



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Ausarbeitung - Anwendungen 2

Mosawer Ahmad Nurzai

Action Planning bei Assistenzrobotern

*Fakultät Technik und Informatik
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science
Department of Computer Science*

Mosawer Ahmad Nurzai

Action Planning bei Assistenzrobotern

Ausarbeitung - Anwendungen 2 eingereicht im Rahmen von Anwendungen 2

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

Gutachter: Prof. Dr. Kai von Luck
Gutachter: Prof. Dr. Bettina Buth

Betreuer: Prof. Dr. Andreas Meisel

Eingereicht am: 31. August 2014

Mosawer Ahmad Nurzai

Thema der Arbeit

Action Planning bei Assistenzrobotern

Stichworte

action planning, roboter, Assistenzroboter, planning, ROS, case based, htn, goap, Framework

Mosawer Ahmad Nurzai

Title of the paper

Action Planning for assistant robots

Keywords

action planning, roboter, assistant Robot, planning, ROS, case based, htn, goap, framework

Inhaltsverzeichnis

1 Einführung	5
1.1 Aufbau der Arbeit	5
1.2 Motivation	5
1.3 Ziel der Arbeit	6
2 Verwandte Arbeiten	7
2.1 Action Planning model for autonomous mobile robots	7
2.1.1 Ziel	7
2.1.2 Vorgehen	7
2.1.3 Ergebnisse	9
2.2 Integrating Planning And Execution For ROS Enabled Service Robots Using Hierarchical Action Representations	9
2.2.1 Ziel	9
2.2.2 Vorgehen	10
2.2.3 Ergebnisse	11
2.3 Designing a Cognitive Case-Based Planning Framework for Home Service Robots	12
2.3.1 Ziel	12
2.3.2 Vorgehen	12
2.3.3 Ergebnisse	14
3 Analyse	14
3.1 Vergleich	14
3.2 Bewertung & Relevanz	15
4 Schluss	16
4.1 Zusammenfassung	17
4.2 Ausblick	17
Literaturverzeichnis	18
Abbildungsverzeichnis	19

1 Einführung

In diesem Kapitel wird der Aufbau der Ausarbeitung präsentiert, außerdem wird anhand von Szenarien, die Motivation dargestellt. Die Ausarbeitung bzw. die praktische Umsetzung erfolgt an der HAW Hamburg in der Forschungsgruppe „Robot Vision“, diesbezüglich erfolgt eine kurze Darstellung der Laborumgebung.

1.1 Aufbau der Arbeit

Die Arbeit ist in fünf Kapitel gegliedert. Das erste (1) Kapitel ist eine Einführung zu dieser Arbeit und beschreibt die Motivation und das Ziel dieser Arbeit. Der Hauptteil dieser Arbeit stellt das zweite (2) Kapitel dar. In diesem Kapitel werden verwandte Arbeiten bzw. verwandte Ideen zu dieser Arbeit vorgestellt. Das dritte (3) Kapitel bezieht sich stark auf das zweite (2) Kapitel und zieht ein Vergleich der vorgestellten Arbeiten und der vorliegenden Arbeit. Es erfolgt ebenfalls eine Bewertung und Relevanz zu dieser Arbeit. Das vierte letzte (4) Kapitel stellt den Schluß dar. In dem Kapitel wird die Arbeit zusammengefasst und ein Ausblick, zum weiteren Vorgehen, wird präsentiert.

1.2 Motivation

Die Ausarbeitung soll in diesem Abschnitt motiviert werden. Hierzu erfolgt die Darstellung des folgenden Szenario. In Abbildung 1 ist ein typisches Szenario abgebildet, wenn der Mensch

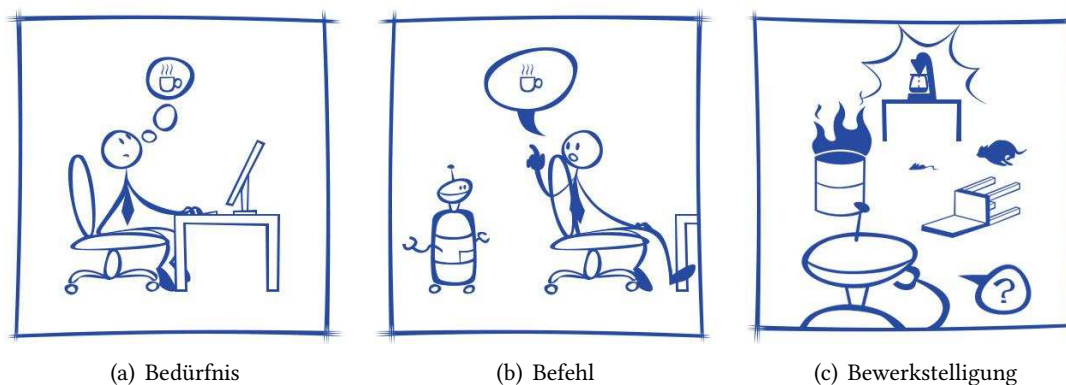


Abbildung 1: Szenario: „Hol mir Kaffee!“

an einen assistierenden/helfenden Roboter denkt. Hinter dem Szenario „Hol mir Kaffee!“ stecken verschiedene Schritte, wie schon in der Abbildung zu erkennen ist. Die drei „B“'s (**B**edürfnis,

Befehl, Werkstellung) müssen vorhanden sein bzw. vollzogen werden. Diese Ausarbeitung konzentriert sich vor allem um das letzte „B“, sprich die Werkstellung. In dieser Phase muss der Roboter an sein Ziel gelangen, in dem er zur Kaffeekanne geht. Es stellt sich nun die Frage, wie macht er es am besten, ohne die Umwelt und sich selber zu beschädigen. Was für Schritte sind hierbei notwendig. Um genauer zu sein wie sieht der Plan und die dazu entstehenden Arbeitsschritte zur Werkstellung dieser Aufgabe aus.

Die Übertragung der präsentierten Vision soll auf den Roboter (2) übertragen werden.

Der Roboter besteht aus der Mobilien Plattform SCITOS G5¹ von der Firma MetraLabs² und einem 5-DOF-Arm von der Firma SCHUNK³. Mit Hilfe des SCITOS G5 kann sich der Roboter bewegen und durch den Roboterarm der Firma SCHUNK besteht die Möglichkeit des Greifens. Am SCHUNK-Arm befindet sich eine Kinect⁴, womit der Roboter seine Umwelt wahrnehmen kann. An der mobilen Plattform befinden sich zwei 2D-Laserscanner (Leuze⁵ und Hokuyo⁶), mit denen eine 2D-Visualisierung der Umgebung erstellt werden kann. Um den Roboter anzu steuern und mit ihm interagieren zu können, wird das spezielle Roboterbetriebssystem ROS (Robot Operating System) verwendet, welches eher ein Software-Framework darstellt⁷.

1.3 Ziel der Arbeit

Ziel der Arbeit ist es einen groben Überblick über verschiedene Verfahren und Möglichkeiten von Action Planning bei Assistenzrobotern zu gewinnen. Die Arbeit dient ebenfalls als Einstieg in diesem Bereich. Erkenntnisse werden in den weiteren Arbeiten bis hin zur Master-Thesis verwertet.



Abbildung 2: Roboter im Robot Vision Labor

¹http://metralabs.com/index.php?option=com_content&view=article&id=70&Itemid=64

²<http://metralabs.com>

³<http://www.de.schunk.com>

⁴<http://www.xbox.com/de-DE/Kinect>

⁵<http://www.leuze-electronic.de>

⁶<http://www.hokuyo-aut.jp>

⁷<http://www.ros.org>

2 Verwandte Arbeiten

In diesem Kapitel werden verwandte Arbeiten vorgestellt. Es wurde darauf geachtet, dass jeder dieser Arbeiten einen anderen Ansatz verwendet oder dass es in der Herangehensweise einen Unterschied gibt. Drei Arbeiten (Roesener u. a. [2007], Janssen u. a. [2013], Jung u. a. [2007]) werden vorgestellt. Nach der Vorstellung werden die jeweiligen Ziele, Vorgehen und Ergebnisse der Arbeiten präsentiert.

2.1 Action Planning model for autonomous mobile robots

In der Arbeit „Action Planning model for autonomous mobile robots“ (Roesener u. a. [2007]) wird ein theoretischer Ansatz präsentiert, um das Verhalten eines Roboters auszuwählen. Die Aussage des Papers ist es, dass durch Regel-basierte Systeme, kein angemessenes Verhalten erzeugt werden kann. Regel-basierte System sind Systeme in denen, Regeln an verschiedene Bedingungen geknüpft sind. Treten die Bedingungen auf, so werden Aktionen ausgeführt. Nachteil dieses Verfahren ist es, dass es nur in Situationen anwendbar ist, wo jedes Statement als Regel (if-then) formuliert werden kann (Yue und de Byl [2006]).

2.1.1 Ziel

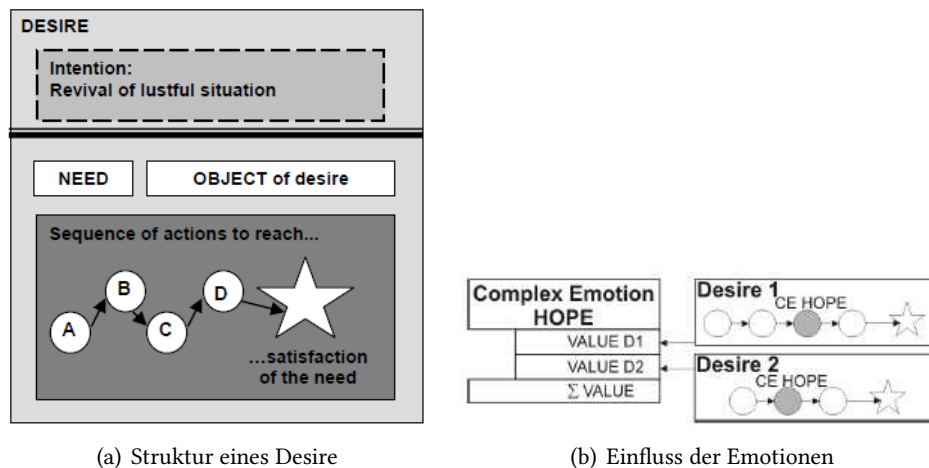
Ziel dieser Arbeit ist es eine Verhaltensarchitektur zu entwickeln. Für die Verhaltensarchitektur werden psychodynamische Annahmen getätigt. Die Annahmen stammen aus der Lehre vom Wirken innerseelicher Kräfte von Menschen. Diese besagen unter anderem aus, dass unbewusste Prozesse (Wut, Gier, etc.) existieren und der Bedarf notwendig ist. Diese Prozesse können ein Konflikt haben und sich gegenseitig hemmen, beispielsweise kann Glück, Wut hemmen.

2.1.2 Vorgehen

Um die Verhaltensarchitektur implementieren und die verschiedenen psychodynamischen Aspekte realisieren zu können, gibt es verschiedene Arten von Datenstrukturen.

- **Image:** Beschreibt die Beziehung zwischen dem Roboter und der Umgebung. Stellt ein Schnappschuss dar und liefert Daten über den aktuellen Zustand des Systems. Es gibt „perceptual“ und „mental“ Images. Perceptual (wahrnehmend) beschreibt aktuellen Zustand der Umgebung, generiert durch Sensoren. Mental (seelisch/geistlich) beschreibt Replikationen von historischen Daten, beispielsweise Events die ähnlich sind zur jetzigen Situation.

- **Episode:** Sind Sequenzen von Images. Durch Episoden können auftretende Veränderungen ermittelt werden, welche notwendig sind um Entscheidungen zu treffen.
- **Emotion:** Liefert Informationen über den momentanen physikalischen Zustand. In dieser Arbeit (Roesener u. a. [2007]) werden nicht alle Emotionen berücksichtigt, sondern es wird sich auf die Emotion „Seeking“ (Begehren) konzentriert.
- **Desire:** Bei einem „Desire“ wird ein individuelles Bedürfnis erfüllt. Desires sind die Elemente die dafür sorgen, dass eine Aktion passiert und sich der Roboter verschieden je nach Situation verhält. Desires werden durch die Episoden und Emotionen beeinflusst. Zu jedem Desire gibt es einen korrespondierenden Action Plan. Ein Desire wird durch die Erkennung einer mentalen Episode (siehe mentales Image) ausgelöst. Somit fungiert die mentale Episode als ein Trigger.



(a) Struktur eines Desire

(b) Einfluss der Emotionen

Abbildung 3: Aufbau der Verhaltensarchitektur

Quelle: Roesener u. a. [2007]

In der Abbildung 3 ist die Struktur eines Desires zu sehen. Wird ein spezifisches Szenario, in dem eine mentale Episode erkannt wurde, wahrgenommen, so hat der Roboter ein Desire, welches er erfüllen will. Der Roboter wählt Aktionen aus und nach jeder erfolgreichen Aktion wird die Wahrscheinlichkeit erhöht für den Erfolg bzw. Erfüllung des Desire.

$$if(\Delta t > t_{max}) \rightarrow S_n = S_{n-1} \quad (1)$$

Um eine Aktion ausführen zu können, müssen Transitionsbedingung erfüllt werden, um somit gleichzeitig in den nächsten Zustand zu gelangen. Wird innerhalb einer Zeit, siehe Formel 1,

der Zustand nicht gewechselt erfolgt ein Timeout und man gelangt zum vorherigen Zustand. Es kann also sein das ein Action Plan variieren kann, um das Desire des Roboters zu erfüllen. Emotionen können einen Einfluss auf die Desire haben und umgekehrt. Emotionen können ein Desire hemmen oder abbrechen. In der Realisierung heißen sie komplexe Emotionen und werden als Singleton-Pattern umgesetzt. Die Veränderung bzw. der Einfluss eines Zustandes je Desire auf einer komplexen Emotion kann definiert werden. Beispielsweise wird die Emotion „Hunger“ von zwei Desire, „Essen“ und „PC spielen“ unterschiedlich beeinflusst. Dadurch, dass „PC spielen“ einen stärkeren (negativen) Einfluss auf Hunger hat, sinkt der Hunger und somit wird das Desire „Essen“ gehemmt bzw. abgebrochen.

2.1.3 Ergebnisse

In der Arbeit wurde ein abstrakter Entwurf einer Verhaltensarchitektur für Action Planning präsentiert. Es soll das reine reaktive (Regel-basiert) Konzept verbessern. Die Verhaltensarchitektur bedient sich Aspekten der Psychodynamik, sprich es basiert auf psychoanalytischen fundierten Prinzipien. Der berechnete Action Plan soll ein Desire erfüllen.

2.2 Integrating Planning And Execution For ROS Enabled Service Robots Using Hierarchical Action Representations

In der Arbeit „Integrating Planning And Execution For ROS Enabled Service Robots Using Hierarchical Action Representations“ (Janssen u. a. [2013]) geht es um die Entwicklung eines Frameworks, welches die Integration einer Planungs- und Ausführungskomponente vorsieht. Es wird die Planungskomponente SHOP2 (Nau u. a. [2003]) und die Ausführungskomponente CRAM (Beetz u. a. [2010]) verwendet. Das Framework soll das „Geplante“ versuchen zu interpretieren und per Aktion ausführen. Replanning-Methoden sollen verwirklicht werden. SHOP2 ist ein Planer der Kategorie HTN (hierarchical task network).

Bei HTN wird ein Plan in Form eines Netzwerks erstellt. Die Abhängigkeiten zwischen Aktionen werden durch das Netz präsentiert (Menkovski und Metafas [2008]). Das Netz besitzt eine hierarchische Struktur. Die Knoten im Netz sind Tasks, welche Sub-Tasks enthalten können. Die Knoten werden durch die Kanten partiell geordnet. In der unteren Ebene werden die Sub-Tasks zerlegt, so das der finale Plan mit primitiven Aktionen entstehen kann (4).

2.2.1 Ziel

Ziel der Arbeit Janssen u. a. [2013] ist es ein funktionsfähiges Framework zu entwickeln, welches die Integration eines hierarchisches Task Network - Planer vorsieht. Der Planer soll Plane

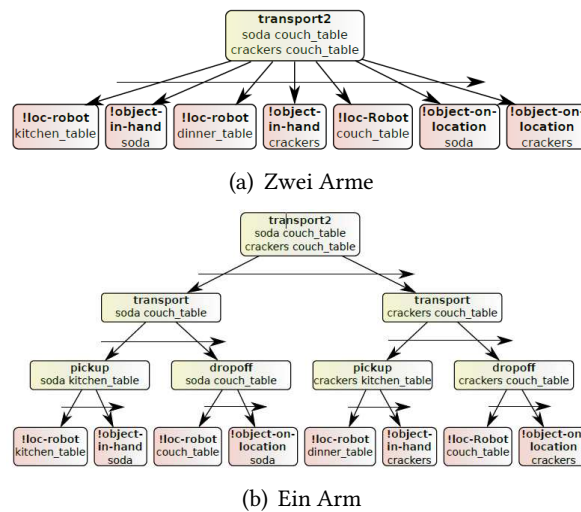


Abbildung 4: HTN-Plan: Transport von Objekten
 Quelle: Janssen u. a. [2013]

auswählen können und mit Hilfe einer hoch reaktiven und semantischen Ausführungskomponente soll der Plan ausgeführt werden. Im Laufe der Arbeit soll klarer werden, wie dies funktioniert. Das Framework soll die Funktionalität bieten, den Umgebungszustand zu analysieren, um beim Planen darauf Rücksicht zu nehmen, um ggf. ein Re-planning zu starten.

2.2.2 Vorgehen

Das Projekt benutzt eine open source, frei zugängliche Datenbank, namens RoboEarth¹, um sogenannte „Action Recipes“ zu verwenden. RoboEarth ist eine Austauschplattform für Roboter. In ihr sind beispielsweise Informationen über Karten (Navigation), Objekt Modelle, Objekt Ontologien und Task Beschreibungen (Action Recipes). Ein Action Recipe ist eine Task Beschreibung, die eine statisch, vorprogrammierte Ausführung von primitiven Action enthält. Der Planer (SHOP2) erstellt anhand der Action Recipes einen Plan für ein Task. Ziel des Planers ist es ein Plan P zu finden, welches eine Aufgabe T im Zustand S_0 durch eine Planning Domain D erfüllt. Die Domain D stellt verschiedene Operatoren und Methoden dar. Operatoren sind beispielsweise Roboter, Objekt in Hand, Objekt in Ort und Methoden sind beispielsweise navigieren und greifen. Mit der Domain kann beispielsweise folgende Aktion durchgeführt werden: Navigiere (Methode) zu Objekt A in Ort X (Operator).

Der Planer SHOP2 kann verschiedene Pläne erstellen, in der die Grundlage das selbe Action

¹<http://roboearth.org>

Recipes ist. Wenn man [Abbildung 4](#) betrachtet, stellt man fest, dass beide Pläne einen Transport beschreiben. Der Plan [4](#) wurde bei einem Roboter erstellt welcher ein Arm hat und bei [Plan 4](#) mit zwei Armen. Mit Hilfe der Kostenoptimierung, welche SHOP2 anbietet, wird der [Plan 4](#) mit weniger Schritten erstellt, da der Roboter nur einmal zum Objekt hin muss, um die zwei Gegenstände zu greifen und zu transportieren.

Der Plan, der von SHOP2 erstellt wird, muss zur nächsten Komponente, der Ausführungskomponente übergeben werden. Die Ausführungskomponente CRAM nimmt den Plan in Form einer symbolischen Actionsequenz entgegen, um diesen Plan auszuführen. Die Ausführung wird in Form der Sprache CPL formuliert. CRAM bedient sich dem Planungsansatz der reaktiven Ausführung, sprich es wird versucht bei unvorhersehbaren und dynamischen Veränderungen in der Umgebung einzugehen. Es werden sogenannte Fluents eingesetzt. Die Fluents werden ständig aktualisiert und beobachtet. Fluents beschreiben Veränderungen in der Umgebung und werden getriggert, falls eine Veränderung auftritt, somit werden bei Anomalien, welche nicht durch die Ausführungskomponente gelöst werden können, ein Replanning ausgelöst und die Planungskomponente erstellt einen neuen Plan. Das Replanning ist somit ein Recovery-Mechanismus.

2.2.3 Ergebnisse

Es wurde in dieser Arbeit ([Janssen u. a. \[2013\]](#)) ein Framework zum Planen und Ausführung von Task für ROS-betriebene Roboter entwickelt. Die [Abbildung 5](#) zeigt die Architektur des Frameworks. Das System besteht wie bereits erklärt aus der Action Recipe Datenbank (RoboEarth), dem Planer (SHOP2), der Ausführungskomponente (CRAM) und Komponenten, um die Umgebung zu beobachten und Intern aktuell zu halten.

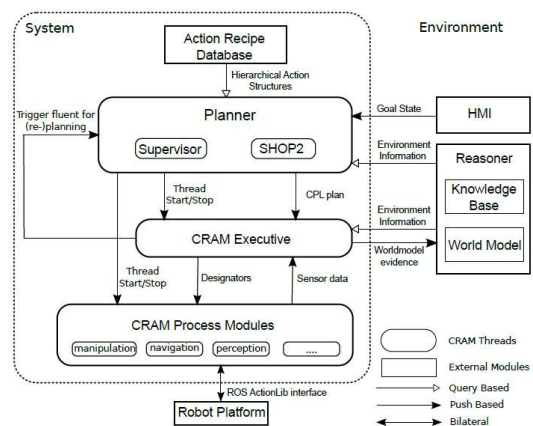


Abbildung 5: Architektur des Frameworks

Quelle: [Janssen u. a. \[2013\]](#)

2.3 Designing a Cognitive Case-Based Planning Framework for Home Service Robots

In der Arbeit „Designing a Cognitive Case-Based Planning Framework for Home Service Robots“ (Jung u. a. [2007]) geht es ebenfalls um die Entwicklung eines Planning-Frameworks für ein Home Service Roboter. Das Planen soll case-based sein. Die Behauptung, die sie aufstellen ist, dass lern-basiertes Task Planning, welches auf Human-Robot Interaction (HRI) basiert, die beste Methodik ist, um mit Veränderlichkeit/Wandelbarkeit und Unklarheit/Ungewissheit in der Umgebung umgehen zu können. Der Case-Based oder Case-Based Reasoning (CBR) Ansatz geht von der Annahme aus, dass klassische Planungsmethoden (z.B. nur HTN) ohne passendes/angemessenes Vorwissen zu komplex sind, um einen geeigneten Plan zu erstellen. Es werden Abfrage-/Wiederfindungsprozesse in CBR angewandt. Im Laufe der Arbeit soll dies klarer werden.

2.3.1 Ziel

Ziel der Arbeit (Jung u. a. [2007]) ist es ein Framework zu erstellen, welches Pläne mit dem CBR-Ansatz erzeugen kann. Hierbei wird angemerkt, dass der CBR-Ansatz gut fürs Planen geeignet ist, da er Vorkenntnisse, beim Planen, mit einbezieht und somit weniger komplex ist als klassische Planungsansätze.

2.3.2 Vorgehen

Es werden verschiedene Elemente benutzt, um das Framework zu implementieren, der CBR-Ansatz, ein kognitives Modell, Robot Task Description Language (RTDL) und Robot Task Manager (RTM). Per Benutzer-Eingabe wird dem Roboter ein Task übermittelt. Der Roboter findet den nächstmöglichen Task von bereits vorhandenen Tasks. Der Plan wird aufgestellt und ggf. modifiziert, sprich die Actionsequenz wird an die Situation angepasst.

- **CBR-Ansatz/RTM:** Im CBR-Ansatz werden Cases/Fälle/Tasks mit einander verglichen, um Ähnlichkeiten zu finden. Es werden neu auftretende Probleme gelöst, in dem man sie mit Problemen vergleicht, die bereits gelöst wurden. Im RTM wird der CBR-Ansatz umgesetzt. Es findet eine Analyse von Attributen statt, um Cases auf Gleichheit zu prüfen.

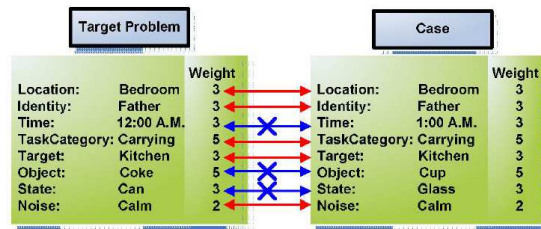


Abbildung 6: CBR: Vergleich zweier Cases

Quelle: (Jung u. a. [2007])

In Abbildung 6 werden zwei ähnliche Cases miteinander verglichen. Folgende Schritte werden vom RTM vollzogen. Findet er ein ähnlichen Case, so wird die Sequenz modifiziert, um sie an die jetzige Situation anzupassen. Gibt es kein ähnlichen Case, so findet eine Mensch-Roboter Interaktion statt. Der Roboter fragt den Menschen, was er genau möchte und so wird der Case neu erstellt (siehe Kognitives Modell).

$$case_dist = \sqrt{weight_1 \times dist_1^2 + weight_2 \times dist_2^2 + \dots weight_n \times dist_n^2} \quad (2)$$

Der Vergleich findet durch eine gewichtete euklidische Distanz statt, die sogenannte Case-Distanz (2). RTM kontrolliert den Flow (Fluss) der Tasksequenz, indem es auf die Vollendung der Sub-Tasks bzw. atomaren Aktionen (siehe RTDL) und auf Sensorinformationen achtet.

- **Kognitives Modell:** Die Verständigung zwischen Roboter und Mensch muss ebenfalls geregelt werden, damit ein leichtes kommunizieren möglich ist. Hierfür wird ein kognitives Modell benutzt, damit Roboter und Mensch das gleiche Set für die Kommunikation benutzen. Ein flexibles Planen (anpassen an die Situation) ist dadurch ebenfalls möglich. Es gibt vier kognitive Modelle: „Task“, „Interaction“, „Needs“, „User“. Ein „Task“ wird durch Kommunikation zwischen Mensch und Roboter eingeleitet. „Interaction“ beschreibt, eine notwendige Interaktion zwischen Roboter und Mensch, welches einen Dialog beschreibt. Beispielsweise fragt der Roboter, ob eine Dose Cola auch ok wäre, statt ein Glas. „Needs“ beschreibt Bedingungen und Anforderungen des Roboters, um die Grenzen des Verhaltens des Roboters zu bestimmen (beispielsweise Roboter hat nur ein Greifer). In „User“ sind individuelle Informationen über Benutzer gespeichert, welches ein individuelles Verhalten des Roboters hervorrufen kann.

Durch dieses Modell kann beispielsweise eine Task-Sequenz-Anpassung statt finden und Parameter können angepasst werden, wenn innerhalb eines Task Mehrdeutigkeit

oder/und unvollständige Informationen herrschen (Interaktion), ebenfalls kann durch eine Interaktion neue Tasks per Dialog hinzugefügt werden. Die Tasks werden somit auf Basis der Kontextinformation angepasst.

- **RTDL:** Um das CBR für das Robot Task Planning anwenden zu können, muss die Struktur einer Task verständlich gestaltet sein. Dies ist notwendig, um ein Task zu planen, Anzahl von Cases zu speichern, Sequenzen von Task zu ändern und letztendlich den finalen Plan zu einem Task aufzustellen. RTDL wird benutzt um die Struktur eines Cases zu definieren. RTDL hat eine 3-Stufen Hierarchie. Der Case/Task ist in Sub-Task unterteilt und ein Sub-Task ist wiederum in atomare Aktionen aufgeteilt. Atomare Aktionen sind Aktionen, die die primitiven Aktionen des Roboters beschreiben, beispielsweise Bewegung, Greifen, etc. Atomare Aktionen wurden vom Autor selbst definiert.

2.3.3 Ergebnisse

Diese Arbeit (Jung u. a. [2007]) hat ein Framework entwickelt, welches den CBR-Ansatz nutzt, um einen Plan zu einem Task erstellen zu können. Beim CBR-Ansatz ist es möglich neue antreffende Cases hinzuzufügen ohne die Systemarchitektur fundamental zu ändern. Mehrdeutigkeit wird, während der Aktionsausführung, durch das kognitive Modell gelöst und die Wiederbenutzung der Task zeichnet diesen Ansatz aus.

3 Analyse

In diesem Kapitel sollen die bisher vorgestellten Arbeiten untereinander verglichen werden. Zusätzlich zum Vergleich der bisher vorgestellten Arbeiten, erfolgt der Vergleich auch zum Vorgehen zukünftiger Arbeiten. Die vorgestellten Arbeiten werden bewertet, indem ein abschließendes Fazit erfolgt, außerdem wird die Relevanz der Arbeiten zu dieser und zu zukünftigen Arbeiten auf gestellt.

3.1 Vergleich

Der Vergleich der bisherigen Arbeiten und dieser Arbeit bzw. dem zukünftigen Vorgehen erfolgt anhand einer Tabelle (3.1).

Die Tabelle zeigt die Ziele, wie geplant wird, wie die Ausführung des Plans aussieht bzw. mit welcher Methodik der Plan ausgeführt wird, welche Aspekte bzw. was für Funktionalitäten benutzt werden, damit der Roboter seine Umgebung wahrnehmen kann und zuletzt wie eine Mensch-Roboter-Interaktion (Human-Roboter-Interaction [HRI]) vollzogen wird. Hierbei

	Roesener u. a. [2007]	Janssen u. a. [2013]	Jung u. a. [2007]	Eigener Ansatz
Ziele	Entwicklung einer Verhaltensarchitektur	Integration Planungs- und Ausführungskomponente, Nutzung von hierarchischen Actions	Entwicklung Cognitive Case-Base Planning Framework	Eine für ROS entwickelte Plan-basierte Architektur
Planning	Desire-getrieben, Replanning möglich	SHOP2(HTN), Replanning als Recovery-Mechanismus	Ähnlich wie HTN zusätzlich Wiederverwendung von Vorwissen, Update Case-Base	HTN, GOAP
Ausführung	Verschiedene Typen von Action, Actionpattern	CRAM, Reagieren auf dynamische Umgebung	Robot Task Manager	-
Wahrnehmung	Psychodynamische Aspekte (Image, Episode)	Reasoner PFEIL zwei Modelle (Knowledge Base & World Model)	Case-Based Reasoning System (CBRS)	Case-Based Reasoning
HRI	-	Sprachkommandos PFEIL Ziel-Task	Sprachdialog (cognitive)	Sprach-, GUI-Kommandos

Tabelle 1: Vergleich der Arbeiten

werden die Arbeiten untereinander verglichen und der eigene Ansatz wird ebenfalls vorgestellt. Die Tabelle ist selbsterklärend, da die Arbeiten schon vorgestellt wurden sind, deshalb wird nur auf den eigenen Ansatz eingegangen.

Im Laufe dieser Arbeit hat sich das Ziel, „eine für ROS entwickelte plan-basierte Architektur“ zu implementieren, entwickelt. Dabei soll als Planungsmechanismus HTN und/oder GOAP (Goal-Oriented Action Planning) verwendet werden. Wie die Ausführung aussieht ist bislang unklar, es erfordert weitere Studien und Forschung. In Projekt 2 soll dies angestrebt werden. Durch die bisher vorgestellten Arbeiten hat sich der Case-Based Reasoning-Ansatz als gut geeignete Methodik herausgestellt, um den Wahrnehmungsaspekt zu behandeln. Vorteile hierzu werden im nächsten Abschnitt dargestellt. Die Mensch-Roboter-Interaktion erfolgt durch Sprach- und GUI-Kommandos. In Projekt 1 wird bereits ein Sprachsteuerungsinterface implementiert, welches erweitert werden kann, um Task-Befehle umsetzen zu können.

3.2 Bewertung & Relevanz

In diesem Abschnitt soll kurz die vorgestellten Arbeiten bewertet und deren Relevanz zum eigenen Ansatz dargestellt werden.

- **Roesener u. a. [2007]**: Diese Arbeit benutzt menschenähnliche Elemente, um eine Verhaltensarchitektur zu implementieren. Die große Vision ist es, dass Roboter sich so verhalten und agieren sollen wie Menschen, daher ist dies ein Schritt in die richtige Richtung, wenn auch nur in kleinen Ansätzen. Die Arbeit ist sehr abstrakt dargestellt, welches positiv und negativ zu sehen ist. Durch die abstrakte Darstellung gewinnt man einen schnellen Überblick über deren Ansatz und das Konzept, jedoch fehlen konkrete Reali-

sierungsansätze. Es wird nicht gezeigt, wie ein Aktionsplan aufgestellt wird.

Diese Arbeit spielt kaum eine Rolle für den weiteren Verlauf des eigenen Ansatz, es zeigt zwar eine gute Möglichkeit, wie der Roboter menschenähnlicher gemacht werden kann, jedoch fehlt eine konkrete Implementierung.

- **Janssen u. a. [2013]**: In dieser Arbeit ist positiv zu vermerken, dass es die Funktionalität hat eine Kostenoptimierung beim Planen durch zu führen, sprich bei der gleichen Aufgabe, kann der Planer verschiedene Pläne erstellen, die verschieden gut (Laufzeit, Anzahl Sub-Tasks) sind. Es gibt es Mechanismus, der auf Umweltveränderung reagiert, so ist es möglich ein Replanning durchzuführen. Das Replanning dient hier jedoch nur als Recovery-Mechanismus, sprich bei Fehlern, wird nochmals geplant. Die Planungs- und Ausführungskomponente sind nur in der Sprache LISP verfügbar. Es erfordert also einen zusätzlichen Aufwand, um diese Komponenten in seinem System einzufügen, da LISP keine gängige Sprache ist.

Durch diese Arbeit hat sich das Ziel des eigenen Vorgehen ein wenig gewandelt. Die Erstellung einer Architektur wird nun angestrebt. Hierbei soll eventuell, genau wie in der Arbeit (**Janssen u. a. [2013]**), HTN benutzt werden, um hierarchische Pläne aufzustellen.

- **Jung u. a. [2007]**: Diese Arbeit hat das eigene Vorgehen ebenfalls stark beeinflusst. Vorteil dieser Arbeit ist, dass eine Wiederverwendung von bereits bekannten Tasks/Cases statt findet. Dieses Vorgehen gleicht dem Vorgehen des Menschen. Es findet ein On-the-Fly Planning statt, sprich im Ausführungsprozess kann ein Plan und dessen Ausführung angepasst werden, wenn beispielsweise äußere Einflüsse dies bedingen. Die Tasks werden anhand der euklidischen Distanz verglichen, dabei kann der Vergleich anhand von Heuristiken unterschiedlich ausgehen, beispielsweise bei verschiedenen Präferenzen von Benutzern. Nachteil dieser Methodik (CBR) ist, dass für ein Vergleich vordefinierte Tasks/Cases notwendig sind. Diese müssen in der Initialisierungsphase angelegt werden.

Der Case-Base-Reasoning Ansatz soll im eigenen Ansatz angewendet werden. Vorbild dient diese Arbeit (**Jung u. a. [2007]**). Es sollen also Tasks/Cases angelegt und wieder werden bzw. die Möglichkeit dies zu tun soll geboten werden.

4 Schluss

In diesem Kapitel soll diese Arbeit kurz zusammengefasst werden und ein Ausblick wird präsentiert.

4.1 Zusammenfassung

In dieser Arbeit wurden verschiedene Ansätze zu Action Planning bzw. Architekturen, die Action Planning umsetzen präsentiert. Hierzu wurde ein Bezug zum eigenen Vorgehen genommen. Zu Anfang wurde die Arbeit eingeleitet, in dem die Motivation und das Ziel der Arbeit dargestellt wurden. Im Hauptteil der Arbeit wurden verwandte Arbeiten vorgestellt. Dabei wurde von jeden der Arbeiten das Ziel, ihr Vorgehen und die Ergebnisse vorgestellt. Jeder der Arbeit enthält einen anderen Ansatz, um mit Action Planning und der Ausführung des Plans umzugehen. Im Analyse Kapitel wurden alle Arbeiten und der eigene Ansatz miteinander verglichen, dabei wurden die Arbeit ebenfalls bewertet und die Relevanz der Arbeiten bzw. der Einfluss der Arbeiten aufs eigene Vorgehen wurde vorgestellt. In diesem Kapitel erfolgt noch der Ausblick des eigenen Vorgehens.

4.2 Ausblick

Diese Arbeit hat einen ersten Einblick über verschiedene Architekturen geliefert. Der Einfluss dieser Arbeiten auf das eigene Vorgehen ist stark, da angestrebt wird eine für ROS entwickelte Plan-basierte Architektur für den Assistenzroboter im Robot-Vision-Labor zu entwickeln. In Projekt 2 sollen erste Entwürfe und Überlegungen über die Architektur gemacht werden. Die Architektur soll in der Master-Thesis weit wie möglich implementiert werden. Der Einfluss der vorgestellten Arbeiten auf das eigene Vorgehen kann in [3.2](#) nach gelesen werden. Im Master-Seminar wird versucht der Ansatz zu erklären, Aufwand und Risiken sollen ebenfalls präsentiert werden.

Literatur

- [Beetz u. a. 2010] BEETZ, M. ; MOSENLECHNER, L. ; TENORTH, M.: CRAM A Cognitive Robot Abstract Machine for everyday manipulation in human environments. In: *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct 2010, S. 1012–1017. – ISSN 2153-0858
- [Janssen u. a. 2013] JANSSEN, R. ; MEIJL, E. van ; DI MARCO, D. ; MOLENGRAFT, R. van de ; STEINBUCH, M.: Integrating planning and execution for ROS enabled service robots using hierarchical action representations. In: *Advanced Robotics (ICAR), 2013 16th International Conference on*, Nov 2013, S. 1–7
- [Jung u. a. 2007] JUNG, Yuchul ; PARK, Hogun ; CHOI, Yoonjung ; MYAENG, Sung-Hyon: Designing a Cognitive Case-Based Planning Framework for Home Service Robots. In: *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, Aug 2007, S. 827–832
- [Menkovski und Metafas 2008] MENKOVSKI, Vlado ; METAFAS, Dimitrios: AI Model for Computer Games Based on Case Based Reasoning and AI Planning. In: *Proceedings of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*. New York, NY, USA : ACM, 2008 (DIMEA '08), S. 295–302. – URL <http://doi.acm.org/10.1145/1413634.1453242>. – ISBN 978-1-60558-248-1
- [Nau u. a. 2003] NAU, Dana ; ILGHAMI, Okhtay ; KUTER, Ugur ; MURDOCK, J. W. ; WU, Dan ; YAMAN, Fusun: SHOP2: An HTN planning system. In: *Journal of Artificial Intelligence Research* 20 (2003), S. 379–404
- [Roesener u. a. 2007] ROESENER, C. ; LANG, R. ; DEUTSCH, T. ; GRUBER, A. ; PALENSKY, B.: Action planning model for autonomous mobile robots. In: *Industrial Informatics, 2007 5th IEEE International Conference on* Bd. 2, June 2007, S. 983–988. – ISSN 1935-4576
- [Yue und de Byl 2006] YUE, Billy ; BYL, Penny de: The State of the Art in Game AI Standardisation. In: *Proceedings of the 2006 International Conference on Game Research and Development*. Murdoch University, Australia, Australia : Murdoch University, 2006 (CyberGames '06), S. 41–46. – URL <http://dl.acm.org/citation.cfm?id=1234341.1234350>. – ISBN 86905-901-7

Abbildungsverzeichnis

1	Szenario: „Hol mir Kaffee!“	5
2	Roboter im Robot Vision Labor	6
3	Aufbau der Verhaltensarchitektur	8
4	HTN-Plan: Transport von Objekten	10
5	Architektur des Frameworks	11
6	CBR: Vergleich zweier Cases	13