der theoretisch verfügbaren Informationen in Bezug auf die Handlung vollzogen werden, in der sie erfasst werden. Daraus folgt dann nicht nur eine dynamische Erfassung sondern ferner eine auf die Situation abgestimmte

Selbstauskunft. Anstatt einer im Vorhinein festgelegten, konstruierten Datenarchitektur könnte also ein anpassungsfähiges Wahrnehmungskonzept zu einer zutreffenderen Auffassung von situativem Kontext führen. Neben den Auswirkungen auf die Qualität der Datenerfassung ist aus technischer Sicht aber fraglich, ob mit einer derartigen Anpassungsfähigkeit die tatsächliche Rechenleistung und (Energie-)Effizienz des Systems verbessert werden kann oder ob dafür im Gegenteil sogar umfangreichere Ressourcen notwendig sind. Auch lässt sich insbesondere im Hinblick auf den zeitlichen Rahmen des hier beschriebenen Projektes ein Kompromiss zwischen dynamischem Verhalten des Systems und aufwandsschonender Festlegung von Systemlimitationen nicht vermeiden. Ohnehin kann durch eine geschickte Verknüpfung statischer Daten durchaus auch in einem fixierten System noch eine gewisse Anpassung an die Situation erreicht werden. Zusammenfassend sollte bei der Entwicklung des Systems bzw. bei der Formulierung von Anforderungen neben einer systematisch-technischen Perspektive auch die soziologische und interaktive Dimension beachtet werden. Durch die Betrachtung des menschlichen Verhaltens im Zusammenhang mit dem technisch angereicherten Objekt können viele Hinweise auf relevante Informationen und neue Nutzungsszenarien gewonnen werden. Diese Szenarien können zwar keine Sicherheit geben, die zur Erfahrungssammlung untersuchten Interaktionen zwischen dem Menschen und seinem digitalisierten Umfeld stets korrekt zu erfassen oder ausschließlich relevante Informationen bereitzustellen. Dennoch können sie genutzt werden, um die nicht-technischen Überlegungen durch die Ermittlung einer gemeinsamen Hardwarebasis in technische Anforderungen zu übersetzen. Zu diesem Zweck werden in Kapitel 3 zunächst denkbare Einsatzszenarien der Tasse gesucht, aus denen mit Hilfe von Kriterien eine umsetzbare Auswahl bestimmt wird. Anschließend können die Anforderungen an das sich ergebende Gesamtsystem ausformuliert werden. Zuvor wird jedoch eine kurze Übersicht bisheriger Studien sowie kommerzieller Produkte mit Ähnlichkeit zu diesem Projekt gegeben, nicht zuletzt weil daraus ebenfalls Szenarien abzuleiten sind.

2.3. Projektumfeld und vorangegangene Studien

Wie schon erwähnt, nahm der *MediaCup* des *TeCo* gewissermaßen eine Pionierrolle hinsichtlich der Forschung mit *Augmented Everyday Objects* ein. In späteren Studien, seien es Folgeprojekte desselben Teams oder Ansätze anderer Institutionen, werden häufig Teilaspekte wieder aufgegriffen oder die Entwicklung in eine andere Richtung fortgesetzt. Als Methode, aus der unüberschaubaren Menge an Ausarbeitungen zum Thema *UbiComp* solche mit sinnvollem Bezug zur

vorliegenden Arbeit herauszufiltern, wurden daher sozusagen die Verästelungen des durch den *MediaCup* begründeten Forschungsbaumes untersucht (siehe Abbildung 1).

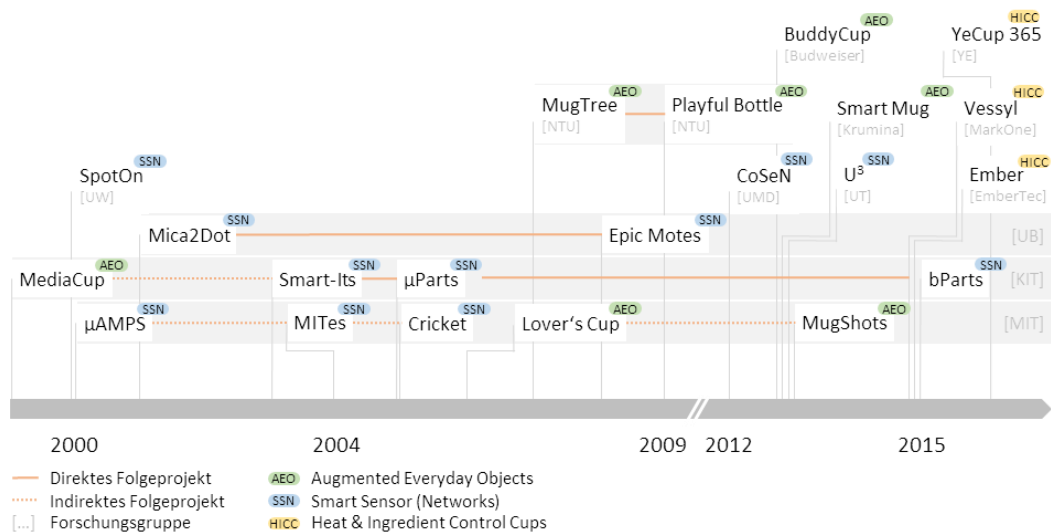


Abbildung 1: Forschungsumfeld mit Fokus auf Augmented Everyday Objects

Es ergeben sich im Wesentlichen vier Schwerpunkte im Projektumfeld: Studien zu *Augmented Everyday Objects* (AEO) im Sinne der Ausführungen im ersten Abschnitt diesen Kapitels, zu *Smart Sensors* bzw. *Sensor Networks* (SSN) sowie zu *Heat & Ingredient Control Cups* (HICC). Hinzu kommen in den letzten Jahren verschiedene kommerzielle Produkte, bei denen hauptsächlich ästhetische oder designbezogene Merkmale im Vordergrund stehen und die hier unberücksichtigt bleiben sollen. Interessant ist festzustellen, dass der Mehrzahl an AEO Projekten keine Weiterentwicklungen an derselben Institution folgen. Auch zielen viele Projekte lediglich darauf ab, die Machbarkeit eines bestimmten Szenarios zu beweisen; abstrakte Fragestellungen zu den Auswirkungen der versteckten Computerisierung gehören nicht zu ihrem Umfang. Auf experimenteller Ebene wurden Trinkgefäße mit Zusatzfunktionen ausgestattet, um isolierte Nutzungsszenarien zu erproben: Sei es die Signalisierung eines Trinkvorgangs bei einem gekoppelten Gefäß [CHUNG-06], die kameragestützte Erfassung des Füllstands einer Tasse [KO-07, CHIU-09] oder die Anzeige personalisierter Bilder auf einem integrierten Bildschirm [KAO-14]. Dies unterstreicht die Relevanz weiterer Forschung und den Bedarf einer Plattform, auf der unterschiedliche Szenarien untersucht werden können. Mit einer solchen Plattform sind zwar auch stärker auf großflächige oder industrielle Anwendungen ausgerichtete Projekte denkbar, die Forschungsergebnisse im Bereich der SSN bieten hier jedoch wesentlich spezialisiertere und nicht zuletzt technisch ausgereifere Lösungen [BEIGL-03, BEIGL-05, BEIGL-01, BERKELEY-01, TAPIA-04, NISSANKA-04, KAWAHARA-03]. Für Szenarien eines aktiven Begleiters sind vor allem die oftmals kommerziell

entwickelten HICC interessant (*Ember, Vessyl, YeCup 365*). Funktionen wie die Erwärmung von Flüssigkeiten oder die Bestimmung ihrer Zusammensetzung scheinen allerdings selbst bei oberflächlicher Betrachtung der technischen Machbarkeit die Grenzen dieser Arbeit zu überschreiten. Zusammenfassend können dem Projektumfeld einige konkrete Anwendungen einer computerisierten Tasse entnommen werden, die im nächsten Abschnitt gemeinsam mit neu erarbeiteten Szenarien zur Definition von Anforderungen dienen.

3. Anforderungsanalyse

Der theoretische Rahmen sowie die Orientierung an themenverwandten Studien werfen viele Fragestellungen bzw. spezifische Anwendungsszenarien auf. Darüber hinaus konnten in Gesprächen im Projektumfeld weitere Möglichkeiten eruiert werden, den Wirkungen einer auskunftsfähigen Tasse nachzugehen. Eine gängige Methode, die verschiedenen Ansätze zu notieren und zugleich neue Ideen zu generieren, ist das Mindmapping. Sie ist in gewisser Hinsicht ein qualitatives Mittel, da sie nur einen Ausschnitt aller potenziell realisierbaren Anwendungen aufdecken kann, abhängig vom Personenkreis. Andererseits kann deutlich schneller als bei anderen Kreativmethoden eine hohe Quantität unterschiedlicher Ideen gesammelt werden. Das Resultat (dargestellt im Appendix A, S. 34) zeigt deutliche Parallelen zur zuvor geführten Diskussion über den Fokus der Nachforschungen: Zum einen wurden viele Szenarien eines helfenden Begleiters erwähnt wie etwa der Einsatz der Tasse als Navigationsgerät durch die Bestimmung des eigenen Aufenthaltsortes, als Terminassistent verbunden mit dem Smartphone oder als Vermittler zu anderen intelligenten Geräten wie Kaffeemaschinen. Zum anderen entstanden Ideen dazu, welche Mehrwerte bestimmte Informationen aus der direkten Umgebung der Tasse oder über sie selbst erbringen könnten. Im CSTI werden grundsätzlich Forschungen in allen Lebensbereichen des Menschen umgesetzt, in Bezug auf die Tasse wurde allerdings ein leichter Fokus auf das Arbeitsumfeld gelegt. Hinzu kam durch die enge Verzahnung mit dem Living Space, dem Labor für Forschungen im Bereich *Smart Home* an der HAW Hamburg, die Berücksichtigung des Wohnumfeldes. Als Schlussfolgerung lassen sich die erzeugten Ideen den beiden Polen des aktiven Begleiters oder des stillen Objektes zur Erzeugung einer *Digital Persona* zuordnen und zugleich in einem Spektrum zwischen Arbeiten und Wohnen verorten.

Aus den aufgedeckten Szenarien können nun erste Schätzungen über die jeweils zur Umsetzung benötigten Hardwarekomponenten abgeleitet werden, zu dem Zweck, anhand von Gemeinsam-

keiten und Überschneidungen die Menge an Möglichkeiten auf ein vernünftiges Maß zu reduzieren. Im Hinblick auf die spätere Systemkonzeption bietet sich eine Trennung zwischen Sensoren, Aktoren und Komponenten zur Erzeugung von Konnektivität mit anderen Systemen, im Folgenden Konnektoren genannt, an. Als Basisanforderung soll die Tasse in die bestehende Infrastruktur des CSTI integrierbar sein sowie mit anderen Tassen interagieren (vorzugsweise über etablierte Standards). Naheliegend sind hier also Konnektoren auf WLAN und Bluetooth Basis. Daneben soll eine Auskunft über den Füllstand und die Temperatur der eingefüllten Flüssigkeit, die Position im Raum, Bewegungen des Anwenders mit der Tasse sowie die Anwesenheit anderer Tassen möglich sein. An dieser Stelle sei vorgegriffen, dass die Unterstützung von Indoor Positioning aufgrund der zum aktuell noch unzureichend entwickelten Technologie im Rahmen dieser Arbeit nur begrenzt umsetzbar ist. Hohe Genauigkeiten sind hier nach derzeitigem Forschungsstand nur mit aufwändigen, zusätzlich zu installierenden Geräten erreichbar und vervielfachen die Komplexität sowohl auf der Hardware- als auch der Software-Ebene [BARTELS-17]. Daher sollen zwar – sofern sinnvoll und realisierbar – alle für eine Lokalisierung nutzbaren Sensoren eingebunden werden, die Datenfusion und softwareseitige Verarbeitung liegt jedoch außerhalb des Projektumfangs. Nicht zuletzt um die Erfahrungstiefe mit der Tasse im Vergleich zu vorherigen Studien zu erweitern, sollen neben den grundlegenden Anforderungen weitere Szenarien auf eine gemeinsame Hardwarestruktur und die Umsetzbarkeit geprüft werden. In Absprache mit den Projektbetreuern soll für den ersten Prototyp stärker das Konzept eines passiven, für interessante Aussagen über den Anwender oder sein Verhalten nutzbaren Objektes verfolgt werden. Für spätere Prototypen können zusätzliche Anwendungsfälle vorgemerkt werden; wiederum andere werden ausgeschlossen. Aus der Gegenüberstellung der verschiedenen Szenarien können zunächst Mehrfachverwendungen derselben Sensoren aufgedeckt und Gruppen gebildet werden. Danach muss eine Bewertung des (geschätzten) Aufwands einer Integration in das Gesamtsystem erfolgen, vor allem hinsichtlich der zukünftigen Erweiterbarkeit und der Verfügbarkeit der elektronischen Komponenten im CSTI. So kann eine schnelle und kostenschonende Prototypenentwicklung begünstigt werden. Der vollständigen Tabelle in Appendix A auf den Seiten 35ff ist zu entnehmen, dass die meisten der Szenarien einen Sensor zu Erfassung von Bewegung (IMU) und des Gewichts erfordern, Bluetooth und WLAN nutzen und ggf. Signale über LEDs geben würden. Mit diesen Komponenten ließen sich der Einschätzung nach die aus den Basisanforderungen resultierenden Szenarien umsetzen. Daneben lassen weitere Szenarien die Messung der Temperatur, der Helligkeit (am Boden der Tasse) und der Geräusche in der Umgebung als sinnvoll erscheinen. Die Konnektoren RFID und NFC sind vor allem für die Positionserkennung interessant. Als zusätzlicher Akteur könnte ein Vibrationsmotor eingesetzt werden. Diese Komponenten sind zur Erfüllung der meisten in den Gesprächen mit den Projektbetreuern

aus dem CSTI genannten Anforderungen nicht zwingend notwendig – dennoch ist eine Unterstützung darüber hinausgehender Szenarien für zukünftige Entwicklungen reizvoll. Da bei der Verwendung von I²C-fähigen Komponenten der Mehraufwand überschaubar bleibt, besteht vorerst der Anspruch, auch die genannten Sensoren in das System zu integrieren. Zudem soll mit zwei unterschiedlichen Sensortypen für die Temperaturmessung experimentiert werden: sowohl ein herkömmlicher Sensor als auch ein digitaler Infrarotsensor werden verbaut. Hintergrund ist hier schlicht der persönliche Erfahrungsgewinn und die Möglichkeit, in späteren Systemen eine fundierte Auswahl aus den redundanten Sensoren zu treffen. Unvermeidbar bleibt an einigen Stellen eine Unsicherheit über die zu erwartenden Aufwände oder den Mehrwert, ein weiteres Szenario einzubinden, bestehen. Entscheidendes Kriterium ist in diesen Fällen letztlich die Hardwareverfügbarkeit oder die Tendenz, mit Blick auf zukünftige Weiterentwicklungen eher eine umfangreichere Lösung einer nicht einschätzbaren Limitation vorzuziehen. Auch kann bei der nun folgenden Festlegung von Anforderungen durch eine Priorisierung der Freiheitsgrad verdeutlicht werden, aufgrund von gesammelten Erfahrungen im Zuge der iterativen Realisation von Prototypen zu einem späteren Zeitpunkt Teile des Systems zu ändern oder wegfallen zu lassen. Insbesondere die Wahl einer geeigneten Datenverarbeitung (Mikrocontroller) hat wesentliche Auswirkungen auf die Umsetzbarkeit der formulierten Anforderungen. Nichtsdestotrotz sollen vorerst alle aus jetziger Sicht sinnvollen Anforderungen definiert werden, um bei der Entwicklung der Prototypen zielgerichtet vorgehen und die jeweiligen Aufwände einschätzen zu können. Es lassen sich drei Ebenen an Anforderungen aufgliedern: auf Projektebene, in Bezug auf die Hardware sowie softwareseitig. Ersteres beschreibt im Wesentlichen die Projektumgebung, insbesondere den Zeithorizont, das Budget und die Rahmenbedingungen des Arbeitsprozesses. Die beiden untergeordneten Ebenen sind dagegen auf das Entwicklungsziel bezogen und legen bereits erste Merkmale der Lösung fest (wobei die konkrete Lösung nicht Teil der Definition von Anforderungen sein darf). Die folgenden Tabellen Tabelle 1, Tabelle 2 und Tabelle 3 stellen die aus den zuvor genannten Quellen abgeleiteten Anforderungen dar.

Anforderungsanalyse

<i>ID</i> <i>Bezeichnung</i>	<i>Kurzbeschreibung</i>	<i>Priorität</i>
P-001 TIME 1 Zeitraumen	Als Laufzeit des Projektes sind sechs Monate angesetzt. Verlängerungen dieses Zeitraumes erfordern eine gesonderte Absprache.	<i>mittel</i>
P-002 BUDG 1 Budget	Ziel des Projektes ist die kostengünstige Entwicklung eines Prototypen. Das Budget sollte daher 100 € nicht überschreiten.	<i>niedrig</i>
P-003 DOCU 1 Dokumentation	Jeder Entwicklungsschritt ist nachvollziehbar zu dokumentieren und nach Projektende im Rahmen der Studienarbeit zu erläutern.	<i>hoch</i>

Tabelle 1: Anforderungen auf der Projekt-Ebene

<i>ID</i> <i>Bezeichnung</i>	<i>Kurzbeschreibung</i>	<i>Priorität</i>
H-001 CONN 1 Konnektivität	Die Tasse soll über WLAN Technologie mit der Infrastruktur des CSTI kommunizieren können.	<i>hoch</i>
H-002 CONN 2 Konnektivität	Die Tasse soll mit anderen Tassen in der Nähe (<3m) kommunizieren können.	<i>hoch</i>
H-003 CONN 3 Konnektivität	Die Tasse soll andere Tassen im selben Raum erkennen können.	<i>hoch</i>
H-004 ENGY 1 Energiemanagement	Die Tasse soll mindestens einen Tag ohne Ladevorgang auskommen.	<i>mittel</i>
H-005 SELF 1 Selbstauskunftsfähigkeit	Die Tasse soll ihren Füllstand erkennen.	<i>hoch</i>
H-006 SELF 2 Selbstauskunftsfähigkeit	Die Tasse soll die Temperatur der eingefüllten Flüssigkeit erkennen.	<i>hoch</i>
H-007 SELF 3 Selbstauskunftsfähigkeit	Die Tasse soll ihre Bewegung wahrnehmen.	<i>hoch</i>
H-008 SELF 4 Selbstauskunftsfähigkeit	Die Tasse soll die Anzahl der Füllungen erfassen.	<i>niedrig</i>
H-009 COMP 1 Computing	Die Tasse soll für eine hohe technische Akzeptanz auf einer der beiden Plattformen Arduino oder STM32 basieren.	<i>hoch</i>
H-010 DEV 1 Entwicklung	Die Programmierung bzw. das Debuggen der Tasse soll über USB möglich sein.	<i>hoch</i>

Tabelle 2: Anforderungen auf der Hardware-Ebene

<i>ID</i>	<i>Kurzbeschreibung</i>	<i>Priorität</i>
<i>Bezeichnung</i>		
S-001 INFC 1 Softwareinterface	Die Tasse soll in regelmäßigen Abständen genormte Statusmeldungen versenden können. Der Richtwert für den zeitlichen Abstand der Meldungen ist 3 Sekunden.	<i>hoch</i>
S-002 PLAT 1 Plattform	Um eine einfache Weiterentwicklung zu gewährleisten, soll die Software der Tasse in die Arduino IDE eingebunden werden können.	<i>hoch</i>
S-003 STRC 1 Strukturierung	Die Software der Tasse soll modularisiert entwickelt werden.	<i>hoch</i>
S-004 STRC 2 Strukturierung	Die Software soll in ihrer Struktur insbesondere in Bezug auf die Sensorik erweiterbar bzw. beschränkbar sein.	<i>mittel</i>
S-005 TEST 1 Testing	Es sollen Unit Tests für jeden Sensor durchgeführt werden.	<i>niedrig</i>

Tabelle 3: Anforderungen auf der Software-Ebene

Einige Anforderungen auf der Hardware-Ebene ließen sich auch an die Software stellen. Die Überlegungen, welche Sensoren, Aktoren und Konnektoren verbaut werden sollen, bestimmen allerdings die Gestaltung der Software mit, sodass diese Anforderungen der üblicherweise nachfolgenden Softwarekonzeption ohnehin immanent sind.

4. Konzeption

Grundsätzlich soll die Realisation zwar in mehreren Schritten erfolgen, einige Festlegungen können allerdings schon zum Projektbeginn formuliert werden. Eine konstruktive Entscheidung ist die Unterbringung sämtlicher Elektronik in einem ablösbaren Fuß. Damit soll einerseits die nach dem Ansatz des *UbiComp* für eine ungehinderte Integration in das natürliche Umfeld des Menschen wichtige Tassenform weitestgehend erhalten bleiben. Andererseits sind dafür pragmatische Gründe, wie etwa die Spülmaschineneignung der Tasse, zu nennen. Weiter wurden den Anforderungen und berücksichtigten Szenarien des dritten Kapitels bereits konkrete Sensoren, Aktoren und Konnektoren zugeordnet, womit ein gewisser technischer Rahmen ebenso festgelegt wurde. Die Realisation des Systems erfolgt in drei Iterationen: Zuerst soll die prinzipielle Funktionalität des Systems, insbesondere der Schnittstellen, ermittelt werden. Darauf aufbauend ist eine Annäherung an die spätere Bauform sinnvoll, sodass Fragestellungen zur Anordnung

und Verdrahtung der Komponenten beantwortet werden. Im letzten Schritt erfolgen die Zusammenführung in einem funktionierenden, fertig konstruierten Prototyp und eine grundlegende Datenauswertung.

4.1. Vorüberlegungen zur Hardwareauswahl

In der Analyse im vorigen Kapitel wurde an einer Stelle besonders stark von dem Grundsatz abgewichen, in Anforderungen keine Lösungen vorzuschreiben: Viele der bisherigen Projekte mit Bezug auf IoT im CSTI wurden interdisziplinär umgesetzt und nutzten die Arduino Technologie für einen einfachen Einstieg in die Programmierung von digitalen Gegenständen. Daher wurde im Zusammenhang mit diesem Projekt der Wunsch geäußert, auch hier auf der hinlänglich bekannten Plattform aufzubauen. Dennoch kann hinterfragt werden, ob dies auch für den ersten Prototyp erforderlich bzw. sinnvoll ist. Um lange Lieferzeiten und Einarbeitungsprozesse zu vermeiden, ist generell ein Prototypenbau basierend auf verfügbaren Ressourcen bzw. mit minimalem Aufwand wünschenswert. Erst wenn der Systementwurf validiert und eine weitere Anpassung etwa einzelner Bauteile nicht mehr absehbar ist, werden dediziert für die eigene Anwendung hergestellte Platinen sinnvoll. Bei vielen Sensorik-Themen liegt der Gedanke nahe, die in heutigen Smartphones integrierten, zahlreichen Sensoren für erste Erkenntnisse zu nutzen, anstatt ein komplettes System basierend auf einem Mikrocontroller aufzubauen. Die diversen Mikrocontroller weisen durchaus oftmals eine große Entwicklergemeinde und teilweise hohe Verbreitung im Projektumfeld auf – ein Systementwurf erscheint allerdings im Vergleich mit dem vermeintlich einfachen Auslesen der Sensordaten eines Smartphones mit deutlich mehr Aufwand verbunden. Dem steht dann wiederum der Aufwand gegenüber, zu einem späteren Zeitpunkt die Plattformen zu wechseln. Um den besten Ansatz für eine rasche Prototypenentwicklung zu finden, werden daher beide Möglichkeiten kurz beleuchtet. Danach wird mit Hilfe eines Abgleichs mit den zuvor entwickelten Anforderungen, der zu erwartenden Unterstützung im CSTI und an der HAW allgemein sowie einer Aufwandsschätzung für die Einarbeitung und Entwicklung die beste Alternative ausgesucht.

Smartphone-basierte Lösungsansätze

Eine oberflächliche Suche nach existierenden Frameworks, die Sensoren eines Smartphones als Datenquelle für den ersten Prototyp der Tasse zu nutzen, ergibt im Wesentlichen zwei dokumentierte, quelloffene Projekte: Zum einen können mit einer speziellen Android Applikation (A-

marino) die Sensordaten des Smartphones ausgelesen und über Bluetooth an einen entsprechend ausgerüsteten Mikrocontroller (aufgeführt wird offiziell nur ein Arduino) gesendet werden [AMARINO-10]. Zum anderen kann ein auf Mozilla OS basierendes Betriebssystem, JanOS, auf die Platine eines Smartphones aufgespielt werden, welches nativ die Programmierung in Javascript unterstützt oder eine Linux-ähnliche Umgebung für die dort lauffähigen Sprachen bietet [JANOS-14]. Gängige Sensoren in Smartphones liefern (teilweise sogar bereits grundlegend aufbereitete) Daten über die Lage, Beschleunigung, Temperatur, den Druck und die Lichtintensität im Umfeld des Gerätes (Infrarot) sowie über WLAN-, Bluetooth- und GPS-Signale. Überdies sind oftmals zwei Kameras und mehrere Mikrofone integriert [LANE-10]. Eine Erweiterung der Systeme um zusätzliche Komponenten ist allerdings nicht realisierbar. Die größten Aufwände bei den Smartphone-basierten Lösungsansätzen stellen die Einarbeitung sowie die Erstellung der Software dar. Dem gegenüber stehen Vereinfachungen, die sich durch die systembedingte Erfüllung einiger Anforderungen wie H-001, H-002 oder H-010 ergeben. Auch übernehmen die jeweiligen Frameworks die andernfalls manuell zu programmierende Kommunikation mit den einzelnen Sensoren, Aktoren und Konnektoren, sodass bei der Softwareentwicklung an dieser Stelle kaum Aufwand entsteht. Welche der zuvor aufgezählten Komponenten über das Framework auslesbar oder steuerbar sind, ist unterschiedlich – beide erwähnten Projekte haben diesbezüglich Limitationen. Bei JanOS werden zusätzlich nur bestimmte Smartphone-Platinen unterstützt. So besteht bei diesen Ansätzen die Gefahr, bei dem ersten Prototyp manche Anforderungen noch nicht berücksichtigen zu können und überdies höhere Aufwände bei der Beschaffung der Hardware hinnehmen zu müssen.

Entwickeln mit Mikrocontrollern, Shields und Break-Out-Boards

Durch andere Projekte in den technischen Fakultäten der HAW Hamburg oder in der Makerszene (deren Veröffentlichungen auf Internetplattformen zwar keinen wissenschaftlichen Ansprüchen genügen, nichtsdestotrotz interessante Impulse liefern) entstand ein unvermeidlicher Fokus auf die dort vorherrschenden Plattformen. Dazu zählen insbesondere die diversen Arduino Modelle, die meist auf Atmel Mikroprozessoren basieren, der STM32 und darauf aufbauende Systeme wie der Particle Photon sowie größere, leistungsfähigere Ein-Platinen-Computer wie der Raspberry Pi. Die Aufzählung erhebt keinen Anspruch auf Vollständigkeit, eine detaillierte Auseinandersetzung mit der Vielzahl an Mikrocontrollern auf dem Markt ist allerdings auch nicht Gegenstand dieser Arbeit. Neben der häufigen Verwendung in verschiedensten IoT-Projekten spielt bei der Bewertung, ob sich eines der Systeme besonders gut für den Prototypenbau eignet bzw. welche Aufwände zu erwarten sind, die Verfügbarkeit von kompatiblen Sensoren, Aktoren und Konnektoren eine wesentliche Rolle. Seien es Shields zum Aufstecken auf den Controller oder

Break-Out-Boards, auf denen alle für den Betrieb der jeweiligen Komponente erforderlichen elektrischen Bauteile bereits integriert sind. Beides vereinfacht den Entwicklungsprozess und setzt zudem weniger elektrotechnisches Fachwissen voraus. Shields haben üblicher Weise größere Abmessungen, sodass Break-Out-Boards hinsichtlich des geringen Bauraumes im Inneren des Tassenfußes präferiert werden. Ein weiterer Vorteil solcher Boards ist die oftmals gegebene Unterstützung von Bussystemen wie SPI oder I²C. Da die meisten Mikrocontroller nur über eine begrenzte Anzahl an (analogen) Eingangspins verfügen, erscheint die Anbindung vieler unterschiedlicher Sensoren andernfalls schwierig. Alle zuvor genannten Plattformen weisen mindestens eines der beiden Bussysteme auf, daher müssen für eine Entscheidung die Aufwände sowie die voraussichtliche Akzeptanz im Projekt abgewägt werden. Letztere ist bei den Arduino-Modellen hoch (ebenso die Verfügbarkeit), während sehr hardwarenahe Systeme weniger Unterstützung erwarten lassen. Der Particle Photon erfordert zwar deutlich weniger Auseinandersetzung mit rudimentären Strukturen wie es bei der Arbeit mit einem einfacheren System auf STM32-Basis der Fall wäre, der genaue Einarbeitungsaufwand kann mangels persönlicher Erfahrung jedoch nicht eingeschätzt werden. Die noch in Diskussion stehenden Ein-Platinen-Computer fallen schließlich aufgrund ihrer Abmessungen aus der Betrachtung heraus. Damit ist vor allem zu prüfen, welcher Arduino die festgelegten Anforderungen erfüllen kann. Ein erster Vergleich der verschiedenen Modelle anhand der Merkmale Abmessungen, Rechenleistung, verfügbarer Schnittstellen und Ressourcenverbrauch ergab eine potenzielle Eignung des Nano, Mini sowie MKR1000. Der Mini kann aufgrund der Anforderung H-010 ausgeschlossen werden, da dieses Modell über keine USB-Schnittstelle verfügt [ARDUINO-MINI]. Weiter wird der MKR1000 gegenüber dem Nano aus mehreren Gründen präferiert: der MKR ist etwas leistungsfähiger, bietet den Vorteil eines integrierten WLAN Moduls (wodurch Anforderung H-001 ohne die Notwendigkeit zusätzlicher Peripherie erfüllt ist) und vor allem läuft das System auf 3.3V wie die Mehrzahl der (im CSTI verfügbaren) Sensoren [ARDUINO-MKR, ARDUINO-NANO]. Dies erübrigt Spannungswandler und ermöglicht eine einheitliche Versorgungsstruktur.

Beide aufgeführten Lösungsansätze weisen Stärken und Schwächen auf. Die wohl größten Vorteile eines Smartphone-basierten Ansatzes sind die validierte Kompatibilität der einzelnen Hardwarekomponenten sowie die Vereinfachung der Auswertung einzelner Sensoren. Nachteilig sind dagegen die Limitationen der Frameworks, nur bestimmte Komponenten ansteuern zu können, sowie die Beschränkung auf bestimmte Platinen. Auch ist schwer einschätzbar, wieviel Einarbeitungsaufwand zu Beginn anfällt. Größere Aufwände sind zwar ebenso bei der eigenständigen Entwicklung eines Systems auf Basis eines Mikrocontrollers zu erwarten, doch kann hier argu-

mentiert werden, dass die Ergebnisse des ersten Prototyps für die darauf aufbauenden Entwicklungen nutzbar bleiben. Davon abgesehen ist (in der Theorie) eine unkomplizierte Verbindung von Peripherie und Mikrocontroller durch ein Bussystem möglich, die Kommunikation kann durch mitgelieferte Bibliotheken vereinfacht werden. Inwiefern diese Annahmen realistisch sind, wird im nächsten Kapitel erläutert. Zusammenfassend geht als vielversprechender Lösungsansatz die Kombination eines Arduino MKR1000 mit I²C-kompatiblen oder analogen Sensoren hervor. Für den ersten Prototyp sind zunächst noch alle aus den berücksichtigten Szenarien hervorgehenden Komponenten vorgesehen.

4.2. Systemabbildung in der Software

Nachdem die Vorüberlegungen zur Hardwareauswahl zu einer eigenen Systemkonzeption beginnend ab dem ersten Prototyp führten, stellt sich die Frage, wie die zugehörige Software strukturiert sein soll. Für den wissenschaftlichen Erkenntnisgewinn ist eine einfache Umsetzung verschiedener Szenarien mit Hilfe der Tasse wichtig. Als Grundlage dafür wurde bereits die Verwendung der weit akzeptierten Arduino IDE gefordert. Darüber hinaus soll die Software übersichtlich und modular aufgebaut werden – es bietet sich an, eine dedizierte Bibliothek für die Tasse zu entwickeln. Diese kann einfach in individuellen Programmen (Sketches) eingebunden werden, sodass der Zugriff auf die Funktionen der Tasse ohne weiteren Aufwand möglich ist. Das Verstecken von Programmteilen, die für den Nutzer nicht relevant sind (*Information Hiding*), ist zusammen mit dem Bestreben, in sich geschlossene Funktionen zu isolieren (*Encapsulation*) ein grundlegendes Designmuster. Um Außenstehenden das Verständnis des Systems zu erleichtern, soll die Software der Tasse das reale System abbilden, d.h. die einzelnen Komponenten werden im Objektmodell als eigene Klassen dargestellt. Eine übergeordnete Klasse vereint alle Objekte und bietet Methoden, auf die Daten und Funktionen des Systems zuzugreifen. Die Arduino IDE nutzt einen eigenen Sprachsyntax, unterstützt jedoch die wesentlichen Konstrukte der Programmiersprache C++. Genau genommen übersetzt die IDE den Arduino-Code in C++, verwendet hierfür allerdings eine eigene Compile-Struktur, sodass sich bei der unveränderten Übertragung von C++-Programmen bei der Verwendung von Bibliotheken und bei Abweichungen von der vorgeschriebenen Dateistruktur Schwierigkeiten ergeben können. Davon abgesehen, lässt sich prinzipiell ein Objektmodell in der Arduino IDE realisieren. Als Resultat aller bisherigen Überlegungen lässt sich das System der Tasse gemäß Abbildung 2 in drei Schichten abstrahieren: Auf der untersten Schicht werden Daten von Sensoren akquiriert (*Listen* [engl.]) und Aktionen mit

Konnektoren oder Aktoren ausgeführt (*Act*). Die erfassten Informationen können auf einer darüber liegenden Schicht aufbereitet werden (*Compute*) und damit auf der obersten Schicht komplexere Szenarien abgebildet werden.

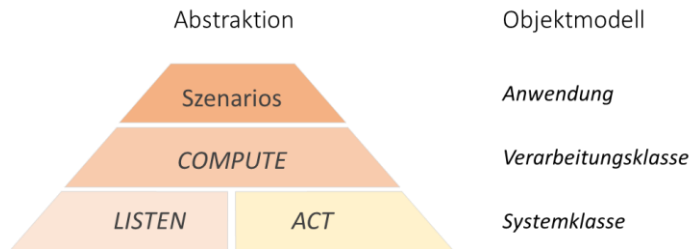


Abbildung 2: Abstraktionsebenen des Systems und Zusammenhänge zum Objektmodell

Im Rahmen der vorliegenden Arbeit werden nur die Funktionen *Listen* und *Act* umgesetzt, als Grundlage für zukünftige Weiterentwicklungen. Für eine kleine Demonstration wurde ergänzend ein rudimentäres Beispiel auf der *Compute*-Ebene erzeugt (siehe im beigefügten Code auf der CD), allerdings ohne das Objektmodell dahingehend zu erweitern. Die genauere Betrachtung der Softwarestruktur führt im nächsten Schritt auf ein Klassendiagramm, dargestellt in Abbildung 3. An dieser Stelle sei darauf hingewiesen, dass aus Entwicklersicht die Verwendung von Interface-Klassen für die Sicherstellung elementarer Funktionalitäten der drei Komponententypen Sensor, Aktor und Konnektor sinnvoll ist, eine dementsprechende Implementation in der Arduino IDE jedoch nicht realisiert werden konnte (dazu Kapitel 5.1). Daher wird im Folgenden das Objektmodell ohne eine solche Struktur beschrieben, wobei sich die von Komponenten des gleichen Typs geteilten Funktionen nun in jeder einzelnen Klasse wiederfinden. Zusätzlich greift das Klassendiagramm ein wenig der Realisation vorweg, da aufgrund der inkrementellen Entwicklung manche Ausprägungen des Systems zu Beginn noch anders vorgesehen waren als sie im hier abgebildeten Endzustand resultierten (insbesondere die fehlende Softwareunterstützung des verbauten RFID Moduls sowie der gänzliche Wegfall des NFC Moduls, doch dazu später mehr). Die Dreiteilung der elektronischen Komponenten ist im Klassendiagramm mit unterschiedlichen Farben markiert. Alle Klassen haben eine Methode zur Serialisierung des jeweiligen Objektes wie sie z.B. von Java nativ unterstützt wird. Sie gibt eine JSON Objekt-Repräsentation zurück und kann vor allem zum Debugging oder der menschenlesbaren Datenausgabe genutzt werden. Darüber hinaus werden alle Daten der Objekte über entsprechende Getter und Setter verwaltet, um unbeabsichtigten Manipulationen vorzubeugen. Die wichtigste Methode der Sensor-Komponenten dient der Datenaktualisierung: mit *refresh()* werden die jeweils zur Aktualisierung notwendigen Prozesse gestartet. Dazu können bei vier Sensoren (der IMU, den beiden Temperatursensoren sowie dem Lichtintensitätssensor) die mitgelieferten Bibliotheken einge-

bunden werden. Die individuellste Implementation ist für die Steuerung der Konnektoren notwendig, insbesondere müssen bei der Initialisierung des Bluetooth Moduls bzw. der entsprechenden seriellen Schnittstelle bestimmte Abläufe eingehalten werden. Die detaillierte Ausgestaltung des Codes ist schließlich das Resultat der im nächsten Kapitel beschriebenen Prototypenentwicklung.

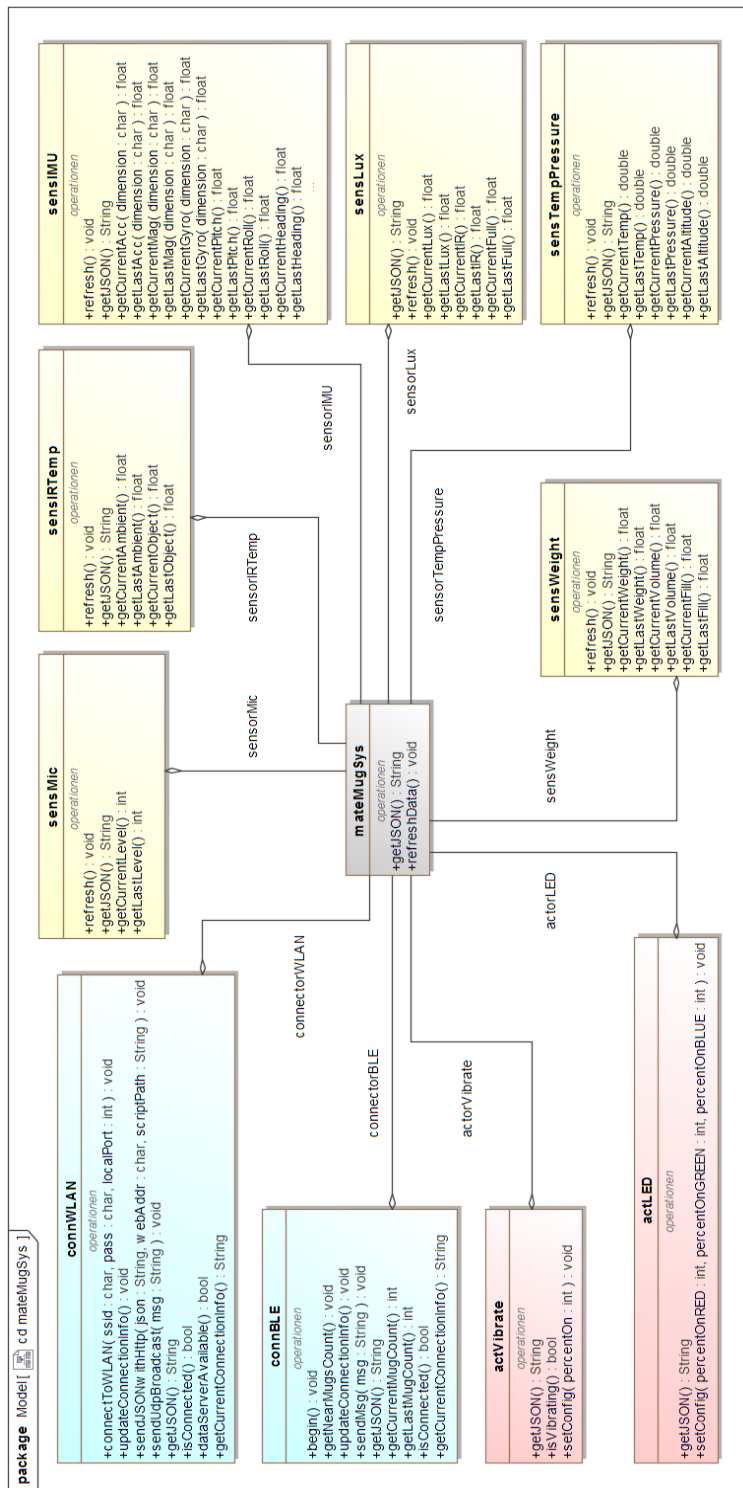


Abbildung 3: Klassendiagramm des Gesamtsystems (mateMugSys)

5. Realisation

Aus den Hard- und Softwarekonzepten gehen bereits einige Merkmale der Umsetzung hervor: Grundidee ist, Break-Out-Boards der gewünschten Sensoren in geeigneter Weise an den MKR 1000 anzuschließen und durch eine platzsparende Anordnung in einem abnehmbaren Tassenfuß unterzubringen. Dafür ist insbesondere zu klären, welche Rahmenbedingungen bei der Unterbringung zu beachten sind und wie die Versorgungs- und Datenleitungen verlegt werden müssen. Einige Limitationen führen einer bestimmten Lage der jeweiligen Break-Out-Boards im Tassenfuß: Für Sensoren ist vor allem relevant, in welcher Position sie eingebaut werden müssen, um ihre Funktion erfüllen zu können. Beispiele sind hier die IMU, die vorzugsweise nahe der Hochachse der Tasse liegen sollte, oder die Temperatursensoren, die nur unmittelbar unterhalb des Bodens der angefügten Tasse sinnvolle Messergebnisse liefern. Ähnliche Forderungen lassen sich für die Sensoren für den Druck- bzw. das Gewicht sowie die Lichtintensität formulieren. Bei den Konnektoren ist insbesondere die gegenseitige Störung zu minimieren, was z.B. durch einen möglichst großen Abstand oder eine orthogonale Orientierung zueinander erreicht werden kann. Die Aktoren schließlich müssen wie die Sensoren an der Stelle eingebracht werden, an der sie ihre Wirkung entfalten können (bei LEDs etwa in einem sichtbaren Bereich). Diese Überlegungen führen letztlich auf eine Unterbringung in drei Schichten, wobei die mittlere Schicht den Akkus vorbehalten ist. Nach einer überschlägigen Verbrauchsrechnung und unter Berücksichtigung der geometrischen Beschränkungen wurden zwei Lithium-Polymer Akkus mit je 1000mAh verschaltet. Die endgültige Ausführung wird in Abschnitt 5.2 beschrieben, wurde aber schon im zweiten Prototyp maßgeblich entwickelt. Grund hierfür ist unter anderem, dass die Lage der Komponenten den Aufbau der Versorgungsstrukturen wesentlich bestimmt. So wurde für den zweiten Prototyp eine halboffene Ringstruktur erarbeitet, bei der die Spannungs- und Datenleitungen (I²C) am Rand des Tassenfußes entlang gelegt werden und parallele Abzweigungen zu den Komponenten führen. Welche Erkenntnisse neben der offensichtlichen Beobachtung, dass der selbst bei einer großen Tasse winzige Bauraum alle Arbeiten an der Elektronik erschwert, gewonnen werden konnten und welche Änderungen an der zuerst geplanten Konfiguration erfolgen mussten, beschreibt der folgende Abschnitt.

5.1. Erfahrungen aus dem Entwicklungsprozess der Prototypen

Die Idee einer inkrementellen Entwicklung der Prototypen ist, in möglichst kurzer Zeit die Erkenntnis zu erlangen, ob die Konzeption in dieser Form umsetzbar ist bzw. an welchen Stellen

Anpassungen erfolgen müssen. Zwar ist im Hinblick auf das Studienziel nur der funktionierende, letzte Prototyp von Interesse, doch lassen sich durchaus einige Schlüsse für zukünftige Projekte oder Weiterentwicklungen aus den Zwischenschritten ziehen. Deshalb werden nun jeweils kurz der Funktionsumfang der entwickelten Prototypen sowie die größten Arbeitsaufwände beschrieben.

Nachdem eine oberflächliche Einarbeitung in die Entwicklungsumgebung und Funktionsweise des Arduinos erfolgte, lag der Fokus des ersten Prototyps auf dem sogenannten *Proof of Idea*: Alle beschafften Komponenten wurden auf einem Steckbrett an den MKR angeschlossen und mit Hilfe der mitgelieferten Bibliotheken erste Daten erfasst. Schnell wurde allerdings deutlich, dass diese Form an den entscheidenden Stellen keine Aussage über die Machbarkeit einer sinnvollen Erfassung der für den Kontext der Tasse relevanten Daten ermöglicht. Zu stark sind die Messergebnisse von einer dem späteren Aufbau ähnlichen Lage und einer Kombination der Daten abhängig. Weiterhin konnte ohne zusätzlichen Verdrahtungsaufwand kein Szenario für einen Integrationstest erzeugt werden, sodass an dieser Stelle die Entwicklung des ersten Prototyps beendet wurde. Das Ergebnis der grundsätzlichen Auseinandersetzung mit der Funktionsweise des Systems floss dann in den zweiten Prototyp ein.

Um die späteren Dimensionen des Tassenfußes besser einschätzen zu können, wurde eine Pappform erstellt, in der die Komponenten gemäß der zuvor erläuterten Rahmenbedingungen provisorisch in Schichten eingeordnet werden konnten. Das erarbeitete, ringförmige Verdrahtungskonzept wurde mit einigem Aufwand umgesetzt, sodass schließlich die Sensoren für Temperatur, Druck, Licht, Bewegung und Geräusche, ein Vibrationsmotor sowie die Konnektoren für WLAN, Bluetooth Low Energy, RFID und NFC an den MKR verlötet werden konnten. Bei einem rudimentären Test auf Basis der bestehenden Bibliotheken fielen einige Sensoren überraschend aus – die Fehlersuche gestaltete sich allerdings äußerst schwierig. Sobald tiefergehende Fragestellungen in Bezug auf die Funktion oder den Aufbau der jeweiligen Sensoren und Bibliotheken geklärt werden müssen, reichen die Dokumentationen auf den Herstellerseiten oder der arduino.cc Entwicklerplattform nicht mehr aus – oft war die einfache Frage nach den standardmäßig zugewiesenen Hardwareadressen des I²C-Protokolls kaum zu beantworten. Erst mit Hilfe eines digitalen Oszilloskops konnte die Fehlerursache, ein schadhaftes NFC Modul, aufgedeckt werden. Nachdem diese zeitaufwändige Hürde überwunden (und das Modul entfernt) war, konnte der Rest des Systems zufriedenstellend getestet und in Betrieb genommen werden.

Beim dritten Prototyp lag der Schwerpunkt auf der konstruktiven Ausgestaltung des Tassenfußes sowie der Integration des verdrahteten Systems des vorherigen Prototyps. Eine Herausforderung war dabei der Umgang mit den eng beieinander liegenden Komponenten: die Unterbringung in der dreischichtigen Konstruktion erforderten eine Fixierungen einerseits und eine Kabelführung zwischen den Schichten andererseits. Deshalb wurden eine Gitterstruktur und eine Verschraubung der Grundplatten jeder Schicht mit der äußeren Hülle entwickelt, die nachträgliche Änderungen erleichtern. Umgesetzt wurde die Konstruktion mit Hilfe von 3D-Druck aus Stratasys ABSplus-P430. Als letzte Änderung an der Hardware wurde ein Ring aus vier RGB-LEDs integriert, allerdings eher im Hinblick auf die Nutzung als aktiven Begleiter oder in Showcases als zur Bestimmung der *Digital Persona*. Anschließend sollte die Software finalisiert werden, wobei sich erneut erhebliche Schwierigkeiten ergaben. Optimal wäre eine Umsetzung aller tassenbezogenen Funktionen in einer für den ungeübten Programmierer oder Forscher aus einer anderen Disziplin leicht einzubindenden Arduino-Bibliothek. Doch selbst die umfangreiche Beschäftigung mit dem Build-Prozess der Arduino IDE und die Auseinandersetzung mit diversen Alternativen zu der nativen Entwicklungsumgebung ergaben keine Möglichkeit, eine gut strukturierte eigene Bibliothek mit gleichzeitiger Verwendung von Drittbibliotheken zu realisieren. Nach dieser Erkenntnis wurde stattdessen ein Lösungsansatz entwickelt, bei dem der Programmcode (und das Objektmodell) der Tasse in einem separaten Tab eines Arduino Sketches liegt. Damit kann zumindest ein Minimum an Übersichtlichkeit erreicht werden und der Nutzer unkompliziert auf die Funktionen der Tasse zugreifen. Die konstruktive Gestaltung des Prototyps und ein Auszug aus dem zu Testzwecken entwickelten Showcase werden als Ergebnis der gesamten Prototypenentwicklung im nächsten Abschnitt ergänzend zu den bisherigen Erläuterungen etwas ausführlicher dargestellt.

5.2. Beschreibung und Visualisierung des abschließenden Projektstands

Das Resultat der Entwicklung ist eine an die Form der Tasse angepasste Struktur, deren drei Böden bei Bedarf mit Zylinderstiften verbunden und an die die Break-Out-Boards angeschraubt werden können. Der Stapel aus den verbundenen Ebenen lässt sich dann in die äußere Hülle einschieben und durch Verdrehen arretieren. Die Abbildung 4 zeigt das CAD-Modell, welches zur Herstellung des Tassenfußes konstruiert wurde, die Abbildungen Abbildung 5 und 6 veranschaulichen den realen Prototyp. Die Lage der einzelnen Komponenten auf den drei Schichten wird zusammen mit dem Verdrahtungskonzept in Appendix B (S. 38) visualisiert, sodass die Vervielfältigung der Tasse erleichtert wird.

Realisation

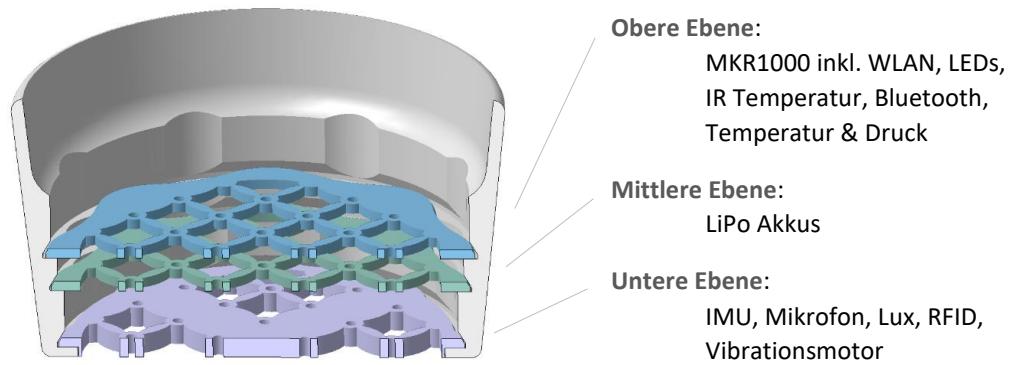


Abbildung 4: Querschnitt durch den Tassenfuß



Abbildung 5: Hülle des konstruierten Tassenfußes



Abbildung 6: Komponentenschichten

Die Software der Tasse wird als Arduino Sketch distribuiert und ist in drei Tabs aufgeteilt. Im ersten Tab werden die genutzten Drittbibliotheken eingebunden und der Nutzer hat die Möglichkeit, weitere Bibliotheken hinzuzufügen. Der zweite Tab beinhaltet die eigentliche Programmierung der Tasse in Form einer Objektstruktur wie sie in der Konzeption beschrieben wurde. Zuletzt ist das Hauptprogramm des Sketches (bestehend aus den standardmäßig von der Arduino IDE eingeführten `setup()` und `loop()` Routinen) in einen Tab ausgelagert. Hier können nun die konkreten Anwendungsszenarien umgesetzt werden. Das Vorgehen dazu beschreibt der nachfolgende Auszug aus dem Showcase:

```

1  #include "Arduino.h"
2  /***** INITIATE MATE MUG OBJECT CONTAINING ALL DATA *****/
3  mateMugSys myMug;
4
5  void setup() {
6      mugSystemInit();
7      Serial.begin(9600);
8      myMug.connectorWLAN.connectToWLAN(homeWLAN_ssid, homeWLAN_pass, local_port);
9      myMug.connectorBLE.begin();
10 }
11
12 void loop() {
13     String status;
14     myMug.refreshData();
15     status = myMug.getJSON();
16     // Output the system status via Serial Monitor
17     Serial.println(status);
18     // Send the data to a remote device with bluetooth
19     myMug.connectorBLE.sendMessage(status);
20 }

```

Abbildung 7: Beispielcode zur Nutzung der MateMug Bibliothek

Es sind im Wesentlichen nur zwei Voraussetzungen zu erfüllen: Erstens muss eine Instanz der *mateMugSys*-Klasse erzeugt werden (Zeile 3) um später auf die Funktionen und Daten der Tasse zugreifen zu können. Zweitens muss in der *setup()* Routine das System einmalig durch den Aufruf der global zur Verfügung gestellten Funktion *mugSystemInit()* initialisiert werden (Zeile 6). Danach können in jedem Zyklus neue Daten von den Sensoren erfasst (Zeile 14) und das Objektbild im JSON Format z.B. über die serielle Schnittstelle des MKRs oder über Bluetooth ausgegeben werden (Zeilen 17 bzw. 19).

5.3. Testen

Am Ende eines Hardware-Projektes besteht der Anspruch von Zuverlässigkeit und Sicherheit bei der Nutzung des Systems. Im Vorfeld sind dazu eine klare Definition von Anforderungen und ein gut strukturiertes Konzept hilfreich, zudem sollten während der Umsetzung systematisch Tests durchgeführt werden. Eine Schwierigkeit dabei wurde im vorherigen Abschnitt beschrieben: Oft ergeben sich Fehler erst bei der Kombination mehrerer, für sich allein funktionsfähiger Systemteile. Die dann erforderlichen Integrationstests lassen sich im Gegensatz zu Unit Tests kaum automatisieren, sodass meist nur der explizite Anwendungsfall, nicht aber potenzielle Missbrauchsmöglichkeiten untersucht werden können. Während des Übergangs zwischen dem ersten und zweiten Prototypen blieb ein solcher Integrationstest aller Komponenten auf dem Steckbrett aus, weswegen der im Nachhinein entstandene Aufwand der anschließenden Fehlersuche hätte vermieden werden können. Im Allgemeinen wurden allerdings kontinuierliche Tests mit Hilfe der seriellen Schnittstelle des Arduinos durchgeführt. Unabhängig davon musste nicht zuletzt aufgrund des zeitlichen Rahmens des Projektes die Annahme getroffen werden, dass die von den Sensorherstellern mitgelieferten Bibliotheken den Qualitätsansprüchen genügen und

getestet sind. Schließlich sind mit Blick auf das Ziel dieser Arbeit, einen Prototyp und kein fertiges Produkt zu entwickeln, die Ansprüche dahingehend etwas zu relativieren, dass, sofern keine Verletzungsgefahr besteht, kleinere Systemfehler hinnehmbar sind. Das System verwendet nur niedrige Spannungen, die Akkus sind von außen einsehbar und leicht auf etwaige Schäden zu prüfen, die Verkabelung wurde so weit wie möglich versiegelt. Eine letzte oberflächliche Prüfung erfolgte im Showcase, in dem einige der wesentlichen Funktionen des Systems nochmals aus Anwendersicht getestet und über einen längeren Zeitraum ausgeführt wurden. Damit sollte es, unter Zuhilfenahme der erstellten Dokumentation, möglich sein, in zukünftigen Weiterentwicklungen auftretende Fehler schnell zu beheben.

6. Auswertung und Erkenntnisse

Bevor eine Bewertung der dargestellten Lösung aus technischer und gesellschaftlicher Perspektive erfolgen kann, sollte eine Revision der eingangs festgelegten Anforderungen und ihrem Erfüllungsgrad erfolgen. Dies dient unter anderem dazu, den Weiterentwicklungsbedarf zu eruieren sowie ggf. als nicht zielführend zu bewertende Forderungen zu streichen. Auf Projektebene kann festgestellt werden, dass der Zeitrahmen (P-001) leicht überschritten wurde, hauptsächlich aufgrund der unvorhergesehenen Aufwände des Debuggings des I²C-Busses und der Softwarearchitektur im zweiten und dritten Prototyp. Dies ist in zukünftigen Projekten zu berücksichtigen. Nichtsdestotrotz konnte im Einvernehmen mit den Projektbetreuern eine beidseitige Akzeptanz der von der Planung abweichenden zeitlichen Umsetzung erreicht werden. Das Budgetziel (P-002) wurde mit Bestellungen im Wert von etwa 80€ (unter Verwendung vorhandener Komponenten) erfüllt, eine Bewertung der Dokumentation, zu der nicht zuletzt diese Ausarbeitung gehört, steht noch aus. Aus Hardwaresicht können die Anforderungen H-001 bis H-003 sowie H-009 und H-010 als erfüllt angenommen werden, da die Verwendbarkeit des WLAN bzw. Bluetooth Konnektors und des im MKR integrierten USB Ports durch den Showcase aufgezeigt wurden. Einige Anforderungen spielen auf konkrete Szenarien wie die Erfassung von Flüssigkeitseigenschaften oder Bewegungen an (H-005 bis H-008), sodass hier zwar die potenzielle Umsetzbarkeit durch die Integration der entsprechenden Sensoren sichergestellt werden konnte, ein dedizierter Test wegen der Fokussierung auf die unterste Abstraktionsebene des Systems (siehe Kapitel 4.2) jedoch ausblieb. Die einzige unzureichend erfüllte Anforderung ist das Energiemanagement (H-004); die mit den in Reihe geschalteten Akkus erreichbare Betriebszeit wurde im Rahmen des Showcases auf nur etwa 3 Stunden unter hoher Auslastung (LEDs, Kommunikation über WLAN und BLE) bestimmt. Inwiefern im Angesicht des stark begrenzten Bauraumes hier

Verbesserungen möglich sind, ist noch zu prüfen. Als letztes lässt sich bezüglich der Anforderungen an die Software ein insgesamt hoher Erfüllungsgrad feststellen. Die Implementation in der Arduino IDE konnte, bis auf die Einschränkung der fehlenden Unterstützung einer nativen Bibliothek, gut strukturiert umgesetzt und dem Nutzer eine einfach zu bedienende Programmierschnittstelle zur Verfügung gestellt werden (S-001 bis S-004). Obgleich wie in Kapitel 5.3 erläutert umfangreichere Integrationstests sowie automatisierte Unit Tests aus Zeitgründen ausblieben, wurde dennoch regelmäßig die Funktionalität des Systems überprüft. Daher kann S-005 mit einer ausreichenden Erfüllung hinsichtlich des Prototypencharakters der umgesetzten Lösung bewertet werden. Zugleich wird auf die Notwendigkeit weiterer Tests insbesondere zur Sicherheit im Sinne des englischen Begriffs *Security* hingewiesen. Aufgrund des sehr persönlichen Charakters der von *Augmented Everyday Objects* erfassten Daten ist für die Wahrung von Privatsphäre und die Vermeidung einer Kompromittierung zu schädlichen Zwecken dieses Thema äußerst kritisch und gewinnt zunehmend an Bedeutung in der Forschung [KUSEN-16]. Zusammenfassend resultiert also eine zufriedenstellende Erfüllung der meisten Anforderungen, die verbleibenden Abweichungen sind entweder den Rahmenbedingungen des Projektes oder einer bewussten Schwerpunktsetzung zuzuschreiben.

Aus der technischen Perspektive lassen sich einige wesentliche Erkenntnisse festhalten: Auf die Hardware bezogen, bewies sich die Wahl eines Konzeptes auf Basis eines Mikrocontrollers und Break-Out-Boards als effiziente Lösung zur Prototypenentwicklung. Kritisch bleibt wie erwähnt das Energiemanagement, wobei für die Suche nach Alternativen zur erarbeiteten Realisation mehr Fachwissen und eine tiefgründigere Beschäftigung mit den Verbrauchsstrukturen auf unterster Ebene notwendig sind. Dies wird dadurch erschwert, dass bei Detailfragen etwa zu Verbrauchsoptimierungen häufig die Dokumentationen der Komponenten versagen. Eine weitere Erkenntnis aus diesem Projekt ist, für komplexere Projekte keinen Arduino bzw. nicht die native Entwicklungsumgebung zu nutzen. Die Limitationen der IDE erhöhen die Programmieraufwände enorm und verhindern die Umsetzung bewährter Designmuster. Inwiefern sich die Arduino Mikrocontroller an sich für alternative Verwendungen eignen, muss gesondert begutachtet werden. Von alledem abgesehen scheint eine Tasse prinzipiell gut geeignet, um Thematiken wie *Digital Persona* oder *Augmented Everyday Objects* als aktive Begleiter des Menschen zu untersuchen. Dies hat auch die Resonanz auf den im Rahmen der offiziellen Eröffnung des CSTI ausgestellten Showcase gezeigt: Schnell konnte ein Verständnis des Grundthemas und teils positives, teils skeptisches Feedback von den Besuchern erlangt werden. Bereits in diesem Rahmen ergaben sich also wertvolle Impulse dazu, wie Menschen auf digitalisierte Gegenstände reagieren könn-

ten. Meist äußerten die Besucher Gedankengänge, die die Tasse als aktiven Begleiters identifizierten, einige der in dieser Arbeit erwähnten, abstrakteren Szenarien wurden jedoch ebenso genannt. Abschließend kann der vorliegende Prototyp als Beweis aufgefasst werden, dass schon heute die Konzepte des *UbiComps* realisierbar sind, womit die Relevanz weiterführender Studien nochmals unterstrichen wird.

7. Ausblick

Der realisierte Prototyp verkürzt den Weg bis zur Beantwortung der Frage, wie Menschen auf eine unsichtbar vernetzte Umwelt reagieren bzw. welche Implikationen sich für ihren Alltag ergeben. Die Plattform erfüllt die Basisanforderungen und darüber hinaus weitere interessante Anwendungsszenarien. Aus der jetzigen Hardwarekonfiguration gehen allerdings auch potenzielle Arbeitsaufträge zukünftiger Weiterentwicklungen hervor. So wurde etwa ein RFID (und NFC-fähiges) Modul integriert, aus Zeitmangel blieb aber dessen Inbetriebnahme aus. Des Weiteren ist, wie an mehreren Stellen erwähnt, das Energiemanagement unzureichend umgesetzt, sodass hier der Bedarf nach besseren Konzepten sowie ggf. nach der Erweiterung um eine kontaktlose Ladetechnologie bestehen bleibt. Das eingebundene Experiment zu geeigneten Messmethoden für die Temperatur hat aufgezeigt, dass zwar beide Sensoren (auf Dehnung bzw. auf Infrarot basierend) nur die Wärme an der Unterseite des Tassenbodens erfassen können, gleichwohl liefert der IR Sensor diese Information deutlich schneller. Daher könnte im nächsten Prototyp der herkömmliche Temperatursensor aus der Konfiguration entfernt werden. In den rudimentären Versuchsreihen im Rahmen des Showcases fiel auf, dass die Füllstandsbestimmung mit Hilfe des analogen Drucksensors leider recht ungenaue Ergebnisse liefert. Dies kann zum einen durch die verbesserungsfähige konstruktive Lösung, zum anderen durch die Messmethode selbst begründet sein. Um eine fundierte Entscheidung diesbezüglich treffen zu können, müssten daher weitere Überlegungen und Tests durchgeführt werden. Generell gilt es, eine vollständigere Prüfung der erfassten Daten insbesondere im Hinblick auf eine womöglich angebrachte Glättung anzustreben. Neben der Optimierung des erarbeiteten Prototyps sind zur Erforschung der Vernetzung von *Augmented Everyday Objects* per se weitere Objekte nötig, zu deren Herstellung durchaus die beschriebene oder eine ähnliche Konfiguration auf einen anderen Gegenstand übertragen werden könnte. Bewegliche, nah am Menschen befindliche Objekte wie Flaschen, Taschen oder Kleidung sind dabei ebenso denkbar wie Möbel.

Abgesehen von diesen Hardwareaspekten lassen sich verschiedene Perspektiven für die Anwendung des Prototyps und aufbauende Forschungsziele aufzeigen. Denn letztlich konnte bisher nur die Grundlage einer Einflussanalyse von *Ubiquitous Computing* geschaffen werden. Der eingangs formulierten Frage, wie sich menschliches Verhalten und allgemein die Lebensbedingungen in Zukunft technologiebedingt verändern könnten, ist erst eine Vorahnung entgegenzustellen und viele weitere Fragen, etwa nach Sicherheitsaspekten oder gesellschaftlichen Gefahren, kamen neu hinzu. Antworten sind in den unterschiedlichsten Fachgebieten zu finden: beginnend bei sozialwissenschaftlichen Untersuchungen, die den Menschen im Angesicht der zunehmenden Digitalisierung beobachten, über Gedankengänge aus der Informatik bis hin zu der anfangs angeführten Integration in vernetzte Umgebungen wie dem Living Place an der HAW Hamburg. Ideen liefern die im Rahmen der Anforderungsanalyse aufgedeckten Szenarien oder weiterführende Anwendungsfälle. Während z.B. eine Gestensteuerung eher den Ansatz eines aktiven Begleiters vergegenwärtigen kann, erfordern Überlegungen zur *Digital Persona* hingegen eine gezielte Datenfusion, sodass die Auswertung dem Kontextbegriff gerecht wird. Die Aussicht auf eine konsequente Umsetzung des letztgenannten Ansatzes macht auch bewusst, dass es den unsichtbaren Computern an Schnittstellen zum Menschen mangelt – daher ist in diesem Zusammenhang die Einbindung in Projekte zur Virtuellen oder Erweiterten Realität (VR/AR) denkbar.

Zusammenfassend konnten im Rahmen der Projektarbeit viele Erkenntnisse zur entwickelten Hardware einerseits und zu dem zugrundeliegenden Forschungsthema andererseits gewonnen werden. Die Vielzahl an Fragen verdeutlicht die Relevanz erster Schritte zu deren Klärung und darüber hinaus den Bedarf an aufbauenden Forschungen. Der hohe Vernetzungsgrad des entwickelten Prototyps lässt sich an der Menge angrenzender Themengebiete ablesen, unter anderem der Aufdeckung neuartiger Anwendungen in VR oder AR. Die Komplexität einer zielführenden Erfassung des Kontexts eines Menschen konnte durch diese Arbeit dargestellt und ein Ansatz zu ihrer Bewältigung gegeben werden.

8. Quellen

- [AMARINO-10] KAUFMANN, B., BUECHLEY, L.: Amarino: a toolkit for the rapid prototyping of mobile ubiquitous computing. In: Proceedings of the 12th international conference on Human computer interaction with mobile devices and services (MobileHCI 2010), ACM New York, USA, 2010, S. 291-298. Online verfügbar unter <http://amarino-toolkit.net>.
- [ARDUINO-MINI] Arduino Mini Produktbeschreibung, <https://www.arduino.cc/en/Main/ArduinoBoardMini> (zuletzt aufgerufen am 10.04.17).
- [ARDUINO-MKR] Arduino MRK1000 Produktbeschreibung, <https://www.arduino.cc/en/Main/ArduinoMKR1000> (zuletzt aufgerufen am 10.04.17).
- [ARDUINO-NANO] Arduino Nano Produktbeschreibung, <https://www.arduino.cc/en/Main/ArduinoBoardNano> (zuletzt aufgerufen am 10.04.17).
- [BARTELS-17] BARTELS, L.,: Indoor Positionierung mit Smartphones zur Kontextererkennung auf Basis vorhandener Infrastruktur. Masterarbeit an der HAW Hamburg, 2017, S. 89. Online verfügbar unter <https://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/bartelt.pdf> (zuletzt aufgerufen am 10.04.17).
- [BEIGL-05] BEIGL, M., DECKER, C., KROHN, A., RIEDEL, T., ZIMMER, T.: μ Parts: Low Cost Sensor Networks at Scale. Telecooperation Office (TecO), Karlsruher Institut für Technologie, 2005. Projektseite unter <http://particle.teco.edu/upart/> (zuletzt aufgerufen am 10.04.2017).
- [BEIGL-03] BEIGL, M., GELLERSEN, H.: Smart-Its: An Embedded Platform for Smart Objects. Smart Objects conference (sOc), 2003. Online verfügbar unter <https://www.teco.edu/~michael/publication/soc2003.pdf> (zuletzt aufgerufen am 10.04.2017).
- [BEIGL-01] BEIGL, M., GELLERSEN, H.W., SCHMIDT, A.: MediaCups: Experience with Design and Use of Computer-Augmented Everyday Artefacts. In: Computer Networks: The International Journal of Computer and Telecommunications Networking, Pervasive Computing, Band 35, Ausgabe 4, März 2001, S. 201-409. Online verfügbar unter http://www.teco.edu/~michael/publication/mediacup_full.pdf (zuletzt aufgerufen am 10.04.2017).
- [BERKELEY-01] BERKELEY WEBS: Wireless Embedded Systems, Projektseite unter <http://smote.cs.berkeley.edu/motescope/> (zuletzt aufgerufen am 10.04.2017).
- [BERNING-15] BERNING, M., BUDDE, M., RIEDEL, T., BEIGL, M.: bPart - A Small and Versatile Bluetooth Low Energy Sensor Platform for Mobile Sensing. In: Proceedings of the 13th Annual International Conference on Mobile Systems, Applications and Services (MobiSys '15), ACM New York, USA, 2015, S. 473. Online verfügbar unter http://www.teco.edu/~berning/papers/mobisys2015_bpart.pdf (zuletzt aufgerufen am 10.04.2017).

- [CHIU-09] CHIU, M.C., CHANG, S.P., CHANG, Y.C., CHU, H.H., ET AL.: Playful Bottle: a Mobile Social Persuasion System to Motivate Healthy Water Intake. In: Proceedings of the 11th international conference on Ubiquitous Computing, ASM New York, USA, 2009, S. 185-194. Online verfügbar unter http://mll.csie.ntu.edu.tw/papers/PlayfulBottle_Ubicomp09.pdf (zuletzt aufgerufen am 10.04.17).
- [CHUNG-06] CHUNG, H., LEE, C.H.J., SELKER, T.: Lover's Cups: Drinking Interfaces as New Communication Channels. In: Extended Abstracts on Human Factors in Computing Systems (CHI EA '06), Montréal, Kanada, 2006. Online verfügbar unter http://dl.acm.org/ft_gateway.cfm?id=1125532&ftid=348443&dwn=1&CFID=922445523&CFTOKEN=27258037 (zuletzt aufgerufen am 10.04.2017).
- [CLARKE-94] CLARKE, R.: The Digital Persona and its Application to Data Surveillance. In: The Information Society, Band 10, Ausgabe 2, Juni 1994, S. 77-92. Projektseite unter <http://www.rogerclarke.com/DV/DigPersona.html> (zuletzt aufgerufen am 11.04.17).
- [DOURISH-04] DOURISH, P.: What We Talk About When We Talk About Context. Personal and Ubiquitous Computing, Band 8, Ausgabe 1, Februar 2004, Springer, London, UK, S. 19-30.
- [JANOS-14] JANOS, 2014. Online verfügbar unter <http://janos.io> (zuletzt aufgerufen am 21.11.2016).
- [KAO-14] KAO, H.L.C., SCHMANDT, C.: MugShots: Everyday Objects as Social Catalysts. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, ASM New York, USA, 2014, S. 75-78. Online verfügbar unter http://ubicomp.org/ubicomp2014/proceedings/ubicomp_adjunct/posters/p75-kao.pdf (zuletzt aufgerufen am 10.04.17).
- [KAWAHARA-03] KAWAHARA, Y., MINAMI, M., MORIKAWA, H., AOYAMA, T.: Design and Implementation of a Sensor Network Node for Ubiquitous Computing Environment. In: Vehicular Technology Conference, 2003. Online verfügbar unter <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.6630&rep=rep1&type=pdf> (zuletzt aufgerufen am 10.04.2017).
- [Ko-07] KO, J.C., HUNG, Y.P., CHU, H.H.: Mug-Tree: A Playful Mug to Encourage Healthy Habit of Drinking Fluid Regularly. In: The Late Breaking Results (LBR) session of Proc. UBICOMP 2007. Online verfügbar unter http://mll.csie.ntu.edu.tw/papers/Ubicomp2007LBR_1569054868_camera_ready.pdf (zuletzt aufgerufen am 10.04.17).
- [KUNIAVSKY-03] KUNIAVSKY, M.: Smart Things: Ubiquitous Computing User Experience Design. Elsevier, Burlington, USA, 2010, S. 287-293.
- [KUSEN-16] KUŠEN, E., STREMBECK, M.: A decade of security research in ubiquitous computing: results of a systematic literature review. In: International Journal of Pervasive Computing and Communications, Band 12, Ausgabe 2, S. 216-259. Online verfügbar unter <https://arxiv.org/pdf/1701.00773.pdf> (zuletzt aufgerufen am 11.04.17).

Quellen

- [LANE-10] LANE, N.D., MILUZZO, E., LU, H., PEBBLES, D., CHOUDHURY, T., CAMPBELL, A.T.: A survey of mobile phone sensing. In: IEEE Communications Magazine, Volume 48, Issue 9, 2010, S. 140-150.
- [NISSANKA-04] NISSANKA, B.P.: Cricket v2 User Manual, 2004. Online verfügbar auf der Projektseite <http://cricket.csail.mit.edu/> (zuletzt aufgerufen am 10.04.17).
- [PARK-02] PARK, S., LOCHER, I., SAVVIDES, A., SRIVASTAVA, B., ET AL.: Design of a Wearable Sensor Badge for Smart Kindergarden. In: Proceedings of the Sixth International Symposium on Wearable Computers (ISWC 2002), 2002. Online verfügbar unter <http://ieeexplore.ieee.org/document/1167252/> (zuletzt aufgerufen am 10.04.2017).
- [TAPIA-04] TAPIA, E.M., MARMASSE, N., INTILLE, S.S., LARSON, K.: MITes: Wireless Portable sensors for Studying Behavior. In: Proceedings of Ubicomp 2004: Ubiquitous Computing, Nottingham, UK, 2004. Online verfügbar unter <http://www.ccs.neu.edu/home/intille/papers-files/MunguiaTapiaETAL04.pdf> (zuletzt aufgerufen am 10.04.2017).
- [WEISER-91] WEISER, M.: The Computer for the 21st Century. In: Scientific American, Ausgabe 9, September 1991, S. 94-104.

Appendix A

A.1 Ideensammlung möglicher Anwendungs-/ Forschungsszenarien

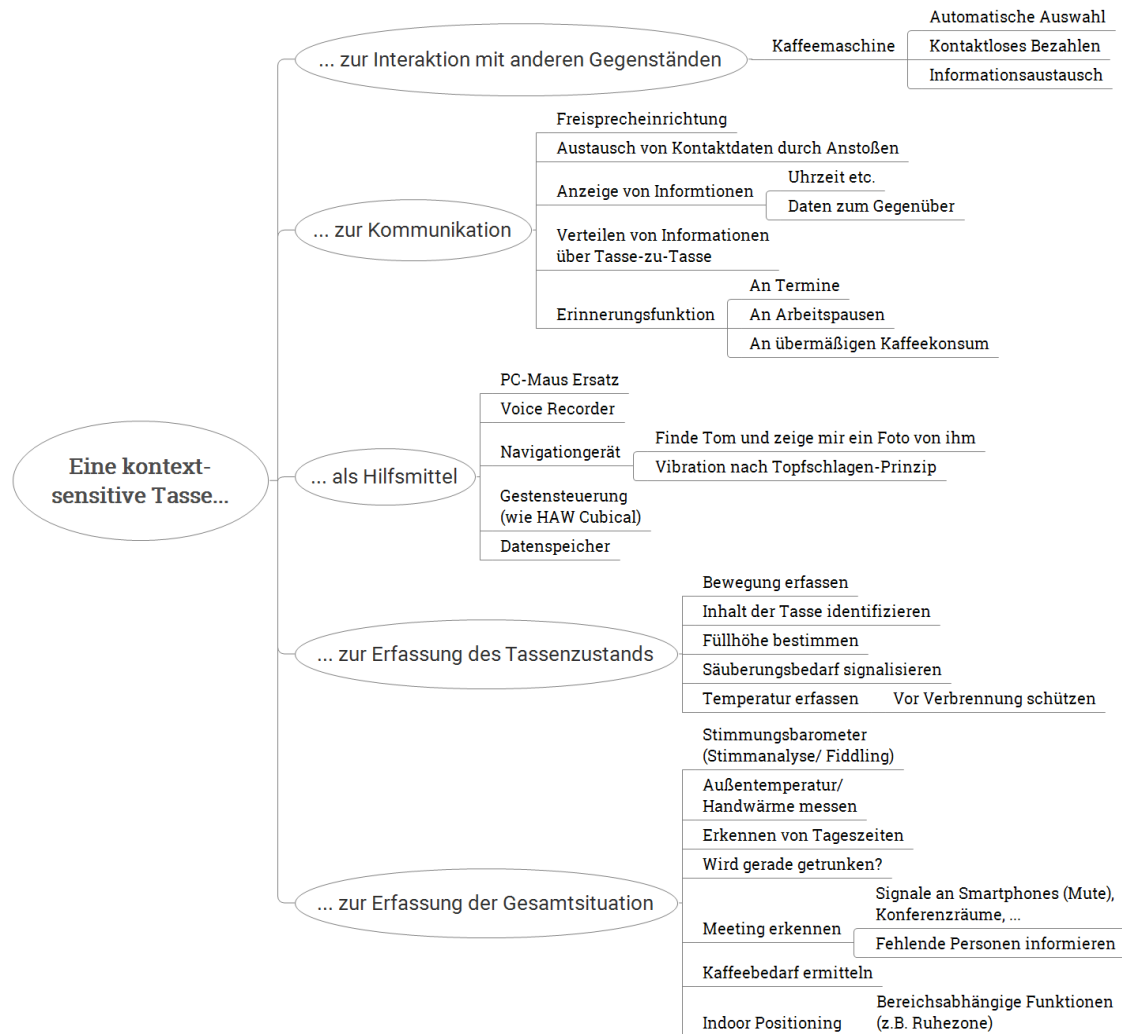


Abbildung 8: Mindmap zur Aufdeckung von Anwendungsszenarien

A.2 Szenarien- und Komponentenauswahl

Aus der Ideensammlung möglicher Szenarien wurde eine Auswahl geeignet erscheinender Anwendungsfälle selektiert, die in den nachfolgenden Tabellen den vermutlich für ihre Umsetzung benötigten Hardwarekomponenten gegenübergestellt werden. Anschließend werden jene Komponenten ausgewählt, die in möglichst vielen Szenarien Anwendung finden, einfach in das System zu integrieren und gut verfügbar sind.

Szenario	Sensoren					
	IMU	Temperatur	Kraft/ Gewicht	Mikrofo n	IR	Licht (Intensität)
Verbindung mit dem Backbone des CSTI						
Kommunikation mit anderen Tassen						
Andere Tassen im selben Raum erkennen						
Temperatur der Flüssigkeit erkennen		X				
Bewegung wahrnehmen	X					
Indoor Location	X					
Anzahl der Füllungen erfassen	X	X	X			
Trinkvorgang erkennen	X		O			
Stimmungsbarometer	x			x		
Stimmen erkennen				x		
Gestensteuerung	x					
Navigator mit Vibrationsfeedback	x					
Austauschen von Kontakten durch Anstoßen	x					
Verteilen von Informationen über Mug-to-Mug Kommunikation						
Freisprecheinrichtung				x		
Interaktion mit einer Kaffeemaschine (Präferenzen, Bezahlen, ...)						
Meeting erkennen, wer fehlt, Zeitgeber im Meeting						
Datenaustausch mit dem Smartphone						
Terminereinnerungen signalisieren						
Signalgeber ("zu heiß", "Zeit für eine Pause", "zu viel Kaffee heute", Säuberungsbedarf...)						
Tassenfinder, Vibrieren in Abhängigkeit zur Nähe zu einer Tasse ("Topfschlagen")						
Informationen über den Untergrund aufnehmen					x	x
Voice Recorder				x		
Umgebunghelligkeit z.B. für Abschalten wahrnehmen						x
Anzeigen von kontextuellen Informationen						
Füllstand erkennen			X			
Akkustand monitoren						
Anzahl möglicher Szenarien	8	2	3	4	1	2
Aufgewichten der Basiskriterien	1,5	2,0	2,0	1,0	1,0	1,0
Zwischenergebnis	12	4	6	4	1	2
Integrationsaufwand (4: niedrig, 1: hoch)	4	4	2	2	4	4
Verfügbarkeit im CSTI (4: vorrätig, 1: unbestellbar)	3	3	3	3	4	4
Ergebnis	19	11	11	9	9	10
Auswahl (>= 8)	X	X	X	X	X	X

Abbildung 9: Komponentenauswahl anhand von Szenarien und Kriterien (Sensoren)

Szenario	Konnektivität				
	BTLE	WLAN	RFID (passiv)	RFID (aktiv)	NFC (passiv)
Verbindung mit dem Backbone des CSTI		X			
Kommunikation mit anderen Tassen	X	?			
Andere Tassen im selben Raum erkennen	X	?			
Temperatur der Flüssigkeit erkennen					
Bewegung wahrnehmen					
Indoor Location	X	X		X	O
Anzahl der Füllungen erfassen					
Trinkvorgang erkennen					
Stimmungsbarometer					
Stimmen erkennen					
Gestensteuerung					
Navigator mit Vibrationsfeedback	x	x			
Austauschen von Kontakten durch Anstoßen	x				
Verteilen von Informationen über Mug-to-Mug Kommunikation	x				
Freisprecheinrichtung	x				
Interaktion mit einer Kaffeemaschine (Präferenzen, Bezahlen, ...)	O		x	O	x
Meeting erkennen, wer fehlt, Zeitgeber im Meeting	x				
Datenaustausch mit dem Smartphone	x				x
Terminerinnerungen signalisieren					
Signalgeber ("zu heiß", "Zeit für eine Pause", "zu viel Kaffee heute", "Säuberungsbedarf...)					
Tassenfinder, Vibrieren in Abhängigkeit zur Nähe zu einer Tasse ("Topf schlagen")	x	x			
Informationen über den Untergrund aufnehmen					
Voice Recorder					
Umgebungs helligkeit z.B. für Abschalten wahrnehmen					
Anzeigen von kontextuellen Informationen					
Füllstand erkennen					
Akkustand monitoren					
Anzahl möglicher Szenarien	11	6	1	2	3
Aufgewichten der Basiskriterien	1,3	1,4	1,0	1,5	1,3
Zwischenergebnis	14	8	1	3	4
Integrationsaufwand (4: niedrig, 1: hoch)	2	4	2	1	2
Verfügbarkeit im CSTI (4: vorrätig, 1: unbestellbar)	4	4	2	4	4
Ergebnis	20	16	5	8	10
Auswahl (>= 8)	X	X		X	X

Abbildung 10: Komponentenauswahl anhand von Szenarien und Kriterien (Konnektoren)

Szenario	Aktuatoren			
	Vibration smotor	Laut- sprecher	LED	(OLED) Display
Verbindung mit dem Backbone des CSTI				
Kommunikation mit anderen Tassen				
Andere Tassen im selben Raum erkennen				
Temperatur der Flüssigkeit erkennen			o	
Bewegung wahrnehmen			o	
Indoor Location				
Anzahl der Füllungen erfassen				
Trinkvorgang erkennen				
Stimmungsbarometer				
Stimmen erkennen				
Gestensteuerung				
Navigator mit Vibrationsfeedback	x			
Austauschen von Kontakten durch Anstoßen	x			
Verteilen von Informationen über Mug-to-Mug Kommunikation				
Freisprecheinrichtung		x		
Interaktion mit einer Kaffeemaschine (Präferenzen, Bezahlen, ...)			o	
Meeting erkennen, wer fehlt, Zeitgeber im Meeting			x	
Datenaustausch mit dem Smartphone				
Terminerinnerungen signalisieren		o	o	o
Signalgeber ("zu heiß", "Zeit für eine Pause", "zu viel Kaffee heute", Säuberungsbedarf...)	x		x	o
Tassenfinder, Vibrieren in Abhängigkeit zur Nähe zu einer Tasse ("Topf schlagen")	x			
Informationen über den Untergrund aufnehmen				
Voice Recorder				
Umgebungshelligkeit z.B. für Abschalten wahrnehmen				
Anzeigen von kontextuellen Informationen				x
Füllstand erkennen				
Akkustand monitoren				
Anzahl möglicher Szenarien	4	2	6	3
Aufgewichten der Basiskriterien	1,0	1,0	1,3	1,0
Zwischenergebnis	4	2	8	3
Integrationsaufwand (4: niedrig, 1: hoch)	3	2	3	1
Verfügbarkeit im CSTI (4: vorrätig, 1: unbestellbar)	4	3	4	2
Ergebnis	11	7	15	6
Auswahl (>= 8)	X		X	

Abbildung 11: Komponentenauswahl anhand von Szenarien und Kriterien (Aktoren)

Appendix B

B.1 Verdrahtungs- und Lageschema der Komponenten im Tassenfuß

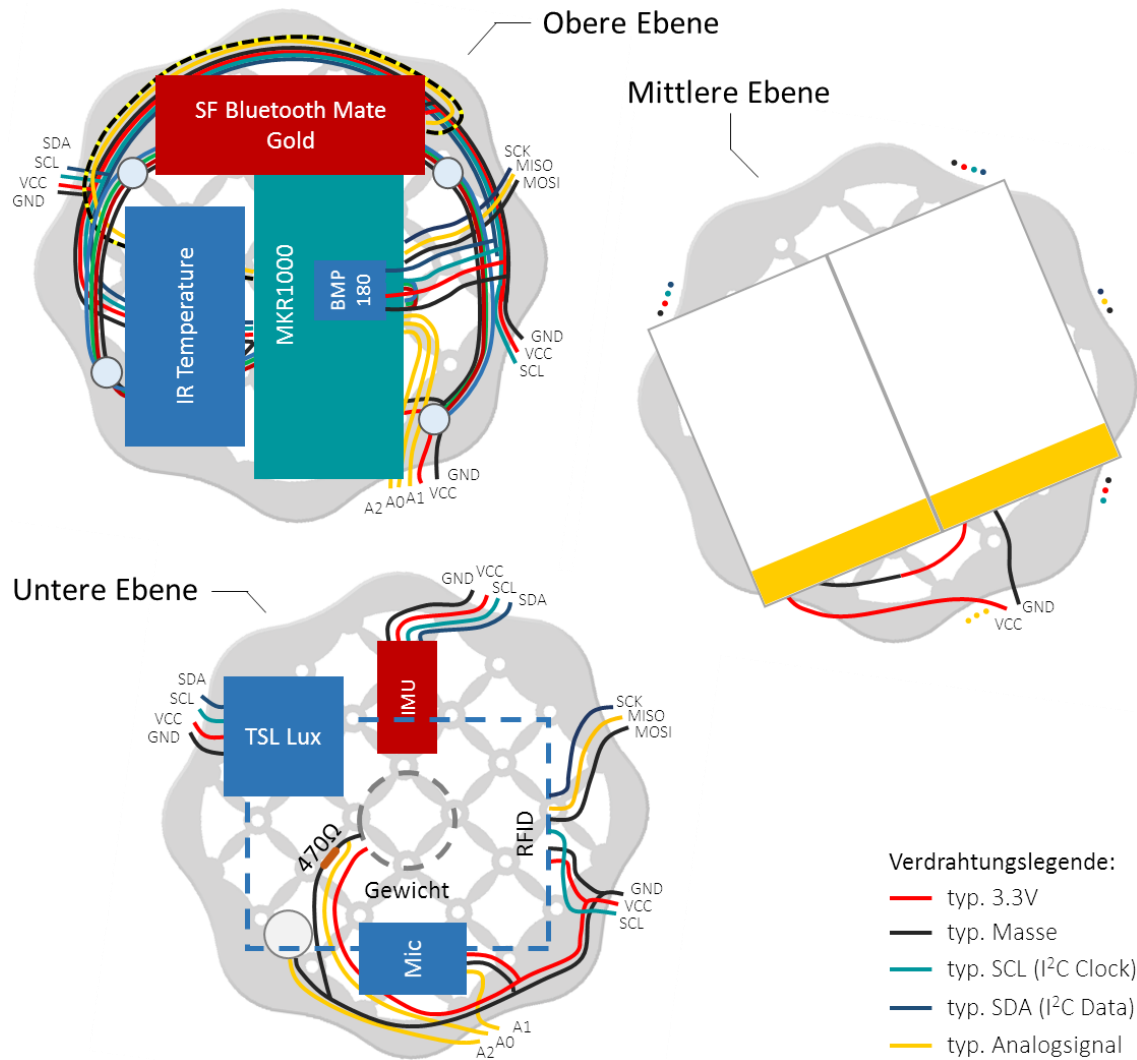


Abbildung 12: Verdrahtungs- und Lageschema

B.2 Komponentenübersicht und Pinbelegung

Pin	Modul	Adresse	Neupreis
Vcc	alle	-	-
GND	alle	-	-
14 Tx	SparkFun Bluetooth Mate Gold (RN-42)	-	~ 33€
13 Rx	SparkFun Bluetooth Mate Gold (RN-42)	-	~ 33€
12 SCL	Bosch BMP180 (GY-68) Temperature & Pressure	0x77	~ 5€
11 SDA	Adafruit TSL2561 Digital Luminosity/Lux/Light	0x39	~ 5€
	SparkFun 9 DoF IMU LSM9DS1	Mag: 0x1E Acc/Gyr: 0x6B	16€
	Digital Infrared Temperature	0x5B	
10 MISO	MFR522 NFC/RFID Controller	-	7€
9 SCK	MFR522 NFC/RFID Controller	-	
8 MOSI	MFR522 NFC/RFID Controller	-	
A0	Short Tail FSR 402	-	8€
A1	Adafruit Electret Microphone Amplifier - MAX4466	-	8€
A2	Pololu Shaftless Vibration Motor 10x2.0mm	-	~ 3€
A3	LEDs - Blau	-	
A4	LEDs - Rot	4x Kingbright L-154A4SURKQBDZGC	~ 5€
A5	LEDs - Grün	-	
JST	LiPo Akkus 1000mAh, 3.7V, 50x32x5.7mm (2x)	-	15€
			~ 105€

Abbildung 13: Pinbelegung, Komponentenbezeichnungen, Kostenübersicht

Appendix C

C.1 Installationsanweisungen

Um den mitgelieferten Code verwenden oder anpassen zu können, sind einige Grundbedingungen zu beachten.

1) *Installation der Arduino IDE*

Die Tassenbibliothek wurde mit der Arduino IDE Version 1.8.1 entwickelt. Es wird daher empfohlen, dieselbe Version zu installieren. Folgen Sie dafür den Anweisungen auf <https://www.arduino.cc/en/main/software>.

2) *Laden der Boardinformationen für den MKR1000*

Da die IDE zunächst die erforderlichen Konfigurationsdateien des MKRs vermissen lässt, müssen diese nachträglich hinzugefügt werden. Eine gute Anleitung dazu ist auf der Seite <https://www.arduino.cc/en/Guide/MKR1000> auffindbar.

3) *Einbindung der Drittbibliotheken*

Die Nutzung einiger Sensoren wurde durch die Einbindung der vom Hersteller mitgelieferten Bibliotheken vereinfacht. Damit die Tassenbibliothek verwendet werden kann, müssen diese Bibliotheken in die IDE eingebunden werden. Weil kleinere Änderungen an den jeweiligen Codes für die Kompatibilität mit dem vorliegenden System notwendig waren, müssen die auf der beigelegten CD zu findenden Dateien manuell eingebunden werden. Dazu muss die IDE geschlossen und alle Unterordner des Ordners *Code/libraries* in das lokale Arduino *libraries* Verzeichnis kopiert werden. Beim nächsten Start der Entwicklungsumgebung können die Bibliotheken verwendet werden.

4) *Verwendung des MateMug_SampleCode Sketches*

Nachdem die Vorbereitungen abgeschlossen wurden, kann eine lokale Kopie des Sketches (zu finden auf der CD im Ordner *Code*) frei bearbeitet werden. Im Sketch sind einige Anmerkungen zur Erleichterung der Nutzung gegeben, darüber hinaus können die Funktionalitäten der Tasse in der nachfolgenden Objektreferenz und dem Klassendiagramm nachgeschlagen werden.

C.2 Objektreferenz

MateMugSys

Beschreibung:

Das mateMugSys stellt das System der Tasse auf der untersten, abstrakten Datenebene dar. Mithilfe dieses Objektes können alle unterstützten *Listen* und *Act* Funktionalitäten erreicht werden.

Initialisierung:

- 1) Erzeugen einer Objektinstanz der Klasse durch die globale Deklaration:
`mateMugSys mySystem;`
- 2) Initialisierung des Systems (letztlich der angeschlossenen Sensoren) in der Arduino `setup()` Routine:
`mugSystemInit();`

Methoden

<code>getJSON()</code>	Liefert die JSON Repräsentation des Objektes und aller enthaltenen Objekte inklusive der aktuellen Daten. <pre>String sysObj = mySystem.getJSON();</pre>
<code>refreshData()</code>	Aktualisiert die Daten aller verbundenen Komponenten. Beispiel: <pre>String sysObj = mySystem.getJSON();</pre>

Attribute

<code>sensorMic</code> <code>sensorIRTemp</code> <code>sensorIMU</code> <code>sensorLux</code> <code>sensorTempPressure</code> <code>sensorWeight</code>	Die einzelnen Attribute ermöglichen den Zugriff auf die verbundenen Objekte der einzelnen Komponenten und deren Funktionen/ Daten. <pre>// Refresh only the temperature sensor's data mySystem.sensorTempPressure.refresh();</pre>
<code>actorVibrate</code> <code>actorLED</code>	<pre>// Read some data from a sensor Float acceleration = mySystem.sensorIMU.getCurrentAcc('x');</pre>
<code>connectorBLE</code> <code>connectorWLAN</code>	<pre>// Set the vibration motor to 80% rumble mySystem.actorVibrate.setConfig(80);</pre>