



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Benjamin-Bodo Salewski

Entwicklung einer interaktiven Pflanze als proaktives
smartes Objekt

Benjamin-Bodo Salewski

Entwicklung einer interaktiven Pflanze als proaktives
smartes Objekt

Bachelorarbeit eingereicht im Rahmen Bachelorprüfung

im Studiengang Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr. Kai von Luck
Zweitgutachter : Dr. Susanne Draheim

Abgegeben am 13.04.2017

Benjamin-Bodo Salewski

Thema der Arbeit

Entwicklung einer interaktiven Pflanze als proaktives smartes Objekt

Stichworte

Smartes Objekt, Companion, Human Computer Interaction, HCI, Pflanze-Companion

Kurzzusammenfassung

Viele Gegenstände im Alltag sind nicht besonders auskunftsfähig und können uns daher auch nicht mitteilen, wenn sie Aufmerksamkeit brauchen. In dieser Arbeit wird eine interaktive Pflanze als proaktives smartes Objekt entwickelt, die sich in ihrem menschlichen Umfeld bemerkbar machen kann, wenn sie Wasser braucht. Es wird damit aufgezeigt wie Objekte durch Proaktivität, Interaktivität und Erwecken von Bedürftigkeit den Benutzer dazu verleiten können, sich um diese kümmern bzw. diese zu warten.

Benjamin-Bodo Salewski

Title of the paper

Development of an interactive plant as a proactive and smart object

Keywords

smart objekt, companion, human computer interaction, HCI, plant-companion

Abstract

The most objects in our everyday life have not the capability to express themselves and cannot tell you if they need attention. In this work an interactive Plant is developed as a proactive and smart object that can become noticeable for its human environment if it needs water. It shows how objects are capable to get people to take care for them though proactivity, interactivity and arousing neediness.

Inhaltsverzeichnis

1	Einleitung	7
1.1	Hinführung	7
1.2	Ziel der Arbeit.....	8
1.3	Überblick über die Arbeit.....	9
2	Analyse	10
2.1	Szenario und Motivation.....	10
2.1.1	Location Awareness	11
2.1.2	Self Awareness	13
2.1.3	Time Awareness	13
2.1.4	Willingness	14
2.1.5	Communication.....	15
2.2	Voraussetzungen.....	16
2.2.1	CSTI (Creative Space for Technical Innovation) – The Lab.....	16
2.2.2	Das Framework	18
2.2.3	Die Methodik.....	18
2.3	Das Multiagentsystem.....	18
2.3.1	Die Agentenmetapher.....	18
2.3.2	Der Basisagent – Selbstauskunfts-fähigkeit	20
2.3.3	Kalendaragent	21
2.3.4	Präferenzagent.....	22
2.3.5	Kommunikationsagent.....	29
2.3.6	Person-Recognition-Agent	30

2.4	Anforderungen	31
2.5	Fazit	35
3	Design	37
3.1	Grobdesign	37
3.1.1	Sense	38
3.1.2	Reason	39
3.1.3	Act	39
3.1.4	Gründe für Sense Reason Act.....	40
3.2	Feindesign	40
3.2.1	Gründe für das MVC Pattern.....	41
3.2.2	Gründe für eine 3 Tier Architektur.....	42
3.3	Designkriterien	42
3.3.1	Stabilität	42
3.4	Die Systemarchitektur	42
3.4.1	Gründe für das agentenbasierte Framework.....	43
3.4.2	Kinect Data Provider	44
3.4.3	Umsetzung des Kinect Data Provider.....	45
3.4.4	Person Recognition Agent.....	45
3.4.5	Gründe für OpenCV?	48
3.4.6	Calendar-Agent	48
3.4.7	E-Mail-Agent.....	49
3.4.8	Gründe für Google Dienste	49
3.4.9	Smart-Plant-Agent.....	50
3.4.10	Ranking List Agent.....	51
3.4.11	Gründe für Angular2	51
3.4.12	Plant Agent.....	53
3.4.13	Gründe für die Verwendung einer C-Bibliothek.....	56
3.4.14	Gründe für die Text-to-Speech-Lösung "festival"	56
3.5	Verhalten der Pflanze bei Wassermangel	56
3.6	Fazit	60
4	Evaluation	61
4.1	Der Ist Zustand	61

4.1.1	Die Pflanze.....	62
4.1.2	Lokalisation und Identifikation.....	62
4.1.3	Time Awareness	63
4.1.4	Intelligenz	63
4.1.5	Die Rankingliste.....	64
4.2	Versäumnisse und Verbesserungen.....	65
4.2.1	Die Pflanze.....	65
4.2.2	Lokalisation und Identifikation.....	65
4.2.3	Time Awareness	66
4.2.4	Kommunikation.....	66
4.3	Fazit	66
5	Zusammenfassung und Ausblick	68
5.1	Zusammenfassung	68
5.2	Ausblick	68
6	Quellen	70

1 Einleitung

1.1 Hinführung

Computer werden immer kleiner, leistungsfähiger und effizienter. Sie nehmen wenig Platz ein, können viel leisten und verbrauchen dabei wenig Strom. Dadurch ist es möglich leistungsstarke Computer in alltäglichen Objekten zu verbauen, die mit einem Akku lange betrieben werden können. Dies hatte bereits Marc Weiser vorhergesehen, indem er spekuliert hat, dass Computer in der Größe von "Tabs" und "Pads" immer mehr Einzug in den Alltag erhalten werden (Weiser, 1999, S. 10). Diese Vorhersage hat sich in Form von Smartphones und Tablets bestätigt.

Die Computer verschwinden dadurch in alltäglichen Objekten, sie treten in den Hintergrund. Dies ist eine Entwicklungsstufe die viele Technologien durchlaufen. Ein Beispiel ist z.B. die Schrift (Weiser, 1999, S. 3). Man findet sie auf diversen alltäglichen Gegenständen. So auch auf Objekten, welche nur eine sehr kurze Lebenserwartung haben, wie beispielweise Bonbonpapier. Damit ist die Schrift nicht nur in den Hintergrund getreten, sondern wurde zu einer Technologie, die wegwerfbar wurde.

Dies geht einher mit der günstigen und massenhaften Produktion von Technologie. Im Fall der Schrift wurde dies durch die Erfindung des Buchdrucks eingeleitet.

Ein weiteres Beispiel sind Elektromotoren, von denen zu Beginn nur ein großer Elektromotor alle Gerätschaften in einer Fabrik antrieb. Später, als Elektromotoren günstiger, kleiner und effizienter wurden, wurde in jeder Maschine mehrere

Elektromotoren eingesetzt (Weiser, 1999, S. 3). Heutzutage findet man Elektromotoren in allerlei Alltagsgegenstände, wie z.B kleine Handventilatoren, deren Wert so gering ist, dass diese wegwerfbar werden.

Diese Kombination aus Preisverfall und Allgenwärtigkeit lässt Technologie in den Hintergrund treten. Sie wird dadurch Teil der Umwelt und wird nicht mehr aktiv wahrgenommen. Wenn man allerdings seiner Umwelt mehr Ausmerksamkeit schenkt, wird man feststellen, wo überall Schrift oder Elektromotoren eingewebt sind.

Diese Entwicklung ist auch bei Computern zu beobachten. Ein Beispiel hierfür ist der Raspberry Pi Zero, der in seiner Rechenleistung mit dem Raspberry Pi Model B+ vergleichbar ist (Raspberry Pi Foundation), dabei aber um ca. 2 cm kleiner (Stimson, 2015) (Adams, 2014) und kostet nur \$5 im Gegensatz zum Raspberry Pi B und seinen Nachfolgern, die bei \$20 bis \$35 liegen (Upton, 2015).

Die Verfügbarkeit von kleinen, leistungsstarken, günstigen und energieeffizienten Computern ermöglicht und vereinfacht die Entwicklung Smart Objects und Social Companions. Diese sind meist auf eine hohe Rechenleistung angewiesen, die einfache Mikrocontroller nicht bieten können. Trotzdem können nicht alle Aktionen lokal abgewickelt werden, wie z.B. rechenintensive Aktionen wie Gesichtserkennung. Da kommt eine andere Neuerung ins Spiel: IoT (Internet of Things). Dies bedeutet jedes "Ding" ist mit dem Internet verbunden (Friedemann Mattern, 2010). Neben den genannten Verbesserungen haben Kleincomputer nun auch die Möglichkeit mittels eingebauter Antenne sich drahtlos mit dem Internet zu verbinden. Dadurch ist es möglich Aufgaben aufzuteilen und zu delegieren. Rechenintensive Aufgaben wie Gesichtserkennung lassen sich dadurch auslagern.

1.2 Ziel der Arbeit

Das Ziel dieser Arbeit ist es eine proaktive und smarte Pflanze zu bauen, welche beim Benutzer ein mentales Modell für ein Objekt dahingehend erzeugt, dass dieses Objekt

(quasi) menschliche Attribute erhält und dadurch Empathie für dieses Objekt erzeugt werden kann.

Dies wird versucht, indem die Pflanze befähigt wird über Sprache, Piktogramme und Schrift zu kommunizieren. Sie soll ihre Befindlichkeit äußern und passende Personen zum Gießen heranziehen können. Dies tut die Pflanze, indem sie Bedürftigkeit bei Personen weckt.

1.3 Überblick über die Arbeit

Im Kapitel "Analyse" werden zuerst das Szenario sowie die Motivation besprochen. Danach werden die Voraussetzungen, die für die Arbeit gegeben waren, betrachtet und im Anschluss werden die theoretischen Grundlagen und verwandte Arbeiten beleuchtet. Zum Ende des Kapitels werden die Anforderungen für eine proaktive und smarte Pflanze definiert.

Das Kapitel "Design" leitet die Architektur des Softwaresystems her. Dabei wird die Betrachtungsweise zum Ende hin immer detaillierter. Am Ende des Kapitels werden die Abläufe im System betrachtet.

Die Erfassung des Ist-Zustands der proaktiven und smarten Pflanze erfolgt im Kapitel "Evaluation". Anschließend werden Versäumnisse betrachtet und Verbesserungsmöglichkeiten aufgezeigt.

2 Analyse

In diesem Kapitel werden die Anforderungen für einen proaktiven Companion erarbeitet. Dieser Pflanzen-Companion soll eine Antwort auf die zentrale Fragestellung dieser Arbeit liefern. Diese lautet:

„Kann man ein mentales Modell für ein Objekt beim Betrachter dahingehend erzeugen, dass dieses Objekt (quasi) menschliche Attribute erhält und dadurch Empathie für dieses Objekt erzeugt werden kann?“

Hierzu wird zunächst das Szenario gezeichnet und Teilaspekte dieses Szenario diskutiert. Anschließend werden verwandte Arbeiten zu diesem Thema betrachtet.

2.1 Szenario und Motivation

Das Szenario besteht aus einer Pflanze und einer oder mehreren Personen, die sich in der Nähe befinden können, aber nicht müssen. Die Pflanze ist proaktiv und kann über verschiedene Kanäle mit Personen kommunizieren. Mit Personen in der Nähe kann sie über Sprache und mit Personen in der Ferne über E-Mail kommunizieren. Sie ist in der Lage Personen in der Nähe zu identifizieren und zu lokalisieren. Wenn die Pflanze nun feststellt, dass sie gegossen werden muss, tritt sie mit Personen in Kontakt. Dabei versucht sie möglichst Personen in der Nähe und Pflanzenliebhaber anzusprechen und diese dazu zu bewegen sie zu gießen.

Die Kommunikation mit Personen lässt sich dem Bereich der Communication zuordnen. Die Pflanze kann ebenfalls erfassen, ob sie gegossen werden muss oder nicht. Dies wiederum umfasst den Aspekt der Self Awareness. Zudem kann die Pflanze feststellen welche Person sich wo in ihrer Nähe aufhält. Die Pflanze kann also Personen in ihrer Nähe identifizieren und zuordnen. Dies fällt in den Aspekt der Location Awareness.

Wenn die Pflanze feststellt, dass sie Wasser braucht, kontaktiert sie eine passende Person. Dies kann sie zum einen darüber entscheiden, ob die Person gerade Zeit hat oder nicht. Dies wird durch den Aspekt der Time Awareness abgedeckt. Wenn die Person Zeit hat, kommen andere Aspekte zum tragen. Inwiefern z.B. ist die Person bereit die Pflanze zu gießen? Wie hat die Person die Pflanze in letzter Zeit behandelt? Hat die Person die Pflanze missachtet und andere sie gießen lassen oder gar übergossen? Diese Fragen spielen neben der Time Awareness eine Rolle und stellen den Aspekt der Willingness dar. Das Szenario setzt sich somit aus fünf Aspekten zusammen:

- Location Awareness (Die Erfassung von Personen in der unmittelbaren Umgebung)
- Self Awareness (Die Erfassung des eignen Zustands)
- Time Awareness (Die Erfassung von zeitlicher Freiheit von Personen)
- Willingness (Die Erfassung der Bereitschaft von Personen)
- Communication (Die Kommunikation mit Personen)

2.1.1 Location Awareness

Zunächst wird die Location Awareness betrachtet. Die Pflanze soll wahrnehmen können, welche Personen in der Nähe sind und wo sich diese befinden. Dazu gibt es verschiedene Lösungsansätze.

Zum einen kann die Lokalisation und die Identifikation über etwas erfolgen, was die betreffende Person bei sich trägt oder tragen muss.

Die erste Möglichkeit wäre AR-Tracking (Advanced Realtime Tracking GmbH). Dabei werden sogenannte AR-Tags mit Hilfe von Infrarotkameras lokalisiert und identifiziert. Die Person müsste hierzu ein AR-Tag sichtbar bei sich tragen. Diese Methode hat allerdings viele Nachteile. Zum einen ist bereits das sichtbare Tragen eines solchen Tags problematisch.

Sobald der Tag verdeckt wird, z.B. durch ein Kleidungsstück, funktioniert das Tracking nicht mehr. Dadurch ist die Methode sehr umständlich für Personen im Umkreis. Der einzige Vorteil, das genaue Tracking im Millimeterbereich, wird zudem in dieser Genauigkeit nicht benötigt.

Die zweite Möglichkeit wäre das Tracking über Funksignale. Dabei wird der Ursprung des Funksignals eines Tags von vier Empfangsstationen geortet und getrackt. Die Person müsste bei dieser Methode ebenfalls einen Tag bei sich tragen. Diese Methode hat den Vorteil, dass der Tag nicht sichtbar sein muss. Allerdings muss gewährleistet sein, dass der Tag nicht funktechnisch abgeschirmt ist. Hierbei bleibt das Problem, dass die Person weiterhin etwas zusätzliches bei sich tragen muss. Der Nachteil der Umständlichkeit besteht also weiterhin.

Eine Variation der zweiten Möglichkeit ist statt eines Tags das Smartphone der Person zu tracken, da Smartphones selbst im Standby in gewissen Abständen Funksignale zur Synchronisation aussenden müssen. Hierbei besteht der Vorteil darin, dass man einen Gegenstand zum Tracking benutzt, den fast jeder bei sich trägt. Somit ist hier der Nachteil der Umständlichkeit nicht gegeben. Allerdings gibt einen anderen Nachteil. Smartphones werden nicht die ganze Zeit am Körper getragen. Manchmal wird es zeitweise abgelegt. Dann kann die Person nicht mehr getrackt werden.

Zum anderen kann die Lokalisation und Identifikation auch über Computer-Vision erfolgen. Die Person wird hierbei über ihr Gesicht erkannt und identifiziert. Die Erkennung des Gesichts ermöglicht auch den Körper zu erkennen und zu tracken. Es besteht also nicht die Notwendigkeit, dass die Person etwas bestimmtes bei sich trägt. Allerdings darf die Person nicht ihr Gesicht verhüllen. Die Identifikation kann allerdings erst dann erfolgen, wenn die Kamera das Gesicht der Person aufnehmen kann. Dies wird insbesondere problematisch, wenn die Person immer mit dem Rücken zur Kamera steht. Dies lässt sich aber durch mehrere Kameras, die den ganzen Raum abdecken, reduzieren. Der Nachteil an dieser Methode ist, dass sie sehr performancelastig ist und ihre Leistung von den Lichtverhältnissen abhängt.

Hier wurde sich trotz der Performancelast für die Gesichtsidentifikation entschieden, da diese für den Benutzer am wenigsten umständlich ist.

2.1.2 Self Awareness

Ein weitere Aspekt ist die Self Awareness. Die Pflanze soll ihren eigenen Zustand erfassen können. Hier gibt zwei Möglichkeiten.

Die erste Möglichkeit wäre die Erdfeuchte zu messen. Wenn die Erdfeuchte einen gewissen Wert unterschreitet, braucht die Pflanze Wasser. Wenn die Erdfeuchte allerdings einen gewissen Wert überschreitet, wurde sie übergossen. Die Methode ist relativ simpel und einfach umzusetzen da nur die Schwellenwerte bestimmt werden müssen. Diese von Pflanze zu Pflanze unterschiedlich sein. Manche Pflanzen brauchen weniger Wasser z.B. Kakteen als andere.

Ein weitere Möglichkeit ist den Zustand der Pflanze direkt zu messen z.B. durch Elektroden. Das dies prinzipiell möglich ist, zeigt Project Florence (Helene Steiner, 2015). Allerdings fehlt es hier an Grundlagen, wie die bioelektrischen Signale zu interpretieren sind, da diese Methode noch nicht ausgereift und erforscht ist und kommt deswegen derzeit nicht in Frage. Sie sei hier aber aufgeführt als zukünftige Alternative zur ersten Möglichkeit.

Hier wurde sich für die Messung der Erdfeuchte entschieden, da dies sehr einfach zu realisieren ist und da die zweite Möglichkeit noch nicht ausgereift ist.

2.1.3 Time Awareness

Der nächste Aspekt ist Time Awareness. Die Pflanze soll erfassen können, ob ihre Besitzer Zeit haben. Personen in der Nähe haben wahrscheinlich Zeit kurz eine Pflanze zu gießen. Aber wie verhält es sich, wenn keine Person in der Nähe ist und die Pflanze eine Person kontaktieren muss? Welche Person hat gerade Zeit und welche hat Termine? Hier gäbe es ebenfalls zwei Möglichkeiten.

Bei der ersten Möglichkeit könnte die Pflanze in den Terminkalendern der Personen schauen und entscheiden, wen sie kontaktiert. Dazu muss der Kalender allerdings online

verfügbar sein und die betreffende Person muss vorher der Pflanze den Zugriff erlaubt haben. Nun gibt es hier den Nachteil, dass die Personen überhaupt einen Online-Kalender führen müssen. Denn trotz der wachsenden Digitalisierung gibt es Menschen, die ihre Termine analog oder garnicht notieren.

Bei den anderen Option könnte die Pflanze Termine auch mündlich entgegennehmen. Die betreffende Person erzählt einfach der Pflanze, dass sie von dann bis dann beschäftigt ist und die Pflanze merkt sich das. Diese Methode hat den Vorteil, dass das Führen eines Online-Kalenders nicht nötig ist. Andererseits ist diese Methode performancelastig und ihre Leistung hängt von den akustischen Verhältnissen, dem Akzent oder dem Sprachfehler der Person ab.

Hier wurde sich für die erste Möglichkeit entschieden, da sich diese einfach umsetzen lässt und es wahrscheinlich scheint, dass durch die wachsende Digitalisierung immer mehr Menschen Online-Kalender benutzen werden.

2.1.4 Willingness

Der nächste Aspekt ist die Willingness. Die Pflanze soll entscheiden können, welche Person am besten für die Aufgabe geeignet ist. Diese Entscheidung hängt von mehreren Faktoren ab. Die eigentliche Eignung lässt sich vernachlässigen, da fast jeder eine Pflanze gießen kann. Interessanter ist die Haltung der Person gegenüber Pflanzen. Mag die betreffende Person Pflanzen oder nicht? Ein weiterer Faktor ist, wie oft hat die Person in der letzten Zeit mit der Pflanze interagiert und ob sie die Pflanze übergossen hat?

Der zweite Faktor lässt sich als Rankingliste sehen, in der die Person einen Punkt bekommt, wenn sie die Pflanze gießt und einen Punkt verliert, wenn sie die Pflanze übergießt. Die generelle Haltung gegenüber Pflanzen könnte die Person vorher angeben. Des Weiteren könnte die Pflanze die aktuelle Stimmung der Person in ihre Entscheidung mit einbeziehen. Dies könnte die Pflanze mit Kameras, die die Mimik der Personen analysieren, bewerkstelligen. Diese Methode ist allerdings sehr performancelastig.

2.1.5 Communication

Der letzte Aspekt ist die Communication. Hierbei geht es um die Kommunikationsmöglichkeiten der Pflanze mit ihrer Umwelt.

Zunächst einmal könnte die Pflanze schriftlich mit der Person in Kontakt treten z.B. über E-Mail oder Instant Messaging. Diese Methode eignet sich besonders, wenn die betreffende Person nicht in der Nähe ist. Diese Art der Kommunikation kann man unidirektional oder bidirektional gestalten. Unidirektional würde bedeuten, die Pflanze sendet ein Gießanfrage ab und wartet eine gewisse Zeit bis sie eine andere Person anfragt. Bidirektional hingegen bedeutet, die Person kann antworten und z.B. absagen. So muss die Pflanze nicht warten und kann sich direkt an eine andere Person wenden. Die bidirektionale Form dieser Methode hat allerdings den Nachteil, dass Textinterpretation nötig wird und dies mit Aufwand verbunden ist. Der Vorteil ist, dass bereits vorhandene Kommunikationskanäle genutzt werden.

Eine Abwandlung der ersten Möglichkeit wäre die Verwendung einer Smartphone-App, über welche Gießanfragen eingehen und diese auch beantwortet werden können. Der Vorteil hieran ist, dass man die Textinterpretation umgeht. Allerdings muss die Person extra eine App installieren und kann nicht bereits vorhandene Kommunikationskanäle verwenden.

Zum anderen kann die Pflanze über Sprache kommunizieren. Dies eignet sich besonders, wenn sich die Person in der Nähe befindet. Diese Art der Kommunikation kann man ebenfalls unidirektional oder bidirektional gestalten. Zudem kann man sie personalisieren. Unidirektional würde bedeuten, dass die Pflanze einfach, wenn eine oder mehrere Personen in der Nähe sind, eine Aufforderung zum Gießen äußert. Um die Aufforderung zu personalisieren könnte die Pflanze noch den Namen der betreffenden Person anfügen, die sie gießen soll. Die Pflanze würde daraufhin eine gewisse Zeit warten und dann eine andere Person anfragen. Ist die Kommunikation nicht personalisiert, könnte die Pflanze die Aufforderung ein letztes Mal wiederholen und dann die schriftliche Kommunikation wählen.

Bei der bidirektionalen Variante könnte die Person antworten und z.B. sagen, dass sie gerade beschäftigt ist. Die Pflanze müsste dann nicht warten, sondern könnte sich an eine andere Person wenden. Diese Art der Kommunikation bedingt eine Personalisierung.

Die sprachliche Kommunikation lässt sich noch durch Mobilität seitens der Pflanze erweitern. Die Pflanze könnte zu der betreffenden Person hinfahren und sie dann ansprechen. Hierbei wäre die Personalisierung nicht mehr so wichtig, da klar ist, wen die Pflanze anspricht. Der Nachteil ist der Aufwand. Es müsste eine fahrbare Plattform für die Pflanze gebaut werden. Es stellt sich die Frage, wie die Pflanze in Räumen navigieren kann, sodass sie nicht gegen Hindernisse fährt oder gar umfällt. Diese und andere Fragen müssten zuerst geklärt werden.

Hier wurde sich für die unidirektionale und personalisierte Variante der E-Mail-Kommunikation und verbalen Kommunikation entschieden, da diese einfach zu realisieren sind.

2.2 Voraussetzungen

2.2.1 CSTI (Creative Space for Technical Innovation) – The Lab

Die Umsetzung erfolgte im CSTI (Abbildung 1). Dabei handelt es sich um ein Labor auf dem HAW Campus (HAW Hamburg). Es beinhaltet unter anderem eine Traverse, in der LED Stage Bars, Studiolautsprecher, dynamische Mikrophone, Infrarottrackingkameras, funkbasierte Trackingsensoren, Wärmekameras, Hochgeschwindigkeitskameras und vieles mehr installiert sind.

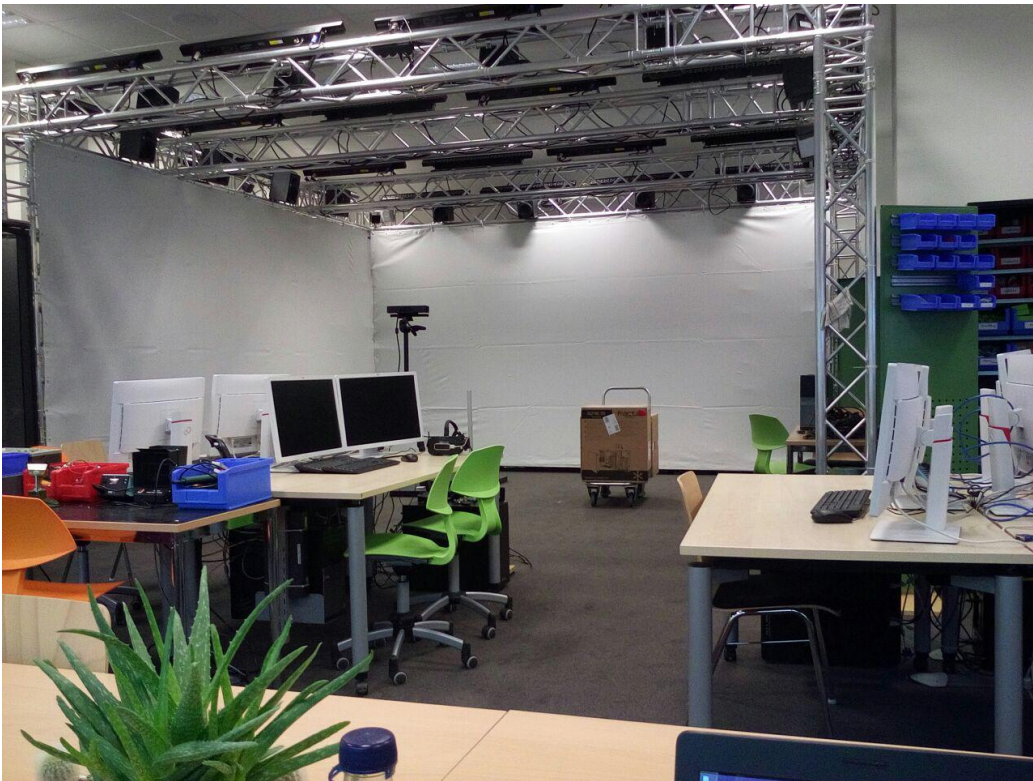


Abbildung 1: Das CSTI

Dieser Aufbau ermöglicht Experimente mit 360°-Surround Sound, Raumbeleuchtung, Erkennung von Emotionen, Gesichtern, Gesten, Sprache und vielem mehr. Neben der Traverse bietet das CSTI leistungsstarke Computer für allerlei Berechnungen und eine große Auswahl an elektronischen Komponenten und Entwicklungsplattformen, die Rapid Prototyping ermöglichen. Das CSTI umfasst zurzeit folgende Forschungsgebiete (HAW Hamburg):

- Companion-Technologie (basierend auf dem Individuum, dem Kontext oder den Emotionen)
- Sense-Reason-Act
- Virtual Reality und Augmented Reality
- Internet of Things und 3D-Druck
- Urban- und Smart-Home-Umgebungen

Damit bot es für diese Arbeit die perfekten Voraussetzungen.

2.2.2 Das Framework

Hinter dem Kulissen des CSTI sorgt ein Framework dafür, dass die diversen Sensoren und Aktoren miteinander kommunizieren können. Entwickelt wurde es von Tobias Eichler und basiert auf einer Publish/Subscribe-Architektur (Eichler, 2014). Die Kommunikation erfolgt über eine Middleware, die alle Nachrichten empfängt und sie dann an die Abonnenten weiterleitet. Die Nachrichten werden in sogenannte Gruppen veröffentlicht und welche sich wiederum abonnieren lassen. So lassen sich wiederverwendbare Module bauen, die auch in anderen Projekten verwendet werden können. Zudem lassen sich damit lose gekoppelte Multiagentensysteme bauen.

2.2.3 Die Methodik

Die Methodik, die hinter dem CSTI steckt, ist die sogenannte „Lifelong Kindergarten“-Methodik, die auf einen spielerischen Ansatz setzt (Resnick, 2001). Das heißt, es wird nicht lange geplant, sondern sofort begonnen einen Prototypen zu bauen, mit dem experimentiert wird. Aus den Experimenten lassen sich Verbesserungen oder neue Ideen für den Prototypen ableiten, auf deren Basis der Prototyp weiterentwickelt wird. Mit diesen weiterentwickelten Prototypen lässt sich wieder experimentieren und somit beginnt der Kreislauf von Experimentieren und Weiterentwickeln von vorne. Damit lassen sich schnell Fortschritte und Ergebnisse erzielen. Diese Methodik passt somit sehr gut zum genannten Framework und Rapid Prototyping.

2.3 Das Multiagentsystem

2.3.1 Die Agentenmetapher

Hinter der Agentenmetapher steckt die Idee von kleinen unabhängigen Programmen, die Services anbieten oder einen kleinen Teil der Arbeit in einem Gesamtsystem übernehmen. Auch Sensorknoten lassen sich als Agenten sehen, die Sensordaten zur Verfügung stellen.

Wenn mehrere Agenten in einem System Services anbieten oder zusammen arbeiten, handelt es sich um ein Multiagentensystem. Dazu hatte bereits Wooldridge geforscht. Er hat drei Grundeigenschaften für einen Agent herausgearbeitet:

- *“Reaktivität – Intelligente Agenten können ihre Umgebung beobachten und auf Änderungen, die in dieser auftauchen, reagieren, um ihren Designanforderungen gerecht zu werden.”* (Wooldridge, 2002, S. 23)
- *“Proaktivität – Intelligente Agenten sind in der Lage ein zielgerichtetes Verhalten zu zeigen, indem sie die Initiative ergreifen, um ihren Designanforderungen gerecht zu werden.”* (Wooldridge, 2002, S. 23)
- *“Soziale Fähigkeiten – Intelligente Agenten sind in der Lage mit anderen Agenten und möglicherweise mit Menschen zu interagieren, um ihren Designanforderungen gerecht zu werden.”* (Wooldridge, 2002, S. 23)

Aber auch das Multiagentensystem hat Eigenschaften, die Wooldridge herausgearbeitet hat (Wooldridge, 2002, S. 189-190).

- Autonomie – Agenten sind zumindest partiell eigenständig, selbstbewusst und autonom
- Lokale Sichten – kein Agent hat eine komplette Sicht auf das System
- Dezentralisation – es gibt keinen zentralen Agenten oder Kontrollpunkt, der alle anderen Agenten kontrolliert.

Das heißt neben dem Sensorknoten, der Daten über die Erdfeuchte liefert, bedarf es noch weiterer Agenten um einen emotionalen Pflanzen-Companion zu bauen. Da wäre zum einen der Kalenderagent, der dafür zuständig ist, Kalenderdaten der betreffenden Person abzurufen und zu entscheiden, ob die Person gerade Zeit hat. Zum anderen wäre da der Person-Recognition-Agent, der Personen im Raum identifiziert und lokalisiert und diese Daten bereit stellt. Außerdem braucht es einen Präferenzagenten, der Personen anhand ihres Umgangs mit der Pflanze einstuft. Und zum Schluss bedarf es einen Kommunikationsagenten, der für die Kommunikation mit dem Menschen zuständig ist.

2.3.2 Der Basisagent – Selbstauskunftsfähigkeit

Damit ein Gegenstand selbstauskunftsfähig wird, muss dieser seinen eigenen Zustand erfassen können. Dies ist mit einem Sensor möglich. Allerdings muss der erfasste Zustand kommuniziert werden. Hier wäre eine Funkverbindung in Form von WLAN oder Bluetooth denkbar. Ein Sensor kombiniert mit einer Kommunikationsmöglichkeit ergibt einen sogenannten Sensorknoten.

Ein erster Sensorknoten, der Motescope (University of California, Berkeley, 2011), wurde bereits in der Soda Hall der Universität Berkeley entwickelt. Diese Sensorknoten lassen sich zu einem Sensorknotennetz zusammenschalten. Dadurch ist es möglich Anwendungen zu entwickeln, die sich auf die Daten von vielen verschiedenen Sensoren stützen. Allerdings bilden hierbei der Sensor und das Kommunikationsmodul eine untrennbare Einheit. Das macht diese Form der Sensorknoten relativ unflexibel.

Man ging dazu über die einzelnen Komponenten zu trennen, aber die Möglichkeit offen zu lassen, diese bei Bedarf wieder zusammenschalten zu können. Ein Beispiel hierfür ist die Prototyping-Plattform Arduino (Arduino, 2017). Der Arduino ist ein Mikrocontroller, an den man relativ einfach weitere Sensoren oder andere Module wie z.B. ein WLAN-Shield anschließen kann. Dadurch ist man sehr flexibel und kann Komponenten für spätere Experimente wiederverwenden.

Im Zuge des Internets der Dinge (IoT) ging man dazu über zumindest den Mikrocontroller und das WLAN-Modul zusammenzufassen. Dadurch muss man sich nicht mehr um den Anschluss des WLAN-Moduls kümmern, wenn man es ohnehin fast immer braucht. Auch wird der experimentelle Aufbau kompakter.

Nachfolgend wird das Erfassen des Zustands der Pflanze betrachtet. Zum einen kann an den äußeren Bedingungen der Zustand der Pflanze abgeschätzt werden, wie z.B. an der Erdfeuchte. Zum anderen kann der Zustand der Pflanze direkt erfasst werden, wie es im Project Florence umgesetzt wurde. Im Project Florence (Helene Steiner, 2015) werden durch Elektroden an den Blättern schwache elektrische Impulse abgefangen, die durch Lichtimpulse ausgelöst wurden, interpretiert. Diese Lichtimpulse sind eine übersetzte

Nachricht vom Benutzer an die Pflanze und die elektrischen Impulse sind eine Nachricht von der Pflanze an den Benutzer, die übersetzt in Worte ausgegeben wird.

Es stellt somit eine Art Pflanzen-Companion dar. In diesen Bereich fällt auch PotPet (Kawakami, Tsukada, Kambara, & Siio, 2011): Ein Blumentopf-Roboter, der sich autonom wie ein Haustier verhält. Es fährt selbstständig an sonnige Plätze und sucht seinen Besitzer auf, wenn es Wasser braucht und freut sich, wenn es gegossen wurde. Ein weiterer Pflanzen-Companion ist EmotiPlant (Angelini, Caparrotta, Khaled, & Mugellini, 2016). EmotiPlant kann ihren Zustand emotional durch ein Display über Smileys äußern. Dadurch soll die Pflege der Pflanze erleichtert werden.

Diese Arbeit setzt im Gegensatz zu Project Florence (Helene Steiner, 2015) bei der Erfassung des Zustands der Pflanze auf die Erdfeuchte, da die Interpretation der bioelektrischen Signale sehr aufwendig ist und Grundlagen dazu fehlen. Des Weiteren grenzt sich diese Arbeit zu PotPet (Kawakami, Tsukada, Kambara, & Siio, 2011) ab, da bei der Proaktivität auf verbale und schriftliche Kommunikation gesetzt wird und nicht auf nonverbale Handlungen als Kommunikation. Die Ideen und Ansätze von EmotiPlant übernimmt diese Arbeit als Teilaspekt des Gesamtsystems. Hierbei wäre die Verwendung von Smileys zum Darstellen des Zustands der Pflanze zu erwähnen. Allerdings bewegt sich diese Arbeit im Bereich von Smart Environments und EmotiPlant hingegen fasst mehr den Bereich Elder Care (Angelini, Caparrotta, Khaled, & Mugellini, 2016, S. 1).

2.3.3 Kalenderagent

Für die Time Awareness ist der Kalenderagent zuständig. Er kann Termine aus einem oder mehreren Online-Kalendern abfragen und beurteilen, ob eine Person Zeit hat. Allerdings ist mehr zu beachten als freie Termine. Dazu hat bereits Erik T. Müller geforscht. In seiner Arbeit präsentiert er einen Kalender mit „Common Sense“ den SensiCal (Mueller, 2000). Dieser intelligente Kalender unterstützt Benutzer, in dem er Benutzereingaben entgegen nimmt, diese analysiert und fehlende Eingaben ergänzt oder intelligente Defaults angibt. Zum Beispiel wenn der Benutzer nur „Dinner“ angibt, wird automatisch zwei Stunden als

Dauer vorgeschlagen. Außerdem werden gewisse Grundannahmen getroffen wie z.B. dass Menschen Nachts schlafen. Wenn nun versucht wird, einen Termin für 3 Uhr zu erstellen, wird eine Warnung ausgegeben. Auch wird beachtet, dass man Reisezeit benötigt um vom einem Ort zum anderen zu gelangen. Neben Annahmen, wie dass eine Person nicht an zwei Orten zur gleichen Zeit sein kann, werden auch logische Schlussfolgerungen, wie z.B. dass ein Steakhouse nicht die beste Wahl ist um mit einem Vegetarier essen zu gehen, getroffen. Dieser „Common Sense“ wäre auch für den Kalenderagent von Vorteil. Zum Beispiel wäre so schon im Vorhinein klar, dass um 3 Uhr Nachts keine Person Zeit zum Gießen hat. Auch könnten Reisezeiten mit einbezogen werden. Wenn zum Beispiel eine Person in naher Zukunft einen Termin hat, könnte der Kalenderagent berechnen, ob es die Person zeitlich schafft, zur Pflanze zu gelangen, sie zu gießen und den Termin wahrzunehmen.

Diese Arbeit übernimmt die Idee des „Common Sense“ für die Time Awareness. Allerdings steht hier nicht im Vordergrund dem Benutzer das Eingeben von Terminen zu erleichtern, sondern der Pflanze neben Terminen noch andere Entscheidungshilfen an die Hand zu geben. Des Weiteren wird auf eine reduzierte Version dieses „Common Sense“ gesetzt, da die Ausrede-Metapher im Mittelpunkt steht. Dies grenzt diese Arbeit von SensiCal ab.

2.3.4 Präferenzagent

Für die Willingness ist der Präferenzagent verantwortlich. Er entscheidet anhand des Verhaltens der Person gegenüber der Pflanze, ob diese fürs Gießen infrage kommt oder nicht. Dabei wird beurteilt, wie die Person die Pflanze in der Vergangenheit gegossen hat: ob zu viel, zu wenig oder trotz Gießaufforderung garnicht gegossen wurde. Auch wird die generelle Haltung gegenüber Pflanzen in die Entscheidung mit einbezogen. Die Pflanze soll also gegenüber der Person Emotionen simulieren.

Der emotionale Bereich umfasst zwei Ebenen. Die erste Ebene ist die emotional-adaptive Ebene. Die Pflanze stellt sich auf die Gefühle und Vorlieben ihres menschlichen Gegenübers ein und bezieht diese in ihr proaktives Verhalten mit ein. Die zweite Ebene ist

ego-emotionale Ebene. Die Pflanze hat selbst Gefühlszustände und äußert diese auch. Hierbei wird später noch betrachtet wie Menschen auf diese Äußerungen reagieren bzw. sich dadurch verändern.

Objekte, die der emotional-adaptiven Ebene angehören, werden auch „Companions“ (Wilks, 2010, S. 11) oder „Socialable Robots“ (Wilks, 2010, S. 3) genannt. Hierzu hatte bereits Yorick Wilks geforscht und sich mit den sozialen, psychologischen, ethnischen und design-technischen Aspekten befasst. Für diese Arbeit von besonderen Interesse sind die design-technischen Aspekte. Hierbei beschreibt Wilks, dass ein Companion ein guter Zuhörer sein muss, damit sein Gegenüber sich verstanden und umsorgt fühlt (Wilks, 2010, S. 131). Außerdem muss er empathisch sein, um auf die Gefühlszustände seines Gegenübers angemessen reagieren zu können. Dabei lässt sich Empathie in drei Arten aufteilen (Wilks, 2010, S. 138-141):

- parallele Empathie
- reaktive Empathie
- absichtliche Empathie

Die parallele Empathie stellt eine Spiegelung des Gemütszustands des Gegenüber dar. Wo hingegen die reaktive Empathie mit einer verwandten Emotion, aber nicht der selben Emotion reagiert. Die absichtliche Empathie hingegen versucht den Gefühlszustand ihres Gegenübers zu verändern. Das heißt, wenn ein Freund traurig ist, würde eine parallele Empathie ebenfalls mit Traurigkeit antworten, eine reaktive Empathie hingegen mit Mitgefühl und eine absichtliche Empathie würde versuchen das Gegenüber aufzumuntern.

Objekte der ego-emotionalen Ebene ahmen menschliche Emotionen nach, um die Kommunikation mit Menschen zu verbessern. Dies funktioniert selbst, wenn die Kommunikation auf Text limitiert ist. Auch die Glaubwürdigkeit des Objekts steigt, wenn es Emotionen äußern kann.

Empirische Studien haben gezeigt, dass Menschen Computer als soziale Akteure behandeln und dass die Meinung des Menschen über den Computer besser ist, wenn dieser ihm Komplimente macht und schlechter, wenn dieser ihn kritisiert (Reeves & Nass, 1996, S. 53-58). Außerdem wurde festgestellt, dass Menschen auf Medien mit der gleichen Varietät an Emotionen reagieren wie in einer Mensch-zu-Mensch-Kommunikation (Reeves & Nass, 1996, S. 131-135).

Aber wie verändern sich Menschen, wenn Roboter oder Objekte Gefühle zeigen? Yorick Wilks stellte fest, dass, wenn Roboter Gefühle zeigen, wir uns um sie kümmern wollen und im Gegenzug auch wollen, dass sie sich auch um uns kümmern. Wir sehen also nicht mehr eine Maschine vor uns, sondern immer mehr ein lebendiges Wesen (Wilks, 2010, S. 4). Hierfür ist Paro, ein kleiner Seehundroboter, ein Beispiel (Wada & Shibata, 2007). Sein niedliches Aussehen veranlasst Menschen sich mit ihm zu beschäftigen. Aber der Mensch isoliert sich dadurch nicht von anderen, sondern knüpft sogar neue Kontakte zu anderen Menschen. In einer Studie wurde Paro in einen Altersheim platziert und die Auswirkungen analysiert (Wada & Shibata, 2007, S. 976-977). Durch das Spielen mit Paro kamen die Bewohner auch mit anderen Bewohnern in Kontakt, die sich auch an Paro interessiert zeigten. Man konnte zeigen, dass Paro das soziale Netzwerk im Altersheim ausdehnte und enger knüpfte.

In der Arbeit von Martin Cooney, Shuichi Nishio und Hiroshi Ishiguro wurde untersucht wie Menschen auf verschiedene Reaktionsstrategien reagieren (Cooney, Nishio, & Ishiguro, 2014). Dazu bauten sie einen kleinen Roboter namens Kakapo (Abbildung 2), der sich optisch, akustisch und haptisch verständigen kann und wie folgt aussieht:



Abbildung 2: Kapako (Cooney, Nishio, & Ishiguro, 2014, S. 19:2)

Der Roboter konnte drei verschiedene Reaktionsstrategien anwenden:

- haptophil (reagiert auf jede Berührung positiv)
- haptoneutral (reagiert, je nachdem welche Stelle berührt wurde, negativ oder positiv)
- haptophobisch (reagiert auf jede Berührung negativ, aber signalisiert versteckt, dass er es mag an machen Stellen berührt zu werden)

Die haptophile Strategie fanden einige Probanden seltsam, da der Roboter positiv reagierte, obwohl er an einer Stelle berührt wurde, die ihm eigentlich unangenehm sein sollte. Die haptoneutrale Strategie empfanden einige Probanden hingegen als unbeständig und reaktiv. Der Roboter reagierte vorhersehbar. Seine schnellen Gemütschwünge ließen einige Probanden zudem glauben, dass er sie nicht wirklich mochte. Die haptophobische Strategie ließ die Probanden in dem Gefühl, dass der Roboter sie nicht mochte.

Im Gegensatz zu Paro oder Kakapo setzt diese Arbeit auf eine berührungslose Interaktion durch Gießen. Auch grenzt sich diese Arbeit zu Paro ab, indem Bedürftigkeit nicht durch ein niedliches Äußeres, sondern durch Smileys und einen Pflanzen-Avatar erzeugt wird. Aus der

Arbeit zu Kakapo wird eine neutrale Reaktionsstrategie übernommen. Das heißt die Pflanze reagiert positiv, wenn sie durstig ist und gegossen wird und negativ, wenn eine Person auf eine Gießanfrage nicht reagiert.

Nun scheint es so, dass die meisten Menschen Pflanzen nicht als Lebewesen, sondern als Objekte wahrnehmen, obwohl nach biologischen Kriterien Pflanzen Lebewesen sind. Ein Beispiel ist die Verwendung als Einrichtungsgegenstand. Nun berufen sich Menschen bei der Einordnung, ob etwas lebendig ist oder nicht, nicht rein auf die biologischen Kriterien, sondern legen sich eigene Kriterien zurecht. Diese verändern sich mit dem Alter und können sich mit biologischen Kriterien decken.

Zu diesem Schluss kommt Sherry Turkle, die dazu geforscht hat (Turkle, 2005, S. 33-64). Sie hatte untersucht, wie Kinder in verschiedenen Altersstufen auf Computer-Spielzeuge reagieren und wie diese auf die Frage antworten, ob das Spielzeug lebendig ist. Dabei hatte sie drei Phasen ausgemacht, die Kinder durchlaufen. In jeder Phase ist eine psychologische Komponente vertreten, die langsam die physische Komponente verdrängt.

In der ersten Phase beurteilen die Lebendigkeit eines Objekts anhand seiner Bewegung (Turkle, 2005, S. 319, 53-54). Im Alter von vier bis sechs Jahren sind das 67 % der insgesamt 25 beobachteten Kinder. Wenn ein Objekt sich aus eigener Kraft heraus bewegen kann, ist es lebendig. Für ein Kind in dieser Phase ist z.B. ein Stein, der eine Rampe runter rollt, lebendig, da er sich scheinbar von selbst bewegt. Dabei wird dem Stein auch die Intention zugesprochen, dass er z.B. nach unten wollte. Hierbei wird die psychologische Komponente lediglich als Begründung für die Bewegung hergenommen. Die physische Komponente, die Bewegung, ist hierbei von zentraler Bedeutung bei der Beurteilung von lebendig und nicht lebendig. Außerdem können bestimmte Aktionen nicht ausgeführt werden, wenn die anatomischen Voraussetzungen dafür fehlen. Ein Tic-Tac-Toe-Spielzeug Namens „Merlin“ (Turkle, 2005, S. 372), ein unscheinbarer Kasten mit einer Eingabefläche für den Benutzer, der gegen den Computer spielen kann, kann aus Augen eines Kindes nicht schummeln, da es keine Hände hat. Hierbei ist anzumerken, dass Merlin durch optimierte Strategien fast immer gewinnt.

In der zweiten Phase ist die physische Komponente und die menschliche Anatomie nicht mehr so wichtig (Turkle, 2005, S. 319, 53-54). Im Fall von Merlin heißt dies, dass Merlin zum Schummeln keine Hände benötigt. In dieser Weise argumentieren 50 % der beobachteten Kinder im Alter von sieben bis acht Jahren. Er hat seinen eigenen Weg die Knöpfe zu drücke. In der dritten Phase spielt die physische Komponente keine Rolle mehr (Turkle, 2005, S. 319, 53-54). Diese Phase haben 80 % der beobachteten Kinder mit neun Jahren oder älter erreicht. Um ein Spiel wie Tic-Tac-Toe zu spielen, kommt es nicht mehr auf die Anatomie oder die Fähigkeit zu Handeln an, sondern auf mentale Zustände. "Um Tic-Tac-Toe zu spielen braucht man kein Gehirn, sondern einen Verstand", wie ein Kind es ausdrückte (Turkle, 2005, S. 53). Im Fall von Merlin heißt dies, dass Merlin sich bewusst sein muss, dass er schummelt. Wenn er sich diesem nicht bewusst ist, schummelt er auch nicht und ist somit auch nicht lebendig.

Mit fortschreitendem Alter wird die psychologische Argumentation feiner und differenzierter. Computer sind nicht lebendig, da sie keinen freien Willen, Emotionen oder Originalität haben. Dabei werden Emotion als Hauptmerkmal für Lebendigkeit gesehen. Allerdings merkt Sherry Turkle an, dass dieses harte Trennen von Gedanken und Gefühlen dazu führt, dass die komplexe Beziehung zwischen diesen beiden Bereichen außer Acht gelassen wird. Dies führt dazu, dass das Kognitive zu einem rein logischen Prozess verkommt, der dadurch kalt, trocken und leblos daher kommt. Aber auch das Affektive wird dadurch auf das Intuitive, Primitive und Unanalysierbare reduziert. Diese scharfe Trennung von Intellekt und Emotionen kann schnell zu einer seichten und sentimental Definition von Gefühlen führen. Gedanke und Gefühle sind nach Turkle untrennbar miteinander verknüpft.

Dies bedeutet Menschen empfinden etwas als lebendig, wenn es Emotionen hat bzw. äußert. Wenn etwas als lebendig empfunden wird, neigen Menschen eher zu Empathie dem gegenüber, als wenn etwas als nicht lebendig empfunden wird.

Nun lässt sich das System aus Pflanze und deren Besitzer auch mit der Akteur-Netzwerk-Theorie (ANT) beschreiben. Die Akteur-Netzwerk-Theorie betrachtet Systeme als ein Netzwerk von menschlichen und nicht menschlichen Akteuren (auch Aktanten), die über Assoziationen miteinander verbunden sind (Belliger, 2006, S. 30-37, 195-211).

Angewendet auf das System aus Pflanze und deren Besitzern bedeutet dies, dass die Pflanze ein nicht menschlicher Aktant und die Besitzer menschliche Aktanten sind. Nun bestehen zwischen den menschlichen Aktanten zwar Beziehungen sogenannte Assoziationen, allerdings sind diese nicht kohärent, sondern fragil. Es können z.B. Konflikte zwischen den Besitzern entstehen, die zur Auflösung des Netzwerkes führen können. Diese assoziierten Entitäten können "sich wie ein Schwarm Vögel zerstreuen" (Belliger, 2006, S. 392). Ein Akteur-Netzwerk ist also nicht sehr stabil. Aktanten sind mit einem „leicht zerreißbaren Band“ miteinander verbunden, wie es Latour beschreibt (Latour, 2002, S. 59). Dieses fragile Band aus Assoziationen zwischen den Besitzern kann durch die Pflanze stabilisiert werden. Sie tritt als nicht-menschlicher Aktant in Assoziationen zwischen den Besitzern auf. Alle Besitzer sind zumindest über die Pflanze verbunden. Durch die Technisierung der Pflanze, erfährt die Pflanze einen Substitutionsvorgang. Durch diesen sich ihre Existenzform ändert. Ihre Existenz erhält dadurch mehr Essenz, da sie ein „Ding der Technik“ wird (Bonz, 2005, S. 4). Diese technisierte Pflanze ist stabiler im Vergleich zu menschlichen Aktanten. Denn sie ist eine Black Box. Sie ist daher mehr als ein Aktant. Wenn man ein Blick in die Black Box wirft, erblickt man weitere Assoziationen von menschlichen und nicht menschlichen Aktanten. Man erblickt ein weiteres Netzwerk, welches durch Blackboxing stabilisiert wurde, indem die Komplexität reduziert und Rollen fixiert wurden. Ein fragiles und instabiles Netzwerk wurde zu einem Makro-Akteur, welcher nach der ANT nicht größer ist als ein Mikro-Akteur (Belliger, 2006, S. 43-44). Man erblickt Teile, Geräte und Dinge, die wiederum weitere Assoziation enthalten, welche nicht zuletzt menschliches Tun in verdinglichter Form darstellen. Handlungen, Ideen, die unter Umständen lang zurück liegen können.

Die Pflanze tritt damit an die Stelle einer kulturellen Funktion, die herkömmlicher Weise der Konvention zugeschrieben wird. Die Pflanze hält also eine Gruppe von Menschen zusammen, wie es herkömmlicher eine Konvention es täte (Bonz, 2005, S. 3).

Dieses Phänomen lässt auch bei anderen Companions beobachten z.B. Paro (Wada & Shibata, 2007). Auch Paro festigt bereits vorhandene Assoziation zwischen den Altersheimbewohnern. Darüber hinaus schafft Paro neue Assoziationen zwischen den Bewohnern und stabilisiert diese.

Nun befinden sich Akteur-Netzwerke im ständigen Wandel und in andauernder Wiedererschaffung (Latour, 2007, S. 62-67). Das heißt, in Akteur-Netzwerken müssen bestimmte Beziehungen wiederholt vollzogen werden, damit diese sich nicht auflösen. Im Fall von Paro müssen sich die Bewohner jeden Tag mit Paro beschäftigen, damit das soziale Netz zwischen ihnen erhalten bleibt. Das Gleiche gilt für die Pflanze. Allerdings ist es auch im Interesse der Pflanze dieses Netz aufrecht zu erhalten, damit sie mit ausreichend Wasser versorgt wird. Dazu wird die Pflanze proaktiv und versucht zumindest eine Person zum Gießen zu bewegen.

2.3.5 Kommunikationsagent

Um die Communication kümmert sich der Kommunikationsagent. Dabei muss er für zwei Kommunikationswege Sorge tragen:

- Near-Distance-Communication
- Long-Distance-Communication

Die Near-Distance-Communication kann durch Sprache realisiert werden. Aber das es auch anders geht, zeigt InfoPlant (Hammerschmidt, Hermann, Walender, & Krömker, 2015). Eine Pflanze die über verschiedene Kanäle mit dem Menschen interagieren und kommunizieren kann. Sie kann durch Drehung ihre Haltung ändern, durch Ventilatoren an ihrer Basis mit ihren Blättern wackeln, durch RGB-LEDs ihr Aussehen ändern, Sound ausgeben und dadurch Output tätigen. Zum anderen kann der Mensch durch drehen der Pflanze oder

durch Ziehen an den Blättern Input tätigen. Eine weitere Art der Kommunikation zeigt Keepon (Michalowski, Sabanovic, & Kozima, 2007). Ein kleiner Roboter, der über rhythmische Bewegungen mit Menschen interagiert und kommuniziert. Der Rhythmus wird aus der Geräuschkulisse und visuellen Stimuli ermittelt und der dominanteste Rhythmus ausgewählt. Der Roboter synchronisiert sich dann zu diesem Rhythmus. Beispielsweise von der Person, die vor ihm tanzt. Wenn keine Person da ist, tanzt der Roboter zum Rhythmus der Hintergrundmusik oder Geräuschkulisse und animiert so vorbeigehende Person zum Mittanzen. Visuelle Stimuli haben also Vorrang zu akustischen Stimuli.

Diese Arbeit grenzt sich zu InfoPlant und Keepon ab, da nicht auf eine taktile Interaktion oder rhythmische Interaktion, sondern auf eine abstrakte Interaktion in Form des Gießens gesetzt wird.

Die Long-Distance-Communication kann über E-Mail, Instant-Messaging oder einer App realisiert werden.

2.3.6 Person-Recognition-Agent

Der Person-Recognition-Agent ist für die Identifikation und Lokalisation von Personen zuständig. Die Lokalisation erfolgt über die Erkennung des Körpers der Person und die Identifikation wird über Gesichtserkennung realisiert.

Dazu muss der Agent erstmal Gesichter lernen. Dabei werden eine bestimmte Anzahl von unterschiedlichen Bildern von Gesichtern einer Person gelernt. Dieses Lernset an Bildern wird je nach Algorithmus anders verarbeitet. Auch die drei bekanntesten Algorithmen Eigenfaces, Fisherfaces und Local Binary Patterns Histograms (LBPH), die auch in OpenCV (OpenCV) enthalten sind, gehen hier verschiedene Wege. Eigenfaces (Turk & Pentland, 1991) findet die dominanten Merkmale eines Lernsets und erhöht damit die Varianz als Ganzes. Fisherfaces hingegen erhöht die Varianz zwischen Klassen und verringert sie innerhalb von Klassen (Belhumeur, Hespanha, & Kriegman, 1997). Eine Klasse ist hierbei ein Set von Bildern von ein und dem selben Gesicht. Zwei Bilder eines Gesichts können durch Lichtverhältnisse und Blickwinkel eine höhere Varianz aufweisen als ein Wechsel in der

Identität es täte. Daher strebt der Fisherfaces-Algorithmus eine Verringerung der Varianz innerhalb von Klassen und eine Erhöhung der Varianz zwischen Klassen an. Während Eigenfaces und Fisherfaces einen ganzheitlichen Ansatz verfolgen, verfolgt der Local-Binary-Patterns-Histograms-Algorithmus einen lokalen Ansatz (He & Wang, 1990). Hierbei wird das Bild erst in sogenannte "Texture Units" aufgeteilt, welche die lokalen Charakteristika beschreiben, welche wiederum in einem "Texture Spectrum" zusammengefasst werden, welches die Verteilung der "Textue Units" im Bild beschreibt. Dadurch reicht schon ein Sample-Bild zur Gesichtsidentifikation und der Fehlerfaktor Lichtverhältnisse wird auch abgemildert, da das relative Verhältnis zu dem Pixeln betrachtet wird.

Wenn der Agent erstmal Gesichter gelernt hat, kann er auf Basis dieser Daten versuchen Gesichter zu erkennen. Wie dies insbesondere in einen Video und in Wohnräumen funktionieren kann, haben bereits Henrik Siebo und Peter Brauer untersucht (Brauer, 2014). In ihrer Arbeit präsentieren sie eine Methode, in der nicht versucht wird über jeden Videoframe eine Gesichtserkennung laufen zu lassen, sondern die Identität einer Person pro Track zu ermittelt. Ein Track ist hierbei eine Sequenz an Positionsdaten des Gesichts mit dem zugehörigen Bild vom Gesicht. Das heißt, auf jedes Bild des Tracks wird die Gesichtserkennung angewendet und das Ergebnis ausgewählt, welches im Track am häufigsten auftaucht.

Diese Arbeit nutzt diese Erkenntnisse, um sie in reduzierter und vereinfachter Weise zur Gesichtsidentifikation zu nutzen.

2.4 Anforderungen

Aus den genannten Arbeiten und Beispielen lassen sich nun Use-Cases definieren. Aus denen im Kapitel "Design" Sequenzdiagramme abgeleitet werden.

Der einfachste Use-Case ist, wenn der Benutzer den Zustand der Pflanze oder seinen Platz in der Rankingliste einsehen möchte. Dies soll er über eine grafische Oberfläche tun können (Abbildung 3).

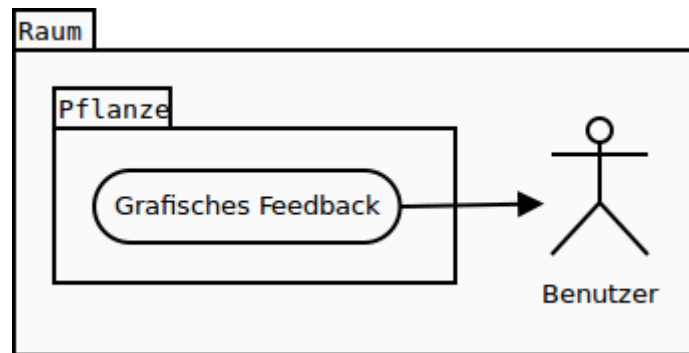


Abbildung 3: Der Benutzer möchte den Zustand der Pflanze und seinen Platz in der Rankingliste einsehen.

In einem weiteren Use-Case möchte der Benutzer kontaktiert werden, wenn die Pflanze gegossen werden möchte. Dieser Use-Case lässt sich in zwei aufteilen. Der eine Use-Case behandelt den Fall, dass der Benutzer sich vor Ort befindet und der andere den Fall, wenn der Nutzer nicht vor Ort ist. Im ersten Fall möchte der Benutzer persönlich von der Pflanze angesprochen werden (Abbildung 4). Im zweiten Fall möchte der Benutzer von der Pflanze über E-Mail oder Instant-Messaging benachrichtigt werden (Abbildung 5). Im ersten Fall steht vor dem Ansprechen noch das Identifizieren der Person, damit die Pflanze die Person mit Namen ansprechen kann.

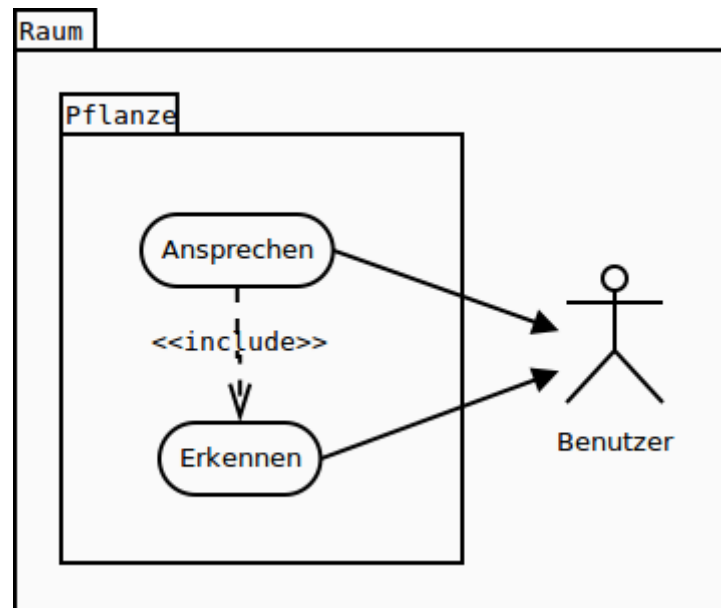


Abbildung 4: Der Benutzer möchte angesprochen werden, wenn die Pflanze gegossen werden will.

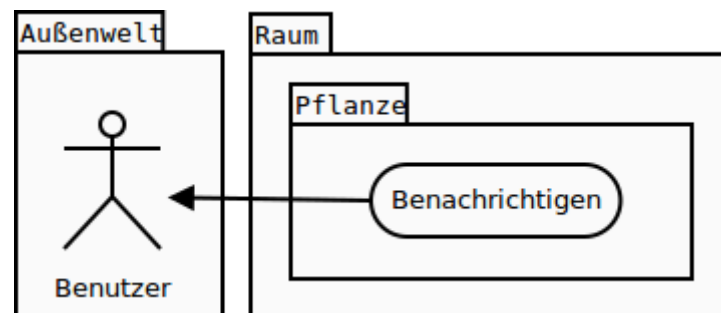


Abbildung 5: Der Benutzer möchte benachrichtigt werden, wenn die Pflanze gegossen werden will.

Wenn der Benutzer kontaktiert wurde, möchte er die Pflanze gießen. Dem vorgelagert ist das Identifizieren der Person, damit die Pflanze weiß wer sie gegossen hat und wem sie einen Punkt in der Rankingliste gutschreiben soll. Nachdem der Benutzer die Pflanze gegossen hat, möchte dieser die Rankingliste einsehen und schauen, wo er im Ranking steht. Außerdem möchte der Benutzer beim Gießen auch ein Feedback über den Zustand der Pflanze bekommen, um zu entscheiden, wann er mit dem Gießen aufhören kann. Beides ist über die grafische Oberfläche möglich (Abbildung 6).

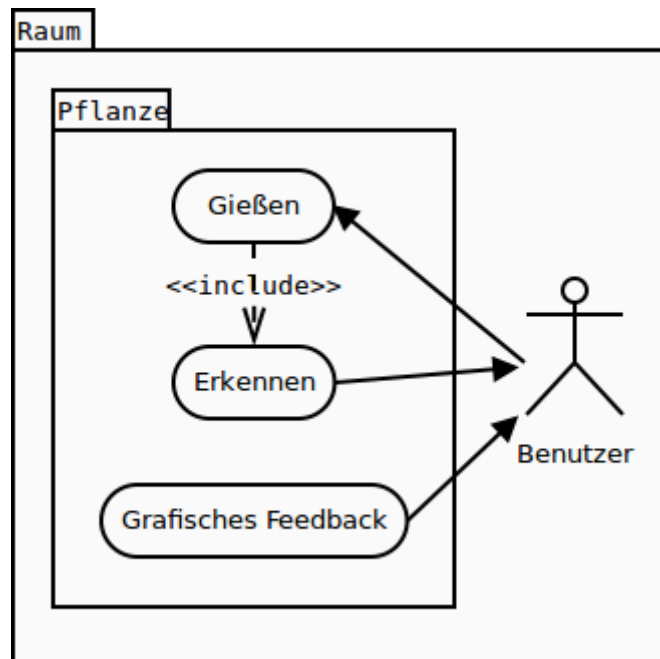


Abbildung 6: Der Benutzer möchte die Pflanze gießen und den Zustand der Pflanze wie auch seinen Platz in der Rankingliste einsehen.

Aus den genannten Arbeiten, Beispielen und den abgeleiteten Use-Cases leiten sich nun folgende Anforderungen für diese Arbeit ab.

Zum einen muss der Zustand der Pflanze erfasst werden. Hierbei wird sich auf die Erdfeuchte als Indikator des Gesamtzustands der Pflanze beschränkt. Zum anderen soll der Benutzer emotional und durch Gamification (Deterding, Dixon, Khaled, & Nacke, 2011) dazu veranlasst werden die Pflanze zu gießen. Dazu wird eine Rankingliste verwendet, in der der Benutzer aufsteigen kann, wenn er sich Punkte verdient. Diese kann er sich durch Gießen verdienen. In der Rankingliste ist auch der aktuelle Zustand der Pflanze einsehbar. Dieser Zustand wird durch Smileys emotionalisiert und die Pflanze kann somit ihren Gemütszustand äußern.

Zudem soll die Pflanze aber auch auf den emotionalen Zustand des Benutzers eingehen. Dies wird allerdings auf die zwei Phänotypen „Pflanzenliebhaber“ und „Pflanzenhasser“ heruntergebrochen. Das heißt, die Pflanze versucht, wenn sie Wasser braucht, möglichst nur Pflanzenliebhaber anzusprechen. Nur in einer Notsituation, wenn kein

Pflanzenliebhaber in der Nähe ist oder keine Zeit hat, wird ein Pflanzenhasser angesprochen oder benachrichtigt. Die Pflanze soll auch ihre Umgebung wahrnehmen können und mit dieser in Kontakt treten. Dazu behält sie mit Kameras die Umgebung im Blick und kann dadurch Personen erkennen und zuordnen. Zudem kann sie mit einer Sprachausgabe Personen in der Nähe ansprechen. Dies soll aber nicht der einzige Kommunikationskanal sein. Wenn keine Person in der Nähe ist, soll die Pflanze in den Google Kalendern (Google) von registrierten Nutzern nachschauen, ob diese Zeit haben und per E-Mail benachrichtigen. Dies lässt sich gut durch eine Ausrede-Metapher verbildlichen. Die Nutzer, die nicht vor Ort sind und einen Termin haben, wenn Pflanze Wasser braucht, haben ein Ausrede, wieso sie die Pflanze nicht gießen können. Neben der Ausrede gibt es noch den Fall, dass einer Bitte zum Gießen nicht nachgekommen wurde. Dem Benutzer muss also eine gewisse Frist eingeräumt werden, bis wann oder wie lange er Zeit hat die Pflanze zu gießen. Sollte der Benutzer der Bitte nicht nachkommen, werden ihm Punkte abgezogen und er steigt in der Rankingliste ab. Auch interessant ist der Fall des Übergießens: Hierbei soll es ebenfalls zu Punktabzug kommen. Aus der Fragestellung der Arbeit ergeben sich Schwerpunkte für die Ausführung, aber auch Bereiche die für Fragestellung von geringerer Bedeutung sind. Hierbei sind vor allem Bereiche wichtig, die für die Modellbildung des Benutzers über die Pflanze verantwortlich sind. Hierzu zählen die Rankingliste und die Kommunikation. Diese Bereiche sind wichtig, damit die Pflanze als lebendige Entität mit Emotionen wahrgenommen wird. Bereiche wie die Personenidentifikation und -verfolgung sind von nicht so großer Bedeutung für die Fragestellung. Hier wird auf fertige Lösungen zurückgegriffen. Außerdem wird in diesem Bereich keine perfekte Lösung angestrebt, sondern vielmehr eine ausreichende.

2.5 Fazit

Es soll eine proaktiver Pflanzen-Companion gebaut werden, der seinen eigenen Zustand über die Erdfeuchte definiert. Dieser Zustand ist in einer Rankingliste als Smiley einsehbar.

In dieser Rankingliste sieht der Benutzer auch seine Beliebtheit bei der Pflanze als Platz in der Liste.

Die Pflanze soll Personen in ihrer Nähe durch Gesichtserkennung identifizieren und lokalisieren. Außerdem soll die Pflanze Personen in der Nähe zum Gießen ansprechen und Personen in der Ferne per E-Mail kontaktieren. Die Pflanze soll durch einen Blick in den Online-Kalender der betreffenden Person herausfinden, ob die Person Zeit hat. Zudem soll der Benutzer vorher angeben können, ob er Pflanzenliebhaber oder Pflanzenhasser ist.

Des Weiteren soll der Benutzer Punkte fürs Gießen bekommen und Punkte abgezogen bekommen, wenn er die Pflanze übergießt oder einer Bitte nach Gießen nicht nachkommt.

3 Design

In diesem Kapitel werden die verwendeten Techniken und Konzepte für die Entwicklung eines proaktiven Pflanzen-Companion dargestellt. Dabei werden im Unterkapitel „Designkriterien“, die für einen proaktiven Pflanzen Companion wichtigen Designkriterien diskutiert. Im Unterkapitel „Systemarchitektur“ wird die Architektur und die Funktionsweise des zu entwickelnden Systems betrachtet.

3.1 Grobdesign

Aus den Anforderungen, die im Analysekapitel definiert wurden, lassen sich Module für einzelne Aufgabenbereiche ableiten. Dabei bildet sich eine Struktur heraus, die sich an der „Sense Reason Act“-Vorgehensweise (Arkin, 1998) orientiert. Das heißt, die Struktur lässt sich wiederum grob in drei Teilmodule aufteilen wie man in Abbildung 7 sehen kann.

Das erste Modul ist das „Sensing“. Dieses Modul ist für die Wahrnehmung der Umwelt und des eigenen Zustands zuständig. Die Daten, die im Sensing erhoben wurden, werden im Modul des „Reasoning“ verarbeitet und analysiert, um dann eine Handlung festzulegen, die im Modul „Acting“ umgesetzt wird.

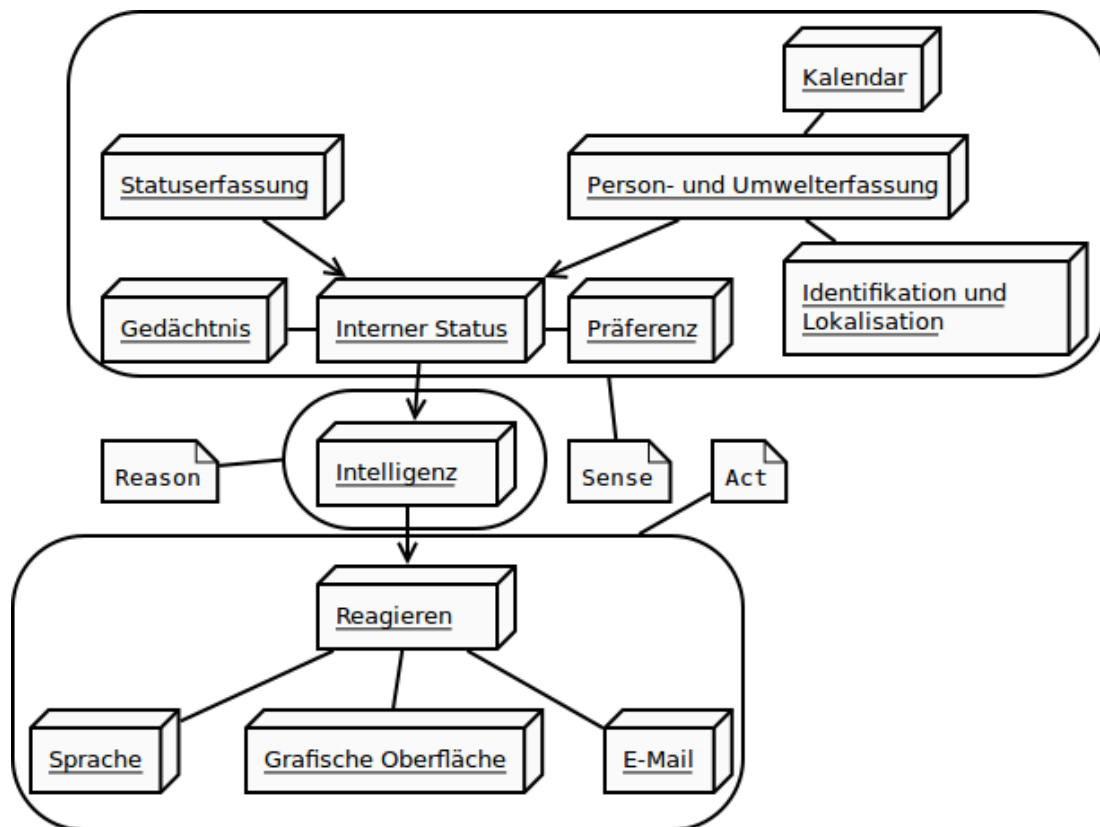


Abbildung 7: Das Grobdesign

3.1.1 Sense

Das Herzstück des Sensing Moduls ist der Interne Status. Dieser speist sich aus der Statuserfassung des Zustands der Pflanze sowie aus der Personen- und Umwelterfassung. Das Modul Personen- und Umwelterfassung ist dafür zuständig Personen zu identifizieren, zu lokalisieren und bei nicht anwesenden Personen zu erfassen, ob diese Zeit haben. Dazu hat dieses Modul zwei Untermodule. Das Untermodul "Kalender" ist dafür zuständig zu erfassen, ob nicht anwesende Personen Zeit haben. Dies bewerkstelligt es mit einem Blick in den Online-Kalender der betreffenden Person. Das Untermodul "Identifikation" und Lokalisation identifiziert und lokalisiert Personen in der Nähe. Dazu greift dieses Modul auf Methoden des maschinellen Sehens zurück. Das Modul "Statuserfassung" ist für die Erfassung des Zustands der Pflanze zuständig. Dies macht es an der Erdfeuchte fest.

Neben diesen zwei Modulen kann der interne Zustand noch auf das "Gedächtnismodul" und das "Präferenzmodul" zurückgreifen. Das Modul "Gedächtnis" ist für die persistente Haltung von Daten zuständig. Dazu gehören gelernte Gesichter, zugehörige Personeneigenschaften und Verhalten von Personen gegenüber der Pflanze. Das Modul "Präferenz" ist für die Auswahl einer Person aus mehreren zuständig. Dies macht es anhand von Personeneigenschaften und Personenverhalten gegenüber der Pflanze fest.

All diese Informationen konzentrieren sich im internen Status, der dann den Zustand der Pflanze und präferierte Personen in der Nähe und Ferne enthält.

3.1.2 Reason

Das Reasoning Modul besteht aus dem Modul "Intelligenz". Dieses Modul verarbeitet und analysiert die Daten aus dem internen Status und bestimmt auf Basis dieser Daten eine Handlung. Das Modul bestimmt, ob ein Erdfeuchtwert optimal oder kritisch ist. Die Standardhandlung ist immer den Status der Pflanze sowie die Rankingliste ausgeben zu lassen. Sollte die Bestimmung ergeben, dass der Erdfeuchtwert kritisch ist, wird zunächst versucht die präferierte Person in der Nähe anzusprechen. Sollte das scheitern, weil keine Personen in der Nähe sind, wird die präferierte Person in der Ferne per E-Mail kontaktiert. Je nachdem, ob der Gießaufforderung nachgekommen wurde oder nicht, wird der betreffende Person ein Punkt gutgeschrieben oder abgezogen. Dies wird sofort auf der Rankingliste dargestellt und im Modul "Gedächtnis" vermerkt.

3.1.3 Act

Das Acting-Modul setzt die Handlungen, die vom Reasoning Modul bestimmt wurden, um. Dazu hat es drei Untermodule. Das Untermodul "Sprache" ist für die Kommunikation mit Personen in der Nähe zuständig. Dies bewerkstelligt es mit Text-to-Speech. Das Untermodul "Grafische Oberfläche" ist für die Darstellung des Zustands der Pflanze und der Rankingliste zuständig. Dazu verwendet es eine Weboberfläche. Das Untermodul "E-Mail" kümmert sich

um die Kommunikation mit Personen in der Ferne. Es bedient sich dabei der API eines E-Mail-Providers um E-Mails zu verschicken.

3.1.4 Gründe für Sense Reason Act

Dies lässt sich auf die Fragestellung dieser Arbeit (im Kapitel Analyse erwähnt) zurückführen.

„Kann man ein mentales Modell für ein Objekt beim Betrachter dahingehend erzeugen, dass dieses Objekt (quasi) menschliche Attribute erhält und dadurch Empathie für dieses Objekt erzeugt werden kann?“

Dass heißt, das Objekt, in diesen Fall die Pflanze, soll (quasi) menschliche Attribute erhalten und ein menschliches Attribut ist Intelligenz. Also muss auch dieses Objekt über eine gewisse Intelligenz verfügen, Intelligenz lässt sich auf das Sense-Reason-Act-Muster herunterbrechen.

3.2 Feindesign

Das Grobdesign stellt noch eine logische Sichtweise auf das System dar. Im Feindesign wird zum ersten Mal eine grobe softwaretechnische Struktur definiert. Hierbei wurde sich für das Model View Controller Pattern (MVC) mit einer Persistenzschicht entschieden (Abbildung 8). Zusammen mit der Persistenzschicht stellt dies eine 3 Tier Architektur dar.

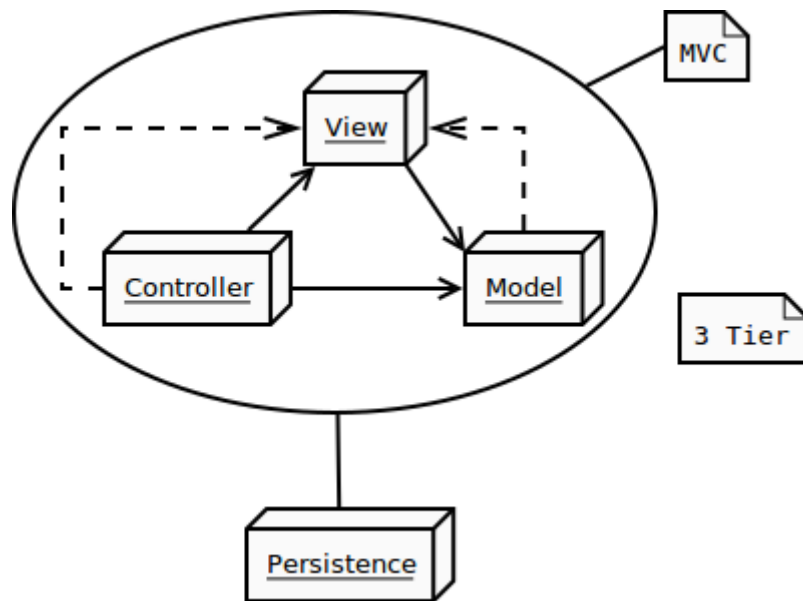


Abbildung 8: Das Feindesign

3.2.1 Gründe für das MVC Pattern

Wie im Analysekapitel schon beleuchtet, stellt jede Interaktion mit der Pflanze eine Form von Input dar, den der Benutzer tätigt. Die Pflanze reagiert auf die Aktion des Nutzers mit einer Reaktion ihrerseits und tätigt damit eine Form von Output. Es handelt sich also hier um eine Art User Interface, welches das Haupteinsatzgebiet vom MVC Pattern ist.

Dabei stellt das Modul "interner Status" das Modell (Model) dar und wird von der Präsentation (View) beobachtet und von der Steuerung (Controller) bei Bedarf verändert bzw. aktualisiert. Das Reasoning Modul stellt die Steuerung dar. Sie wertet Eingaben aus, die sie von der Präsentation bekommen hat und aktualisiert entsprechend das Modell. Die Präsentation setzt sich aus dem Acting Modul und den Erfassungsmodulen des Sensing Modules zusammen, da die Präsentation Eingaben entgegen nehmen und Ausgaben tätigen muss. Die Eingaben werden dabei an die Steuerung weitergeleitet.

3.2.2 Gründe für eine 3 Tier Architektur

Da neben dem MVC noch eine Persistentschicht eingezogen wird, entsteht eine 3 Tier Architektur, also eine Drei Schichten Architektur. Die erste Schicht ist die Präsentation. Sie ist die einzige Schicht die vom Nutzer wahrgenommen wird bzw. werden kann. Die zweite und mittlere Schicht setzt sich aus der Steuerung und dem Modell zusammen. Die Persistenzschicht stellt hierbei das Modul "Gedächtnis" aus dem Grobdesign dar und ist die dritte Schicht.

3.3 Designkriterien

3.3.1 Stabilität

Ein wichtiges Kriterium ist Stabilität. Dies betrifft mehrere Komponenten. Die Gesichtserkennung z.B. sollte Gesichter stabil erkennen und diese identifizieren. Das heißt Ausreißer in der Gesichtsidentifikation sollen ausgeglichen werden, sodass eine Person in den Augen der Gesichtsidentifikation über die Zeit nicht ihre Identität ändert.

Eine weitere Komponente, die dies betrifft, ist die Komponente zur Erfassung der Erdfeuchte. Der Wert der Erdfeuchte sollte nicht zu stark schwanken. Dies könnte Fehlinterpretationen bewirken wie z.B. die Pflanze wurde gegossen, wenn der Wert einen Ausreißer nach unten hat und wieder zum Ausgangswert zurückkehrt. Auch hier gilt es Ausreißer zu minimieren.

3.4 Die Systemarchitektur

Das System besteht aus sieben Komponenten (Abbildung 9). Genauer gesagt Agenten, die mit einander über Nachrichten kommunizieren. Die Richtung der Kommunikation wird durch gestrichelte Pfeile dargestellt. Der Kinect Data Provider stellt hierbei eine Ausnahme dar. Er wurde nicht im agentenbasierten Framework umgesetzt und sendet auch keine Nachrichten, sondern UDP-Pakete. Alle anderen Komponenten sind mit dem

agentenbasierten Framework umgesetzt und kommunizieren über Nachrichten miteinander.

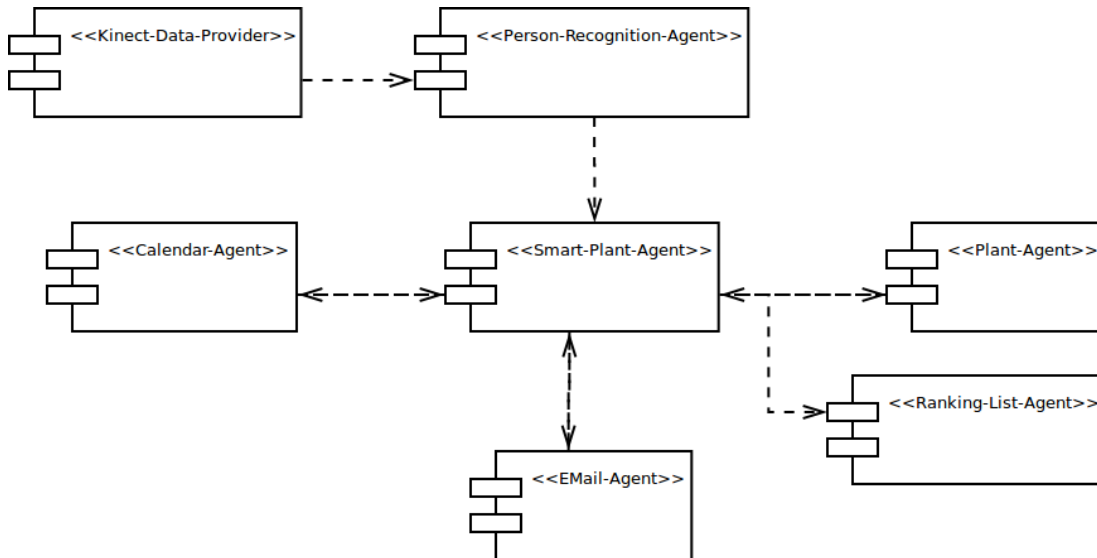


Abbildung 9: Komponentendiagramm des Gesamtsystems

3.4.1 Gründe für das agentenbasierte Framework

Zum einen war das Framework bereits vorhanden und fast alle bisherigen Projekte im CSTI setzten auf dieses Framework. Dies hat den Vorteil, dass das Framework einem diverse Aufgaben, wie beispielsweise das Versenden und Empfangen von Nachrichten, abnimmt. Zum anderen werden Projekttemplates für Java und Scala angeboten, die einen schnellen Entwicklungsstart ermöglichen. Hinzu kommt eine intuitive und schlanke API, die die Entwicklung zusätzlich beschleunigt.

Zudem unterstützt es die Entwicklung von wiederverwendbaren Modulen und verteilten Systemen. Durch die Unterstützung von verteilten Systeme in Form von Agenten, lässt sich ein System auf mehrere Rechner aufteilen. Dadurch lassen sich z.B. leistungshungrige Komponenten auslagern.

3.4.2 Kinect Data Provider

Der Kinect Data Provider dient als Schnittstelle zwischen der Hardware und dem System. Dadurch wird von der Kamera Hardware abstrahiert. Das heißt, dass die Kinect auch gegen eine andere Kamera ausgetauscht werden kann, solange die Schnittstelle implementiert wird.

Konkret bedient sich der Kinect Data Provider der Kinect API (Microsoft), um bei erkannten Personen mit Hilfe von deren Skelettdaten den Kopf der betreffenden Person zu extrahieren. Dabei wird der HeadJoint (Microsoft) aus den Skelettdaten auf das Kamerabild der Kinect gemappt und der Kopf aus dem Kamerabild ausgeschnitten. Zusammen mit der Position des SpineMid (Microsoft) Joint wird das Bild des Kopfes per UDP-Nachricht (Abbildung 10) an den Person Recognition Agent gesendet. Auch benachrichtigt der Kinect Data Provider den Person Recognition Agent per UDP-Nachricht wenn eine Person das Videobild betritt oder verlässt.

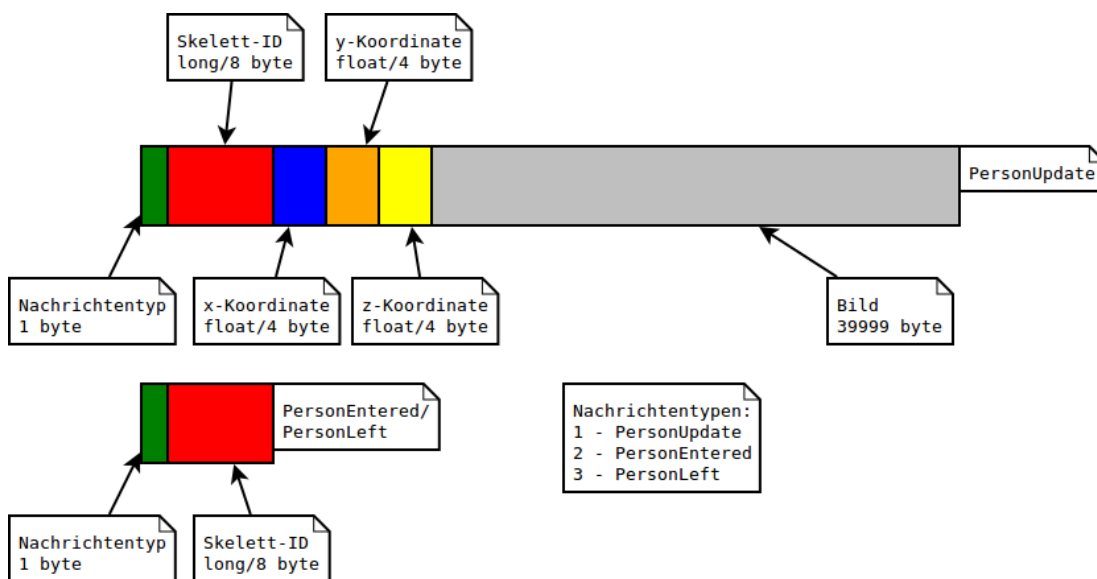


Abbildung 10: UDP-Nachrichten des Kinect-Data-Providers

3.4.3 Umsetzung des Kinect Data Provider

Der Kinect Data Provider wurde nicht als Agent umgesetzt, das das Mapping von Skelettkoordinaten auf das Farbbild der Kinect bei den vorhandenen Java Bibliotheken fehlerhaft ist. Einzig die Bibliothek, die der Hersteller anbietet, ist in der Lage dies fehlerfrei zu leisten.

Allerdings gibt für diese Bibliothek nur Implementationen in C# und C++. Agenten im agentenbasierten Framework lassen sich aber nur in Scala, Java und JavaScript schreiben. Daher ließ sich der Kinect Data Provider nicht als Agent umsetzen. Man konnte daher auch nicht auf das Nachrichtensystem des Frameworks setzen und musste auf UDP Nachrichten zurückgreifen.

3.4.4 Person Recognition Agent

Der Person Recognition Agent nimmt die UDP-Nachrichten entgegen und nutzt insbesondere das Bild vom Kopf um Gesichter zu lernen oder zu identifizieren.

Bevor das Bild zum Lernen oder Identifizieren verwendet wird, wird überprüft, ob das Bild überhaupt ein Gesicht enthält. Denn es könnte sein, dass nur der Hinterkopf oder das Gesicht im Profil zu sehen ist. Diese Bilder eignen sich nicht zur Weiterverwendung und werden verworfen. Dies wird mit einem kaskadierenden Klassifikator für ein Gesicht von vorne bewerkstelligt. Dazu wird auf jedes eingehende Bild dieser Klassifikator angewendet und wenn dieser anschlägt, enthält das Bild ein Gesicht.

Wenn das Bild ein Gesicht enthält, wird es anhand der Augen ausgerichtet und ausgeschnitten. Dann wird über das ausgeschnittene Gesicht von allen Face Recognizern eine Vorhersage eingeholt und die Vorhersage mit der höchsten Übereinstimmung mit dem Gesicht ausgewählt oder keine, wenn das Gesicht in keiner Vorhersage eine Übereinstimmung aufzeigt. Dabei gibt es für jede Identität einen Face Recognizer. Die Identität, die sich aus der Vorhersage ergibt oder „unknown“, falls keine Übereinstimmung gefunden wird, wird als Stimme für die Identität, die hinter der Skelett ID steckt, abgegeben. Wenn eine gewisse Anzahl an Stimmen gesammelt wurde, wird der Skelett ID

die Identität zugeordnet, die die meisten Stimmen erhalten hat und erhalten wird. Dadurch werden Ausreißer in der Gesichtsidifikation ausgeglichen.

Der Person Recognition Agent kann auch neue Gesichter lernen. Dazu geht der Agent in den Lernmodus, wenn dieser von Außen durch eine Nachricht dazu angeregt wird. Diese Nachricht beinhaltet den Namen der Person, dessen Gesicht gelernt werden soll. Der Agent nimmt nun an, dass die Person, die vor der Kamera steht, die Person ist, dessen Gesicht gelernt werden soll. Als nächstes wird ein neuer Face Recognizer erstellt und für jedes Bild eine Vorhersage angefordert. Wenn die Vorhersage keine Übereinstimmung aufzeigt, wird das Gesicht dem neuen Face Recognizer hinzugefügt. Der Lernvorgang ist beendet, wenn dem Face Recognizer eine gewisse Anzahl an Gesichtern hinzugefügt wurde. Dann wird der Face Recognizer persistent weggespeichert.

Zum Lernen und Identifizieren von Gesichtern wird das Computer Vision Framework OpenCV (OpenCV) verwendet. Es bietet drei verschiedene Algorithmen zum Lernen und Identifizieren von Gesichtern. Den Algorithmus von Eigen, den Algorithmus von Fisher und den LBPH Algorithmus (Local Binary Pattern Histogram). Der Person Recognition Agent verwendet den Algorithmus von Eigen. Allerdings kann dieser auch gegen einen der zwei anderen ausgetauscht werden.

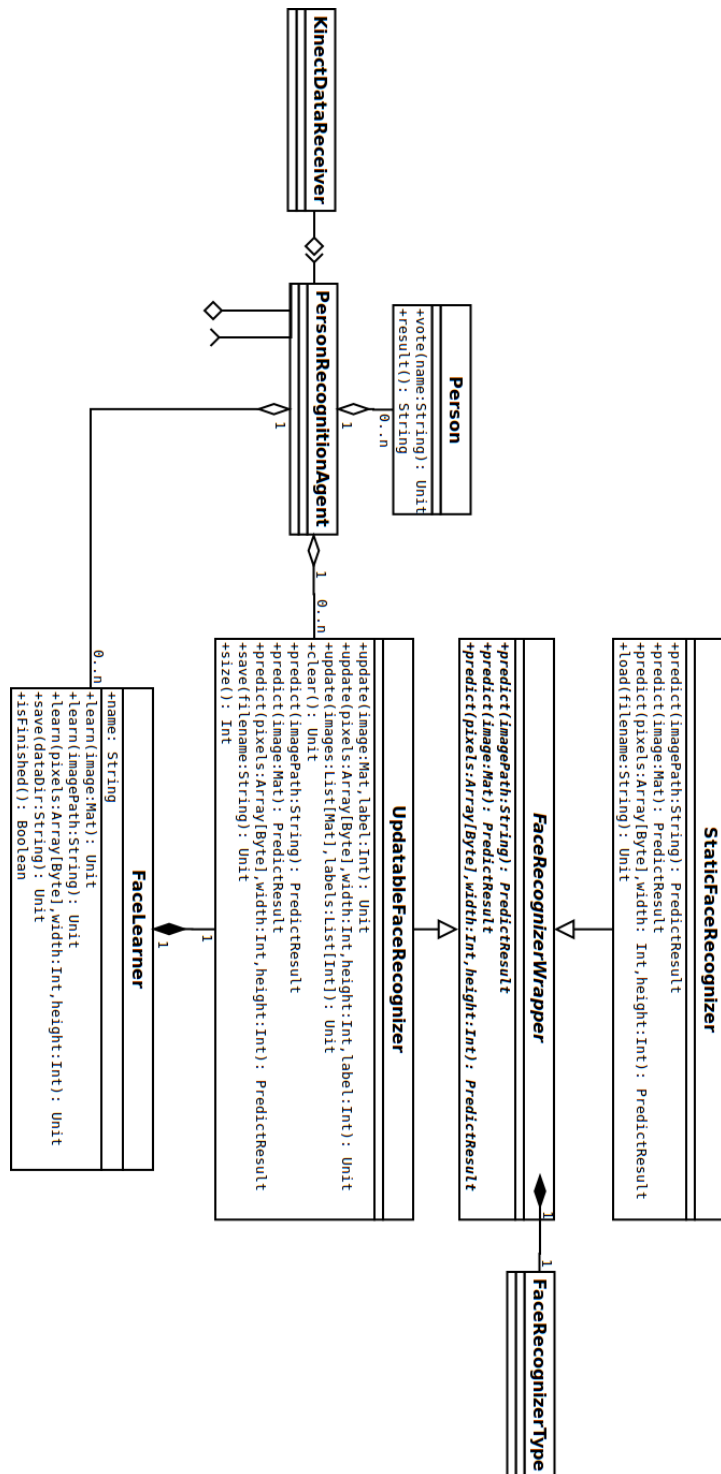


Abbildung 11: Klassendiagramm des Person-Recognition-Agents

3.4.5 Gründe für OpenCV?

OpenCV (OpenCV) wurde unter der BSD Lizenz veröffentlicht und kann ohne größere Restriktionen benutzt werden. Die Bibliothek an sich ist in C/C++ umgesetzt. Allerdings gibt es auch Schnittstellen für Java und Python. Daher ließ es sich leicht im Person Recognition Agent benutzen.

Zur Gesichtsidentifikation wurde es notwendig ein zusätzliches Paket zu installieren, welches nicht standardmäßig in OpenCV enthalten ist. Dieses zusätzliche Paket musste mit OpenCV kompiliert werden, was zu Problemen beim Kompilieren führte. Daher wurde auf eine fertiges Kompilat, welches über Maven (The Apache Software Foundation) eingebunden werden kann, zurückgegriffen. Dieses Kompilat ist allerdings nicht für eine spezielle Grafikkarte kompiliert. Das bedeutet alles Operation für z.B. Gesichtserkennung werden auf der CPU aufgeführt. Das hat zum einen den Vorteil, dass der Agent auf jedem Rechner, der über genügend Rechenleistung verfügt, laufen kann. Dies hat aber allerdings auch den Nachteil, dass man nicht die Vorteile einer Grafikkarte ausnutzen kann. Dies wären z.B. optimierte und dadurch schnellere grafische Berechnungen.

3.4.6 Calendar-Agent

Der Calendar Agent (Abbildung 12) stellt den Zugriff auf die Online-Kalender der Personen sicher, in diesen Fall auf den Google Calendar. Personen können sich beim Calendar-Agent registrieren und erhalten dann eine URL unter der sie den Zugriff auf ihre Online-Kalender autorisieren können. Die bei der Registrierung angegebenen Daten sowie die Credentials, die der Agent von Google erhält, werden persistent gespeichert.

Wenn eine Person beim Agent registriert ist, kann dieser Termine aus allen Online Kalendern abfragen und bereitstellen.

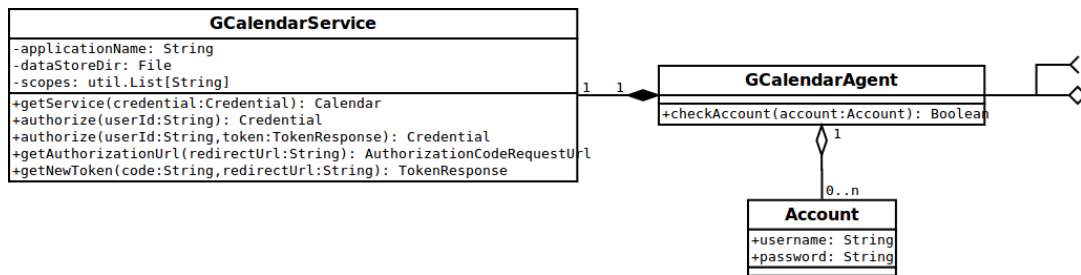


Abbildung 12: Klassendiagramm des Calendar-Agents

3.4.7 E-Mail-Agent

Der E-Mail Agent stellt den Zugriff auf einen E-Mail Account sicher. In diesen Fall auf einen Gmail Account. Eine Person kann sich beim E-Mail-Agent registrieren und erhält dann eine URL unter der sie den Zugriff auf ihren E-Mail Account autorisieren kann. Die bei der Registrierung angegebenen Daten sowie die Credentials, die der Agent von Google erhält, werden persistent gespeichert.

Wenn eine Person beim Agent registriert ist, kann dieser den Account der betreffenden Person nutzen um E-Mails zu verschicken.

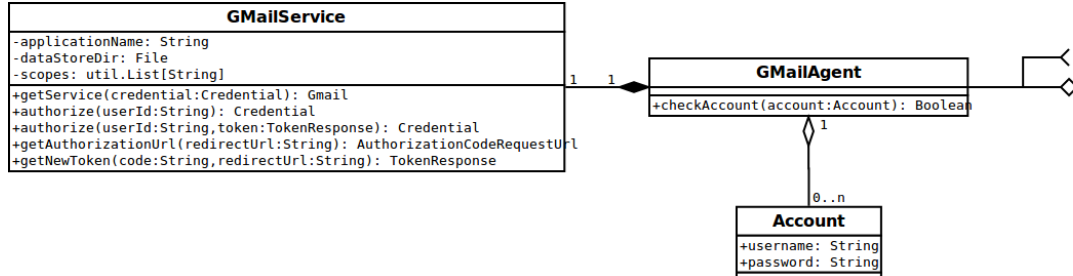


Abbildung 13: Klassendiagramm des EMail-Agents

3.4.8 Gründe für Google Dienste

Google bietet für seine Dienste APIs in diversen Sprachen an, darunter auch Java. Dies macht es besonders einfach diese Dienste in Agenten zu implementieren. Zudem abstrahieren die APIs von der Web Schnittstelle, sodass man sich nicht um HTTP Requests und das Erstellen von JSON kümmern muss.

3.4.9 Smart-Plant-Agent

Der Smart-Plant-Agent ist das Herz des proaktiven Pflanzen-Companions. Er bedient sich an den Funktionalitäten aller anderen Agent, welche diese bereitstellen. Er nutzt den EMail Agent zur Long Distance Communication, den Person-Recognition-Agent zur Lokalisation und Identifikation von Personen in seiner Nähe, den Plant-Agent zur Near Distance Communication und zur Erfassung von der Erdfeuchte, den Calendar-Agent um zu überprüfen, ob nicht anwesende Personen Zeit haben und den Ranking-List-Agent um den Zustand der Pflanze und den Beliebtheitsgrad jeder Person bei der Pflanze darzustellen.

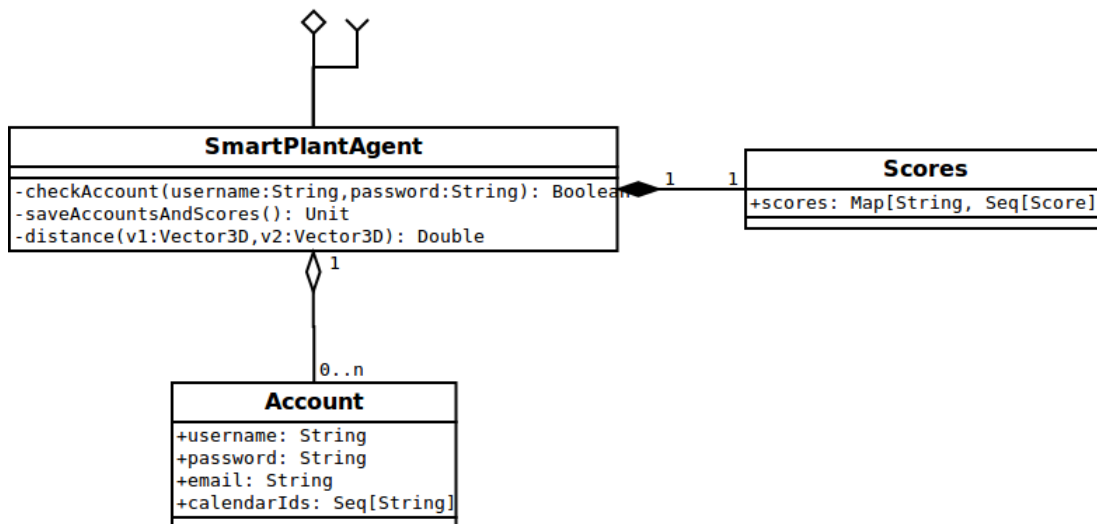


Abbildung 14: Klassendiagramm des Smart-Plant-Agent

Bei dem Smart-Plant-Agent können sich Personen als Besitzer der Pflanze registrieren. Dabei kann die Person neben Benutzernamen und Passwort auch die Kalender wählen, in denen der Agent nachschaut, ob die Person im Falle von Abwesenheit Zeit hat. Zudem kann die Person ihre E-Mail-Adresse angeben, an die Gießanfragen gesendet werden und ihren Phänotypen.

Neben den Accounts verwaltet der Agent auch die Scores, die bei einem Update an den Ranking List Agent gesendet werden. Ein Update erfolgt nur, wenn ein Besitzer Punkte

durch Gießen erzielt. Außerdem leitet der Agent den Status der Pflanze, den er von dem Plant Agent erhält, weiter an den Ranking List Agent. Accounts sowie Scores werden persistent vorgehalten.

3.4.10 Ranking List Agent

Der Ranking List Agent ist für die Visualisierung des Zustands der Pflanze und der Beliebtheit der Personen bei der Pflanze zuständig. Er wurde mit dem Model View Whatever Framework (MVW) „Angular2“ (Angular) umgesetzt. Es handelt sich also hier bei um eine Webanwendung.

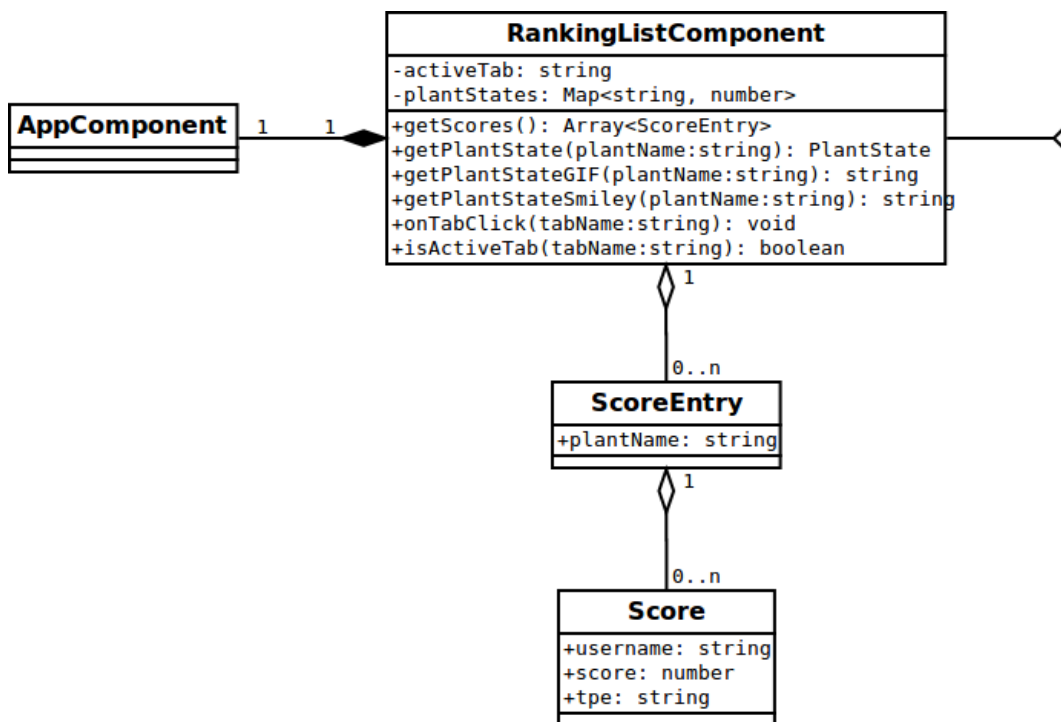


Abbildung 15: Klassendiagramm des Ranking-List-Agent

3.4.11 Gründe für Angular2

Zum einen sind mit Angular2 (Angular) gebaute Web Apps oder GUIs sehr responsiv. Das heißt sie reagieren sehr schnell auf Eingaben und Veränderungen ihres Modells. Es kommt

also nicht zu Verzögerungen, die die Oberfläche behäbig wirken lassen. Da es sich bei Angular2 um ein JavaScript Framework handelt, welches im Browser ausgeführt wird und alles auch im Browser rendert, sind mit Angular2 gebaute Web Apps und GUIs plattformunabhängig.

Ein weiterer Vorteil ist das einfache Bauen von GUIs durch sogenannte Templates. Die Templates folgen im Großen und Ganzen der HTML-Syntax. Zudem kann man im HTML-Code auf Variablen oder Funktionen in der Komponente zugreifen bzw. verweisen. Diese Datenbindung funktioniert auch in die entgegengesetzte Richtung, wodurch Eingaben, welche durch ein Eingabefeld im Template getätigt wurden, auf eine Variable in der Komponente projiziert werden können.

Diese Datenbindung und die HTML-Syntax machen es besonders leicht und schnell Web Apps oder GUIs zu bauen. Hinzu kommt noch, dass Angular2 TypeScript unterstützt, mit dem es sich einfacher entwickeln lässt als mit puristischen JavaScript.

Auch das Erstellen eines Angular2 Projektes gestaltet sich einfach durch bereitgestellte Projekt Templates in Form von Repositories, die man lediglich klonen muss.

Die Darstellung des Zustands der Pflanze erfolgt über einen Smiley neben den Namen der Pflanze und einer Animation neben der Rankingliste (Abbildung 16).

Peter :(

Place	Name	Score
1.	Benny	20
2.	Lars	10
3.	Hans	5

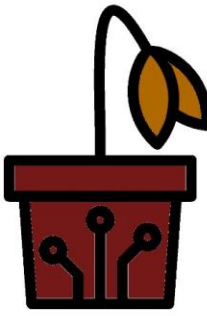


Abbildung 16: Weboberfläche des Ranking-List-Agent

3.4.12 Plant Agent

Der Plant Agent ist für die Erfassung der Erdfeuchte und für die Near Distance Communication zuständig. Bei der Erfassung der Erdfeuchte werden Ausreißer durch die Moving Window Technik ausgeglichen. Dabei wird der Mittelwert aus einer gewissen Anzahl der letzten eingegangenen Werte gebildet. Zudem werden die Werte mit einem großen Abstand abgefragt, um Sensorfehler klein zu halten.

Zur Near Distance Communication verwendet der Agent das Konsolenprogramm „festival“ (Black), um generierten Text als Sprache auszugeben.

Der Plant-Agent läuft auf einen Raspberry Pi (Raspberry Pi Foundation), welcher mit einem externen Lautsprecher und dem Erdfeuchtesensor verbunden ist. Der Lautsprecher verfügt über eine separate Stromversorgung, da das Audiosignal aus dem Raspberry Pi nicht besonders stark ist, was mit der geringen Versorgungsspannung von 5 V, mit der der Raspberry Pi auskommen muss, zusammenhängt.



Abbildung 17: Pflanze mit Raspberry Pi und externem Lautsprecher

Der Raspberry Pi verfügt zwar über GPIOs, darunter befinden sich aber keine analogen Pins. Daher muss zwischen dem Raspberry Pi und dem Erdfeuchtesensor ein Analog-Digital-Converter (ADC) geschaltet werden. Dieser wandelt die analogen Signale des Sensors in digitale Signale um und stellt diese in einem Register zur Verfügung. Dieses Register kann der Raspberry Pi mit der C-Bibliothek „WiringPi“ (WiringPi) auslesen. Da der Agent aber in Scala geschrieben ist, musste noch eine Java-Anbindung geschrieben werden.

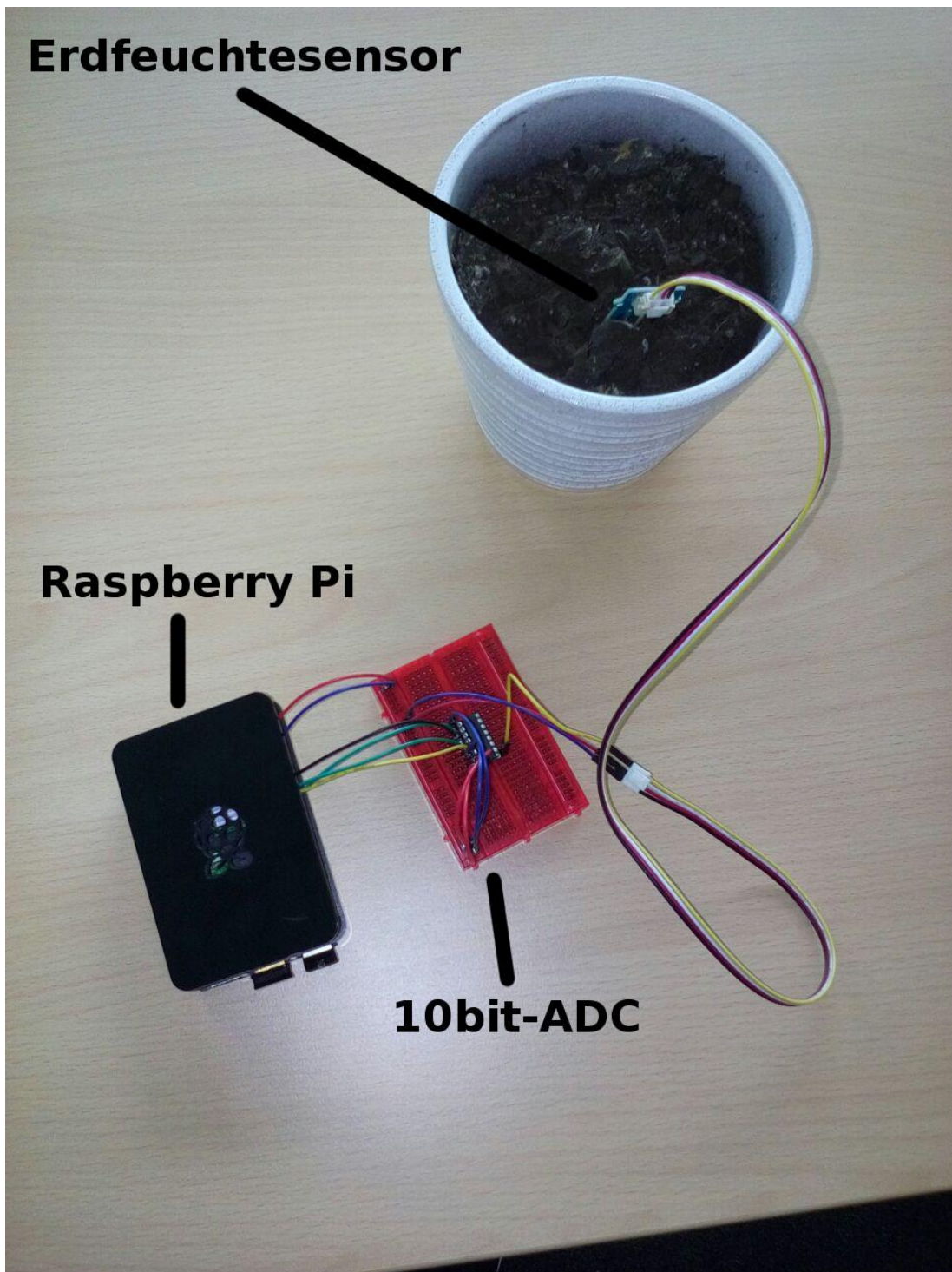


Abbildung 18: Raspberry Pi mit 10bit-ADC und Erdfeuchtesensor

3.4.13 Gründe für die Verwendung einer C-Bibliothek

Da C näher an der Hardware ist, hat es einen Geschwindigkeitsvorteil, den eine Java Bibliothek nicht bieten kann. Das liegt daran, wie der ADC abgefragt wird. Jedes Bit muss einzeln abgefragt werden. Das heißt bei einen 10 bit ABC muss man 10 mal ein Bit abfragen, um einen Wert zu bekommen. Diese Abfragen müssen schnell hintereinander erfolgen, sonst nimmt der ADC an, die Abfrage insgesamt wurde abgebrochen. Java wäre hier schlichtweg zu langsam.

3.4.14 Gründe für die Text-to-Speech-Lösung "festival"

Text-to-Speech hat den Vorteil, dass man keine vorbereiteten Sounddateien vorhalten muss. Zudem lassen sich Sprachausgaben flexibler gestalten. Zum Beispiel kann bei einer Gießaufforderung der Name der betreffenden Person angefügt werden. Das wäre mit Sounddateien nur sehr umständlich möglich.

Es wurde „festival“ (Black) verwendet, da es das einzige Programm unter den getesteten war, was auf dem Raspberry Pi funktionierte. Neben "festival" wurden auch "espeak" (eSpeak Projekt) und "spd-say" (Ubuntu) getestet.

3.5 Verhalten der Pflanze bei Wassermangel

In diesen Abschnitt werden die Verhaltensmuster der Pflanze behandelt, nachdem sie festgestellt hat, dass sie gegossen werden muss. Dabei gibt es verschiedene Szenarien:

- Person in der Nähe
- Keine Person in der Nähe

Wenn keine Person in der Nähe ist, muss die Pflanze anhand vom Phänotyp und ob eine Person Zeit hat, entscheiden, welche Person per E-Mail angeschrieben werden soll. Nachdem der Smart-Plant-Agent vom Plant-Agent eine Nachricht über einen zu niedrigen Erdfeuchtewert erhalten hat, geht dieser in den „lookingForPersonsCloseBy“ Zustand über.

Der Agent behält diesen Zustand eine gewissen Zeit lang bei und nimmt Nachrichten vom Person Recognition Agent über Personen und deren Identität und Standort entgegen. Wenn in dieser Zeit keine Person auftaucht, also der Agent keine Nachrichten vom Person-Recognition-Agent erhält, geht dieser in den „contactPerson“ Zustand. In diesen Zustand holt sich der Agent vom Calendar Agent von allen registrierten Besitzern deren Termine und ermittelt anhand dieser Informationen, welche Person Zeit hat. Wenn mehrere Personen Zeit haben, wird dann anhand von Phänotyp und Ranking entschieden, welche Person kontaktiert wird. Der Agent nutzt dann den Email Agent um der ausgewählten Person eine E-Mail zu schicken und geht anschließend wieder in den normalen Zustand über. Wenn der Agent vom Plant Agent eine Nachricht über einen genau richtigen Erdfeuchtwert bekommt, geht dieser wieder in den „lookingForPersonsCloseBy“ Zustand. Der erkannten Person wird ein Punkt im Ranking gutgeschrieben und das aktuelle Ranking als Nachricht an den Ranking List Agent übermittelt, der dieses sofort darstellt.

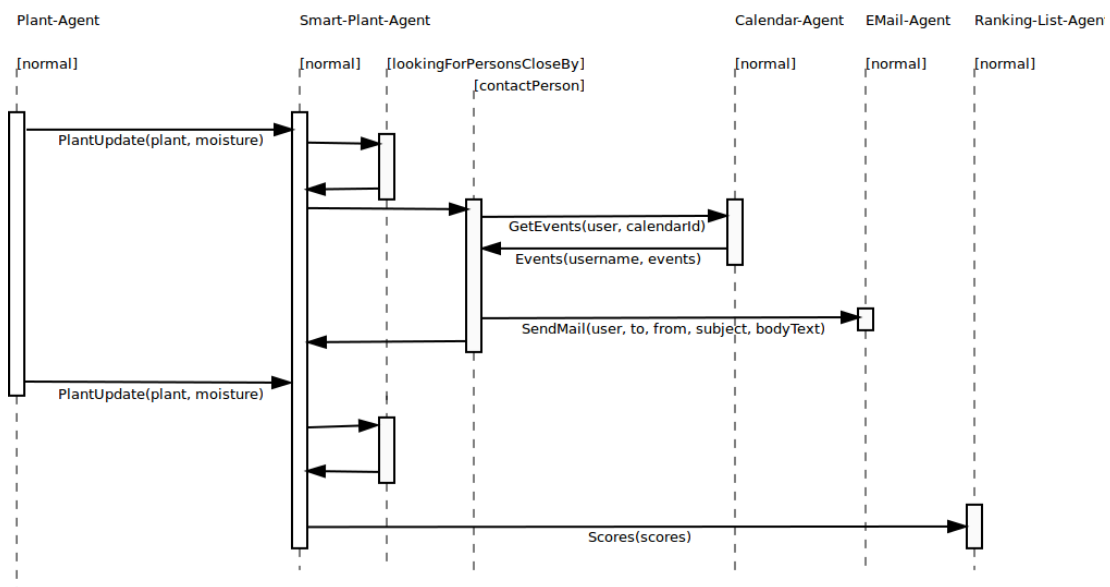


Abbildung 19: Sequenzdiagramm des Szenarios "Keine Person in der Nähe"

Wenn eine oder mehrere Personen in der Nähe sind, muss die Pflanze anhand des Phänotyps entscheiden, welche Person sie anspricht.

Wenn der Smart-Plant-Agent vom Plant-Agent eine Nachricht über einen zu niedrigen Erdfeuchtwert bekommt, geht dieser für eine gewisse Zeit in den „lookingForPersonsCloseBy“ Zustand über. Wenn mehr als eine Person gefunden wurde, muss der Agent anhand des Phänotyps und des Rankings entscheiden, welche Person dieser anspricht. Wenn der Agent eine Person ausgewählt hat, sendet dieser eine Nachricht an den Plant Agent, dass dieser die Person beim Namen ansprechen soll. Wenn der Agent vom Plant Agent eine Nachricht über einen genau richtigen Erdfeuchtwert erhält, geht dieser in den „lookingForPersonsCloseBy“ Zustand über. Die erkannte Person bekommt dann im Ranking einen Punkt gutgeschrieben. Das aktuelle Ranking wird an den Ranking List Agent übermittelt, welcher dieses sofort darstellt.

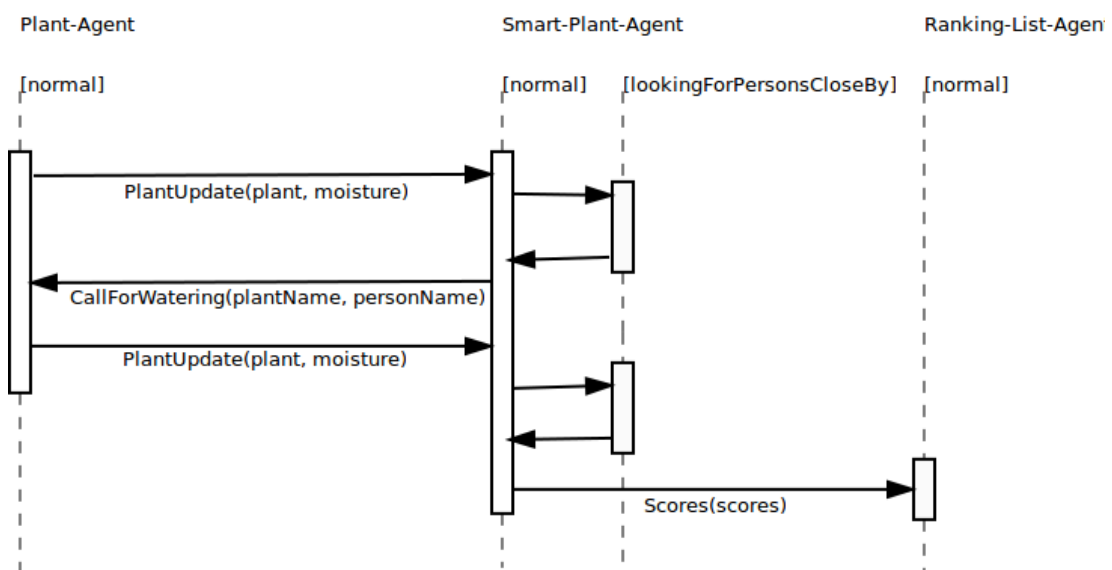


Abbildung 20: Sequenzdiagramm des Szenarios "Person(en) in der Nähe"

Im Falle vom Übergießen der Pflanze wird der betreffenden Person ein Punkt im Ranking abgezogen. Das heißt, wenn der Smart-Plant-Agent vom Plant-Agent eine Nachricht über einen zu hohen Erdfeuchtwert bekommt, geht dieser in den „lookingForPersonsCloseBy“

Zustand über. Der erkannten Person wird ein Punkt im Ranking abgezogen. Das aktuelle Ranking wird wieder an den Ranking-List-Agent übermittelt.

Im Falle, dass eine Person einer Gießaufforderung nicht nachkommt, soll ebenfalls ein Punkt im Ranking abgezogen werden. Dazu räumt der Agent nach dem Versenden einer E-Mail oder nach dem Ansprechen einer Person eine gewisse Reaktionszeit ein. Die Reaktionszeit ist bei nicht anwesenden Person höher als bei anwesenden, da diese noch einen gewissen Weg zur Pflanze überwinden müssen. Sollte die Person der Gießaufforderung nicht nachkommen, wird ihr ein Punkt im Ranking abgezogen und das aktuelle Ranking an den Ranking List Agent übermittelt. Der Agent beginnt in diesen Fall mit der Suche nach einer geeigneten Person von vorne.

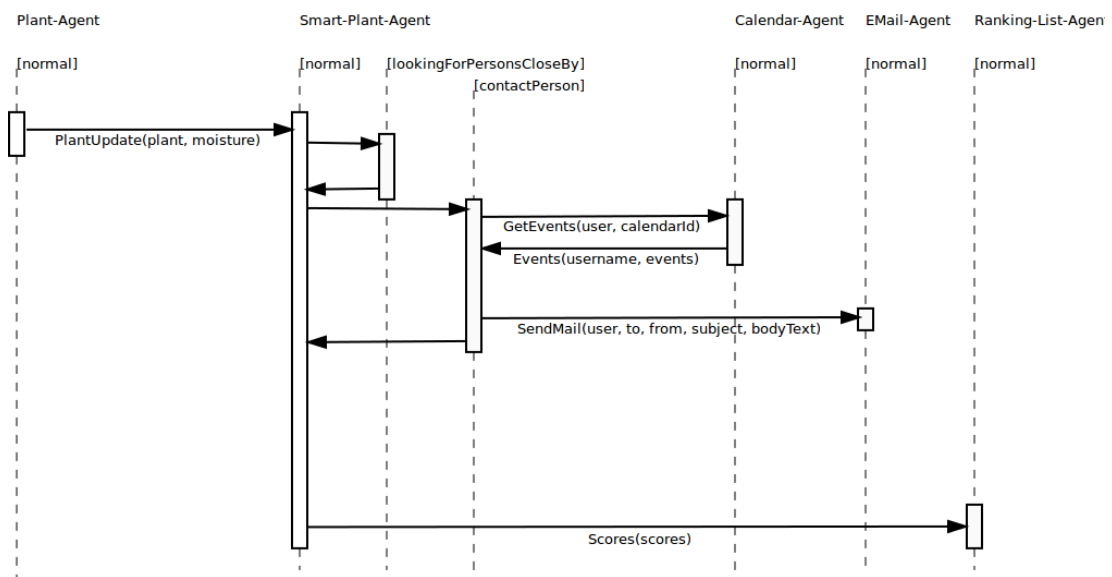


Abbildung 21: Sequenzdiagramm des Szenarios "Entfernte Person kommt der Aufforderung nicht nach"

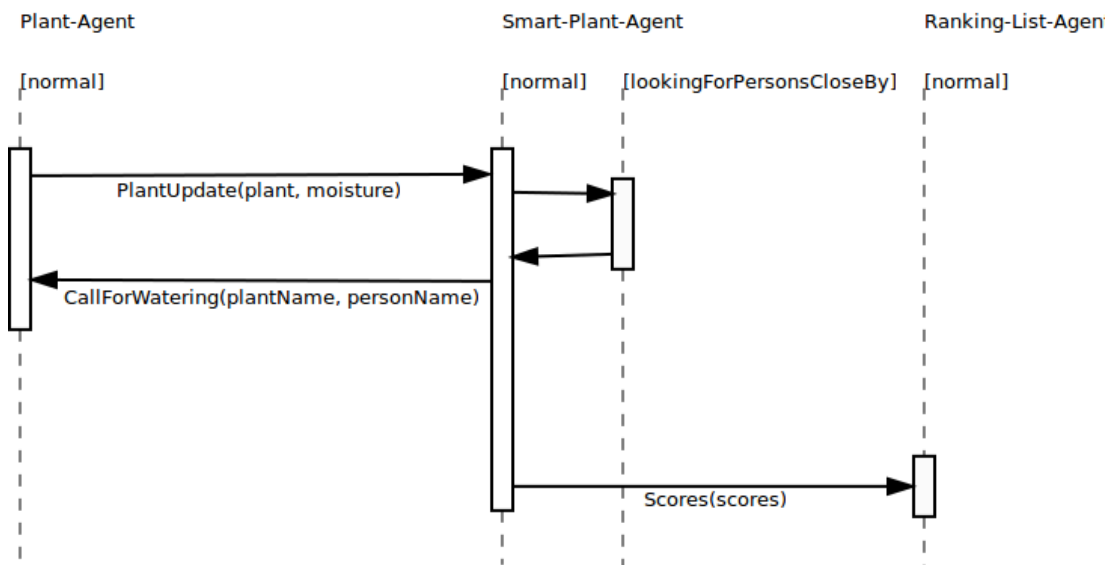


Abbildung 22: Sequenzdiagramm des Szenarios "Person in der Nähe kommt Aufforderung nicht nach"

Diese Szenarien unterscheiden sich zu den ersten zwei durch die fehlende Nachricht des Plant Agent über einen genau richtigen Erdfuchtewert. So wird die Reaktionszeit überschritten und der betreffenden Person einen Punkt im Ranking abgezogen.

3.6 Fazit

In diesem Kapitel wurde dargestellt, wie und mit welchen Mitteln der proaktive Pflanzen Companion umgesetzt wurde.

Dabei bin ich auf das Problem gestoßen, eine passende Java Bibliothek zu finden, um die Kinect (Microsoft) anzusprechen. Am Ende konnte keine zufriedenstellende Bibliothek gefunden werden und es musste auf eine Bibliothek vom Hersteller (Microsoft) zurückgegriffen werden, die in C# geschrieben war. Das Suchen und Testen der Bibliotheken hat sehr viel Zeit in Anspruch genommen und ich würde daher empfehlen die Bibliothek des Herstellers zu verwenden.

4 Evaluation

In diesem Kapitel wird der Ansatz evaluiert. Dazu wird zuerst der Ist Zustand des Systems betrachtet. Anschließend wird analysiert, welche Komponenten, Teile, Features oder Verbesserungen noch fehlen. Zum Schluss wird ein Fazit gezogen, in dem betrachtet wird, welches die nächsten Schritte sind.

4.1 Der Ist Zustand

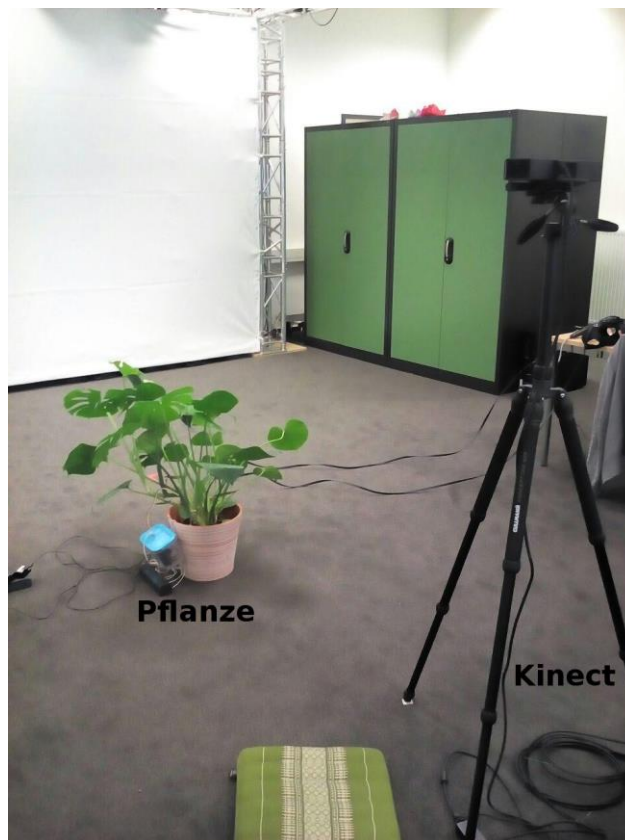


Abbildung 23: Aufbau mit Kinect und Pflanze

Der Aufbau besteht aus einer Kinect und einer Pflanze, die ca. 1,58 m von der Kinect entfernt steht. Die Kinect befindet sich ca. 1,44 m über dem Boden. An der Pflanze selbst befindet sich eine Kunststoffdose, in der sich der Raspberry Pi und der ADC befinden. Unter der Kunststoffdose befindet sich ein externer Lautsprecher, der separat mit Strom versorgt wird und mit dem Raspberry Pi verbunden ist. Hinter der Kinect befindet sich der Computer, mit dem die Kinect verbunden ist. Auf dem Computer befindet sich auch ein Monitor, der die Rankingliste anzeigt.

4.1.1 Die Pflanze

Wie bereits beschrieben, befindet sich an der Pflanze selbst ein Raspberry Pi mit einem Lautsprecher und einem Erdfeuchtesensor. Die Pflanze ist dadurch in der Lage nach einer Vorlaufzeit von 5 s jede Sekunde einen Erdfeuchtwert zu liefern. Dieser Wert ist der Mittelwert aus den letzten fünf ermittelten Werten und weist daher eine gewisse Stabilität auf.

Des Weiteren kann die Pflanze über einen Lautsprecher Sprachausgaben tätigen. Da diese über einen Text-to-Speech-Generator generiert werden, klingen die Sprachausgaben dementsprechend maschinell und steif.

4.1.2 Lokalisation und Identifikation

Die Lokalisation läuft dank der Kinect relativ reibungslos. Einzig anzumerken ist, dass Personen, dessen Gesicht für die Kinect nicht sichtbar ist, nicht von der Kinect erkannt werden. Auch kann es zu Problemen kommen, wenn Körperteile verdeckt werden z.B. wenn sich die Hände hinter dem Rücken befinden. Dann muss die Kinect die Position der Hände schätzen.

Die Identifikation ist noch nicht zuverlässig und performant. Es kann vorkommen, dass eine Identifikation ein falsches Ergebnis liefert. Auch nimmt die Latenz mit den gelernten Gesichtern in der Datenbank zu. Ab drei gelernten Gesichtern ist die Latenz so groß, dass

die Usability stark beeinträchtigt wird, da jedes gefundene Gesicht gegen die gelernten Gesichter verglichen wird.

4.1.3 Time Awareness

Die Pflanze ist in der Lage in den Online-Kalender von registrierten Person nach zu sehen, ob diese Zeit haben. Dabei prüft die Pflanze nur, ob die betreffende Person gerade einen Termin hat. Die Pflanze bezieht also nicht die An- bzw. Abreise mit ein. Dadurch kann es dazu kommen, dass eine Person kurz vor einen Termin eine Gießanfrage bekommt und diese berechtigterweise nicht wahrnehmen kann. Trotzdem wird die betreffende Person mit einen Punktabzug bestraft.

Die Pflanze sendet keine Gießanfragen von 22 Uhr bis 6 Uhr, da sie annimmt, das Menschen normalerweise in diesen Zeitraum ihre Ruhe haben wollen (Ministerium für Inneres und Kommunales Nordrhein-Westfalen).

4.1.4 Intelligenz

Die Pflanze verfügt über die nötige Intelligenz um eine passende Person zum Gießen auszuwählen.

Dazu teilt sie verfügbare Personen in Phänotypen ein. Hierbei wird in zwei Phänotypen unterteilt. Der Pflanzenliebhaber hat generell eine positive Einstellung gegenüber Pflanzen und ist gewillt diese zu gießen. Der Pflanzenhasser hat generell eine negative Haltung gegenüber Pflanzen und gießt diese, wenn überhaupt nur, wenn keine andere Person Zeit hat.

Wenn verfügbare Personen in die zwei Gruppen von Phänotypen eingeteilt wurden, wird zuerst versucht aus der Gruppe der Pflanzenliebhaber die Person mit dem höchsten Score auszuwählen. Sollte dies scheitern, weil keine Pflanzenliebhaber verfügbar sind, wird in der Gruppe der Pflanzenhasser die Person mit dem höchsten Score ausgewählt. Der Score stellt hierbei das bisherige Verhalten der Person gegenüber der Pflanze dar.

Die Pflanze schaut zuerst nach verfügbaren Personen in der Nähe und wenn sie dort nicht fündig wird, schaut sie in die Online-Kalender von registrierten Personen, welche Zeit haben und daher verfügbar sind.

4.1.5 Die Rankingliste

Über die Rankingliste kann die Pflanze ihren Zustand und ihre Haltung gegenüber Personen kundgeben. Ihren Zustand kann sie über einen Smiley und einen Pflanzen Avatar (Abbildung 24) ausdrücken. Die Haltung gegenüber Personen kann die Pflanze über eine Liste äußern (Abbildung 25). Der Platz, den eine Person in der Liste einnimmt, stellt die Beliebtheit der Person bei der Pflanze dar. Die Person auf Platz eins ist somit die beliebteste Person der Pflanze.



Abbildung 24: Die verschiedenen Zustände des Pflanzen-Avatars

	Place Name	Score
1.	Benny	20
2.	Lars	10
3.	Hans	5

Abbildung 25: Die Rankingliste

4.2 Versäumnisse und Verbesserungen

4.2.1 Die Pflanze

Hier wäre ein besserer Sprachsynthesizer zu verwenden, deren Ausgaben lebendiger klingen. Auch könnte man die Sätze etwas variieren und die Pflanze könnte sich, nachdem sie erfolgreich gegossen wurde, bedanken. Auch könnte die Pflanze durch Spracherkennung erweitert werden, sodass z.B. eine Person eine Gießanfrage verbal ablehnen kann.

4.2.2 Lokalisation und Identifikation

Hier müsste die Erkennung von Personen verbessert werden. Dazu könnte man die Daten von mehreren Kinects fusionieren, um Fehler bei der Erkennung und Lokalisation so gering wie möglich zu halten, da die Wahrscheinlichkeit groß ist, dass mindestens eine Kinect das Gesicht einer Person aufnimmt. Auch könnten so Fehler bei der Skelettberechnung ausgeglichen werden.

Ein weiteres Problem ist die Latenz bei der Identifikation. Hier könnte man auf die Verwendung von mehreren Gesichtsidifizierer verzichten und einen Gesichtsidifizierer verwenden der alle Gesichter kennt. Auch die Fehler bei der Identifikation sind ein Problem. Ein Grund könnte die geringe Auflösung der Kinect sein. Aber auch die Gesichtsidifikationsalgorithmen sind nicht ideal geeignet. Algorithmen basierend auf neuronalen Netzen könnten hier bessere Ergebnisse liefern.

Wenn man sich dafür entscheidet die Kinects gegen Kameras mit einer höheren Auflösung auszutauschen, muss man sich selbst um die Erkennung und Lokalisation von Personen kümmern.

4.2.3 Time Awareness

Hier könnte man durch das Einbeziehen von Reisewegen eine intelligentere Auswahl über verfügbare Personen treffen um einem ungerechtfertigten Punkteabzug vorzubeugen. Dazu müsste der aktuelle Standort der und das voraussichtlich verwendete Transportmittel bekannt sein. Das heißt das System müsste den Standort ermitteln und schätzen, welches Transportmittel verwendet werden könnte.

4.2.4 Kommunikation

Neben der Near Distance Communication könnte auch die Long Distance Communication erweitert werden. Zum Beispiel könnte es den Besitzern ermöglicht werden auf schriftliche Gießanfragen zu antworten, um diese z.B. ablehnen zu können. Auch könnte die E-Mail bzw. Instant Messaging Kommunikation durch eine App ausgetauscht werden. Dadurch würde sich eine Textinterpretation erübrigen.

4.3 Fazit

Es konnte ein Prototyp gebaut werden, der die Anforderungen erfüllt. Auch konnten mögliche Verbesserungen und Nachbesserungen für die nächste Iterationsstufe des Prototypens erarbeitet werden. Was noch nicht betrachtet wurde, sind die nächsten Schritte neben Nach- und Verbesserungen des Systems. Hierbei wäre ein Schritt das System um Mimikererkennung zu erweitern, um die Empathie zu verbessern. Bisher wurde die Empathie anhand der generellen Einstellung gegenüber Pflanzen angepasst, welches sehr statisch ist.

Auch wäre denkbar, dass die Phänotypen nicht statisch sind, sondern sich verändern, je nachdem wie sich der Benutzer in der Vergangenheit gegenüber der Pflanze verhalten hat.

5 Zusammenfassung und Ausblick

5.1 Zusammenfassung

In dieser Arbeit wurde gezeigt, dass es möglich ist einen proaktiven und smarten Pflanzen-Companion zu bauen, der mit seiner Umwelt kommunizieren kann.

Dazu wurde ein verteiltes System entworfen, dessen Komponenten mit Hilfe eines Frameworks über eine Publish/Subscribe-Architektur (Eichler, 2014) miteinander kommunizieren. Dieses System ist lose gekoppelt und dessen Komponenten lassen sich bei Bedarf auslagern. Manche Komponenten sind zudem so generisch, dass diese wiederverwendet werden können oder durch die Offenheit der Publish/Subscribe-Architektur auch anderen Systemen oder Komponenten zur Verfügung stehen wie z.B. die Gesichtsidentifikation.

Dadurch wurde auch gezeigt, dass es möglich ist einen verteilten Companion zu bauen, dessen Komponenten nicht an einem Punkt konzentriert sein müssen und der sich dadurch leicht in eine verteilte Smart-Home-Umgebung integrieren lässt.

5.2 Ausblick

Der Pflanzen-Companion stellt einen Gegenstand dar, der von Zeit zu Zeit menschlicher Aufmerksamkeit bedarf. Diese Aufmerksamkeit bekommt der Pflanzen-Companion, indem er proaktiv wird. Wenn der Companion feststellt, dass er Aufmerksamkeit in Form vom Gießen braucht, spricht er eine passende Person an. Außerdem wird eine Bedürftigkeit geweckt, indem der Zustand der Erdfeuchte als emotionaler Zustand verpackt wird und über Smileys und einen animierten Avatar geäußert wird.

Dieses Grundprinzip eines Objekts, welches ein gewisses Maß an Aufmerksamkeit braucht, lässt sich auch auf andere Objekte des alltäglichen Lebens übertragen. Beispiele wären hier die Kaffeemaschine oder der Wasserkocher, die von Zeit zu Zeit entkalkt bzw. geputzt werden wollen. Der Wasserkocher könnte sich hierbei an eine passende und verfügbare Person wenden und durch Emotionen diese zum Putzen bewegen. Auch könnte der Wasserkocher seinen emotionalen Zustand durch einen Smiley auf einen Display ausgeben. Dies lässt sich aber nicht nur auf Objekte des Wohnbereichs übertragen, sondern auch auf Objekte außerhalb diesem wie z.B. Autos, Motorräder und Fahrräder. Ein Auto könnte sich z.B. bei seinem Besitzer melden, wenn Wartungsarbeiten anstehen. Dabei könnte das Auto durch Emotionen seiner Forderung Nachdruck verleihen oder jemanden anderen fragen, der gerade zeitlich nicht gebunden ist, wie z.B. der oder die Partner/in des Besitzers. Es gibt aber auch Anwendungsgebiete außerhalb des Consumerbereichs wie z.B. im Professionalbereich. Denkbar wären hier Eisenbahnweichen, die Bescheid geben, wenn sie drohen defekt zu gehen. Hier könnte sich die Weiche an ein Fachpersonal wenden, das gerade Zeit hat und ihrer Forderung durch Emotionen zusätzlich Nachdruck verleihen.

6 Quellen

Literaturverzeichnis

- Adams, J. (7. 3 2014). *Raspberry Pi*. Abgerufen am 7. 4 2017 von <https://www.raspberrypi.org/documentation/hardware/raspberrypi/mechanical/Raspberry-Pi-B-Plus-V1.2-Mechanical-Drawing.pdf>
- Advanced Realtime Tracking GmbH. (kein Datum). *ART Advanced Realtime Tracking*. Abgerufen am 7. 4 2017 von <http://www.ar-tracking.com/home/>
- Angelini, L., Caparrotta, S., Khaled, O. A., & Mugellini, E. (2016). EmotiPlant: Human-Plant Interaction for Older Adults. *Proceedings of the TEI '16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction* (S. 373-379). New York, NY, USA: ACM. doi:10.1145/2839462.2856548
- Angular. (kein Datum). *One framework. - Angular*. Abgerufen am 9. 4 2017 von <https://angular.io/>
- Arduino. (7. 4 2017). *Arduino*. Von <https://www.arduino.cc/> abgerufen
- Arkin, R. C. (1998). *Behavior-based robotics*. MIT press.
- Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (Jul 1997). Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19, 711-720. doi:10.1109/34.598228
- Belliger, A. (2006). *ANThology: Ein einführendes Handbuch zur Akteur-Netzwerk-Theorie*. transcript Verlag.
- Black, A. W. (kein Datum). *Festvox: Festival*. Abgerufen am 9. 4 2017 von <http://festvox.org/festival/>
- Bonz, J. (5. Juli 2005). *docslide.us*. Abgerufen am 1. April 2017 von docslide.us/documents/ant-in-zehn-minuten1.html
- Brauer, H. S. (2014). *Camera based Human Localization and Recognition in Smart Environments*. Ph.D. dissertation, dissertation, University of the West of Scotland, 2014.[Online]. Available: <http://users.informatik.haw-hamburg.de/ubicomp/arbeiten/phd/brauer.pdf>.
- Cooney, M., Nishio, S., & Ishiguro, H. (december 2014). Affectionate Interaction with a Small Humanoid Robot Capable of Recognizing Social Touch Behavior. *ACM Trans. Interact. Intell. Syst.*, 4, 19:1--19:32. doi:10.1145/2685395
- Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From Game Design Elements to Gamefulness: Defining "Gamification". *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (S. 9-15). New York, NY, USA: ACM. doi:10.1145/2181037.2181040

- Eichler, T. (2014). Agentenbasierte Middleware zur Entwicklerunterstützung in einem Smart-Home-Labor. Hamburg, Germany, Hochschule für Angewandte Wissenschaften Hamburg, Diplomarbeit.
- eSpeak Projekt. (kein Datum). *eSpeak: Speech Synthesizer*. Abgerufen am 9. 4 2017 von <http://espeak.sourceforge.net/>
- Friedemann Mattern, C. F. (26. 1 2010). *The Distributed Systems Group*. Abgerufen am 7. 4 2017 von <http://www.vs.inf.ethz.ch/publ/papers/Internet-der-Dinge.pdf>
- Google. (kein Datum). *Google Kalender*. Abgerufen am 9. 4 2017 von <https://www.google.com/intl/de/calendar/about/>
- Hammerschmidt, J., Hermann, T., Walender, A., & Krömker, N. (Oct 2015). InfoPlant: Multimodal augmentation of plants for enhanced human-computer interaction. *Cognitive Infocommunications (CogInfoCom), 2015 6th IEEE International Conference on*, (S. 511-516). doi:10.1109/CogInfoCom.2015.7390646
- HAW Hamburg. (kein Datum). *CSTI*. Abgerufen am 7. 4 2017 von <https://csti.haw-hamburg.de/>
- He, D.-C., & Wang, L. (1990). Texture unit, texture spectrum, and texture analysis. *IEEE transactions on Geoscience and Remote Sensing*, *28*, 509-512.
- Helene Steiner, P. J. (3. 8 2015). *Project Florence - Microsoft Research*. Abgerufen am 3. 4 2017 von <https://www.microsoft.com/en-us/research/project/project-florence/>
- Kawakami, A., Tsukada, K., Kambara, K., & Siio, I. (2011). PotPet: Pet-like Flowerpot Robot. *Proceedings of the Fifth International Conference on Tangible, Embedded, and Embodied Interaction* (S. 263-264). New York, NY, USA: ACM. doi:10.1145/1935701.1935755
- Latour, B. (2002). *Die Hoffnung der Pandora : Untersuchungen zur Wirklichkeit der Wissenschaft* (1. Aufl. Ausg.). Frankfurt, am, Main: Suhrkamp.
- Latour, B. (2007). Eine neue Soziologie für eine neue Gesellschaft: Einführung in die Akteur-Netzwerk-Theorie, aus dem Französischen von Gustav Roßler. *Frankfurt am Main: Suhrkamp*.
- Michalowski, M. P., Sabanovic, S., & Kozima, H. (2007). A Dancing Robot for Rhythmic Social Interaction. *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction* (S. 89-96). New York, NY, USA: ACM. doi:10.1145/1228716.1228729
- Microsoft. (kein Datum). *Entwickeln mit Kinect*. Abgerufen am 9. 4 2017 von <https://developer.microsoft.com/de-de/windows/kinect/develop>
- Microsoft. (kein Datum). *JointType Enumeration*. Abgerufen am 9. 4 2017 von <https://msdn.microsoft.com/en-us/library/microsoft.kinect.jointtype.aspx>
- Microsoft. (kein Datum). *Kinect API Overview*. Abgerufen am 9. 4 2017 von <https://msdn.microsoft.com/de-de/library/dn782033.aspx>
- Ministerium für Inneres und Kommunales Nordrhein-Westfalen. (kein Datum). *Gesetze und Verordnungen | Landesrecht NRW*. Abgerufen am 9. 4 2017 von https://recht.nrw.de/lmi/owa/br_bes_detail?sg=0&menu=1&bes_id=3620&anw_nr=2&aufgehoben=N&det_id=381714

- Mueller, E. T. (2000). A Calendar with Common Sense. *Proceedings of the 5th International Conference on Intelligent User Interfaces* (S. 198-201). New York, NY, USA: ACM. doi:10.1145/325737.325842
- OpenCV. (kein Datum). *OpenCV library*. Abgerufen am 8. 4 2017 von <http://opencv.org/>
- Raspberry Pi Foundation. (kein Datum). *Raspberry Pi*. Abgerufen am 7. 4 2017 von <https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835/READEME.md>
- Raspberry Pi Foundation. (kein Datum). *Raspberry Pi*. Abgerufen am 9. 4 2017 von <https://www.raspberrypi.org/>
- Reeves, B., & Nass, C. (1996). The Media Equation: How people treat computers, television, and new media like real people and places. *CSLI Publications and Cambridge*.
- Resnick, M. (2001). Lifelong kindergarten. *presentation delivered at the annual symposium of the Forum for the Future of Higher Education, Aspen, Colorado*.
- Stimson, M. (23. 9 2015). *Raspberry Pi*. Abgerufen am 7. 4 2017 von https://www.raspberrypi.org/documentation/hardware/raspberrypi/mechanical/rpi-zero-v1_2_dimensions.pdf
- The Apache Software Foundation. (kein Datum). *Maven*. Abgerufen am 9. 4 2017 von <https://maven.apache.org/>
- Turk, M. A., & Pentland, A. P. (Jun 1991). Face recognition using eigenfaces. *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (S. 586-591). doi:10.1109/CVPR.1991.139758
- Turkle, S. (2005). *The second self: Computers and the human spirit*. Mit Press.
- Ubuntu. (kein Datum). *Ubuntu Manpage: spd-say - send text-to-speech output request to speech-dispatcher*. Abgerufen am 9. 4 2017 von <http://manpages.ubuntu.com/manpages/trusty/man1/spd-say.1.html>
- University of California, Berkeley. (4. 1 2011). *UC Berkeley Motescope*. Abgerufen am 3. 4 2017 von <http://smote.cs.berkeley.edu/motescope/>
- Upton, E. (26. 11 2015). *Raspberry Pi*. Abgerufen am 8. 4 2017 von <https://www.raspberrypi.org/blog/raspberry-pi-zero/>
- Wada, K., & Shibata, T. (Oct 2007). Living With Seal Robots #x2014;Its Sociopsychological and Physiological Influences on the Elderly at a Care House. *IEEE Transactions on Robotics*, 23, 972-980. doi:10.1109/TRO.2007.906261
- Weiser, M. (July 1999). The Computer for the 21st Century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3, 3-11. doi:10.1145/329124.329126
- Wilks, Y. (2010). *Close engagements with artificial companions: key social, psychological, ethical and design issues* (Bd. 8). John Benjamins Publishing.
- WiringPi. (kein Datum). *WiringPi*. Abgerufen am 9. 4 2017 von <http://wiringpi.com/>
- Wooldridge, M. (2002). *An introduction to multiagent systems*. John Wiley & Sons.

Versicherung über Selbstständigkeit

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

Hamburg, den _____