

Regine Becher



# Semantische Objektmodellierung mittels multimodaler Interaktion





Regine Becher

**Semantische Objektmodellierung mittels multimodaler  
Interaktion**





# **Semantische Objektmodellierung mittels multimodaler Interaktion**

von  
Regine Becher



---

universitätsverlag karlsruhe

Dissertation, Universität Karlsruhe (TH)  
Fakultät für Informatik, 2008

## Impressum

Universitätsverlag Karlsruhe  
c/o Universitätsbibliothek  
Straße am Forum 2  
D-76131 Karlsruhe  
www.uvka.de



Dieses Werk ist unter folgender Creative Commons-Lizenz  
lizenziert: <http://creativecommons.org/licenses/by-nc-nd/2.0/de/>

Universitätsverlag Karlsruhe 2008  
Print on Demand

ISBN: 978-3-86644-295-5





# Semantische Objektmodellierung mittels multimodaler Interaktion

Zur Erlangung des akademischen Grades eines

**Doktors der Ingenieurwissenschaften**

von der Fakultät für Informatik

der Universität Fridericiana zu Karlsruhe (TH)

genehmigte

## Dissertation

von

Regine Becher

aus Karlsruhe

Tag der mündlichen Prüfung:

6. Mai 2008

Erster Gutachter:

Prof. Dr.-Ing. R. Dillmann

Zweiter Gutachter:

Prof. Dr.-Ing. J. Beyerer



*„The art of writing is the art of applying the seat of one's trousers to the seat of one's chair.“*

Kingsley Amis

## Danksagung

Wie die meisten Dinge im Leben ist auch eine Promotion nur möglich mit der Hilfe vieler anderer Menschen.

Meinem Doktorvater, Herrn Prof. Dillmann, danke ich sehr herzlich für seine Bereitschaft, mich als Fachfremde an seinem Lehrstuhl aufzunehmen, für all die Freiheiten, die er mir während meiner Arbeit gelassen hat und für das ausgezeichnete Arbeitsklima an seinem Lehrstuhl. Meinem Korreferenten, Herrn Prof. Beyerer, bin ich sehr dankbar für seine Bereitschaft, das Korreferat zu übernehmen, für sein Interesse an meinem Arbeitsgebiet und für seine durchdachten und hilfreichen Anmerkungen.

Meinen Kollegen bin ich nicht nur sehr dankbar für den Zusammenhalt und die gute Atmosphäre in der Gruppe, ich habe auch viel von Ihnen gelernt – nicht nur über Roboter...

Oli R. hat mir das Wildschwein als intelligentes Tier nähergebracht. Markus ist trotz aller Bemühungen daran gescheitert, meinen modischen Horizont zu erweitern. Tobias hat mich davon überzeugt, dass man Trompete auch leise spielen kann. Von Björn habe ich zwei wesentliche Dinge gelernt: improvisieren kann jeder, irgendwie, und der Fehler liegt nie in den BGTools. Auch wenn ich mich sehr bemüht habe, von Tilo zu lernen, werde ich sein Niveau an bunter Bebilderung und Arbeitseffizienz doch niemals erreichen können. Tamim hat mich in die wunderbare Welt der arabischen Küche eingeführt. Von Oli T. habe ich mein Wissen über Datenbanken – für die Defizite ist er nicht verantwortlich...

Alex hat mir bewusst gemacht, aus wie vielen einzelnen Teilen ein Segelschiff besteht, und sich auch dann noch für die Objektmodellierung begeistert, als ich das nicht mehr konnte. Nela, Christine und Isa sind das heimliche Herz des Lehrstuhls, der ohne sie vermutlich längst zusammengebrochen wäre. In hygienischer, organisatorischer und sozialer Hinsicht und bei den Reiseanträgen wären wir ohne sie verloren. Von den Feinheiten des Bierbrauens über Serienbriefe in Word bis hin zur strategischen Planung von Verwaltungsakten waren sie ein Quell unerschöpflicher Weisheit.

Raouls unerschrockene Laptop-Reparaturen sind mir ein größeres Vorbild geworden als seine zahlreichen sportlichen Aktivitäten, auch wenn letzteres sicher nicht seine Schuld ist. Er hat sich seit dem Tag meines Vorstellungsgesprächs am Lehrstuhl meiner Person und meiner Arbeit angenommen, und ohne seine bereitwillige Hilfe bei kleineren und größeren Problemen wäre ich wohl einige Male steckengeblieben.

Peter schließlich war mir nicht nur ein kompetenter Informant zu so unterschiedlichen Themen wie Steuererklärungen, Energieeinsparverordnung, die Vorzüge und Nachteile von Pelletheizungen, Planung erfolgreicher Nachzuchten bei Shelties, der Einrichtung von Meerwasseraquarien sowie dem Sortiment von Baumärkten unter besonderer Berücksichtigung der Bewässerungselemente von Gardena. Er hat auch meine Arbeit von Anfang an inhaltlich begleitet, mit mir Ideen entwickelt und teils wieder verworfen, und war mir nicht nur bei meiner Promotion der beste Zimmernachbar, den ich mir vorstellen kann.

Euch allen herzlichen Dank für Eure Unterstützung, für die vielen Ermutigungen, das entspannte und humorvolle Arbeitsklima und Eure bereitwillige Hilfe, teils zu den unmöglichsten Tages- bzw. Nachtzeiten. Peter, Raoul und Tilo haben unter viel Mühe, Zeitdruck und vermutlich auch unter einigem Gelächter meine Arbeit Korrektur gelesen – danke!

Ohne meine kompetenten und motivierten Studenten wäre meine Arbeit sicher nicht das geworden, was sie heute ist. Besonders danke ich Alexander Kasper (noch einmal), Björn Hamann, Florian Fuchs, Florian Hübner, Florian Eisenberg, Benjamin Noack, Dominik Berttram, Klaus Steinbach, Mihai Streza, Florian Rabe, Patrick Rössler und Martin Hoffmann, die sich weit über das übliche Maß hinaus und auch außerhalb der gängigen Arbeitszeiten engagierten.

Von meinen Freunden habe ich gelernt, dass man mit den begehrten zwei Buchstaben so glücklich leben kann wie ohne, und auf dem Weg dahin sowieso, und dass es nicht darauf ankommt, wie steinig oder verschlungen dieser Weg ist, solange man ihn gehen will. Danke an Maike, Geli, Christian, Ralf, Marianne, Sven, Marc, Achim, Bassi, Vera und Mischa für Eure Unterstützung und Freundschaft, das geteilte Leid und dafür, dass nach Pfingsten alles besser wird. Bestimmt.

Meine Familie hat mich in all den Jahren immer unterstützt, auch bei den ungewöhnlichsten Vorhaben. Für diesen Rückhalt, die Hilfe bei allen praktischen und weniger praktischen Problemen und für unseren liebevollen Umgang herzlichen Dank meinen Eltern, meinen Großeltern und meinen Geschwistern – ihr seid die beste Familie der Welt!

Steffen, Emma und inzwischen auch Tillmann haben diese Arbeit mit all ihren Höhen und Tiefen geduldig ertragen und mitgetragen. Danke, dass ihr da seid.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Problemstellung . . . . .	2
1.3	Lösungsansatz und Beitrag der Arbeit . . . . .	3
1.4	Gliederung der Arbeit . . . . .	5
<b>2</b>	<b>Stand der Forschung</b>	<b>7</b>
2.1	Fragestellungen und Diskussionsbedarf . . . . .	7
2.2	Semantische und funktionale Ansätze zur Objektmodellierung . . . . .	8
2.2.1	Ansätze zur Objektmodellierung im Rahmen des Programmierens durch Vormachen . . . . .	9
2.2.2	Funktionale Objektmodellierung . . . . .	13
2.2.3	Objekt-Handlungs-Komplexe . . . . .	18
2.2.4	Semantische bzw. kategorial orientierte Ansätze zur Objektmodellierung	20
2.3	Lernen von Objektmodellen . . . . .	23
2.4	Benutzerinteraktion zur (Objekt-)Modellierung . . . . .	28
2.4.1	Graphische Benutzerschnittstellen . . . . .	28
2.4.2	Sprachliche Interaktion . . . . .	29
2.4.3	Interaktion mittels Gesten, Griffen und anderer Benutzerbewegungen	31
2.4.4	Interaktion in Virtueller und Erweiterter Realität . . . . .	35
2.4.5	Multimodale Interaktion . . . . .	38
2.5	Bewertung der Arbeiten im Hinblick auf die gegebene Problemstellung . . .	40
2.6	Zusammenfassung . . . . .	40

<b>3</b>	<b>Semantische Objektrepräsentation</b>	<b>41</b>
3.1	Begriffsklärung . . . . .	42
3.2	Anforderungen an die semantische Objektrepräsentation . . . . .	46
3.3	Entwurf einer semantischen Objektrepräsentation . . . . .	48
3.3.1	Struktur und Komponenten der Repräsentation . . . . .	48
3.3.2	Attribute, Attributwerte und Werttypen . . . . .	50
3.3.3	Objektzustände . . . . .	53
3.3.4	Handlungen . . . . .	54
3.3.5	Eigenschaften . . . . .	55
3.3.6	Objektklassen . . . . .	57
3.3.7	Objektinstanzen . . . . .	59
3.4	Diskussion . . . . .	59
3.5	Zusammenfassung . . . . .	60
<b>4</b>	<b>Das interaktive Objektmodellierungssystem IOMOS</b>	<b>63</b>
4.1	Anforderungen an das Modellierungssystem . . . . .	63
4.2	Systementwurf und Ablauf der Objektmodellierung . . . . .	64
4.3	Datenverwaltung, Speicherung und Darstellung von Objektmodellen . . . . .	68
4.3.1	Verwaltung der Objektinformationen . . . . .	69
4.3.2	Objektmodelldatenbank und Speicherung von Objektmodellen . . . . .	69
4.3.3	Objektinstanzen zur Laufzeit . . . . .	71
4.4	Implementierung der Objektrepräsentation . . . . .	71
4.4.1	Schablonen . . . . .	73
4.4.2	Wertklassen . . . . .	74
4.4.3	Besonderheiten von Attributen . . . . .	75
4.5	Zusammenfassung . . . . .	76
<b>5</b>	<b>Sensorgestützte interaktive Objektmodellierung</b>	<b>77</b>
5.1	Grundlagen der Mensch-Maschine-Interaktion . . . . .	77
5.2	Anforderungen an die interaktiven Komponenten des Modellierungssystems . . . . .	79
5.3	Aufbau der Modellierungsumgebung . . . . .	80
5.4	Benutzerinteraktion zur Objektmodellierung . . . . .	89

5.4.1	Graphische Benutzerschnittstelle auf berührungssensitivem Bildschirm	89
5.4.2	Spracheingabe . . . . .	91
5.4.3	Joystick mit Inertialsensor . . . . .	92
5.4.4	Datenhandschuh mit Kraftsensorik . . . . .	93
5.4.5	Magnetfeldbasierter Positionssensor . . . . .	94
5.5	Interaktive bzw. semi-autonome Ermittlung von Objektattributen . . . . .	95
5.5.1	3D-Geometrie . . . . .	96
5.5.2	Objektansichten . . . . .	97
5.5.3	Silhouetten . . . . .	98
5.5.4	Stabile Flächen . . . . .	99
5.5.5	Hauptachsen . . . . .	103
5.5.6	Bewegungseinschränkungen bei der Manipulation . . . . .	105
5.6	Systemintegration . . . . .	110
5.7	Zusammenfassung . . . . .	111
<b>6</b>	<b>Die Domäne „Küche“ als Anwendungsszenario</b>	<b>113</b>
6.1	Grundlagen des Ontologieentwurfs und Entwurfsentscheidungen . . . . .	114
6.2	Besonderheiten der Domäne <i>Küche</i> . . . . .	116
6.3	Entwurf einer Spezifikation der Domäne <i>Küche</i> . . . . .	117
6.3.1	Objekte und Objekthierarchie . . . . .	118
6.3.2	Relationen zwischen Objekten . . . . .	118
6.3.3	Eigenschaften . . . . .	122
6.3.4	Attribute . . . . .	122
6.3.5	Werttypen . . . . .	122
6.3.6	Handlungen . . . . .	122
6.3.7	Zustände . . . . .	126
6.4	Exemplarische Darstellung einer Objektklasse . . . . .	128
6.5	Zusammenfassung . . . . .	129

<b>7</b>	<b>Evaluation und Bewertung</b>	<b>131</b>
7.1	Evaluation der semantischen Objektrepräsentation . . . . .	131
7.2	Evaluation des multimodalen interaktiven Modellierungssystems . . . . .	133
7.2.1	Grundsätzliche Überlegungen . . . . .	133
7.2.2	Interaktive Modellierung stabiler Flächen mit dem Inertial-Joystick . . . . .	135
7.2.3	Interaktive Modellierung von Manipulationseinschränkungen mit dem magnetfeldbasierten Positionssensor . . . . .	139
7.3	Evaluation der resultierenden Objektmodelle . . . . .	140
7.3.1	Bewertung der Qualität der Objektmodelle . . . . .	140
7.3.2	Verwendungsmöglichkeiten und Nutzung der erstellten Objektmodelle . . . . .	142
7.4	Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE . . . . .	146
7.5	Bewertung der Ergebnisse . . . . .	152
7.6	Zusammenfassung . . . . .	152
<b>8</b>	<b>Abschließende Betrachtungen</b>	<b>153</b>
8.1	Zusammenfassung . . . . .	153
8.2	Diskussion . . . . .	155
8.3	Ausblick . . . . .	156
<b>A</b>	<b>Technische Daten der verwendeten Hardware</b>	<b>159</b>
A.1	Farbkameras <i>Marlin 145C2</i> von Allied Vision Technologies . . . . .	159
A.2	Laserscanner Minolta <i>VI-900</i> . . . . .	160
A.3	Rotationseinheit Isel <i>RF-1</i> . . . . .	160
A.4	Rotationsgelenk Amtec <i>PowerCube PR 090</i> . . . . .	161
A.5	Inertialsensor Intersense <i>InertiaCube2</i> . . . . .	162
A.6	Plasmadisplay Panasonic <i>TH-50PF9EK</i> mit berührungssensitivem Aufsatz <i>TY-TP50P8-S</i> . . . . .	162
A.7	Datenhandschuh Immersion <i>CyberGlove</i> . . . . .	163
A.8	Magnetfeldbasierter Positionssensor Polhemus <i>Fastrack</i> . . . . .	164
	<b>Abbildungsverzeichnis</b>	<b>165</b>
	<b>Tabellenverzeichnis</b>	<b>170</b>
	<b>Algorithmenverzeichnis</b>	<b>171</b>
	<b>Literaturverzeichnis</b>	<b>172</b>

# Kapitel 1

## Einleitung

### 1.1 Motivation

Seit der Einführung der ersten Industrieroboter haben sich viele Neuerungen in der Robotik ergeben: Der Bereich Servicerobotik rückte zunehmend in den Fokus des wissenschaftlichen Interesses, Roboter zeigten zuerst in den Medien, in den letzten Jahren auch zunehmend als reale Maschinen menschliche Züge und Verhaltensweisen, Serviceroboter sind populär geworden und beginnen auf einer wenig komplexen Ebene wie z. B. als Staubsaugerroboter, Bereiche des menschlichen Alltags zu erobern. Dies hat dazu geführt, dass zum einen die Erwartungen an Roboter und deren Verhalten deutlich gestiegen sind, zum anderen Roboter zunehmend in Umgebungen agieren sollen, die sehr komplex angelegt und auf menschliche Bedürfnisse zugeschnitten sind. Gleichzeitig steigt somit auch die Schwierigkeit der Aufgaben, die ein Roboter lösen soll.

Inzwischen sind die Herausforderungen, die ein Serviceroboter im menschlichen Alltag (oft betrachtet am Beispiel *Haushalt*) zu bewältigen hat, Gegenstand zahlreicher Forschungsprojekte; erwähnt seien hier exemplarisch der Sonderforschungsbereich 588 „Humanoide Roboter – lernende und kooperierende multimodale Roboter“ (Dillmann et al. (2004)) und der „Bielefeld Robot Companion“ BIRON (Haasch et al. (2004)). Im Rahmen dieser und ähnlicher Arbeiten haben sich einige Forschungsgebiete als besonders interessant und anspruchsvoll herausgestellt. Besonders augenfällig sind dabei Bereiche wie z. B. das zweibeinige Laufen oder Greifstrategien, aber auch Interaktions- und Kooperationsaufgaben bieten noch viele Herausforderungen für die Forschung der nächsten Jahre.

Insgesamt besteht ein Serviceroboter z. B. für Anwendungen im Haushalt aus unterschiedlichen Modulen, die durch definierte Schnittstellen miteinander kommunizieren und Daten austauschen bzw. auf den gleichen Datenbestand zugreifen. Ein wichtiger Datenbereich sind dabei die Objektmodelle, die so unterschiedlichen Modulen wie beispielsweise der Objekterkennung oder der Greifplanung als Grundlage dienen. Gleichzeitig stammen die Daten, die in einem Objektmodell zusammengeführt werden, aus verschiedenen Quellen: aus Vorwissen des Roboters, aus der sensorischen Umwelterfassung oder aus der Interaktion mit dem Benutzer. Hierdurch ergeben sich komplexe und vielschichtige Informationen für jedes Objekt, die in einem Modell zusammengeführt und für verschiedene Komponenten der Robotersysteme verfügbar gehalten werden müssen. Die Anforderungen an die Objektmodellierung

steigen zudem mit der Komplexität und Dynamik der Umgebung. Bei menschenzentrierten Orten wie einem Haushalt handelt es sich in der Regel um hochdynamische Umgebungen. In einer solchen Umgebung können Objekte in einer Szene nicht nur verschwinden und wieder auftauchen, sondern es können auch neue Objekte hinzukommen, Attribute oder das Erscheinungsbild von Objekten können sich verändern oder es können gänzlich neue Objekte, vielleicht sogar mit bisher unbekannter Funktionalität, auftreten. All diesen Besonderheiten hat die Objektmodellierung Rechnung zu tragen. Die Objektrepräsentation als zentraler Bestandteil der Wissensbasis eines solchen Roboters muss somit hinreichend ausdrucksstark, flexibel und erweiterbar sein.

## 1.2 Problemstellung

Wie bereits erläutert, sind die Anforderungen an die Objektmodellierung für Serviceroboter in menschenzentrierten Umgebungen wie Haushalten sehr vielschichtig. Neben der Dynamik der Umgebung sollen auch Veränderungen an den Objekten<sup>1</sup> abgebildet und neue Objekte modelliert werden. Zusätzlich greifen Funktionskomponenten des Serviceroboters auf die Objektmodelle zu, müssen Informationen über Objekte erhalten und neue Objektinformationen beisteuern können.

Bei der Objektmodellierung ist auch zu beachten, dass bestimmte Objektinformationen in solchen Umgebungen nur mit Unsicherheiten autonom durch den Roboter bestimmbar sind, da sie sensorisch schwer zu erfassen, verrauscht oder fehlerhaft sind. Es handelt sich hierbei um benutzerspezifisches Wissen wie die Lieblingstasse eines bestimmten Benutzers, komplexes Handlungswissen wie unterschiedliche Bediensequenzen eines Backofens oder um Normen der menschlichen Gesellschaft wie das korrekte Eindecken des Tisches für unterschiedliche Anlässe und Personenzahlen.

Weiterhin soll die Objektmodellierung für solche Anwendungen auch die Interaktion des Roboters mit dem Menschen und ihre bidirektionale Kommunikation möglichst erleichtern. Hierzu ist es von Vorteil, wenn Mensch und Roboter die gleichen Begriffe verwenden und in den gleichen Kategorien „denken“. Dies macht es dem menschlichen Benutzer möglichst einfach, den Roboter zu verstehen. Zusätzlich wird dadurch die Aufgabe des Roboters erleichtert, menschliche Äußerungen innerhalb seines Systems einzuordnen, zu verstehen und umzusetzen. Die Akzeptanz eines Robotersystems wird hierdurch wesentlich erhöht – denn auf eine Anweisung wie „Bring mir etwas Warmes zu trinken.“ hin erwartet der Mensch in der Regel nicht, eine Tasse voll heißen Wassers serviert zu bekommen. Menschliche Begriffe und Kategorien müssen also in der Objektrepräsentation des Roboters wiedergespiegelt werden und auch als Steuerungskriterien für seine Wahrnehmungskomponenten verwendet werden. Die zu repräsentierenden Objektbestandteile sind insbesondere Eigenschaften und Attribute von Objekten, die Zustände, die sie einnehmen können, sowie mit ihnen ausführbare Handlungen. Durch den Einsatz dieser Objektinformationen wird eine verbesserte und robustere Verarbeitung und Interaktion zwischen Mensch und Roboter erreicht.

---

<sup>1</sup>Im Rahmen dieser Arbeit wird der Begriff „Objekt“ anstelle des Begriffes „Gegenstand“ verwendet, um zu verdeutlichen, dass die vorliegenden Ergebnisse sich nicht nur auf starre (nicht deformierbare) Körper anwenden lassen, sondern auch auf Flüssigkeiten, Pulver etc. Dies ist besonders relevant für die gewählte Domäne „Küche“, in der viele solcher Objekte auftreten bzw. andere Gegenstände durch äußere Einwirkungen verändert werden können, z. B. durch Zerschneiden.

Zu den Objektinformationen, die nur teilweise automatisch extrahierbar sind, gehört auch funktionales Objektwissen. Es dient als wichtige Grundlage für Planung, Problemlösung und Umsetzung von Anweisungen des Benutzers. Beispielsweise ist es hilfreich zu wissen, dass jedes Gefäß notfalls als Gießkannenersatz verwendet werden kann, wenn gerade keine Gießkanne zur Hand ist. Das jeweils relevante Wissen ist dabei in hohem Maße von der konkreten Domäne und den gewünschten Anwendungen abhängig – nicht jede Information ist in jedem Kontext sinnvoll, und teilweise unterscheidet sich das menschliche Verständnis von Objekten je nach Kontext. So hängt z. B. die Verwendung der Begriffe „heiß“ und „kalt“ von der jeweiligen Anwendung ab, und je nach Verwendungszweck und anstehenden Aufgaben werden Objekte von Menschen unterschiedlichen Kategorien zugeordnet. Konkrete Objektmodelle können also immer nur im Hinblick auf eine gegebene Domäne, einen bestimmten Kontext oder die gewünschten Anwendungen erstellt werden.

Aus diesen Überlegungen ergeben sich folgende Probleme und Herausforderungen für die Modellierung von Objekten beispielsweise für einen Serviceroboter im Haushalt:

- Dynamische, menschenzentrierte Umgebung
- Individualität des Benutzers
- Datenaustausch mit anderen Systemkomponenten
- Autonom nicht bestimmbares Objektwissen
- Gegenseitige Verständlichkeit der Äußerungen und Handlungen von Roboter und Benutzer
- Erfassung und Repräsentation funktionalen Objektwissens
- Kontext- und Anwendungsabhängigkeit der Objektmodelle

Um einen erfolgreichen Einsatz von Servicerobotern in menschenzentrierten Umgebungen zu ermöglichen und eine entsprechende Akzeptanz bei ihren menschlichen Benutzern herzustellen, ist also ein Konzept für die Objektmodellierung zu entwickeln, das diese Punkte in Betracht zieht.

## 1.3 Lösungsansatz und Beitrag der Arbeit

Als Lösungsansatz für die gegebene Problemstellung wird eine semantische Objektrepräsentation vorgeschlagen. Hierbei wird semantisches Objektwissen unter Erhalt der für das menschliche Verständnis relevanten Objektstruktur in Objektmodelle abgebildet. Diese semantischen Objektmodelle enthalten unter anderem funktionale Objektinformationen und sind in ihrem semantischen Gehalt spezifisch für eine bestimmte Anwendung oder Domäne.

In diesem Lösungsansatz wird das relevante Wissen über eine Domäne nicht nur insoweit modelliert, als es für die unmittelbare Verarbeitung auf einem Roboter notwendig ist. Darüber hinaus wird auch Objektinformation in die Modelle integriert, die im alltäglichen Umgang

von Menschen verwendet wird und z. B. zum Problemlösen oder zur Handlungsplanung notwendig ist. Diese Integration erleichtert die Interaktion mit dem Menschen, da der Roboter in den gleichen Begriffen über Objekte „denken“ und reden kann, wie Menschen dies tun. Sowohl die Darstellung von Roboterwissen oder aktuellen Zuständen für den Menschen als auch die Instruierung durch den Menschen bzw. die Kommunikation mit dem Benutzer werden hierdurch erleichtert oder überhaupt erst ermöglicht. Durch die Modellierung semantischer Objektinformation werden auch funktionale Objektaspekte repräsentiert, die wiederum eine direkte Verknüpfung von Objektwissen mit Handlungswissen darstellen und hierdurch erst eine erfolgreiche Handlungsplanung und -erkennung ermöglichen. Gleichzeitig wird Wert darauf gelegt, die Objektrepräsentation in ihrer Datenstruktur möglichst flexibel und erweiterbar zu halten. Die Objektmodelle können dadurch entsprechend den jeweiligen Gegebenheiten an die Umwelt angepasst werden, da starre oder sogar fest implementierte Modelle der Realität in Alltagsumgebungen nicht gerecht werden können.

Der vorgeschlagene Lösungsansatz legt besonderes Augenmerk auf die Generierung der Objektmodelle. Weite Teile des zu modellierenden semantischen Objektwissens können nur schwer oder überhaupt nicht automatisch durch ein Robotersystem extrahiert werden. Zur Objektmodellierung ist also in jedem Fall die Hilfe eines menschlichen Benutzers notwendig. Dies stellt im Rahmen der hier vorgeschlagenen Vorgehensweise keinen Nachteil dar, da der Mensch trotzdem weitestgehend von Routinetätigkeiten entlastet wird, die das Robotersystem automatisch durchführt. Interaktion mit dem Menschen wird möglichst nur eingesetzt, um besonders interessante, domänenspezifische oder nicht automatisch extrahierbare Informationen in die Objektmodelle einzufügen bzw. um automatisch erstellte Modellbestandteile zu korrigieren oder nach den Bedürfnissen des Benutzers zu ergänzen. Es ist hierbei allerdings zu beachten, dass die Objektmodelle in der Regel nicht a priori von Programmierexperten erstellt werden, wie es in vielen Modellierungsanwendungen üblich ist. Bei der hier angestrebten Benutzergruppe handelt es sich vornehmlich um nicht speziell vorgebildete Anwender, für die die Interaktion mit dem System so einfach und intuitiv wie möglich gestaltet werden soll. Die Integration des Benutzers in das System und die Verbindung des von ihm eingefügten Wissens mit den automatisch extrahierten Informationen stellen also auf der einen Seite eine Herausforderung dar, ermöglichen es aber auf der anderen Seite auch dem Benutzer, gezielt Schwerpunkte in der Objektmodellierung zu setzen, die dem Umfeld bzw. den für den Roboter relevanten Aufgabenstellungen am besten entsprechen.

Ausgehend von diesen Überlegungen wird zunächst eine allgemeine semantische Objektrepräsentation vorgeschlagen, die domänenunabhängig eingesetzt werden kann. Danach wird das **Interaktive ObjektMO**dellierungs**S**ystem **IOMOS** vorgestellt, in dem Objekte interaktiv mit dem Benutzer modelliert werden können unter Nutzung semi-autonom akquirierter Objektinformationen. Es stellen sich somit sowohl die Problematik der autonom bzw. semi-autonom zu gewinnenden Objektmodellbestandteile und einer sinnvollen und möglichst nutzbringenden Interaktion mit dem Menschen, als auch die der Integration dieser beiden Bestandteile. Die hierbei generierten Objektmodelle werden in der semantischen Objektrepräsentation abgelegt. Abschließend wird das System im Hinblick auf eine konkrete Anwendung unter Verwendung einer im Rahmen dieser Arbeit erstellten Taxonomie von Begrifflichkeiten für die Domäne „Küche“ evaluiert.

Die Ergebnisse dieser Arbeit können in verschiedener Hinsicht verwendet werden: Die semantische Objektrepräsentation ist so allgemein, dass sie auch auf andere Domänen angewendet



werden kann. Es bieten sich hierbei insbesondere Anwendungen in menschenzentrierten und dynamischen Umgebungen an, bei denen ein hoher Anteil an Interaktion mit dem Menschen zu erwarten ist. Das interaktive Modellierungssystem IOMOS generiert interaktiv erstellte semantische Modelle von konkreten Objekten, die in anderen Systemkomponenten eines Roboters wie Szenenrepräsentation, Objekterkennung oder Greifplanung Verwendung finden.

Die Kernthesen der Arbeit stellen sich wie folgt dar:

- Objektmodelle für komplexe Robotersysteme, die in enger Kooperation mit dem Menschen interagieren, müssen ein hohes Maß an semantischer Information beinhalten.
- Eine semantische Objektmodellierung in menschenzentrierten Umgebungen ist nur in Interaktion mit dem menschlichen Benutzer erreichbar.
- Die Mensch-Maschine-Schnittstelle kann derart gestaltet werden, dass interaktive Objektmodellierung auch für ungeübte Benutzer möglich ist.
- Durch bestmögliche automatische Nutzung von sensoruell gewonnener Information kann der menschliche Benutzer weitestgehend vor Routineaufgaben bei der Objektmodellierung bewahrt werden, so dass er nur noch überwachend tätig wird oder um dem System semantische Informationen zur Verfügung zu stellen, die nur mit dem Hintergrundwissen des Benutzers zu erschließen sind.

Als wissenschaftliche Ergebnisse der vorliegenden Arbeit sind insbesondere der Entwurf einer flexiblen semantischen Objektrepräsentation hervorzuheben sowie Entwurf und Implementierung des interaktiven Objektmodellierungssystems IOMOS, mit dessen Hilfe gezeigt werden konnte, dass eine Objektmodellierung in Interaktion mit dem Menschen nicht nur für bestimmte Anwendungen notwendig ist, sondern auch mit angemessenem Aufwand und guten Ergebnissen realisiert werden kann, wenn interaktive und autonome Verarbeitungskomponenten miteinander verknüpft werden.

## 1.4 Gliederung der Arbeit

Das folgende Kapitel befasst sich mit dem Stand der Forschung im Bereich der Objektmodellierung, soweit dieser die vorliegende Arbeit thematisch berührt. Dabei werden insbesondere Arbeiten zur semantischen und funktionalen Objektmodellierung, zum Lernen von Objektmodellen sowie Arbeiten zur interaktiven Objektmodellierung und hierfür geeignete Interaktionsformen vorgestellt und kritisch untersucht.

In Kapitel 3 wird dann die in dieser Arbeit entwickelte semantische Objektrepräsentation erläutert. Hierzu werden zunächst notwendige Grundbegriffe geklärt und die Anforderungen an eine solche Repräsentation definiert. Danach erfolgt die detaillierte Vorstellung der semantischen Objektrepräsentation und ihrer Bestandteile.

Kapitel 4 befasst sich mit dem interaktiven Objektmodellierungssystem IOMOS, das im Rahmen dieser Arbeit entworfen und aufgesetzt wurde. Es werden zunächst die Anforderungen an ein solches System geklärt, bevor dann die Architektur von IOMOS und der Ablauf

der Modellierung vorgestellt und schließlich die Datenhaltung sowie die Implementierung der Objektrepräsentation als grundlegende Bestandteile des im darauf folgenden Kapitel dargestellten interaktiven Modellierungssystems erläutert werden.

Kapitel 5 ist der sensorgestützten interaktiven Objektmodellierung gewidmet. Es werden zunächst einige wesentliche Grundlagen der Mensch-Maschine-Interaktion geklärt und die Anforderungen an die Interaktion zwischen Mensch und Maschine bei der interaktiven Objektmodellierung erarbeitet. Danach werden der Aufbau der Modellierungsumgebung zur interaktiven Objektmodellierung und die in ihr integrierten Sensor- und Aktorsysteme erläutert. Den Schwerpunkt des Kapitels bilden zum einen die Ausführungen zu den Interaktionsmodi in IOMOS und deren Verwendungsmöglichkeiten, zum anderen die detaillierte Darstellung der interaktiven bzw. semi-autonomen Ermittlung unterschiedlichster Objektattribute in diesem System. Kapitel 5 schließt mit einer kurzen Darstellung der Systemintegration.

In Kapitel 6 wird die Domäne *Küche* als Anwendungsbeispiel für die interaktive Objektmodellierung vorgestellt. Für diese Domäne wurde eine Taxonomie von Begrifflichkeiten zur Anwendung auf Servicerobotern erstellt, die in IOMOS in konkrete Objektmodelle umgesetzt werden kann. In diesem Kapitel werden zunächst die wichtigsten Grundlagen des Ontologieentwurfs kurz erläutert und einige Besonderheiten der Domäne *Küche* dargestellt. Danach werden beispielhafte Ausschnitte aus der entwickelten Domänenspezifikation vorgestellt, um einen Einblick in die Anwendung des Systems zu ermöglichen und eine Grundlage für die Evaluation im darauf folgenden Kapitel zu schaffen.

Zur Evaluation der erzielten Ergebnisse wird in Kapitel 7 zunächst die entwickelte semantische Objektrepräsentation auf die Erfüllung der gestellten Anforderungen hin untersucht. Danach erfolgt eine Evaluation des interaktiven Objektmodellierungssystems IOMOS, zum einen indem auch IOMOS anhand der in den vorhergehenden Kapiteln identifizierten Anforderungen bewertet wird, zum anderen anhand der detaillierten Untersuchung zweier Modi zur interaktiven Modellierung von Attributwerten. Es handelt sich hier zum einen um eine innovative Methode, mittels eines Inertial-Joysticks stabile Objektflächen zu modellieren, zum anderen um den Einsatz eines magnetfeldbasierten Positionssensors zur interaktiven Modellierung von Bewegungseinschränkungen bei der Manipulation, die jeweils in einer Untersuchung mit Probanden evaluiert wurden. Nach einer Diskussion der Tauglichkeit unterschiedlicher möglicher Bewertungskriterien erfolgt die Bewertung der Qualität der erstellten Objektmodelle im Hinblick auf die Verwendungsmöglichkeiten und die Nutzung der resultierenden Objektmodelle. Sodann wird eine komplette Modellierungssequenz für ein Objekt noch einmal im Überblick vorgestellt. Das Kapitel schließt mit einer kurzen Bewertung der im Rahmen der Arbeit erzielten Ergebnisse.

Im letzten Kapitel werden dann der Inhalt der Arbeit noch einmal zusammenfassend diskutiert und abschließend mögliche zukünftige Arbeiten zur Verbesserung und Erweiterung des Systems aufgezeigt. Anhang A schließlich zeigt die technischen Daten der verwendeten Geräte im Überblick.

# Kapitel 2

## Stand der Forschung

Die vorliegende Arbeit leistet Beiträge auf unterschiedlichen wissenschaftlichen Gebieten, insbesondere in den Bereichen *Objektmodellierung* und *Interaktion*. Bei der Darstellung des Standes der Forschung wird somit zunächst die Frage nach der Realisierung einer semantischen Objektmodellierung und der Modellierung funktionaler Aspekte von Objekten erörtert. Im darauf folgenden Abschnitt werden einige Ansätze aus dem Bereich des Lernens von Objektmodellen vorgestellt, und schließlich wird der Stand der Technik im Bereich interaktiver Objektmodellierung anhand der wichtigsten Interaktionsmodi vorgestellt. Dabei werden, insbesondere zu Fragen der Interaktion und der unterschiedlichen Interaktionsmodi, auch Arbeiten betrachtet, die sich nicht speziell mit Objektmodellierung befassen, soweit diese von Interesse für die gegebenen Fragestellungen sind.

Weitere Forschungsgebiete, die von der vorliegenden Arbeit berührt werden, insbesondere zur Diskussion des Modellbegriffs, zur Mensch-Maschine-Interaktion, zur automatischen Extraktion von Objektattributen aus Sensordaten und zur Ontologieerstellung, werden im weiteren Verlauf der Arbeit jeweils in den relevanten Kapiteln in dem Umfang diskutiert, wie sie für das Verständnis dieser Arbeit notwendig sind.

### 2.1 Fragestellungen und Diskussionsbedarf

Entsprechend der vorgegebenen Problemstellungen ergeben sich unterschiedliche Fragen bzw. Diskussionspunkte, die auf die zu betrachtenden Forschungsarbeiten angewandt werden müssen. Hierfür wurden die Arbeiten in drei Hauptbereiche eingeteilt, da unterschiedliche Forschungsgebiete berührt werden: in die Diskussion verschiedener Ansätze zur Objektmodellierung, in die Untersuchung und Evaluation von Arbeiten zum Lernen von Objektmodellen sowie in Arbeiten im Bereich der interaktiven Objektmodellierung. Diese drei Forschungsgebiete werden unter jeweils verschiedenen Gesichtspunkten untersucht und die zitierten Arbeiten dabei auf ihre Tauglichkeit für die vorgegebene Aufgabe bewertet:

#### **Welches Wissen über Objekte wird modelliert?**

Je nach geplanter Anwendung unterscheiden sich die in Objektmodellen enthaltenen

Informationen über Objekte deutlich. Allein in der Robotik sind sehr verschiedene Arten von Objektmodellen gängig, von rein geometrischen oder rein ansichtsbasierten Modellen (z. B. zur Navigation oder Objekterkennung) bis hin zu komplexen Objektmodellen, die eine einfache Handlungsplanung oder -erkennung erlauben. Im Folgenden wird ein kurzer Überblick über unterschiedliche Ansätze zur Objektmodellierung gegeben. Ausgehend von den im vorherigen Kapitel diskutierten Anforderungen an die Arbeit liegt der Schwerpunkt hierbei auf den komplexeren Objektmodellen, die auch semantische oder funktionale Informationen beinhalten.

### **Werden Objektmodelle gelernt und wenn ja wie?**

Wird Wissen über Objekte in Form von Objektmodellen dem System a priori vorgegeben oder wird es gelernt? Und falls Objektmodelle gelernt werden, in welcher Form findet dieser Lernprozess statt (autonom oder interaktiv, welche Lernverfahren werden gegebenenfalls verwendet, etc.)?

### **Werden Objektmodelle interaktiv erstellt und wenn ja wie?**

Ist Benutzerinteraktion eine wesentliche Grundlage zur Erstellung der Objektmodelle? Und welche Interaktionsmodi und -verfahren werden in diesem Fall verwendet? Zu unterscheiden hiervon ist einerseits der Versuch, Objektmodelle automatisch zu erstellen, andererseits aber auch die (a priori) Erstellung von Objektmodellen durch einen Experten in einem entsprechenden Expertenmodus bzw. als fest implementiertes Wissen, da die Interaktion in diesen Fällen nicht mit durchschnittlichen Systembenutzern erfolgt und zudem kein Augenmerk auf die Art und Ausgestaltung der Schnittstelle zur Interaktion zwischen Mensch und Modellierungssystem gelegt wird.

Diese Fragestellungen können dabei nicht strikt voneinander getrennt werden, sondern zeigen inhaltlich teilweise deutliche Überschneidungen. Ein Teil der Arbeiten entstammt dabei dem Bereich der Robotik, da als potentielle Anwendung, wie in Kap. 1 beschrieben, das Szenario eines Serviceroboters im Haushalt zur Evaluation und Darstellung der vorliegenden Arbeit dient. Einige Arbeiten entstammen auch Forschungsgebieten außerhalb der Robotik, da sie wertvolle Einblicke in den Stand der Forschung im Bereich Modellierung und Interaktion bieten. Die diskutierten Arbeiten in Abschnitt 2.4 über interaktive Modellierung sind nach den jeweiligen Interaktionsmodi gegliedert. Hier werden auch Forschungsergebnisse vorgestellt und kritisch betrachtet, die sich nicht oder nicht ausschließlich mit Objektmodellierung befassen, soweit sie im Rahmen der vorliegenden Arbeit von Interesse sind. Dieser Abschnitt bietet somit gleichzeitig einen Überblick über die wichtigsten gängigen Interaktionsmodi und Ein-/Ausgabegeräte.

## **2.2 Semantische und funktionale Ansätze zur Objektmodellierung**

Die überwiegende Mehrzahl der in der Robotik angewandten Objektmodelle beschränkt sich auf die Nutzung von wenigen Objektinformationen mit Relevanz vor allem für die Objekterkennung, -klassifikation und -detektion. Diese Modelle beinhalten oft ausschließlich geometrische oder ausschließlich ansichtsbasierte Objektinformationen. Bei beiden Ansätzen müssen

die Objektmodelle a priori modelliert werden und die Modellierung kann nur von einem Experten durchgeführt werden. Zudem sind in der Regel keine weiterführenden Objektinformationen zur Funktionalität oder Semantik der Objekte enthalten, wie sie beispielsweise zur Handlungsplanung in einer komplexen Umgebung benötigt werden. Es ist klar, dass diese eingeschränkten Objektmodelle für komplexe Anforderungen, wie sie der Gegenstand dieser Arbeit sind, nicht ausreichen. Es soll daher hier nur kurz auf diese Arten von Objektmodellen eingegangen werden. Im Folgenden werden dann vorwiegend Arbeiten diskutiert, die Ansätze zu einer komplexeren, funktionalen oder semantischen Objektmodellierung aufweisen.

Eine Klassifikation der gebräuchlichsten Methoden zur bildbasierten Objekterkennung wird in Azad et al. (2006) vorgenommen, wobei die Autoren zwischen globalen Methoden (modellbasierten Ansätzen, globalen ansichtsbasierten Ansätzen sowie histogrammbasierten Ansätzen) und ansichtsbasierten Methoden auf der Basis von lokalen Merkmalen trennen. Die Arbeit liefert einen ausführlichen Überblick über den Stand der Forschung sowie über die Vor- und Nachteile der einzelnen Ansätze. Leibe und Schiele (2003) bietet einen Vergleich zwischen ansichts- und konturbasierten Methoden zur Objektkategorisierung und gleichzeitig ebenfalls einen guten Überblick über den Stand der Forschung. Dort wurde eine Datenbank von 80 Objekten aus acht verschiedenen Kategorien mit insgesamt 3280 Bildern erstellt, um verschiedene Methoden zu vergleichen. Sowohl Azad et al. (2006) als auch Leibe und Schiele (2003) konnten die besten Kategorisierungsergebnisse dabei durch eine Kombination aus verschiedenen Methoden erzielen. Einen detaillierten Überblick über modellbasierte Objekterkennung gibt Pope (1994), der auch interessante Anmerkungen zu grundsätzlichen Entwurfsentscheidungen und Problemen enthält.

### **2.2.1 Ansätze zur Objektmodellierung im Rahmen des Programmierens durch Vormachen**

Das Ziel des so genannten „Programmierens durch Vormachen“ ist es, dass ein Robotersystem Handlungen aus Benutzervorführungen lernt und sie danach selbst ausführen kann. Hierzu werden Objektmodelle benötigt, die über eine rein geometrische oder ansichtsbasierte Modellierung hinausgehen, um die Objekte adäquat mit dem zu lernenden Handlungswissen zu verknüpfen: „Die Integration einer Objekthierarchie und für den Benutzer relevanter Eigenschaften wie etwa Farbe, Größe oder Gewicht werden ebenfalls von Friedrich (1998), Riepp (1997), Friedrich et al. (1998) vorgeschlagen. Durch Modellierung dieser Attribute konnte die Interpretationsfähigkeit und Generalisierungsfähigkeit des in diesen Arbeiten entwickelten Systems entscheidend gesteigert werden.“ (Rogalla (2003), S. 48f).

Friedrich (1998) verwendet zu diesem Zweck Objektmodelle, in denen einerseits die Objektgeometrie modelliert wird (oberflächenbasiert oder durch parametrisierte Volumenprimitive) und andererseits ihre technischen und physikalischen „Eigenschaften“, in diesem Fall z. B. Gewicht, Dichte, Oberflächenrauheit etc. Die Modellierung erfolgt a priori durch einen Experten mit Hilfe eines CAD-Systems, die Objektgeometrie wird mittels des Visualisierungssystems KAVIS dargestellt (Schaude (1997)). Im dort vorgeschlagenen System werden alle ein Objekt bzw. eine Objektklasse charakterisierenden Eigenschaften als eine Liste von Attribut-Wert-Paaren modelliert. Die Vorteile dieses Ansatzes sind laut Friedrich (1998), dass die Anzahl betrachteter Objekteigenschaften je Objekt/Objektklasse variabel ist, sowie

dass die betrachteten Eigenschaftstypen an eine bestimmte Aufgabenstellung angepasst werden können. Da es sich im Rahmen dieser Arbeit stets um Manipulationsaufgaben handelt, müssen jeweils nur diejenigen modelliert werden, die für die vorliegende Manipulationsaufgabe von Bedeutung sind.

Folgende Definitionen werden bei Riepp (1997) sowie analog hierzu bei Friedrich (1998) getroffen:

„Definition 4.4: **Eigenschaftstyp**: Ein Eigenschaftstyp ist die abstrakte Beschreibung einer Eigenschaft. Besitzt ein Objekt eine bestimmte Eigenschaft, so besitzt es immer genau eine Ausprägung des entsprechenden Eigenschaftstyps.

Beispiel 4.1: *Beispiele für Eigenschaftstypen für Objekte der realen Welt sind etwa: Inhalt, Legierung, Ersatzteilnummer, Farbe, Typ, Objektinstanz, usw. Beispiele für Ausprägungen z. B. des Typs Inhalt sind: NaCl, HCl<sup>+</sup>, H<sub>2</sub>O, etc.*

Um die Ausprägung von Eigenschaftstypen für Objekte zu bestimmen, werden Eigenschaftsfunktionen verwendet.

Definition 4.5: **Eigenschaftsfunktion**: Eine Eigenschaftsfunktion  $E_{ET}(O)$  definiert für das Objekt  $O$  die genaue Ausprägung des Eigenschaftstyps  $ET$ . Der Wert einer Eigenschaftsfunktion bestimmt sich anhand folgender Funktion:

$$E_{ET}(O) = \begin{cases} x & \text{falls } O \text{ hat Eigenschaft } ET \text{ mit Wert } x \\ \text{nil} & \text{sonst} \end{cases}$$

Eigenschaftsfunktionen können auf verschiedenste Weisen den Wert zu einem Eigenschaftstyp und einem Objekt bestimmen. Die Bestimmung kann auf der Basis des Umweltmodells, vor anderem Hintergrundwissen, wie z. B. Produktionsplänen oder direkt aus Sensordaten erfolgen.

Beispiel 4.2: Beispiele für Eigenschaftsfunktionen und deren Werte sind

$$E_{Typ}(Puma260) = \text{„Manipulator“}$$

$$E_{Inhalt}(Beutel) = \text{„NaCl“}$$

$$E_{Inhalt}(Balken) = \text{„nil“}$$

$$E_{Farbe}(Tomate) = \text{„rot“}$$

$$E_{Ersatzteilnummer}(Schraube) = \text{„k-043285“} \text{.} \text{ (Friedrich (1998), S. 53).}$$

Es fällt auf, dass unter diese Definition von Eigenschaften im wesentlichen alle Objektinformationen fallen können. Dies bedeutet aber, dass die Objektmodelle hier immer noch stark auf eine geometrische Objektmodellierung bezogen sind, da diese besonders herausgehoben und behandelt wird, obwohl sie im Sinne der oben zitierten Definition durchaus dem Eigenschaftskonzept entsprechen würde. Die möglichen Arten von Eigenschaftstypen werden nicht weiter diskutiert. Insgesamt enthalten diese Arbeiten zwar erste Ansätze zu einer erweiterten Objektmodellierung, dies aber nur insoweit, wie es für die unmittelbar anstehende

Manipulationsaufgaben notwendig ist. Eine strukturierte Modellierung der Objektsemantik und -funktionalitäten ist hier noch nicht gegeben.

Auf die Ergebnisse von Friedrich baut die Arbeit von Rogalla (2003) auf, deren Ziel es ist, eine Abbildung von Benutzerdemonstrationen auf variable Roboterkonfigurationen im Rahmen des Programmierens durch Vormachen zu entwickeln. Hier enthält das Weltmodell die Objektmodelle und die räumlichen Beziehungen zwischen den Objekten, wobei ebenfalls zwischen einem geometrischen Objektmodell, das die Lage der Objekte in der Welt und ihre Geometrien enthält, und einem Eigenschaftsmodell getrennt wird. Der Begriff „Eigenschaft“ wird hier ebenfalls im Sinne technischer und physikalischer Eigenschaften gebraucht. Das Eigenschaftsmodell strukturiert bei Rogalla Modelle in eine Taxonomie mit fest definierten, in Form von Attributen repräsentierten Objekteigenschaften, beispielsweise ob ein Objekt beweglich ist oder zu welcher Oberklasse es gehört. Dies bedeutet, dass die so genannten Eigenschaften vor dem Modellierungsprozess festgelegt werden, die Menge aller Eigenschaften also nicht während der Anwendung flexibel erweiterbar ist. Wie oben bei Friedrich bereits angemerkt, umfasst der Begriff „Eigenschaft“ auch hier alle Objektinformationen, die irgendwie für die gegebene Anwendung relevant erscheinen, bietet aber mit Ausnahme einer im voraus festgelegten Objekttaxonomie keinerlei semantische Information.

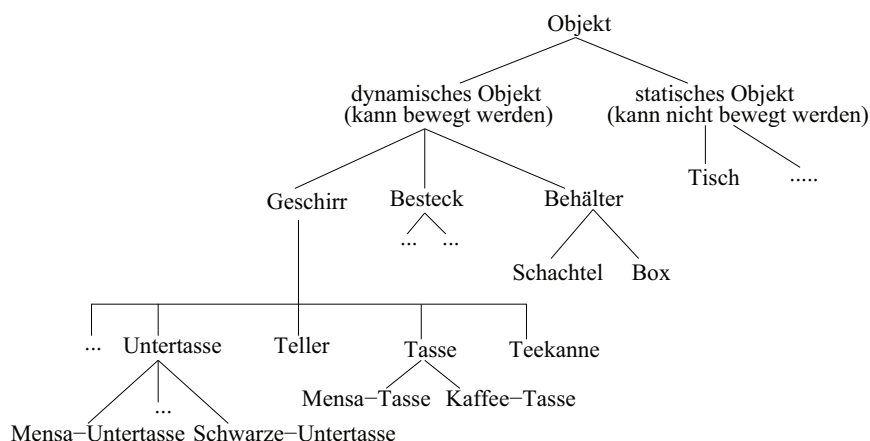


Abbildung 2.1: Beispielhafte Objekttaxonomie für Manipulationsaufgaben (Rogalla (2003), S. 78)

Abb. 2.1 stellt ein Beispiel für eine solche Objekttaxonomie dar. Gut zu sehen ist, dass eine solche starre Taxonomie im Voraus angelegt werden muss und bereits kleinere Änderungen oder Ergänzungen unter Umständen größere Veränderungen an anderen Stellen nach sich ziehen. Die beispielhaft angegebene Taxonomie ist speziell auf Manipulationsanwendungen zugeschnitten, ein wesentliches Unterscheidungsmerkmal ist der Unterschied zwischen statischen und beweglichen Objekten. Für andere Anwendungen muss somit ein völlig neues Eigenschaftsmodell erstellt werden. Zudem werden die Eigenschaften in diesem Ansatz nicht konsistent repräsentiert – teilweise sind sie bereits in der Objekttaxonomie enthalten („statisch“ vs. „dynamisch“), teilweise werden sie in der Klassendefinition der Objekte modelliert: „Abbildung 5.4 [in der vorliegenden Arbeit wiedergegeben als Abb. 2.1] zeigt beispielhaft eine Objekttaxonomie für Haushaltsgegenstände; die zusätzlichen Eigenschaften wie beispielsweise Größe und Farbe werden in der Klassendefinition der Objekte beschrieben.“ (Rogalla (2003), S. 77).

Die Arbeiten von Ehrenmann (z. B. Ehrenmann (2003)) bauen auf dem gleichen Objektmodell auf wie die vorherigen Arbeiten. Im Rahmen der vorliegenden Arbeit sind zwei bei Ehrenmann (2003) erwähnte Attributtypen besonders interessant, da sie eine explizite Verbindung zwischen der Robotersensorik und -aktorik und den Objektmodellen herstellen: Dies sind zum einen Attribute für bildbasierte Verfahren zur Objekterkennung. Hier werden spezielle Modelle benötigt (z. B. Objektansichten, Konturmerkmale), die zusammen mit der spezifischen Eignung einer Methode zur Detektion eines Objekts in einem Attributtyp gespeichert werden. Zum anderen handelt es sich um a priori-Wissen über für ein Objekt sinnvolle Grifftypen. Ein Beispiel für solche Objektattribute ist in Abb. 2.2 wiedergegeben. Ansonsten gelten auch hier die Anmerkungen, wie sie oben bereits in Bezug auf die Arbeiten von Friedrich und Rogalla formuliert wurden.

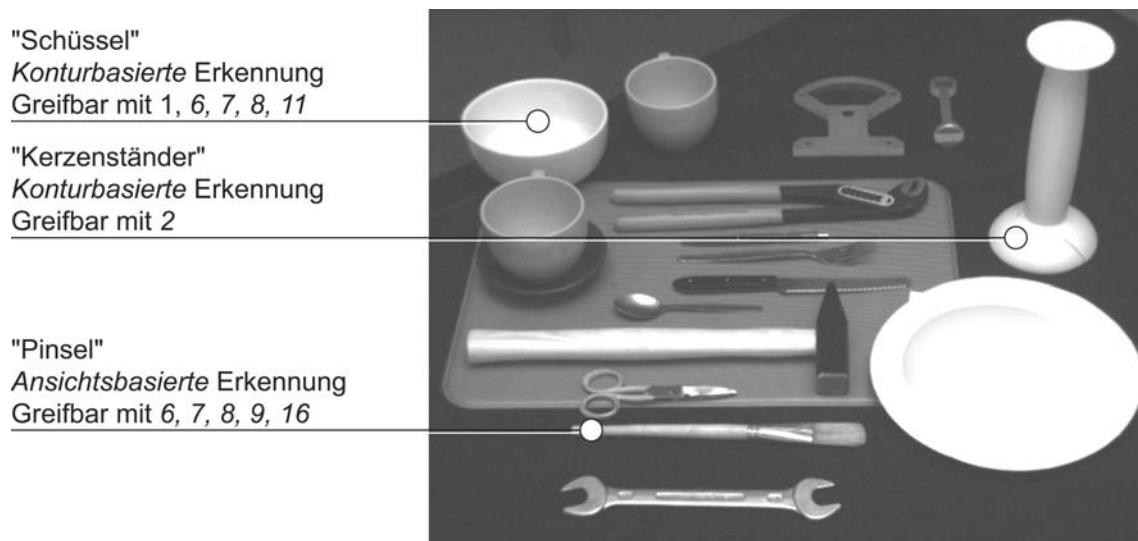


Abbildung 2.2: Beispielhafte Darstellung von Objekten und deren möglichen Attributen (Ehrenmann (2003), S. 59)

Zöllner (2005) erweitert die bereits vorgestellten Objektmodelle um Objektfunktionalitäten. Ein Objektmodell besteht dann aus drei Teilen:

1. einem Geometriemodell, bestehend aus Objektgeometrie und Lage des Objektes in der Welt. Insbesondere sind enthalten Oberflächendefinitionen, z. B. Dreiecksnetze, und parametrierbare Volumenprimitive, die Verknüpfung von beweglichen Elementen durch parametrisierte Verbindungsstrukturen sowie Information über die Objektlage in Form einer homogenen Transformation vom Objekt- zum Weltkoordinatensystem. Auch in Zöllner (2005) werden diese Modelle a priori von einem Experten in einem CAD-System modelliert und dann im Visualisierungssystem KAVIS dargestellt.
2. einem Eigenschaftsmodell, das physikalische Eigenschaften enthält in Form von Attribut-Wert-Paaren. Diese repräsentieren physikalische Eigenschaften wie Gewicht, Dichte oder stabile Ablageflächen. Die Anzahl der Eigenschaften ist theoretisch unbegrenzt, ihre Art beliebig definierbar. Das Eigenschaftsmodell wird wie auch in den Vorgängerarbeiten zum Aufbau einer Objekttaxonomie genutzt, wobei auch hier die grundlegende Unterteilung in bewegliche vs. unbewegliche Objekte vorgenommen wird, was durch die Fokussierung auf Manipulationsaufgaben bedingt ist.



3. einem in dieser Arbeit neu eingeführten Funktionsmodell, das eine Beschreibung funktionaler Aspekte des Objektes enthält. Das Funktionsmodell ist ein Spezialfall des Eigenschaftsmodells und beschreibt Funktionen, die ein Objekt während einer Handhabung einnehmen kann in Form von so genannten „Rollen“. Beispiele für solche Rollen sind „transportiert“, „befüllt“, „geleert“.

Die gebildeten Objektklassen und deren Hierarchie werden zur Generalisierung von beobachteten Handhabungen genutzt. Das Funktionsmodell erlaubt die Zuweisung von Rollen zu Objekten. Die Menge der potentiellen Rollen eines Objekts während einer Handhabung wird dann zur Bestimmung von Teilzielen in der Benutzerdemonstration verwendet. Interessant sind insbesondere die Interaktionen zwischen zwei Objekten anhand derer Rollen, beispielsweise bei der Benutzung von Werkzeugen. In dieser Arbeit wird zwar versucht, auch funktionale Aspekte von Objekten zu modellieren, aber das Funktionsmodell stellt nur eine Erweiterung des Eigenschaftsmodells dar und bringt dadurch auch alle Probleme des Eigenschaftsmodells mit sich.

Insgesamt existieren im Rahmen des Programmierens durch Vormachen zwar erste Ansätze zu einer Objektmodellierung mit funktionalen Elementen, das Hauptaugenmerk dieser Arbeiten lag aber bisher auf der Modellierung von Handlungen und die Objektmodelle genügen nicht den für die vorliegende Arbeit spezifizierten Anforderungen. Die Interaktion mit dem Benutzer erfolgt in diesen Arbeiten ausschließlich zum Erwerb von Handlungswissen, das Objektwissen über die beteiligten Objekte muss hierzu bereits vorliegen. Allerdings könnten erweiterte semantische Objektmodelle mit einer konsistenten Modellierung von Objektsemantik und -funktionalität sowie der Möglichkeit, Objektmodelle auch zur Laufzeit noch zu ergänzen und anzupassen, auch für das Paradigma des Programmierens durch Vormachen einen großen Gewinn bedeuten.

### 2.2.2 Funktionale Objektmodellierung

Die Funktionalität von Objekten bezeichnet ihre potentielle Verwendung bezogen auf eine bestimmte Anwendung oder Aufgabe, ist also immer kontext- oder domänenspezifisch zu sehen: „Functionality of an object can be identified with its purpose and utility in a specific environment. Its purpose depends on the intention of an agent and the utility denotes its applicability in a particular task. Although functionality can be defined abstractly, to be identified in a specific object it needs to be explored in the context of an environment.“ (Bogoni und Bajcsy (1994), S. 2). Chandrasekaran (1994) gibt einen guten Überblick über grundlegende Diskussionen zur Verknüpfung von Objekten mit Funktionen. Insbesondere vertritt er die These, dass eine strikte Trennung zwischen intrinsischen und extrinsischen Funktionen von Objekten nicht getroffen werden kann. Funktionen in diesem Sinne können also nicht vollständig neutral und objektiv sein, sondern hängen vom Standpunkt des Benutzers ab und davon, wie er ein Objekt verwendet. Im Folgenden werden Arbeiten vorgestellt und diskutiert, die sich überwiegend oder ausschließlich mit der Modellierung von Objektfunktionalitäten beschäftigen. Dabei wird unterschieden zwischen Arbeiten, die die Funktionalität von Objekten aus deren Form, und solchen, die sie aus damit möglichen Bewegungen bzw. Handlungen ableiten.

## Funktionalität aus Form

Einen rein funktionalen, global orientierten Ansatz zur Modellierung von Funktionalität aus Form verfolgen Stark und Bowyer (1991). Ihre Objektmodelle enthalten, obwohl sie für die Objekterkennung konzipiert wurden, keinerlei geometrische Modelldaten. Die Grundidee dieser Arbeit ist, dass Funktion sich in der Form äußert und auch durch diese erschlossen werden kann. Der Vorteil eines solchen Ansatzes ist, dass bei Vorliegen entsprechend guter Ergebnisse auch bisher unbekannte Objekte kategorisiert werden können, sofern sie in eine bekannte Kategorie fallen. Als funktionale Elemente von Objekten werden in dieser Arbeit nur Objektoberflächen betrachtet wie z. B. die Sitzfläche eines Stuhls. Die Beschreibungen von Objektkategorien sind hierarchisch angelegt. Zu Evaluationszwecken wurde die Kategorie „Stuhl“ untersucht. Die Objektmodelle, die die Ausgangsbasis für die Objekterkennung bilden, sind dreidimensionale Polyeder, die mittels einer CAD-Software angelegt wurden. Insgesamt ist das System für reale Anwendungen wenig geeignet, selbst die Evaluation auf einer sehr kleinen Objektmenge fand nicht unter realen Bedingungen statt. Es ist zweifelhaft, ob das System bei Verwendung echter Sensordaten mit den zu erwartenden Messfehlern oder mit Verdeckungen umgehen kann. Auch ein Lernen von Objektwissen oder eine interaktive Erstellung durch einen ungeübten Benutzer sind hier nicht möglich.

Rivlin et al. (1995) kombinieren eine Menge von funktionalen Primitiven und deren Beziehungen mit einer Menge von abstrakten volumetrischen Formprimitiven und deren Beziehungen zum Zweck der Objekterkennung. Die Ergebnisse dieser Arbeit werden demonstriert an einer eingeschränkten Klasse von Handwerkzeugen und deren Erkennung aus 2D-Bildern. Die Autoren grenzen sich gegen globale Ansätze zur funktionalen Modellierung (wie beispielsweise bei Stark und Bowyer (1991)) ab und schlagen stattdessen einen lokalen Ansatz vor, um so das Problem von Verdeckungen in realen Szenen zu umgehen. Betrachtet wird hier deshalb die Funktionalität von Objektteilen und deren Interaktionen, wie relative Orientierung, Größe, Form oder Bewegung. Ziel ist die Entwicklung einer Objektrepräsentation, die Form und Funktion integriert, und gleichzeitig das Problem zu lösen, wie Form und Funktion aus 2D- und 3D-Bilddaten gefolgert werden können. Die Autoren schlagen daher die in Abb. 2.3 dargestellte Objektrepräsentation vor.

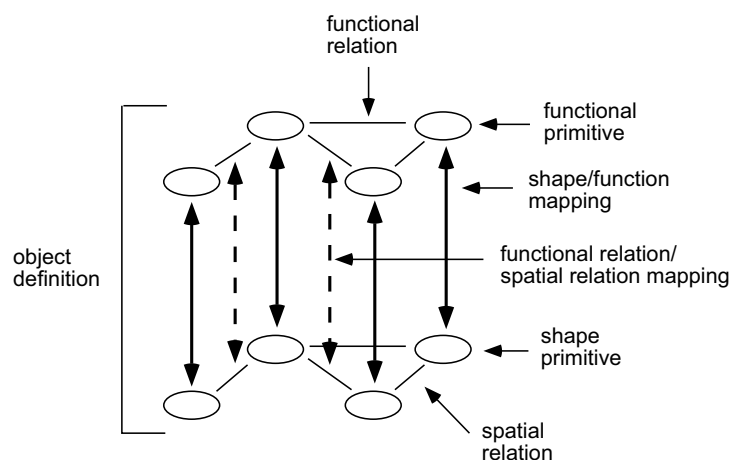


Abbildung 2.3: Repräsentation von Objektfunktionalitäten bei Rivlin et al. (Rivlin et al. (1995), S. 5)

Ein solches funktionales Objektmodell besteht dann aus zwei Ebenen, der Form-Ebene und der Funktions-Ebene. Auf der Form-Ebene existieren Form-Primitive, mit deren Hilfe Objekte als Konstruktionen aus volumetrischen Grundformen beschrieben werden. Auf der Funktions-Ebene existieren hierzu korrespondierende funktionale Primitive, wobei die Autoren für die von ihnen gewählte Domäne „Manipulationsaufgaben“ nur Endeffektoren und Handhabungseinheiten als funktionale Teile von Objekten verwenden. Weder die funktionalen Relationen noch die von den Autoren propagierte Verbindung zwischen Form und Funktion ist formalisiert und klar dargestellt. „Der Einfachheit halber“ beschränken sie sich auf Objektmodelle mit einer eindeutigen Abbildung zwischen Form und Funktion, geben aber selbst an, dass diese Abbildung in vielen Fällen nicht eindeutig sein wird. Für praktische Anwendungen wie einen Serviceroboter ist der hier gewählte Ansatz aus mehreren Gründen nicht anwendbar: zum einen ist die Repräsentation teilweise noch nicht formal ausgearbeitet, zum anderen sind die funktionalen Primitive von Hand so gewählt, dass sie zu den Formprimitiven passen, was in einer realen Anwendung nicht immer möglich sein wird. Auch die Abbildung zwischen funktionalen und Form-Relationen muss jeweils mit Expertenwissen a priori angelegt werden. Eine automatische Modellierung oder interaktive Modellierung durch Laien ist nicht sinnvoll möglich. Die experimentelle Evaluation wird nur auf einer sehr kleinen Objektmenge in einer eingeschränkten Domäne durchgeführt. Für komplexe Alltagssituationen in realen Umgebungen eignet sich das System nicht, da dort zusätzlich zu den bereits erwähnten Schwächen des Ansatzes noch zusätzliche Probleme z. B. aus Sensorungenauigkeiten zu erwarten sind.

### **Objektklassifikation und -erkennung aus menschlichen Bewegungen**

Die Arbeit von Moore et al. (1999) hat als Ziel die Ausnutzung von Wissen über menschliche Bewegungen und Objektkontext zur Handlungserkennung und Objektklassifikation. Die Objektklasse wird hier bestimmt durch extrahierte einfache Objektmerkmale (sog. „low-level Objektfeatures“) und durch mittels eines naiven Bayes-Klassifikators detektierte Handlungen. Es soll gezeigt werden, dass die Verbindung zwischen menschlichen Handlungen und Objekten genutzt werden kann, um Objekte zu detektieren und zu klassifizieren. Die Objektmodelle enthalten unter anderem Verweise auf Handlungen, die als Hidden Markov Modelle repräsentiert werden. Zum Beispiel enthält das Modell „Buch“ Verweise auf die Handlungen „vorwärts blättern“ und „rückwärts blättern“ bzw. die sie repräsentierenden Hidden Markov Modelle. Bei Vererbung zwischen Objektklassen erben die Kinder die Eigenschaften ihrer Eltern, es können aber auch neue Eigenschaften und Methoden hinzukommen. Zur Evaluation wurden Objektmodelle erstellt für Büro-, Küchen- und Automobilumgebungen. Der Küchenumgebung zugeordnet sind insgesamt 12 Objekte, davon 4 ohne assoziierte Handlungen. Bei den evaluierten Beispielen handelt es sich um wenige, stark eingeschränkte Modelle, die von Hand von Experten modelliert wurden. Die Objekt- und Handlungserkennung sind relativ einfach gehalten, da nur das Prinzip dieses Ansatzes demonstriert werden soll. Die Autoren selbst räumen Probleme mit der Skalierbarkeit ihres Ansatzes ein, vor allem wenn jede Handlung mit jedem Objekt verknüpft wird. Sie hoffen allerdings, dies durch wohldefinierte und beschränkte Domänen zu vermeiden – in seiner Komplexität entspricht dies aber dann nicht mehr annähernd den Anforderungen in dynamischen realen Umgebungen.

Zu den Arbeiten, die überwiegend, aber nicht ausschließlich funktionale Objektmodelle einsetzen, gehört Kitahashi et al. (2005). Das Ziel dieser Arbeit ist die Erkennung von menschlichen Bewegungen und damit verbundenen Objekten. Besonderes Augenmerk dieser Arbeit liegt auf der Interaktion zwischen Mensch und Objekt bzw. zwischen Objekten, die laut Ansicht der Autoren in den meisten Arbeiten zur Szenenerfassung vernachlässigt werden. Hierzu werden so genannte „funktionale Attribute“ wie beispielsweise „essbar“ oder „tragbar“ eingeführt, die ein Objektmodell zusätzlich oder anstelle von konventionellen ansichtsbasierten Attributen enthält. Die Objekt- und Handlungshierarchie sowie die Objektmodelle sind von Hand angelegt und fest implementiert und daher für reale Anwendungen in komplexen dynamischen Szenen ungeeignet. Zudem haben die so genannten „funktionalen Attribute“ in diesem Ansatz eine größere Bedeutung für die Objektmodelle als konventionelle Attribute. Durch die speziellen und stark verzahnten funktionalen Attribute wird eine Skalierung der Arbeit auf größere Domänen aufwendig.

Higuchi et al. (2004) gehen sogar davon aus, dass geometrische Modelle zumindest zur Objekterkennung überflüssig sind: „The key notion of this research is that objects can be recognized without any individual shape model.“ (Higuchi et al. (2004), S. 5). Wie in Kitahashi et al. (2005)) wird auch hier versucht, menschliche Bewegungen zur Objekterkennung zu nutzen. Allerdings ist dieser Ansatz nicht auf alle Objekte anwendbar, und immer nur wenn ein Benutzer ein Objekt tatsächlich für eine Handlung nutzt, da die Objekte sonst statisch und mit diesem Vorgehen nicht aus der Szene separierbar sind. Außerdem ist dieser Ansatz aufwendig und fehleranfällig (Schlussfolgerungen können falsch sein, ein Benutzer kann etwas ungewöhnliches tun etc.). Hinzu kommt, dass Objekte nur in der Szene verfolgt werden können, wenn der Benutzer einmal etwas mit ihnen getan hat und sie danach die ganze Zeit im Blickfeld der Kameras sind: Falls z. B. der Roboter hinter dem Rücken des Benutzers eine Handlung durchführt und dabei ein Objekt verschiebt, kann das nachträglich nicht mehr erfasst werden. Falls das Objekt danach erneut vom Benutzer verwendet wird, wird es vermutlich als ein anderes, weiteres Objekt erkannt und das ursprüngliche Objekt immer noch an der alten Position vermutet. Die Modellerstellung wird bei einer größeren Menge an potentiellen Objekten und Handlungen bzw. bei komplexeren Domänen aufwendig und möglicherweise inkonsistent.

Auch in Bogoni und Bajcsy (1994) und Bogoni und Bajcsy (1995) geht es darum, Gegenstände zu erkennen und zu klassifizieren, die eine bestimmte funktionale Rolle erfüllen. Das Fernziel dieser Forschungsarbeiten ist ein System mit Untersuchungsprozeduren, das Objekte basierend auf ihren funktionalen Attributen klassifizieren und erkennen kann. Beispielhaft werden hier Werkzeuge genannt. Verwendet wird hier eine Repräsentation basierend auf der diskreten Ereignissystemtheorie (sowohl für Untersuchungsaufträge als auch für die Funktionalitäten eines Werkzeugs) und dem Paradigma der Aktiven Perzeption. Bei diesem Paradigma geht es darum, dass ein (Roboter-)System selbständig die Welt untersucht (vgl. Erläuterungen in Abschnitt 2.3), in diesem Fall um Objekte zu klassifizieren bzw. zu erkennen. Hierzu werden Positionssensoren, Kraftsensoren und bildverarbeitende Sensoren verwendet. Exemplarisch untersucht wird der Ansatz anhand des „Einstechens“ (engl.: piercing) in Objekte. Die Ergebnisse sind sehr spezifisch für die verwendeten Sensoren und die durchgeführte Handlung. Eine Übertragung auf andere Anwendungen ist daher nicht einfach. Die Repräsentation wird von Hand erstellt und ist daher nur mit großem Aufwand auf größere Domänen übertragbar, ein Lernen der Modelle ist nicht vorgesehen.

In Bogoni und Bajcsy (1995) finden sich zusätzlich interessante grundsätzliche Überlegungen zur Funktionalität von Objekten und zu deren Darstellung in Objektmodellen. Der Grundgedanke dieser Arbeit lässt sich wie folgt zusammenfassen:

„Functionality almost universally involves an interaction. Hence, the representation for an object must include not only its intrinsic properties (e.g. geometric, material, kinematic and dynamic) but also how it is used. Many objects may be recognized as suitable implements for satisfying a particular role. The analysis of their properties allows us to identify what can be labeled as the „functional-features“ – features which make the object suitable for a specific task. Hence, investigating the functionality of an object addresses the object’s purpose in a particular setting.“ (Bogoni und Bajcsy (1995), S. 1).

Im Rahmen dieser Arbeit werden mögliche Ausprägungen und die Repräsentation von Funktionalität untersucht. Als mögliche Arten von Objektfunktionalität werden intendierte, von außen aufgezwungene und intrinsische Funktionalität genannt, wobei eine Klassifizierung immer nur in Bezug auf einen Anwendungskontext möglich ist. Kritisch lässt sich hier anmerken, dass eine klare Trennung dieser verschiedenen Arten von Funktionalität in realen Alltagsumgebungen nicht immer möglich sein wird und für praktische Anwendungen auch nicht unbedingt notwendig ist. Eine weitere wichtige Unterscheidung ist, ob die Eigenschaften und die Beziehung der Objekte, die an einer funktionalen Interaktion beteiligt sind, konstant bleiben oder sich verändern. So hängen z. B. manche Funktionalitäten von räumlichen Beziehungen zwischen Objekten ab. Abb. 2.4 stellt die hier vorgeschlagene Objektrepräsentation im Überblick dar, besonders im Hinblick auf die mit einem Objekt assoziierten Funktionalitäten. Die Funktionalität eines Objekts wird nach Bogoni und Bajcsy (1995) wie folgt repräsentiert:

„FUNCTIONALITY(*Object*) = STRUCTURE + CONTEXT + APPLICATION

where:

- the STRUCTURE is associated with the intrinsic properties of an object.
- The CONTEXT identifies an *agent*, possibly the *recipient* of an action, and the *environment* in which the functionality of the object is expressed.
- The APPLICATION denotes *how* the object is to fulfill the desired function.“ (Bogoni und Bajcsy (1995), S. 9f).

## Fazit

Insgesamt lässt sich festhalten, dass zwar einige Ansätze zur Modellierung von Objektfunktionalitäten sowie deren Verknüpfung mit menschlichen Bewegungen existieren, dass die meisten sich jedoch nicht oder nur mit großer Mühe auf reale Umgebungen übertragen lassen, die eine gewisse Komplexität und Dynamik sowie eine entsprechende Mengen an Objekten mit sich bringen. Das Lernen von Objektmodellen wird in diesen Ansätzen meist

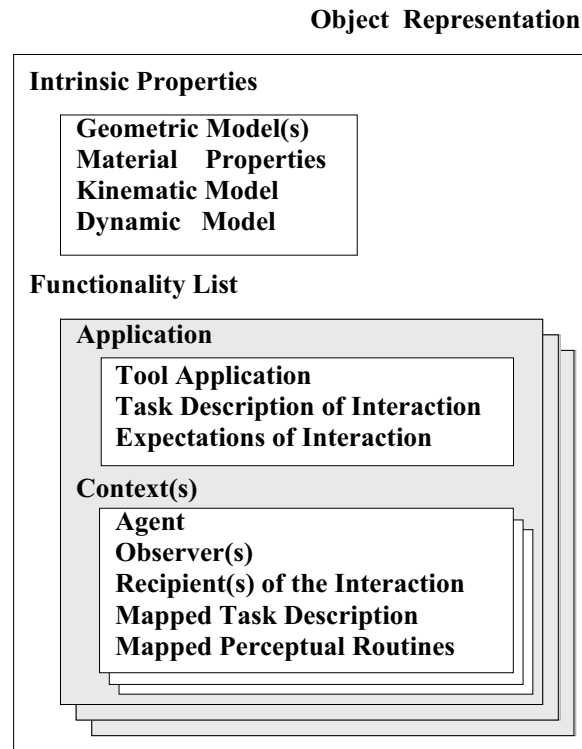


Abbildung 2.4: Repräsentation eines Objektes und seiner Funktionalitäten bei Bogoni und Bajcsy (Bogoni und Bajcsy (1995), S. 30)

nicht betrachtet. Stattdessen werden die Modelle a priori von einem Experten erstellt, und oft sind die Zusammenhänge zwischen den einzelnen Objektmodellen so kritisch, dass auch nur ein entsprechend vorgebildeter Experte überhaupt in der Lage sein wird, konsistente und bedeutungsvolle Objektmodelle für eine Anwendung zu erstellen. Die rein funktional gehaltenen Ansätze bringen zudem das Problem mit sich, dass sie sehr speziell auf bestimmte Anwendungen ausgerichtet sind. Andere Aufgaben wie die Planung von Greifsequenzen oder Armtrajektorien werden sich mit rein funktionalen Objektmodellen ohne zusätzliches Wissen nicht durchführen lassen. Diese Ansätze sind daher für Anwendungen wie einen Serviceroboter als alleiniges Objektmodell schon aus diesem Grund nicht geeignet. Zusätzlich werden bei den Arbeiten zur Modellierung von Funktion rein aus Form andere Objektfunktionen, wie beispielsweise Funktionen aus Bewegungen von Objektteilen (wie das Öffnen einer Schranktür), nicht berücksichtigt, so dass sie im Bereich Servicerobotik nur für wenige Anwendungen tauglich sein werden. Die Arbeiten zur Modellierung von Funktion aus Bewegungen bringen dafür praktische Probleme beim Erkennen und Klassifizieren von Bewegungen sowie bei der Zuordnung von Objekten zu erkannten Bewegungen mit sich.

### 2.2.3 Objekt-Handlungs-Komplexe

Im Rahmen des Projektes PACO-PLUS, dessen Ziel die Entwicklung eines kognitiven Robotersystems ist, wird ein interessanter Ansatz zur Verknüpfung von Objekten mit Handlungen verfolgt (PACO-PLUS (2006)). Zentral für dieses Projekt ist die Annahme, dass Objekte und

Handlungen im Rahmen so genannter Objekt-Handlungs-Komplexe (engl. „Object-Action-Complexes“, OAC) untrennbar miteinander verbunden sind. In diesem Konzept werden Kategorien dadurch bestimmt, welche Handlungen ein Agent damit ausführen kann und welche Attribute der Welt wahrgenommen werden können. Handlungen definieren somit die Bedeutung eines Objektes, und umgekehrt suggerieren Objekte Handlungen. Die menschlichen Verarbeitungsmechanismen, die diesem Paradigma zugrunde liegen, werden in Hommel (2004) und vor allem Hommel et al. (2001) ausführlich vorgestellt und diskutiert. Insbesondere argumentieren die Autoren dafür, dass sowohl wahrgenommene Objekte als auch Handlungspläne durch die gleichen kognitiven Codes repräsentiert werden.

Formal definiert werden die vorgeschlagenen OACs bei Geib et al. (2006). Die Autoren argumentieren damit, dass die üblicherweise verwendeten Repräsentationen in der Robotersteuerung und -regelung einerseits und der diskreten symbolischen Künstlichen Intelligenz andererseits sich deutlich unterscheiden, und dass es nicht einfach ist, diese Ebenen zusammenzubringen. Eine solche Zusammenführung erscheint allerdings notwendig, wenn eine wirklich menschenähnliche Verhaltenssteuerung erreicht werden soll. Sie schlagen daher die Benutzung von Objekt-Handlungs-Komplexen als Zwischenebene vor, die einerseits eine klare Semantik bieten, andererseits aber auch auf beiden Ebenen einfach manipulierbar sein muss. Für eine formale Beschreibung der OACs definieren die Autoren zunächst instanziierte Zustandsübergangsfragmente (engl. „instantiated state transition fragment“, ISTF). Es handelt sich hierbei um eine situierte Paarung eines Objekts mit einer Handlung, die ein kleines, aber vollständig instanziiertes Fragment der Zustandsübergangsfunktion der Planungsdomäne abdeckt. Formal sind ISTFs definiert als ein Tupel  $\langle s_i, mp_j, Obj_{mp_i}, s_{i+1} \rangle$ , wobei  $s_i$  der initial erfasste Zustand der Welt ist,  $mp_j$  die Instanz eines motorischen Programms,  $Obj_{mp_i}$  das gesamte Objekt, das die Komponente enthält, relativ zu der das motorische Programm definiert war, und  $s_{i+1}$  der Zustand, der aus der Ausführung des motorischen Programms resultiert.

Diese Zustandsrepräsentationen enthalten alle Informationen, die ein Robotersystem über die entsprechenden zwei Zustände der Welt besitzt, wobei manche Informationen für das Ergebnis der Handlung irrelevant sein können. Ein Lernmodul muss dann diese irrelevante Information aus den ISFTs entfernen, um somit davon abstrahierte OACs zu generieren. Die OACs enthalten dann schließlich nur die relevante instanziierte Information, die benötigt wird, um die Anwendbarkeit der Handlung und ihre wahrscheinlichen Auswirkungen effektiv vorherzusagen. Für diesen Abstraktionsschritt werden mehrere reproduzierbare ISFTs benötigt, d. h. das Robotersystem muss wiederholt mit der Welt interagieren, um eine genügende Menge hinreichend ähnlicher ISFTs zu erhalten, die dann in OACs überführt werden können. In diesem Sinne entspricht dieses Vorgehen dem Lernprozess im Paradigma des Programmierens durch Vormachen. Ein OAC ist dann definiert als generalisiertes ISFT-Tupel:  $\langle S_i, MP_j, Obj_k, S_{i+1} \rangle$ , mit zwei abstrakten Zuständen  $S_i$  und  $S_{i+1}$ , einer Menge von motorischen Programmen  $MP_j$  sowie einer Objektklasse  $Obj_k$ . Der Initialzustand der Welt  $S_i$  ist so abstrahiert, dass er nur die Eigenschaften enthält, die notwendig sind zur Ausführung eines der motorischen Programme der Menge  $MP_j$  auf einem Objekt der Klasse  $Obj_k$ , die in einem der abstrakten Zustandsdefinition  $S_{i+1}$  genügenden Zustand endet.

Die OACs enthalten somit alle Informationen, die auch in den Repräsentationen der KI-Planungskomponente enthalten sind. Die Unterschiede zwischen diesen beiden Ebenen sind zum einen, dass die OACs Informationen über Vorbedingungen, Nachbedingungen und Objekte

enthalten, die normalerweise an verschiedenen Stellen der Repräsentation gespeichert werden. Zum zweiten macht die Objektzentriertheit der OACs sie zu einer idealen Grundlage für einen Planungsalgorithmus, und dritten beschränken die OACs das lernbare Wissen auf eine einfache Form, so sind z. B. keine Handlungsrepräsentationen mit konditionalen Effekten möglich. Die Autoren schreiben selbst, dass bisher fast alle ihrer Behauptungen noch unbewiesen sind und in diesem Sinne eher ein Forschungsprogramm darstellen.

Der Gedanke, dass Handlungen helfen, die Natur eines Objektes näher zu bestimmen, ist nicht neu (vgl. Abschnitt 2.2.2). Auch im Projekt PACO-PLUS wird von einer solchen handlungszentrierten Objektrepräsentation ausgegangen. Die Information zur Repräsentation der Objektklassen ist nicht detailliert genug, um im Rahmen der hier gegebenen Aufgabenstellung erfolgreich eingesetzt zu werden. Auch die Frage, inwieweit die Objektmodelle selbst (also nicht die OACs) interaktiv gelernt werden können, bleibt ungeklärt (zum autonomen Lernen vgl. die Darstellung der Arbeit von Krüger et al. (2002) in Abschnitt 2.3). Festzuhalten bleibt, dass die Autoren von einer handlungsorientierten Ausrichtung ausgehen und Informationen wie Zustandsbeschreibungen der Welt vor und nach einer Handlung sowie die Handlungen selbst separat von den Objektmodellen ablegen. Diese unterschiedlichen Informationen werden dann erst in den Objekt-Handlungs-Komplexen zusammengeführt.

#### **2.2.4 Semantische bzw. kategorial orientierte Ansätze zur Objektmodellierung**

In diesem Abschnitt werden Arbeiten zur Objektmodellierung vorgestellt und kritisch bewertet, die sich explizit um die Einführung von semantischen Aspekten in die Objektmodelle bemühen oder um eine kategoriale Darstellung von Objekten, z. B. um semantische Informationen zwischen Objekten vererben zu können. Dabei erweist sich das Verständnis von „Semantik“ in den einzelnen Arbeiten als unterschiedlich ausgeprägt, teilweise beschränkt sich der semantische Anteil auf Dinge wie Relationen zwischen Objekten oder Objektteilen, teilweise werden auch semantische Aspekte eingeführt, ohne sie explizit als solche zu kennzeichnen.

Eine einfache Form der Modellierung von Objektsemantik findet sich in Bauckhage et al. (1998). Der Schwerpunkt dieser Arbeit liegt auf der Modellierung von zusammengesetzten Objekten, wobei als „zusammengesetzt“ hier ausschließlich Objekte bezeichnet werden, deren Teile fest miteinander verbunden sind, also nicht Relationen wie Topf-Deckel (das eine kann ohne das andere bewegt werden) oder Schrank-Schrankschranktür (die gegeneinander beweglich sind). Insofern ist schon die Ausgangsmenge an für eine Modellierung in Frage kommenden Objekten für reale komplexe Szenen in menschenzentrierten Umgebungen bei weitem nicht ausreichend. Zusätzlich sind die Annahmen und das der Modellierung zugrunde liegende semantische Netzwerk in ihrer Komplexität eingeschränkt, da die Domäne einfach gehalten ist und nur Baufix-Teile und deren Zusammensetzungen enthält. Ein Übertragen auf andere, komplexere Domänen ist somit schwierig, und das Lernen der zusammengesetzten Objektmodelle ist nicht vorgesehen, dieses Wissen wurde a priori vorgegeben. Es handelt sich hierbei um ein semantisches Netz, das eine Grammatik möglicher Objektverbindungen beschreibt. Diese Repräsentation ist auf reale Domänen nicht anwendbar, da sie diese zu vereinfacht darstellen würde.



Einen Versuch, Objekte nach ihren funktionalen Qualitäten in Kategorien einzuteilen, unternehmen Winston et al. (1983): „The first step is to describe the cup concept in terms of functional qualities such as liftability, stability, and ability to serve as an accessible container.“ (Winston et al. (1983), S. 6). Die Modellierung dieser funktionalen Objektqualitäten und potentieller Schlussfolgerungen daraus erfolgt mittels eines semantischen Netzwerkes. Festzuhalten ist, dass die hier eingeführten „funktionalen Qualitäten“ zwar außerordentlich wichtig für eine semantische Modellierung von Objekten sind, dass in dieser Arbeit allerdings die weit reichenden semantischen Möglichkeiten dieser Kategorien nicht erkannt werden, und sie stattdessen als rein funktionale Unterteilung von Objekten angesehen werden.

Das Ziel der Arbeit von Mann und Jepson (1998) ist es, Handlungen zwischen Objekten zu verstehen, wobei unter anderem auch die menschliche Hand als Objekt zählt. Hierzu werden den Objektrepräsentationen so genannte Objekteigenschaften zugewiesen, die ein Bindeglied zwischen dem jeweiligen Objekt und den Handlungen, die es ausführen kann, bilden. Im Rahmen der Arbeit wurden vier Objekteigenschaften spezifiziert: FLYER( $o$ ) (Objekt kann selbst beliebige Kräfte und Drehmomente generieren), DRIVER( $o$ ) (Objekt kann beliebige Kraft tangential entlang jeder seiner Kanten ausüben), ROTOR( $o$ ) (Objekt kann beliebiges Drehmoment an jeder seiner Ecken ausüben) und GRASPER( $o$ ) (Objekt kann sich mit jedem beliebigen Objekt, das es in der Szene berührt, verbinden). Objekte können dann auch mehrere dieser Eigenschaften haben, z. B. kann eine Hand sowohl Flyer als auch Grasper sein. Die Autoren weisen selbst darauf hin, dass ihre Darstellung der Objekteigenschaften stark vereinfachend ist. Insbesondere ist die Annahme, dass Objekteigenschaften unabhängig voneinander sind, nicht zutreffend, und die Menge der Objekteigenschaften bleibt bei diesem Ansatz über eine komplette Sequenz fest, so dass weder die eigentlichen Eigenschaften noch die beteiligten Objekte zeitlich veränderlich sind bzw. hinzukommen und verschwinden können. Zusätzlich ist das Problem festzuhalten, dass diese Arbeit zwar einen interessanten Ansatz zur Modellierung von Objektfunktionalitäten zeigt, dieser Ansatz aber zu vereinfachend ist. Insbesondere bleibt die semantische Ebene, d. h. die Bedeutung dieser „Objekteigenschaften“, unberücksichtigt. Zudem können auch hier Objektmodelle nur a priori von einem Experten erstellt werden. Eine Anwendung auf realistische, d. h. größere und komplexere Domänen, erscheint schwierig oder sogar unmöglich, da die so genannten Eigenschaften und die Objekte dann sorgfältig gewählt werden müssen, um Inkonsistenzen zu vermeiden. Insgesamt vernachlässigt der Ansatz, Objekte mit elementaren Formen von Funktionen (Ausübung von Kräften und Drehmomenten in bestimmte Richtungen) zu verknüpfen, die Bedeutung der modellierten Objekte und Handlungen und macht die Verknüpfungen damit gleichzeitig für den Menschen unübersichtlich und schwierig anzulegen.

Bley et al. (2006) stellen ein Objektmodell zur Objektdetektion, Objekterkennung und zum Greifen vor, das über einen rein geometrischen bzw. rein ansichtsbasierten Ansatz hinausgeht. Sie begründen dies damit, dass das üblicherweise modellierte Wissen a priori von einem Experten erstellt werden muss und auf bestimmte Objekte beschränkt ist. In ihrem Ansatz hingegen werden Objektkategorien modelliert (z. B. mit Variationen in Größe und Farbe), um mehr unterschiedliche Objekte mit einem Modell darstellen zu können. Diese generischen Kategorienbeschreibungen enthalten ansichtsbasierte Attribute (in Form geometrischer Primitive) ebenso wie zusätzliches Kategorienwissen (funktionale Eigenschaften, mechanische Attribute und Anweisungen für den Umgang mit dem Objekt). Greifinformation wird a priori festgelegt, da eine Bestimmung zur Laufzeit zeitaufwendig ist. Die Autoren stellen

weiterhin fest, dass für das Greifen in Haushaltsumgebungen eine Wissensbasis benötigt wird, die außer dem üblichen geometrischen bzw. ansichtsbasierten Objektwissen noch weitere Informationen enthält wie Informationen über die durchzuführende Aufgabe (Ausgangs- und Endzustand der Umgebung, auszuführende Handlungen, benötigte Objekte), über die Spezifika des Manipulatorarms und des Endeffektors und über Eigenschaften des Zielobjekts (Form, potentiell durchführbare Handlungen, mechanische Eigenschaften wie Größe, Gewicht etc., sowie potentielle Greifpunkte). Ihr Objektmodell enthält folgende Informationen für jede Objektkategorie: Ansicht, Funktion, Mechanik und bevorzugte Kontaktpunkte. Anstelle von Farbe und Textur wie in klassischen Ansätzen wird die äußere Form verwendet, da nach Ansicht der Autoren ihre Varianz für unterschiedliche Objekte der gleichen Kategorie weniger hoch ist und gleichzeitig eine Silhouettdetektion robuster ist als die Feststellung von Farbinformationen unter wechselnden Beleuchtungsverhältnissen. Verwendet wird die von Biedermann (1987) vorgeschlagene Repräsentation von Objekten als Geone (vgl. Abb. 2.5). Im Rahmen des hier eingeführten zusätzlichen Kategorienwissens beinhaltet funktionales Wissen das Wissen über Handlungen, die mit dem Objekt ausgeführt werden können. Verwendungsanweisungen entsprechen dann Bewegungseinschränkungen für den Endeffektor wie z. B. vorsichtige oder schnelle Bewegung, und als mechanische Eigenschaften werden Gewicht, Schwerpunkt und Zerbrechlichkeit angeführt.

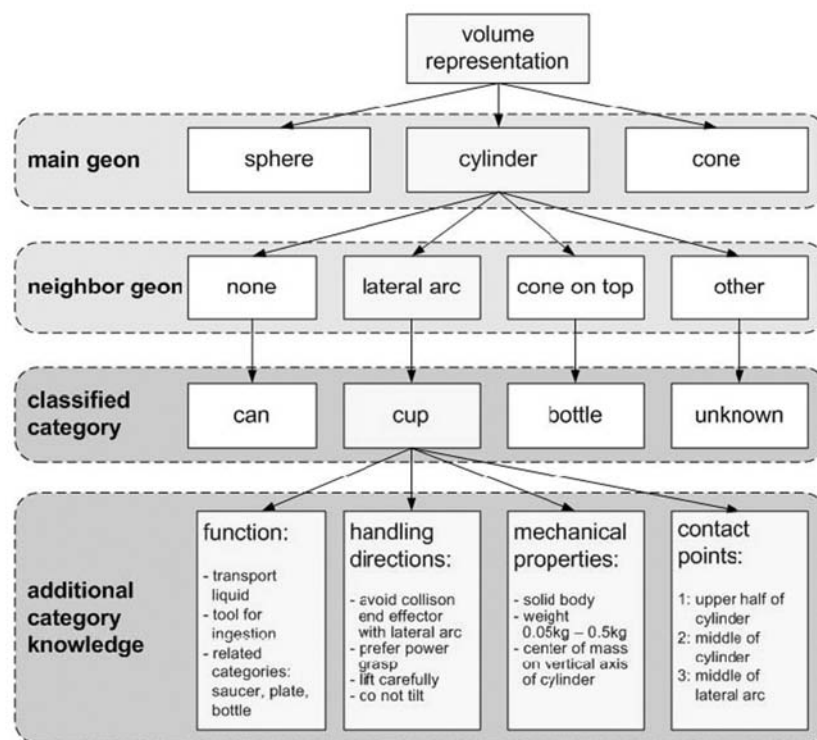


Abbildung 2.5: Darstellung der Wissensbasis bei Bley et al. (Bley et al. (2006), S. 414)

Trotz des interessanten Ansatzes ergeben sich einige problematische Punkte an dieser Arbeit. So ist es nicht verständlich, warum die Autoren sich auf *eine* Art von Objektinformation zur Objekterkennung beschränken. So schreiben sie selbst: „The most critical aspect of the proposed methodology is the performance of the object recognition algorithm.“ (Bley et al. (2006), S. 415). In dieser Hinsicht wäre es sinnvoller, möglichst vielfältige Informationen

über Objekte zur Verfügung stellen, von denen dann je nach Situation (z. B. dem in Frage stehenden Objekt, der Beleuchtung, der zu bewältigenden Aufgabe etc.) die hilfreichste verwendet wird. Zudem ist der Ansatz zur Einbindung zusätzlicher funktionaler und anderer Objektinformationen über die Nutzung von Objektkategorien nicht konsequent ausgearbeitet. Weder wird die Kategorisierung systematisch durchgeführt und Kategorienwissen hierarchisch ausgenutzt, noch werden die relevanten Informationen folgerichtig eingebunden; derzeit existiert nur eine Auflistung aller Informationen, die den Autoren sinnvoll erschienen. Zwar ist hierdurch eine Erweiterbarkeit der Modelle gegeben, aber keine strukturell systematische Modellierung. Ein weiteres Problem ist, dass derzeit die Objektmodelle a priori durch einen Experten erstellt werden müssen, eine interaktive Erstellung durch Laien ist so nicht möglich. Weiterhin räumen die Autoren potentielle Probleme beispielsweise in der Handlungsplanung ein, weil sie ihre Objektmodelle derzeit nur für Aufgaben aus den Bereichen Objektklassifikation und Greifen anlegen. Ob eine Erweiterung mit vertretbarem Aufwand zu sinnvollen Modellen führt, muss zunächst noch überprüft werden. Schließlich unterscheidet dieser Ansatz nicht zwischen einzelnen Objekten: eine Tasse kann nur als „Tasse“ erkannt werden, es kann aber nicht zwischen verschiedenen Tassen unterschieden werden – hier fehlt eine hierarchische Komponente und die Berücksichtigung unterschiedlicher Objektinstanzen (z. B. einer gefüllten und einer ungefüllten Tasse) in einer realen Umgebung.

Die in diesem Abschnitt vorgestellten Arbeiten zeigen zwar interessante Ansätze zu einer Modellierung von Objektsemantik, es fehlt allerdings jeweils eine konsequente Struktur, in der diese semantischen Informationen folgerichtig und systematisch eingebunden werden können. Auch sind die vorgeschlagenen Vorgehensweisen teilweise nur mit großem Aufwand zu realisieren, so dass sie für eine reale Alltagsumgebung nicht in Frage kommen. Das gleiche gilt für eine interaktive Modellierung von Objekten unter Einbindung semantischer Informationen durch einen nicht entsprechend ausgebildeten Benutzer: hierfür sind die vorgeschlagenen Objektmodelle nur eingeschränkt oder teilweise gar nicht geeignet.

## 2.3 Lernen von Objektmodellen

Wie bereits mehrfach angedeutet, beschränkt die überwiegende Mehrzahl der Forschungsarbeiten im Bereich „Objektmodellierung“ sich auf fest vorgegebene Objektmodelle, und nur ein geringer Teil der Arbeiten befasst sich mit der Frage, wie sie gelernt werden können. Gelernte Modelle werden dabei oft als Basis für autonome Navigation eingesetzt, seltener für Anwendungen wie Greifen, Handlungsplanung etc. Es existieren sowohl Arbeiten zum unüberwachten Lernen von Objektmodellen als auch Arbeiten zu interaktiven Lernansätzen. Einige ganze Reihe von Arbeiten verfolgt das Paradigma der „aktiven Perzeption“, entsprechend werden einige Arbeiten, die aktive Perzeption zum Lernen von Objektmodellen einsetzen, hier unter Berücksichtigung der für die vorliegende Arbeit vorgegebenen Anforderungen untersucht.

Mit der automatischen Erstellung dreidimensionaler Objektmodelle befasst sich Fautz (2002), der die häufig übliche Modellerstellung von Hand als zu langsam und aufwendig kritisiert. In seiner Arbeit wird die Erstellung eines dreidimensionalen Oberflächenmodells sowie eines Texturmodells eines Objektes mit guten Ergebnissen automatisch mittels einer auf einen

Roboterarm montierten Kamera durchgeführt. Dieser Aufbau erlaubt es, die zur Modellbildung zu verwendenden Objektansichten flexibel zu wählen. Die Geometrie des Objektes wird mittels eines Volumenschnittverfahrens rekonstruiert, wobei ein Oberflächenmodell des Objektes erzeugt wird. Der Vorteil der vorgeschlagenen Methode ist eine hohe Verarbeitungsgeschwindigkeit. Das Texturbild wird aus den Kamerabildern rekonstruiert. Diese Arbeit ist ein schönes Beispiel für die Versuche, den Prozess der Objektmodellerstellung zu automatisieren, wobei allerdings mit der vorgeschlagenen Methode nur die Rekonstruktion der visuellen Hülle des Objektes möglich ist, also keine beliebig geformten dreidimensionalen Objekte verwendbar sind. Wie bei Fautz (2002) gesehen werden kann, gibt es Objektattribute, die sich gut für eine autonome Erfassung durch ein maschinelles System eignen. Wie oben bereits argumentiert, existieren aber andererseits auch Objektinformationen wie kontext- oder benutzerspezifische Informationen, die nur mit großem Aufwand oder überhaupt nicht von einem System ohne menschliche Hilfe extrahiert werden können. Arbeiten wie die hier beschriebene von Fautz können daher bei den gegebenen Anforderungen immer nur als Ergänzung gesehen werden, um die Modellerstellung soweit als möglich zu automatisieren und dem Menschen damit ermüdende und sich häufig wiederholende Arbeiten abzunehmen. Selbst bei den hierfür in Frage kommenden Objektattributen ist aber gelegentlich die Unterstützung des Menschen erforderlich, z. B. wenn von einem Objekt unter bestimmten Voraussetzungen eine besondere Aufnahmedichte aus Winkeln erforderlich ist, die maschinell nicht abgeleitet werden kann, beispielsweise eine besonders dichte Punktwolke im Bereich einer später besonders häufig auftretenden Ansicht des Objektes. Es muss also dem Benutzer ermöglicht werden, gegebenenfalls korrigierend oder ergänzend einzuwirken.

Modayil und Kuipers (2004) verfolgen das Ziel des unüberwachten Lernens von Objekten aus Sensordaten. Der Objekt-Lernprozess besteht aus vier Schritten: Individuation, Verfolgung, Bildbeschreibung und Kategorisierung. Beispielhaft wird das Prinzip gezeigt in einer nicht-statischen Indoor-Umgebung mit einem mobilen Roboter, der über ein Laserentfernungsmesssystem verfügt. Dabei muss das System bei der Objektkategorisierung eigene Klassen bilden und inkrementell neue Klassen hinzufügen. Die Ergebnisse des beispielhaft durchgeführten Lernprozesses werden nur qualitativ beschrieben, aus den gegebenen Informationen lässt sich aber ableiten, dass insgesamt nur wenige Objekte vom System bearbeitet wurden. Verarbeitungsdauern oder Erkennungsraten werden von den Autoren nicht angegeben. Insgesamt lässt sich dieses Vorgehen nur als eine von mehreren Informationsquellen nutzen, da hierdurch nicht alle Arten von Wissen akquiriert werden können, die ein Serviceroboter in seinen Objektmodellen benötigt. Positiv zu sehen ist der Ansatz, dass Objektlernen für einen sinnvollen Einsatz von Robotern in menschlichen Alltagsumgebungen notwendig ist. Allerdings ist der Ansatz des autonomen Lernens nicht für alle Arten von Objektwissen und für alle Anforderungen geeignet. Positiv zu bewerten ist auch, dass das System direkt auf einem mobilen Roboter unter Verwendung realer Sensorik eingesetzt und evaluiert wird, auch wenn in dieser Arbeit nur ein eingeschränktes Objektmodell erstellt wird. Bei Einsatz weiterer Sensoren und Extraktion weiterer Objektattribute wird die Verarbeitungszeit vermutlich substantiell ansteigen.

Die Arbeit von Lömker und Sagerer (2002) hingegen verfolgt genau den entgegen gesetzten Ansatz. Ziel dieser Arbeit ist das interaktive Lernen von neuen Objekten unter Nutzung von multimodalen Eingabekanälen. Die Autoren kommen zu dem Schluss, dass ein Serviceroboter in einer Alltagsumgebung nicht mit einer vordefinierten Menge von Objekten und Objekt-

namen arbeiten kann. Ziel ist es, dass ein Mensch einen neuen Serviceroboter erwerben und ihm auf einfache und natürliche Art und Weise neue Objekte „beibringen“ kann. Eine Teilaufgabe für das Robotersystem ist dann das sprachlich unterstützte Lernen, das auch eins der Ziele dieser Arbeit ist: Ein Mensch kann mit deiktischen Gesten und Sprache auf neue Objekte referenzieren, die dann vom Robotersystem mit einer Videokamera erfasst werden. Das System extrahiert visuelle Attribute zur Objekterkennung und Handtrajektorien für die Gestenerkennung. Auf Grundlage dieser Information versucht es, entweder ein bereits bekanntes Objekt zu finden und die Informationen hierzu zu aktualisieren, oder ein neues Objekt zu lernen. Über einen sprachlichen Dialog kann das System weitere Informationen vom Benutzer abfragen. Falls das System im Zielbereich einer Zeigegeste nur ein unbekanntes Objekt entdecken kann, wird dieses gelernt: Hierzu wird eine Objektansicht zusammen mit automatisch extrahierten Attributen in einer Datenbank gespeichert. Zur Lokalisation des unbekanntes Objektes, dessen Grenzen noch nicht bekannt sind, muss der Benutzer das Objekt aus dem Bild entfernen. Getestet wurde das System bisher mit fünf Objekten.

Auch an dieser Arbeit sind mehrere Gesichtspunkte kritisch anzumerken. Zum einen ist es ungünstig, dass das System den Dialog führt, anstatt den Benutzer über die nächsten Schritte der Modellierung entscheiden zu lassen. Weiterhin ist das System rein kamerabasiert und ausschließlich auf sprachliche und bildbasierte Informationen ausgelegt, andere Informationen sind demzufolge schwierig einzugeben bzw. zu integrieren. Die relativ schlechten Evaluationsergebnisse sind allerdings vermutlich auf die Wahl der implementierten Methoden zur Objekterkennung etc. zurückzuführen und sagen daher zunächst nichts über das Grundprinzip aus. Schließlich enthalten die gelernten Objektmodelle keinerlei semantische oder funktionale Information, und die Multimodalität des Systems beschränkt sich auf die „Standardmodalitäten“ Sprache und Zeigegesten.

Der Begriff „Aktive Perzeption“ bezieht sich auf das Konzept, das ein System wie z. B. ein Roboter Bewegung aktiv einsetzt, um neues Wissen über die Umwelt oder Objekte zu erwerben: „Active perception refers to the use of motor action to simplify perception (Ballard (1991)), and has proven its worth many times in the history of robotics. The most well-known instance of active perception is active vision. The term „active vision“ is, in common usage, essentially synonymous with moving cameras. But there is much to be gained by taking advantage of the fact that robots can be actors in their environment, not simply passive observers.“ (Fitzpatrick (2003a), S. 4f). In der Literatur zur Objektmodellierung findet sich eine Reihe von Arbeiten, die sich auf dieses Konzept berufen.

Krüger et al. (2002) schlagen eine Verbindung aus einem aktivem Vision-System mit einem Roboterarm vor. Aus einer Stereobildsequenz des vom Roboterarm gehaltenen Objekts wird ein Kantenmodell extrahiert. Grundgedanke dieser im Rahmen des Projektes PACO-PLUS angesiedelten Arbeit (vgl. Abschnitt 2.2.3) ist, wie auch in vielen anderen Arbeiten auf diesem Gebiet, der so genannte „perception-action cycle“, d. h. die Tatsache, dass Perzeption und Aktion sich gegenseitig unterstützen und voneinander abhängen. Die Autoren merken an, dass eine gewisse Menge an a priori-Wissen vorhanden sein muss, damit ein komplexes vision-basiertes System lernen kann.

Eine ganze Serie von Forschungsarbeiten auf dem Gebiet der aktiven Perzeption wurde mit dem Roboter Cog des Massachusetts Institute of Technology durchgeführt (dargestellt z. B. in Fitzpatrick (2003a)). Der Grundgedanke ist, dass Trainings- und Anwendungsphase beim

Objektlernen nicht getrennt werden sollten, da neue Situationen, für die das System nicht trainiert wurde, dazu führen, dass das gelernte Wissen nicht mehr angewendet werden kann. Daher soll ein System entwickelt werden, das in unbekanntem Situationen autonom neues Wissen akquiriert. Exemplarisch wurde ein Objekterkennungssystem aufgesetzt, das mittels aktiver Segmentierung Objektgrenzen automatisch erkennen kann, indem es Objekte bewegt. Als Evaluations- und Ausführungsplattform hierfür dient der Roboter Cog. Der Roboter soll damit selbst in der Lage sein, neue Erfahrungen zu gewinnen und zu lernen. Hierzu wurde ein „poking behavior“, d. h. ein aktives Anstoßen von Objekten, implementiert, das eine zuverlässigere Segmentierung von Objekten und Hintergrund erlaubt als reine passive Bildverarbeitung. Zur Objekterkennung kann der Roboter mittels aktiver Segmentierung Beispiele für Objektansichten sammeln, die dann verwendet werden könnten, um ein Objekterkennungsmodul zu trainieren, so dass die Objekte zukünftig ohne physischen Kontakt erkannt werden könnten. Es ist zu beachten, dass diese Arbeit vornehmlich als „proof of concept“, d. h. als Nachweis für die Anwendbarkeit der vorgeschlagenen Methode, gedacht ist. Das genaue Vorgehen beim Anstoßen sowie das Verfahren zur Separation von Objekt und Hintergrund und experimentelle Ergebnisse hierzu beschreibt Fitzpatrick (2003b) detaillierter.

Der Ansatz des Anstoßens von Objekten ist zwar interessant, ist allerdings nur in einem eingeschränkten Rahmen möglich (keine zerbrechlichen Objekte, nicht zu viel Kraft, Dinge könnten hinunterfallen etc), nur bei beweglichen Objekten und erlaubt nur die Feststellung der Objektgrenzen und damit die Aufnahme von Objektansichten. Die autonome Erkennung anderer Objektmerkmale wird in den vorgestellten Arbeiten nicht diskutiert. Eine mögliche Interaktion zwischen Mensch und Roboter und insbesondere eine Kombination aus Interaktion und Autonomie, je nach Situation, Objekt und zu lernendem Objektattribut, wird nicht in Erwägung gezogen. Ungeklärt ist in den Arbeiten von Fitzpatrick et al. auch, woher der Roboter die Position eines potentiellen Objektes kennt, denn dazu ist ja bereits einiges Wissen über die Umgebung und evtl. auch über das Objekt notwendig. Schließlich wird auf die Repräsentation der Objekte in diesen Arbeiten nur insofern eingegangen, als es für die Erläuterung der Objekterkennungsmethoden etc. notwendig ist.

Die Idee, Objekte anzustoßen, um Objektgrenzen etc. zu lernen, ist auch Ausgangspunkt der Arbeiten von Arsenio (2003) und Arsenio (2004), wobei der Schwerpunkt hier allerdings mehr auf dem Lernen von Handlungen und Handlungssequenzen als auf dem Lernen von Objektwissen liegt. Die Probleme dieses Ansatzes bleiben allerdings die gleichen wie oben bereits angemerkt. Als Versuchsplattform wird in dieser Arbeit wiederum der Roboter Cog eingesetzt.

Wie Fitzpatrick et al. verwenden auch Natale (2004) Roboteraktionen zum Objektlernen. Die verwendeten Roboteraktionen waren zum einen das Ziehen und Schieben von Objekten auf einem Tisch um herauszufinden, was jeweils ihre Zielposition sein wird. Dabei wurden die Objekte immer von einem Menschen wieder zurück in die Tischmitte gestellt. Mit dem erworbenen Wissen sollte der Roboter danach einfache Aufgaben lösen, bei denen das Objekt an eine bestimmte Zielposition zu bewegen war. Zum zweiten wurde, im Gegensatz zu Fitzpatrick et al., eine Roboterhand verwendet, die mehr als einfache Anstoßoperationen ausführen kann. Hiermit sollen Objekte dann gegriffen, die jeweiligen Griffe gelernt und die Objekte später anhand der Griffparameter wieder erkannt werden. In den in dieser Arbeit beschriebenen Experimenten wurden noch keine visuellen Informationen verwendet, dies er-

folgt dann in den hierauf aufbauenden Arbeiten von Natale et al. (2005b) und Natale et al. (2005a).

In Natale et al. (2005b) und Natale et al. (2005a) wird eine ganze Entwicklungssequenz für einen humanoiden Roboter vorgeschlagen, die es ihm erlauben soll, sukzessive Dinge über sich selbst und seine Umwelt zu lernen. Im Rahmen der gegebenen Problemstellung ist hier von der Entwicklungsschritt von Interesse, in dem Informationen über Objekte erworben werden sollen. Ausgangspunkt ist dabei entweder, dass der Roboter zufällig greift, bis er ein Objekt erreicht, oder dass ihm ein Objekt in die Hand gegeben wird. Der Roboter dreht das Objekt dann mehrere Male und nimmt dabei Kamerabilder auf. Mit Hilfe dieser wird ein Objektmodell aus Blobs zusammengesetzt. Die Steuerung für das solcherart mit dem Roboter Babybot durchgeführte Experiment war teilweise fest implementiert, so z. B. die Bewegungen der Roboterhand und des Armes. Die Sequenz war daher in dieser Art nur durchführbar aufgrund der Elastizität der Roboterhand. Am Ende der Lernsequenz ermittelt der Roboter noch das Gewicht des Objekts und seine Konsistenz als haptische Informationen. Diese Arbeiten von Natale et al. betrachten nur eine geringe Menge von potentiellen Objektattributen. Die Lokalisation eines zu lernenden Objektes ist relativ umständlich, oder es wird menschliche Hilfe benötigt, was dem eigentlich verfolgten Ansatz des autonomen Lernens widerspricht. Die fest einprogrammierten Robotertrajektorien für den Lernprozess schränken die Möglichkeiten des Systems beispielsweise in Bezug auf Objektformen und -größen stark ein. Schließlich stellt sich die Frage, ob es nicht sinnvoller wäre, die Objekte in einer eigens hierfür konzipierten Modellierungsumgebung zu lernen. Hierdurch könnten die Nachteile der Handhabung und Erfassung durch den Roboter (Verdeckungen, keine hochauflösende Sensorik, etc.) vermieden werden. Die Vorteile der Erfassung direkt durch den Roboter z. B. in Bezug auf flexible Bewegungen werden in den Arbeiten von Natale et al. nicht genutzt und es ist zusätzlich die Hilfe eines Menschen zur geeigneten Platzierung von Objekten notwendig. In einer Modellierungsumgebung könnten dann zusätzlich auch noch andere Objektattribute gelernt werden, die nicht von den Greiffertigkeiten des Roboters abhängen.

Moll und Erdmann (2001) orientieren sich ebenfalls am Paradigma der aktiven Perzeption, speziell am autonomen Lernen von taktil erfassbaren Objektattributen. Der Roboter manipuliert hierzu ein Objekt in seinem Greifer und stellt hierdurch dessen Form, Gewicht und Schwerpunkt fest. Dieses Vorgehen bringt allerdings in der durchgeführten Form und mit dem gewählten Greifer Schwierigkeiten mit sich bei Objekten mit konkaven Flächen. Es erfordert zudem eine große Fingerfertigkeit und Geschicklichkeit vonseiten des Roboters, und ist daher mit den meisten gängigen Robotergriffern nur schwerlich durchführbar. Zudem werden bisher keine Reibung etc. berücksichtigt, so dass die Ergebnisse nur eingeschränkt aussagekräftig sind. Trotzdem handelt es sich hier um einen interessanten Ansatz, auch wenn hiermit nur wenige Objektattribute überhaupt gelernt werden können. Ein ähnliches Konzept verfolgen auch Okamura und Cutkosky (2001), deren Arbeit die haptische Exploration unbekannter Objekte mit Roboterfingern zum Zwecke der Objektmodellierung zum Ziel hat. Hierbei soll der Roboter die Objektoberfläche abtasten, um kleine Erhebungen, Vertiefungen etc. zu spüren und in ein Objektmodell zu übernehmen. Obwohl der Grundgedanke dieser Arbeit interessant ist, ist das Abtasten von Objekten mit den derzeit zur Verfügung stehenden Roboterhänden und Sensoren noch nicht praktisch anwendbar. Hinzu kommt, dass der Erkenntnisgewinn für Anwendungen in realen Umgebungen gering ist, da ähnliche Oberflächeninformationen zuverlässiger und weit weniger aufwendig aus anderer Sensorik, wie

etwa einem Laserscanner, gewonnen werden können.

Die aktuellen Arbeiten zum Thema Lernen von Objektmodellen zeigen zwar teilweise interessante Ansätze und Vorgehensweisen, insgesamt sind sie aber noch weit vom Erwerb eines ausreichend bedeutungsvollen, gut strukturierten Objektmodells unter realen Bedingungen entfernt. Einige der Arbeiten bemühen sich um ein autonomes Lernen von Objekten, das nicht ausreicht für das Lernen von Objektmodellen, die genügend komplex für Alltagsanwendungen sind. Betrachtet werden jeweils nur kleine Teilaspekte, und in der Regel existiert kein sinnvoll strukturiertes Konzept für den Aufbau der Objektmodelle. Auch semantische Objektinformationen werden in diesen Arbeiten nicht weiter betrachtet.

## 2.4 Benutzerinteraktion zur (Objekt-)Modellierung

Wie oben schon erwähnt, wird in der überwiegenden Mehrzahl der Arbeiten zur Objektmodellierung keine Benutzerinteraktion im eigentlichen Sinne eingesetzt. Stattdessen werden die Objektmodelle durch einen Experten angelegt, meist vor Beginn der Anwendung, und sind oft fest einprogrammiert oder nur durch wenig benutzerfreundliche Schnittstellen veränderbar (vgl. Steinfeld (2004)). Der vorliegende Abschnitt befasst sich mit Arbeiten zur interaktiven Modellierung sowie mit den hierfür in Frage kommenden Interaktionsmodi. Unterschiedliche Modi zur Mensch-Roboter-Interaktion und speziell zur interaktiven Objektmodellierung werden vorgestellt. Dabei werden die existierenden Forschungsarbeiten zur interaktiven Modellierung im Zusammenhang mit den jeweiligen Interaktionsmodi diskutiert.

Generell lässt sich festhalten, dass auch in interaktiven Systemen der Informationsfluss in Richtung Mensch-Roboter oft viel umfangreicher ist als in umgekehrter Richtung. In Bezug auf Objektmodellierungssysteme bedeutet dies, dass schwerpunktmäßig die Übertragung von Benutzerinformationen in das Robotersystem betrachtet wird und weniger intensiv die Darstellung der roboterinternen Informationen für den Menschen. Letzteres erfolgt meist über eine graphische Benutzerschnittstelle oder ein anderes graphisches Anzeigesystem, seltener auch über Sprachausgabe, wird aber in der überwiegenden Zahl der Fälle in den Forschungsarbeiten nicht detailliert betrachtet. Ausnahmen bilden z. B. die Arbeit von Minar (2004) zur Visualisierung von Roboterzuständen oder von Lopes und Teixeira (2000) mit sprachlichen Dialogen. Im Folgenden wird dieser Schwerpunktsetzung in der Literatur Rechnung getragen und daher hauptsächlich der Informationsfluss vom Menschen zum Roboter mittels unterschiedlicher Interaktionsmodi diskutiert.

### 2.4.1 Graphische Benutzerschnittstellen

Graphische Benutzerschnittstellen (engl. „graphical user interface“, GUI) sind zu einem Standard-Interaktionsmodus geworden für nahezu alle Arten von Anwendungen auf dem Gebiet der Mensch-Maschine-Interaktion und entsprechend verbreitet auch für Modellierungsanwendungen. Es existiert eine umfangreiche Literatur zu den Grundlagen und Entwurfsprinzipien für graphische Benutzerschnittstellen. Eine gute Einführung bieten Galitz (1997), Dahm (2006) oder Dix et al. (2006). Für die Interaktion zwischen einem Modellierungssystem und



einem Benutzer gelten dabei die gleichen Regeln wie für alle anderen graphischen Dialogsysteme: Aufgabenangemessenheit, Selbstbeschreibungsfähigkeit, Steuerbarkeit, Erwartungskonformität, Fehlertoleranz, Individualisierbarkeit und Lernförderlichkeit (aus der DIN EN ISO 9241 „Ergonomische Anforderungen für Bürotätigkeiten mit Bildschirmgeräten“, Teil 10 „Grundlagen der Dialoggestaltung“, zitiert nach Dahm (2006)).

Ein Beispiel für die Verwendung einer GUI zur interaktiven Modellierung von Handlungen findet sich bei Dillmann et al. (2002), wo der Benutzer in einem Prozess des Programmierens durch Vormachen Systemhypothesen annehmen oder verwerfen kann. Diese Arbeit wird detailliert diskutiert in Abschnitt 2.4.5. Insgesamt wird die Verwendung graphischer Benutzerschnittstellen in der Literatur häufig nicht weiter thematisiert, obwohl sie in vielen Systemen eingesetzt werden. Dies hängt vermutlich damit zusammen, dass graphische Benutzerschnittstellen als Standard und nicht weiter diskutierenswert betrachtet werden, und dass das Hauptaugenmerk in der Robotik und Modellierung nicht auf ergonomischer Schnittstellengestaltung liegt.

### 2.4.2 Sprachliche Interaktion

Sprache ist nach den graphischen bzw. textuellen Schnittstellen wohl die am häufigsten verwendete Interaktionsmodalität zwischen Mensch und Roboter. Dementsprechend existiert auch eine Vielzahl von Publikationen, in denen die Verwendung von sprachlicher Interaktion erläutert wird. Dieser Abschnitt kann daher nur einen kleinen Einblick in die aktuellen Arbeiten geben. Im Gegensatz zu manchen anderen Interaktionsmodi, die eine aufwendige technische Ausstattung voraussetzen (wie Datenhandschuhe oder Durchsichtbrillen), werden zur sprachlichen Interaktion nur verhältnismäßig günstige und einfach zu beschaffende und zu integrierende Komponenten wie ein Mikrofon, eine Soundkarte und für eine Sprachausgabe zusätzlich Lautsprecher benötigt. Auf technische Details dieser Komponenten wird daher im Folgenden nicht weiter eingegangen, da die hierfür zur Verfügung stehenden Produkte als allgemein bekannt vorausgesetzt werden.

Generell lässt sich festhalten, dass sich viele Forschungsgruppen darum bemühen, eine Form der sprachlichen Interaktion zwischen Mensch und Roboter aufzusetzen. Diese Interaktion findet aber oft nur auf einer einfachen Ebene statt. Wrede et al. (2004) stellten in einer Benutzerstudie fest, dass eine sprachliche Interaktion mit einem Roboter von Benutzern als sehr wichtig eingeschätzt wird. Zudem empfanden die Benutzer eine Rückmeldung durch den Roboter als hilfreich, wobei die Art der gewünschten Informationen allerdings unterschiedlich war. Die Autoren behaupten, dass menschliche Benutzer mehr auf die Qualität der (sprachlichen) Roboter-Schnittstelle achten als auf die Qualität der Roboter-Funktionalität. Lopes und Teixeira (2000) argumentieren, dass für einen nicht speziell ausgebildeten Benutzer Sprache die einzig hinreichend praktische Schnittstelle zu einem Roboter ist, da nur eine sprachliche Schnittstelle flexibel genug ist, um eine Lösung für das „Symbol grounding problem“, d. h. die Verknüpfung von Symbolen und Dingen in der Welt, zu gewährleisten.

Drygajlo et al. (2003) untersuchen in ihrer Arbeit die Anforderungen, die an die sprachliche Kommunikation eines Roboters gestellt wird, der Besucher durch ein Museum führen soll. Insbesondere identifizieren sie dabei drei Problemschwerpunkte: Erstens muss die sprachliche Interaktion leicht und einfach aufzusetzen sein und zu robusten und schnellen Ergebnissen

führen, zweitens sind in diesem Rahmen nur eingeschränkte Sprachverstehensfähigkeiten vonseiten des Roboters notwendig, und schließlich muss das System der lauten Umgebung mit viel Hintergrundlärm, insbesondere anderen Sprachgeräuschen, Rechnung tragen. Sie schlagen eine Architektur für die sprachliche Schnittstelle vor, die für diese und ähnliche Anwendungen geeignet ist. Im Rahmen der Expo 2002 wurden basierend hierauf Experimente mit Robotern durchgeführt. Leider lassen sich die Ergebnisse dieser Arbeit nur beschränkt auf die hier betrachteten Anforderungen übertragen, da die Umgebung und die Ziele der Kommunikation nicht vergleichbar sind.

Im Folgenden wird anhand einiger beispielhaft ausgewählter Projekte, in denen Serviceroboter sprachlich mit Menschen interagieren, ein Überblick gegeben über den aktuellen Stand der Forschung und die Breite der untersuchten Lösungsansätze und Vorgehensweisen auf diesem Gebiet.



Abbildung 2.6: Interaktion zwischen Mensch und Roboter BIRON (aus: Haasch et al. (2004), S. 1)

Zur Modellierung neuer, vom Benutzer gezeigter Objekte wird die Sprachinteraktions-Komponente des Roboters BIRON verwendet, der in Haasch et al. (2004) und Wrede et al. (2004) vorgestellt wird (vgl. Abb. 2.6). Es handelt sich um einen Serviceroboter, der in menschenzentrierten Alltagsumgebungen wie Büro oder Haushalt eingesetzt werden soll. Die Autoren gehen davon aus, dass Menschen in diesen Szenarien keine speziellen Interaktionsgeräte (wie z. B. Datenhandschuhe) tragen. Der Roboter muss daher mit seiner Sensorik die multimodalen Interaktionsakte des menschlichen Gegenübers erfassen und verstehen. Verwendet werden auf dem Roboter integrierte Mikrophone, d. h. die Qualität der aufgenommenen Sprachsignale ist entsprechend schlechter als bei Mikrophonen, die vom Benutzer am Körper getragen werden. Um die Qualität der Sprachsignale zu verbessern, wird Beamforming eingesetzt. Verarbeitet wird natürliche Spontansprache, wobei die Spracherkennungskomponente von der Aufmerksamkeitsteuerung aktiviert wird, um eine unkontrollierte Verarbeitung zu vermeiden. Zusätzlich wird ein Dialogmanager eingesetzt zur Steuerung des Dialogverhaltens des Roboters, wobei das Zustandekommen der sprachlichen Roboteräußerungen nicht detailliert beschrieben wird.

Ein weiteres Beispiel für einen mobilen Serviceroboter, der sprachlich (allerdings auf japanisch) mit Menschen interagiert, ist der Roboter Jijo-2 (vgl. Fry et al. (1998), Asoh et al.

(2001)). Bisher beherrscht Jijo-2 nur ein festes Vokabular von ca. einem Dutzend Sprachkommandos, d. h. es können nur fest eintrainierte Ausdrücke erkannt werden, nicht Spontansprache wie bei BIRON. Auch hier wurde ein Dialogmanager verwendet, wobei die Autoren bisher das Problem der Generierung von natürlich klingenden sprachlichen Roboteräußerungen nicht näher untersucht haben, es wurden hierfür vorgefertigte Bausteine verwendet. Jijo-2 soll Modelle einer Büroumgebung selbständig oder durch Menschen überwacht lernen. Hierzu werden statistische Lernverfahren zum Lernen von Landmarken und Features der Umgebung (d. h. Lernen von Karten zur Navigation) eingesetzt. Auch hier wird Beamforming zur Verbesserung der Sprachqualität eingesetzt. Besonderes Augenmerk dieser Arbeit liegt auf der Ausgestaltung der Dialogkomponente.

Lopes und Teixeira (2000) stellen den Roboter CARL vor, der natürlichsprachliche Anweisungen von einem Benutzer erhält, die er entsprechend umsetzen soll. Im Rahmen von Dialogen, die auch per Sprachsynthese generierte Äußerungen des Roboters beinhalten, soll CARL dann die Bedeutung von neuen Symbolen wie „um die Ecke gehen“ lernen.

Im Rahmen des Programmierens durch Vormachen verwenden Dillmann et al. (2002) einen einfachen Parser, der das erkannte Verb und, soweit vorhanden, das Subjekt, Akkusativobjekt, Dativobjekt und die Präpositionalphrase einer sprachlichen Äußerung liefert. Ihr Ziel ist die Kommandierung von Robotern und insbesondere das Programmieren durch Vormachen unter Erfassung und Verarbeitung der Sprachkommandos eines Benutzers.

### 2.4.3 Interaktion mittels Gesten, Griffen und anderer Benutzerbewegungen

In den Bereich der Mensch-Roboter-Interaktion mittels Gesten, Griffen und anderer Benutzerbewegungen fallen eine Reihe unterschiedlicher Ansätze und Interaktionsmodi, die hier im Überblick vorgestellt werden. Eine detaillierte Diskussion der einzelnen Verfahren ist dabei im Rahmen der vorliegenden Arbeit nicht möglich und auch nicht notwendig. Es soll vielmehr ein Überblick über die möglichen Interaktionsmodi gegeben werden sowie Verweise auf weiterführende Arbeiten, in denen der Stand der Forschung dann jeweils ausführlicher präsentiert wird.

#### Bildbasierte Verfahren

In diesen Bereich fallen eine Reihe unterschiedlicher Verfahren, die sich durch die Verwendung bildgebender Sensoren auszeichnen, unter anderem Verfahren zur Erkennung von Gesten, insbesondere von Zeigegesten, aber auch von symbolischen und anderen Gesten, Verfahren zur Erkennung von Griffen, von Blickrichtungen basierend auf der Kopfstellung, zur Personenverfolgung, zur Verfolgung speziell von Kopf und Händen oder zur Bewegungserfassung. Entsprechend unterschiedlich sind dann die jeweils verwendeten technischen und algorithmischen Methoden. Dieser Abschnitt kann nur wenige ausgewählte Arbeiten aus dem Bereich der bildbasierten Verfahren vorstellen.

Zeigegesten sind die am häufigsten gewählte Gestenform zur Roboterinstruierung (z. B. bei Wachsmuth und Sagerer (1999)), teilweise werden aber zusätzlich noch andere Gestenformen wie symbolische Gesten oder Greifgesten verarbeitet (vgl. Giesler (2005), Perzanowski

et al. (2001) oder Dillmann et al. (2002)). Grundlage hierfür sind Methoden zur Erkennung und Verfolgung der Hände, basierend auf einer Hautfarbsegmentierung (z. B. bei Nickel (2003)), unter Verwendung magnetfeldbasierter Positionssensoren (vgl. die untenstehenden Erläuterungen) oder auch eine Kombination unterschiedlicher Verfahren, beispielsweise magnetfeldbasierte Handverfolgung, bildbasierte Handverfolgung auf Basis von Konturmerkmalen und bildbasierte Handverfolgung auf Segmentierungsbasis bei Ehrenmann (2003), der auch die Vor- und Nachteile der einzelnen Verfahren beleuchtet.

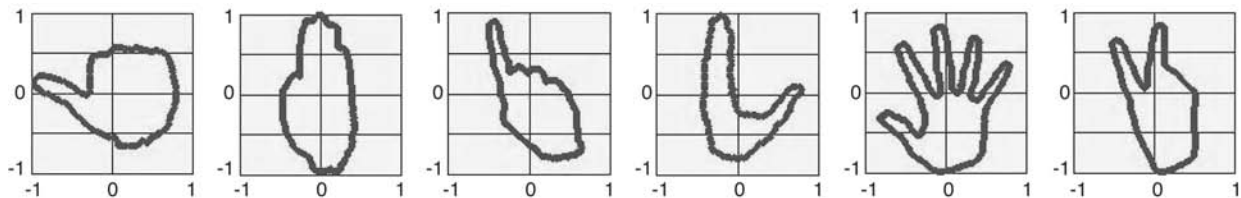


Abbildung 2.7: Silhouetten statischer Referenzgesten zur Gestenerkennung bei Ehrenmann (Ehrenmann (2003), S. 142)

Zeigegesten können erkannt werden durch Verwendung von Hidden Markov Modellen auf den Handtrajektorien, wie bei Nickel (2003). Zur Bestimmung der Zeigerichtung wurden in dieser Arbeit zwei Verfahren eingesetzt: einmal auf Grundlage der Sichtlinie zwischen Auge (hier: Kopfmittelpunkt) und Hand; zum anderen basierend auf der Fortsetzung des Unterarms, da der Unterarm als das einzige Objekt angenommen wird, dass sich während der Haltephase einer Zeigegeste im Umkreis von 40 cm um die zeigende Hand befindet. Ehrenmann (2003) bestimmt symbolische statische Gesten durch den Einsatz von Neuronalen Netzen nach einer Merkmalsextraktion auf den Handsilhouetten (vgl. Abb. 2.7), dynamische Gesten ebenfalls mit Hilfe von Hidden Markov Modellen. Auch eine Gestenerkennung mit Datenhandschuh ist möglich (vgl. Abschnitt 2.4.3). Zur Griffbestimmung wird häufig ebenfalls ein Datenhandschuh eingesetzt, teilweise wird aber auch das Vorhandensein einer Griffgeste aus der Fingerstellung bildbasiert extrahiert (Giesler (2005)). Insgesamt gibt Ehrenmann (2003) einen Überblick über den Stand der Forschung auf den Gebieten bildbasierte Interaktion, Gesten- und Griffenerkennung.

Zur Detektion der Blickrichtung des Benutzers wird oft Information aus der Kopfstellung verwendet (Nickel (2003)), die per Hautfarbsegmentierung ermittelt wurde. Bei entsprechend vorhandener Sensorik erfolgt auch eine markerbasierte Bestimmung, beispielsweise in AR-Systemen (Wachsmuth et al. (2005), Giesler (2005)). Personenverfolgung wird auf unterschiedlichste Arten nicht nur bildbasiert realisiert, z. B. durch hautfarbbasierte Segmentierung von Kopf und Händen, durch Differenzbilder oder durch Tiefenbildanalyse, sondern teilweise auch durch Detektion der Beine in 3D-Punktwolken (durch Verwendung von auf mobilen Plattformen befestigten Laserscannern) oder aus auditiven Richtungsinformationen (Haasch et al. (2004)).

Weiterhin zur Personenverfolgung geeignet sind auch markerbasierte Ganzkörper-Trackingsysteme, die bei entsprechender Vermarkierung auch zur Bewegungserfassung und Erstellung von Bewegungsmodellen genutzt werden können, wie in Stein et al. (2006). Diese Systeme werden von unterschiedlichen Firmen kommerziell vertrieben, beispielsweise das Motion Captor-System von Meta Motion (Meta Motion (2006b)) oder von VICON (Vicon (2006)) (Abb. 2.8). Hierbei wird die zu verfolgende Person mit optischen Markern oder Leuchtdioden



Abbildung 2.8: Markerbasierte Bewegungserfassung mit Kameras. *Links* Motion Captor-System von Meta Motion (aus: Meta Motion (2006b)). *Rechts*: das Trackingsystem von VICON in der Anwendung (aus: Stein et al. (2006), S. 14)

versehen und mittels mehrerer hochauflösender, schneller Kameras erfasst. Nachteilig sind die hohen Kosten dieser Systeme, der große Kalibrieraufwand, je nach Umgebung und Bewegungen der Versuchsperson Probleme mit Verdeckungen bei der Aufnahme sowie der hohe Berechnungsaufwand für die Verarbeitung, der eine Echtzeitauswertung schwierig macht.

### Magnetfeldbasierte Positionssensoren

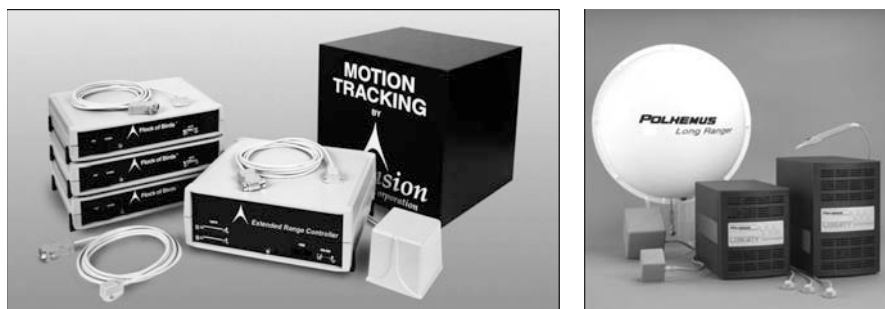


Abbildung 2.9: Magnetfeldbasierte Positionssensoren, *links*: Flock of Birds von Ascension (aus: Ascension (2006)), *rechts*: von Polhemus (aus: Polhemus (2006))

Magnetfeldbasierte Positionssensoren wie der „Flock of Birds“ von Ascension (Ascension (2006)) oder die Systeme von Polhemus (Polhemus (2006)) werden zur Verfolgung von Körper, Kopf und Händen des Benutzers oder von Objekten und zur Bestimmung von deren Position und Orientierung verwendet (vgl. Abb. 2.9). Dabei können sowohl einzelne Sensoren eingesetzt werden als auch ein Ganzkörpertracking, bei dem alle relevanten Gliedmaßen oder Körperteile mit Trackern versehen werden. Problematisch sind dabei die Anfälligkeit dieser Systeme für Störungen z. B. durch metallische Gegenstände sowie der hohe Aufwand für die Kalibrierung.

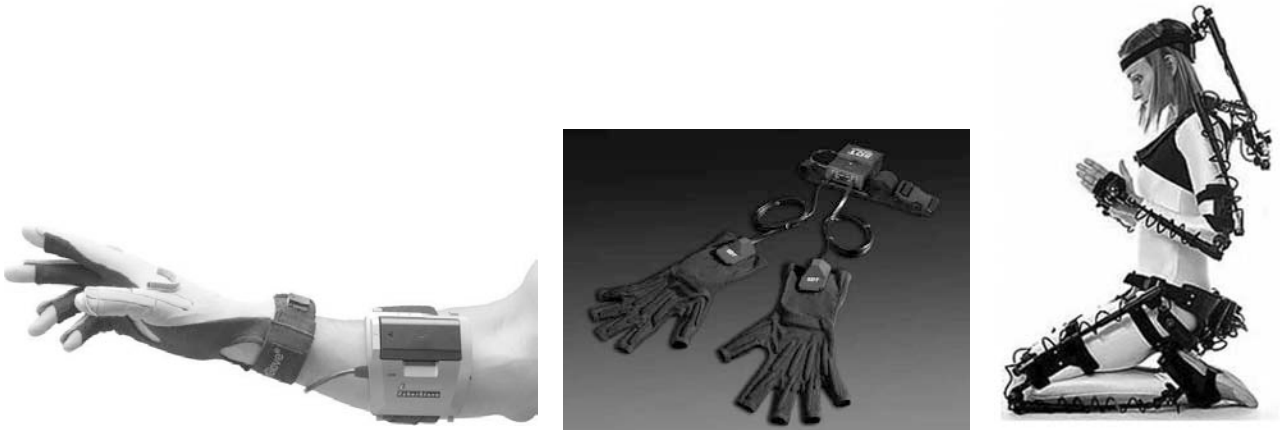


Abbildung 2.10: Datenhandschuhe, *links*: CyberGlove II von Immersion (aus: Immersion (2006)), *Mitte*: Datenhandschuh von Fifth Dimension Technologies (aus: Fifth Dimension Technologies (2006)) und *rechts*: Gypsy-System von Meta Motion als Beispiel für ein Exoskelett (aus: Meta Motion (2006a))

### Datenhandschuhe und Exoskelette

Datenhandschuhe erlauben die Messung von Beugungs- und Spreizwinkeln der Fingergelenke durch Dehnmessstreifen oder Lichtleiter. Beispiele hierfür sind der CyberGlove II von Immersion (2006) oder die Datenhandschuhe der Firma Fifth Dimension Technologies (2006), vgl. Abb. 2.10. Einen guten Überblick über unterschiedliche Datenhandschuh-Systeme und deren technische Daten bietet Mellott (1999). Probleme ergeben sich hier unter anderem durch unterschiedliche Handgeometrien verschiedener Benutzer sowie dadurch, dass durch das Tragen eines Datenhandschuhes möglicherweise der natürliche Bewegungsablauf des Menschen verfälscht wird. Datenhandschuhe erlauben keine Rückschlüsse auf die Position der Hand und werden daher oft in Kombination mit magnetischen Positionssensoren eingesetzt. Die aus einem Datenhandschuh gewonnenen Daten werden, je nach Aufgabenstellung, zur Griffbestimmung oder zur Gestenerkennung genutzt.

Eine Taxonomie von statischen menschlichen Griffen, die zum Standard in der Robotik geworden ist und sowohl zur Griffklassifikation als auch zur Greifplanung verwendet werden kann, schlägt Cutkosky (1989) vor (vgl. Abb. 2.11 links). Es handelt sich hierbei um eine allgemeine hierarchische Einteilung von statischen Griffen, die als solche natürlich nicht nur zur Grifferkennung mittels Datenhandschuh eingesetzt werden kann, sondern auch für andere Formen der Griffklassifikation oder Greifplanung genutzt wird. Als Erweiterung der Arbeit von Cutkosky nimmt Zöllner et al. (2002) eine vergleichbare Einteilung für dynamische Griffe vor, d. h. für Griffe, in deren Verlauf sich die Gelenkwinkelstellungen der Finger ändern (vgl. Abb. 2.11 rechts). Dies ist beispielsweise der Fall beim Öffnen einer Flasche mit Schraubverschluss. Ein Verfahren zur Grifferkennung sowie ein Überblick über den Stand der Forschung findet sich bei Zöllner (2005). Miller et al. (2003) schlagen ein Verfahren zur automatischen Greifplanung vor.

Eine Erweiterung des Prinzips der Datenhandschuhe auf den ganzen Körper bieten Exoskelette wie das Gypsy-System von Meta Motion (Meta Motion (2006a)), vgl. Abb. 2.10. Auch hier ist die Kalibrierung des Anzugs mit Dehnmessstreifen auf die jeweilige Körpergeometrie problematisch.

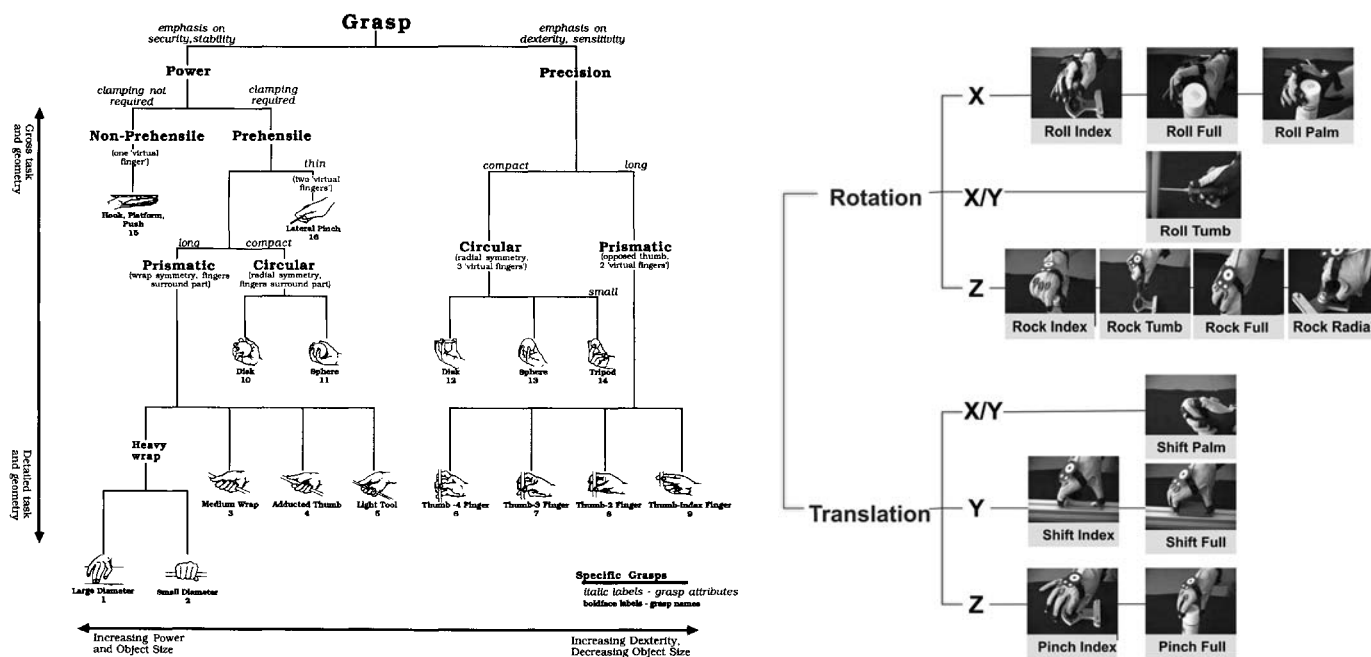


Abbildung 2.11: Links: Taxonomie statischer Griffe aus Cutkosky (1989), S. 273. Rechts: Taxonomie dynamischer Griffe aus Zöllner (2005), S. 80

### Taktile Sensoren

Taktile Sensoren dienen der Erfassung von Berührungen, z. B. von Druck, den der Benutzer auf einen Roboterarm ausübt, um diesen beiseite zu schieben (eine künstliche Haut als taktiles Sensorsystem für einen Serviceroboter stellen u. a. Göger et al. (2006) vor, vgl. Abb.2.12 links). Sie werden auch in Datenhandschuhen eingesetzt, um beispielsweise bei der Verarbeitung von durch den menschlichen Benutzer vorgeführten Griffen Griffansatzpunkte und ausgeübte Kräfte zu erfassen. Der Aufbau von solcherart eingesetzten taktilen Sensoren, ihre Integration in einen Datenhandschuh und die Diskussion der experimentellen Ergebnisse werden in Zöllner et al. (2001) dargestellt (Abb. 2.12 rechts). Hashimoto (2000) stellt einige Entwicklungen im Bereich haptischer Benutzerschnittstellen vor, u. a. Sensorhandschuhe und einen Sensorarm, und nennt außerdem viele weiterführende Verweise zu anderen haptischen Benutzerschnittstellen. In diesem Bereich finden sich auch Arbeiten zur Übermittlung von Informationen von einem Roboter oder anderen technischen System an den menschlichen Benutzer mittels Kraftübertragung, z. B. bei Keyson (1997).

### 2.4.4 Interaktion in Virtueller und Erweiterter Realität

In jüngster Zeit werden zunehmend Techniken der Mensch-Maschine-Interaktion evaluiert, die durch neue technische Entwicklungen erst ermöglicht wurden. Hierzu zählen auch Techniken der Virtuellen Realität (engl. „virtual reality“, VR) und der Erweiterten Realität (engl. „augmented reality“, AR). In Szenarien der Virtuellen Realität bewegt sich ein Mensch völlig in einer virtuellen Umgebung, in der er z. B. die Sicht eines räumlich entfernten Roboters einnehmen und diesen Roboter steuern kann. In AR-Umgebungen werden Elemente aus der virtuellen Welt der realen Welt überlagert. Dies geschieht oft mittels einer Durchsichtbrille,

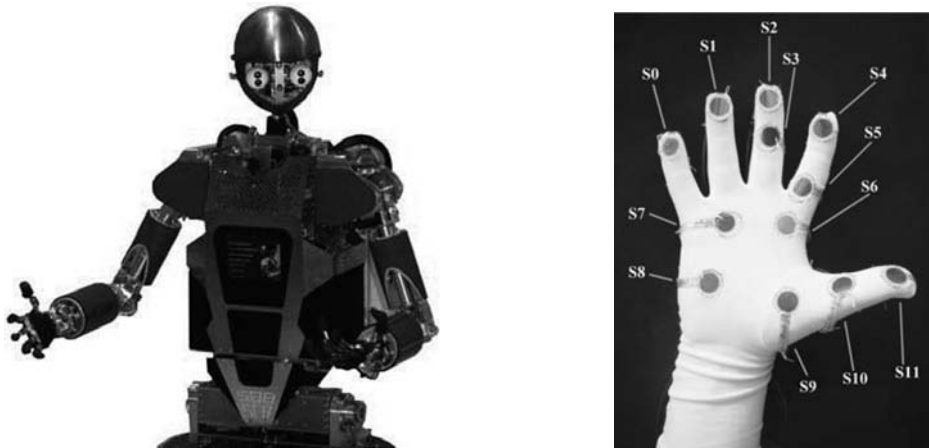


Abbildung 2.12: *Links:* Serviceroboter ARMAR III mit einer künstlichen Haut, aus Göger et al. (2006), S. 6. *Rechts:* taktile Sensoren in einem Datenhandschuh aus Zöllner et al. (2001), S. 69

durch die der Benutzer zwar seine Umgebung noch wahrnehmen kann, in die aber gleichzeitig weitere Informationen eingeblendet werden. Dies wird unter anderem genutzt zur Überlagerung von realen Objekten mit zugehörigen Informationen. Bei beiden Techniken stellen sich jeweils spezifische technische Probleme, auf die hier im Detail nicht eingegangen werden kann (vgl. Salb (2003) und Giesler (2005) für einen Überblick mit Schwerpunkt auf der Erweiterten Realität). Bei der Erweiterten Realität sind maßgebliche Probleme das Tracking der Umgebung, die störungsfreie, korrekt positionierte Überlagerung virtueller und realer Objekte in Echtzeit sowie die Kalibrierung der Überlagerungsoptik.

Da die hier betrachtete Aufgabenstellung die Erstellung von Objektmodellen für reale Umgebungen einschließt, scheint die Verwendung eines VR-Systems zur Benutzerinteraktion nicht sinnvoll, da in VR-Systemen in der Regel keine realen Objekte in realen Umgebungen integriert werden können. Im Folgenden werden daher nur AR-Systeme vorgestellt und bewertet, die dem Benutzer die Modellierung von realen Objekten ermöglichen. Die in diesem Abschnitt vorgestellten Arbeiten überschneiden sich dabei teilweise inhaltlich mit anderen Abschnitten dieses Kapitels, da in der AR diverse andere hier vorgestellte Techniken verwendet werden. Die meisten AR-Systeme lassen sich als Systeme der multimodalen Interaktion bezeichnen. Sie werden hier aufgrund der spezifischen technischen Ausstattung sowie der besonderen Ausrichtung der Interaktion separat diskutiert. Als technische Systeme für die Erzeugung einer erweiterten Realität werden hauptsächlich optische Durchsichtbrillen, Video-Durchsichtbrillen sowie Projektionssysteme verwendet. Technische Details sowie Vor- und Nachteile der unterschiedlichen Techniken werden in Salb (2003) ausführlich dargestellt.

Ein System der Erweiterten Realität zur Interaktion mit einer autonomen mobilen Plattform wird in Giesler et al. (2004) und Giesler (2005) vorgestellt. Dabei wird eine optische Durchsichtbrille genutzt, um die topologischen und geometrischen Daten der mobilen Plattform, Laserscanner-Daten und Planungsdaten zu visualisieren. Der Benutzer kann dann interaktiv eine topologische Karte der Umgebung generieren, indem er umherläuft und auf Punkte auf dem Boden zeigt. Die Positions- und Lagebestimmung erfolgt markerbasiert. Um die Zeigegesten und Manipulationsaktionen des Benutzers intuitiv zu gestalten, wird ein Cursor auf dem Zielpunkt eingeblendet, wann immer der Benutzer auf den Boden zeigt. Die Gesten



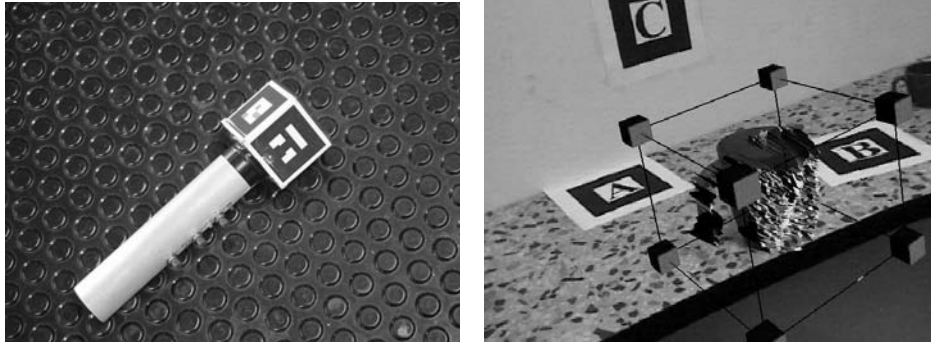


Abbildung 2.13: *Links*: Zeigestab mit Markern aus Giesler (2005), S. 105. *Rechts*: Ergebnis der interaktiven Objektmodellierung mittels Erweiterter Realität in Giesler (2005), S. 147

werden vom Benutzer mit einem Zeigestab mit aufgebrachtten Markern ausgeführt, um die Richtungsbestimmung zu erleichtern (vgl. Abb. 2.13).

Im Rahmen der vorliegenden Arbeit interessant ist auch die Erweiterung des Einsatzgebietes dieses Systems unter anderem auf interaktive Objektmodellierung aus 3D-Messdaten in Giesler (2005). Dem Benutzer wird hierzu eine dreidimensionale Punktwolke der Umgebung immersiv zur Verfügung gestellt. Der Benutzer kann Umweltobjekte durch Zeigen auswählen, wobei sich die automatische Segmentierung der Objekte sowie die Vollständigkeit der resultierenden Modelle als problematisch darstellen. Daher kann der Benutzer das Segmentierungsergebnis interaktiv verbessern. Dabei werden unterschiedliche Interaktionsmodi verwendet: die Erkennung der Benutzerhand über eine Hautfarbsegmentierung, die Erkennung von Zeige- und Griffgesten, soweit möglich wahlweise mit der Hand und mit dem vermarkerten Zeigestab ausgeführt, wobei die Detektion der Zeigerichtung von Zeigegesten über eine Stereorekonstruktion erfolgt, die Detektion von Griffgesten analog bei gekrümmten Fingern. Der Aufmerksamkeitsfokus des Benutzers wird aus dessen Blickrichtung abgeleitet. Da eine okulomotorische Beobachtung der Pupillen wegen der verwendeten Durchsichtbrille nicht realisierbar ist, erfolgt nur eine Lageschätzung des Kopfes, der sich ergebende Sichtkegel wird dann analog zu den Zeigegesten verarbeitet. Eine Spracherkennung ist als Einzelworterkenner zur Erkennung von festen Kommandophrasen eingebunden. Ein Ergebnis eines solchen Modellierungsprozesses ist in Abb. 2.13 wiedergegeben. Die Abbildung zeigt ein Kamerabild aus der Durchsichtbrille, in dem das ausgewählte Objekt durch einen virtuellen Quader markiert wurde. Dem Objekt wurde das Ergebnis einer Objektaufnahme mittels eines dreidimensionalen Laserscanners überlagert.

In Wachsmuth et al. (2005) wird eine Zuordnung von systeminternen Beschreibungen wie Objektmodellen zu in der realen Welt existierenden Objekten angestrebt. Es werden hier also nicht wie sonst häufig a priori per CAD oder mittels hochauflösender Sensorik erstellte Objektmodelle zur Interaktion verwendet, sondern die interaktive Erstellung von Objektmodellen ist das Ziel der Arbeit. Auch hier trägt der Benutzer wie bei Giesler ein am Kopf befestigtes AR-System, die Kopfpose wird ebenfalls markerbasiert bestimmt. Das System geht derzeit davon aus, dass alle von einem Benutzer präsentierten Objekte auf einer Tischoberfläche liegen. Gelernt werden Objekte, die der Benutzer fokussiert, und zur Kontrolle werden die extrahierten Objektansichten dem Benutzer in der AR-Brille präsentiert. Vom System aufgenommene Objektansichten können dann gespeichert werden, um das neu gelernte Objekt künftig ansichtsbasiert zu erkennen. Die Objektmodelle in diesem System

beschränken sich auf reine Objektansichten, und die potentiellen Interaktionsmodalitäten einer AR-Umgebung werden bei weitem nicht ausgenutzt.

Ein Nachteil dieser und ähnlicher Systeme ist der vergleichsweise hohe technische Aufwand, der zu hohen Kosten, praktischen Problemen bei der Integration der technischen Komponenten sowie zu Anpassungs- und Verwendungsproblemen bei Nutzern führen kann. In der Literatur wird unter anderem von Schwindel durch das Tragen von Durchsichtbrillen berichtet (Steinfeld (2004)), insbesondere wenn die Überlagerung der realen mit der virtuellen Realität zeitverzögert oder leicht verschoben stattfindet sowie bei längerem Tragen.

### 2.4.5 Multimodale Interaktion

Ansätze zu einer multimodalen Interaktion zur Modellierung finden sich in unterschiedlichen Varianten, die anhand von Beispielen vorgestellt werden sollen. Unter anderem sind auch Systeme der Interaktion in Erweiterter Realität in der Regel multimodal angelegt, so auch das oben beschriebene System von Giesler (2005). Auch im Programmieren durch Vormachen wird oft multimodale Interaktion eingesetzt, z. B. bei Dillmann et al. (2002) zur Programmierung von Robotern durch Vorführen von Handlungen. Dabei werden folgende Interaktionsmodi eingesetzt: eine kamerabasierte Handverfolgung mittels Hautfarbensegmentierung, eine Gestenklassifikation aus Handkonturen (für einfache symbolische Gesten wie die Stop-Geste, Zeigegesten, Greifgesten), eine Spracheingabe mit einfachem Parsing, ein Datenhandschuh mit taktilen Sensoren an den Fingerspitzen sowie ein magnetfeldbasiertes Trackingsystem zur Positionsbestimmung der menschlichen Hand und zusätzlich Informationen aus dem Stereo-Kamerasystem, um Rauschen in den Positionsdaten zu verringern. Aus diesen Eingaben werden Trajektorien und Griffe segmentiert und dann statische (mittels Neuronaler Netze) und dynamische (mittels Support Vector Machines) Griffe erkannt. Zusätzlich ist ein graphisches Interface in das System eingebunden zur Darstellung von Systeminformationen für den Benutzer, und um dem Menschen Korrekturen und die Eingabe zusätzlicher Informationen zu erlauben. Abb. 2.14 stellt eine Interaktion zwischen Mensch und Serviceroboter mittels Spracheingabe und Zeigegesten dar.



Abbildung 2.14: Sprachliche und gestische Interaktion zwischen Mensch und Roboter bei Dillmann et al. (aus: Dillmann et al. (2002), S. 1)

Ein ähnliches Ziel verfolgen Haasch et al. (2004) und Li et al. (2005) mit dem Serviceroboter BIRON, der als Helfer im Haushalt und ähnlichen Umgebungen eingesetzt werden soll. Auf BIRON werden unterschiedliche Wege der Interaktion eingesetzt wie Personenverfolgung, Dialogmanagement mit Spracherkennungs- und Sprachverstehenskomponente sowie Erkennung von Zeigegesten. Perzanowski et al. (2001) konzentriert sich zur Interaktion mit mobilen Robotern wie einige andere Arbeiten auch auf die Integration von Sprache und Gesten, wobei damit in dieser Arbeit sowohl natürliche Gesten gemeint sind als auch Daten, die in einen Personal Digital Assistant, PDA, eingegeben werden. Betrachtet werden nur Gesten, die Größen andeuten, und Zeigegesten, und die Information hieraus wird mit sprachlich gegebener Information kombiniert, falls z. B. in einer sprachlichen Äußerungen ein Ort fehlt. Die Rückmeldung der Roboter an die Benutzer erfolgt durch Darstellung auf dem PDA, dessen Bildschirmdarstellung dynamisch als adaptive Karte der jeweiligen Umgebung angepasst wird. Abb. 2.15 stellt schematisch das Zusammenspiel der verschiedenen Interaktionskomponenten in diesem System dar.

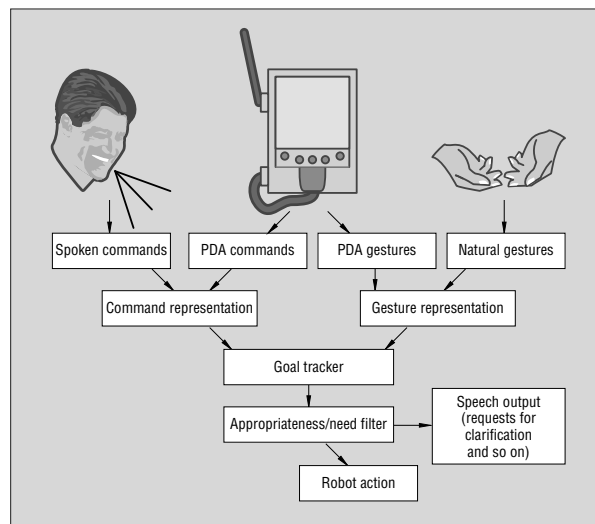


Abbildung 2.15: Schematische Darstellung der multimodalen Benutzerschnittstelle bei Perzanowski et al. (aus: Perzanowski et al. (2001), S. 18)

Konzipiert für eine multimodale Interaktion, insbesondere Kommandierung, ist auch der so genannte XWand von Wilson und Shafer (Wilson und Shafer (2003)). Es handelt sich um einen kabellosen, mit diversen Sensoren bestückten Stab zur intuitiven Steuerung z. B. von mehreren Unterhaltungselektronik-Geräten. Als eingebaute Sensoren dienen Beschleunigungssensoren, ein Gyroskop und ein Magnetometer, sowie zusätzlich zwei fest installierte Kameras im Raum (zur Positionsbestimmung über Aufnahme einer blinkenden LED). Damit kann die Orientierung des Stabes, seine Position (mit Hilfe der beiden Kameras) sowie die Richtung von Zeigegesten sensorisch erfasst werden. Hierdurch soll eine intuitive Interaktion ermöglicht werden mittels Zeigegesten, mittels anderer Gesten, bei denen der Stab bewegt wird, und durch das Drücken eines Knopfes am Stab. Die Autoren erwähnen die potentielle Kombination mit einer Spracherkennungskomponente, um robustere und intuitive Ergebnisse zu ermöglichen.

Insgesamt lässt sich festhalten, dass sich hinter der Bezeichnung „multimodale Interaktion“ oft die Interaktionsmodi Sprache und Gestik verbergen, wobei hier in vielen Fällen auch nur

Zeigegesten betrachtet werden, wie bei Wachsmuth und Sagerer (1999) oder Takahashi et al. (1998). Letztere schlagen zudem noch einen vornehmlich robotergesteuerten Dialog vor, der unter ungünstigen Umständen viele Wiederholungen von Seiten des Menschen verlangt, bis dem Roboter alle gewünschten Informationen vorliegen. Eine etwas elaboriertere Vorgehensweise schlagen Iba et al. (2005) vor. Dort werden außer Zeigegesten auch symbolische Gesten betrachtet, zusätzlich dienen Sprache und ein Datenhandschuh als Eingabequellen.

## 2.5 Bewertung der Arbeiten im Hinblick auf die gegebene Problemstellung

Wie in den vorhergehenden Abschnitten bereits deutlich wurde, berücksichtigen bisherige Arbeiten jeweils schwerpunktmäßig entweder die Mensch-Maschine-Interaktion, die Objektmodellierung oder das Lernen von Objekten. Es existiert somit noch kein umfassender Ansatz, der den gestellten Anforderungen gerecht wird. Auch innerhalb der einzelnen Forschungsbereiche sind die Arbeiten teilweise stark spezialisiert, z. B. auf bestimmte Interaktionsmodi. Diese Konzentration auf wenige Problemfelder wird den Bedürfnissen für eine interaktive semantische Objektmodellierung für reale Anwendungen in menschenzentrierten Umgebungen nicht gerecht. Insofern lassen sich aus den vorgestellten Arbeiten zwar einzelne interessante Ansätze für Teilprobleme der gegebenen Fragestellung isolieren, ein direkter Vergleich der Ergebnisse der vorliegenden Arbeit mit deren Forschungsergebnissen ist aber aufgrund der unterschiedlichen Problemstellungen und Herangehensweisen nur eingeschränkt möglich.

## 2.6 Zusammenfassung

Das vorliegende Kapitel befasst sich mit dem Stand der Forschung auf dem Gebiet der interaktiven semantischen Objektmodellierung und stellt dabei aktuelle Forschungsarbeiten aus diesem Gebiet vor und diskutiert sie kritisch im Hinblick auf die gegebene, in Kap. 1.2 erläuterte Problemstellung. Die existierenden Arbeiten setzen unterschiedliche Schwerpunkte: zum einen werden Ansätze zu einer semantischen oder funktionalen Objektmodellierung betrachtet, dann die Frage des Lernens von Objektmodellen und schließlich Arbeiten zur interaktiven Modellierung. Dabei werden auch jeweils Forschungsarbeiten aus verwandten Gebieten wie der Handlungsmodellierung einbezogen, insoweit sie einen Erkenntnisgewinn für diese Arbeit mit sich bringen. Es zeigt sich, dass kein Ansatz existiert, der das Problem umfassend und systematisch zu lösen versucht, und insbesondere existiert keine Arbeit, die sich gleichzeitig mit den Fragen einer semantischen Modellierung von Objekten, des Lernens solcher Objektmodelle und der Interaktion mit dem Menschen zum Zwecke des Lernens befasst. In vielen Punkten enthalten die vorgestellten Arbeiten jeweils nur Anregungen für einzelne Teilprobleme der gegebenen Forschungsproblematik, die einer vollständigen systematischen Betrachtung nicht gerecht werden.

## Kapitel 3

# Semantische Objektrepräsentation

Die vorliegende Arbeit leistet auf mehreren Gebieten Beiträge zur Forschung im Bereich interaktive Objektmodellierung: zum einen durch den Entwurf einer semantischen Objektrepräsentation im vorliegenden Kapitel, zum anderen durch den Entwurf und die Realisierung eines interaktiven Objektmodellierungssystems (vgl. die folgenden Kapitel) sowie die Erstellung einer Taxonomie von Begrifflichkeiten für die Domäne „Küche“ (vgl. Kap. 6). Die vorgeschlagene abstrakte Objektrepräsentation ist zunächst nicht spezifisch für eine konkrete Domäne, für eine bestimmte Anwendung oder für die Sensorik eines speziellen Objektmodellierungssystems. Insbesondere ist das grundlegende Konzept unabhängig von der Domäne „Küche“, die in dieser Arbeit exemplarisch betrachtet wird, und von dem ebenfalls im Rahmen dieser Arbeit entwickelten interaktiven Objektmodellierungssystem IOMOS. In letzterem wird die in diesem Kapitel vorgestellte Objektrepräsentation implementiert und praktisch angewendet, um ihre Nutzbarkeit zu zeigen. Trotzdem fließen in den Entwurf einer solchen Objektrepräsentation grundsätzliche Überlegungen und Anforderungen mit ein, die hier konkret aus dem Anwendungsszenario „Serviceroboter im Haushalt“ abgeleitet wurden. Somit ist die Objektrepräsentation zwar flexibel und auf andere Anwendungen übertragbar, eignet sich aber besonders für Anwendungen mit ähnlichen Grundbedingungen, also für menschenzentrierte, dynamische Umgebungen mit einer starken Interaktionskomponente zwischen Mensch und Roboter. Weitere potentielle Anwendungen sind beispielsweise Serviceroboter auf Flughäfen oder Bahnhöfen, in Hotels und in ähnlichen Umgebungen, die auf den Menschen ausgerichtet sind und in denen die Anforderungen der Benutzer sowie die zu modellierenden Objekte stark variieren können.

Es wird im Folgenden von der Anforderung ausgegangen, dass ein Serviceroboter z. B. im Haushalt ein Objektmodell benötigt, um seine Aufgaben ausführen zu können. Dieses Objektmodell muss flexibel erweiterbar sein, möglichst auch in Interaktion mit menschlichen Benutzern, um eine dynamische Umgebung wie den Haushalt abzudecken, dessen große Menge an möglicherweise vorhandenen Objekten nicht im Vorhinein modelliert werden kann (vgl. Diskussion in Kap. 1.3). Der Entwurf der Objektrepräsentation erfolgt zunächst auf einer abstrakten Ebene, so dass sie leicht auch auf andere Anwendungen übertragbar ist. Konkretes domänen- bzw. anforderungsspezifisches Wissen kann jederzeit ergänzt werden, und die Konzentration auf eine flexible Modellierung zur intuitiven Interaktion mit dem Menschen macht die vorgeschlagene Objektrepräsentation in vielen Bereichen einsetzbar.

In diesem Kapitel werden zunächst in Abschnitt 3.1 die Verwendung grundlegender Begriffe im Rahmen dieser Arbeit geklärt und der Begriff der „semantischen Objektrepräsentation“ definiert sowie in Abschnitt 3.2 die Anforderungen an eine solche Repräsentation diskutiert. Die Objektrepräsentation selbst wird in Abschnitt 3.3 detailliert vorgestellt. Eine Diskussion der vorgestellten Repräsentation schließt das Kapitel ab.

## 3.1 Begriffsklärung

Die Begriffe „Repräsentation“, „Modell“ und „Semantik“ haben eine lange Verwendungsgeschichte in der abendländischen Wissenschaft. Dabei wurden und werden sie oft in unterschiedlichen Bedeutungen verwendet, selbst innerhalb eines Fachbereiches. Ein vollständiger Überblick über die Verwendungs- und Begriffsgeschichte kann im Rahmen dieser Arbeit nicht geleistet werden. Auch ist es nicht Ziel dieser Arbeit, eine allgemeingültige Definition dieser Begriffe anzugeben. Vielmehr soll in diesem Kapitel die Verwendung dieser grundlegenden Begriffe *im Rahmen dieser Arbeit* geklärt werden, ohne dabei in Diskussionen über eine korrekte oder falsche Bedeutungsdefinition einzutreten (vgl. Strube et al. (1996) für einen ersten Überblick über Gebrauchsvarianten und Bedeutungsnuancen dieser Begriffe). Aus dieser Begriffsklärung werden dann die Definitionen für „semantische Objektrepräsentation“ und „semantisches Objektmodell“ abgeleitet, wie sie für die vorliegende Arbeit Gültigkeit haben.

### Modell

Der Begriff „Modell“ bezeichnet in der Literatur sehr unterschiedliche Konzepte (physikalische Modelle wie das Planetenmodell, architektonische Modelle von Gebäuden etc.) und wird oft ohne weitere Definition in einem vagen umgangssprachlichen Verständnis verwendet (zur Kritik hieran vgl. Brezinka (1984)). Müller (1977) gibt eine kurze Einführung in unterschiedliche Definitionen und Aspekte des Begriffs „Modell“. Die Begriffsgeschichte des Begriffs „Modell“ wird in Müller (2006b) umfassend dargestellt, die Bedeutungsvielfalt des Begriffes in Müller (2006a).

Allgemein werden Modelle oft als vereinfachtes Abbild der Wirklichkeit betrachtet. Eine weitere Komponente der Begriffsbedeutung, die Einschränkung des Modells durch bestimmte Anforderungen oder Ziele, wird unter anderem von Emde eingeführt: „Allgemein werden hier Modelle als an bestimmten Zielen ausgerichtete (formalisierte) Darstellungen von Welt-ausschnitten betrachtet.“ (Emde (1991), S. 64). Eine gründliche Untersuchung des Modellbegriffs nimmt Stachowiak in seiner Allgemeinen Modelltheorie vor. Die vorliegende Arbeit orientiert sich in ihrem Verständnis des Modellbegriffs an dem von Stachowiak, der drei Hauptmerkmale des allgemeinen Modellbegriffs unterscheidet:

**„Abbildungsmerkmal:**

Modelle sind stets Modelle *von etwas*, nämlich Abbildungen, Repräsentationen natürlicher oder künstlicher Originale, die selbst wieder Modelle sein können. [...]

**Verkürzungsmerkmal:**

Modelle erfassen im allgemeinen *nicht alle* Attribute des durch sie repräsentierten Originals, sondern nur solche, die den jeweiligen Modellerschaffern und/oder

Modellbenutzern relevant scheinen. [...]

**Pragmatisches Merkmal:**

Modelle sind ihren Originalen nicht per se eindeutig zugeordnet. Sie erfüllen ihre Ersetzungsfunktion a) für *bestimmte* – erkennende und/oder handelnde, modellbenutzende – *Subjekte*, b) innerhalb *bestimmter Zeitintervalle* und c) unter Einschränkung auf *bestimmte gedankliche oder tatsächliche Operationen*.“ (Stachowiak (1973), S. 131f.).

Bei den abzubildenden Originalen im Stachowiakschen Sinne handelt es sich um Objekte, die „durch natürliche oder maschinelle Informationsverarbeiter perzipiert [werden können]“ (Stachowiak (1973), S. 131), im Rahmen der vorliegenden Arbeit um Haushaltsgegenstände, die durch einen Serviceroboter oder seine menschlichen Benutzer wahrgenommen werden. Die entwickelten Objektmodelle bilden die Wirklichkeit dann insofern verkürzt ab, als eine Konzentration auf eine bestimmte Domäne bzw. vorgegebene Anwendungen erfolgt, wie in Kap. 6 auf die Domäne „Küche“ bzw. auf konkrete Aufgaben, die ein Serviceroboter in der Küche auszuführen hat. Diese Verkürzung nimmt einen Teil des pragmatischen Merkmals von Modellen bereits vorweg: Die solcherart erfolgende Abbildung von Objekten auf Modelle ergibt natürlich nur Sinn für bestimmte Subjekte (unter Bezug auf die Fähigkeiten eines bestimmten Roboters bzw. die Bedürfnisse eines bestimmten Benutzers), innerhalb bestimmter Zeitintervalle (die Ontologie und damit auch die Objektmodelle werden entworfen im Hinblick auf die Möglichkeiten in einer heutigen Küche) sowie unter Betrachtung bestimmter vorgegebener Anwendungen. Eine allgemeine Darstellung der Domäne „Küche“ als Ontologie bzw. in Objektmodellen wäre in diesem Sinne überhaupt nicht erstellbar. Stachowiak fasst dies so zusammen: „Eine pragmatisch vollständige Bestimmung des Modellbegriffs hat nicht nur die Frage zu berücksichtigen, *wovon* etwas Modell ist, sondern auch, *für wen*, *wann* und *wozu* bezüglich seiner je spezifischen Funktionen es Modell ist.“ (Stachowiak (1973), S. 133). Die Beantwortung dieser Fragen im Rahmen der vorliegenden Arbeit erfolgt hauptsächlich in Kap. 6, in dem eine Taxonomie von Begrifflichkeiten der Domäne „Küche“ zur Anwendung für einen Serviceroboter erstellt wird. Diese Domänenspezifikation ist die direkte Grundlage für die Objektmodelle, die im Rahmen dieser Arbeit interaktiv erstellt wurden.

## Repräsentation

Die Begriffsgeschichte des Begriffs „Repräsentation“ stellt Scheerer (1993) dar, einen Überblick über Verwendungsweisen des Begriffs in verschiedenen Fachdisziplinen der Kognitionswissenschaft Strube et al. (1996). Insbesondere wird der Begriff „Repräsentation“ oft verkürzend gebraucht für „mentale Repräsentation“, d. h. für eine mentale Abbildung der Wirklichkeit. Eine nützliche Begriffsdefinition findet sich in Scheerer (1993):

„Dennoch: die kognitive Wende hat auch Neues gebracht, nämlich einen interdisziplinär anwendbaren Begriff der mentalen Repräsentation. Er stammte – man kann dies durch Detailanalysen der terminologischen und sachlichen Transmissionslinien eindeutig nachweisen – aus der „Informatik“, d. h. der Computerwissenschaft und hier speziell aus der Künstlichen Intelligenz (KI)-Forschung. [...]

Man kann also, trägt man das klassische Bedeutungsspektrum von ‘Repräsentation’ an die Repräsentation in KI-Systemen heran, folgendes Schema konstatieren: interne Repräsentationen sind Stellvertreter ([...]), deren Eigenschaft als Stellvertreter durch strukturhaltende Abbildung ([...]) gewährleistet ist und denen semantischer Gehalt beigelegt wird ([...]).“ (Scheerer (1993), S. 35f).

Wesentliche Bestandteile einer Repräsentation nach dieser Definition sind somit ihre strukturhaltende Abbildung sowie deren semantischer Gehalt. Generell ist festzuhalten, dass in der Literatur nicht immer zwischen Repräsentation und Modell getrennt wird, vgl. beispielsweise das oben zitierte *Abbildungsmerkmal* aus Stachowiak (1973). Da es nicht Gegenstand dieser Arbeit ist, eine allgemeingültige Definition der beiden Begriffe zu treffen, wird hier eine Festlegung getroffen, die für den Rahmen dieser Arbeit Gültigkeit hat und dabei einerseits die Ähnlichkeit zwischen diesen Begriffen ausnutzt, andererseits aber durch die beiden unterschiedlichen Begriffe unterschiedliche Aspekte der Problemstellung betont:

Um zwischen dem abstrakten Modellentwurf an sich (unabhängig von konkreten Objekten), wie er in diesem Kapitel vorgenommen wird, und den eigentlichen Objektmodellen, d. h. den Modellen konkreter Objekte (vgl. Kap 6 oder 7), zu unterscheiden, wird mit dem Begriff *Repräsentation* oder *Objektrepräsentation* im Verlauf dieser Arbeit der grundlegende abstrakte Entwurf der Struktur des Objektmodells bezeichnet unabhängig von konkreten Objekten oder Anwendungen, wie er in Kap 3.3 entwickelt wird. Hierbei sind die Eigenschaften der Repräsentation als Stellvertreter durch strukturhaltende Abbildung und ihr semantischer Gehalt (nach der Definition von Scheerer (1993)) gegeben.

Der Begriff *Repräsentation* beschreibt im Rahmen dieser Arbeit die grundlegenden Strukturen und Informationseinheiten des vorgeschlagenen Objektmodells (also das unten erläuterte Konzept der Eigenschaften, Attribute etc.); als *Modelle* bzw. *Objektmodelle* werden die Darstellungen spezifischer Objekte bzw. Objektklassen in Bezug auf eine spezielle Anwendung unter Verwendung dieser Repräsentation bezeichnet. Der Begriff *Modellierung* bezeichnet dann den Prozess der Erstellung eines Modells. Beim Prozess der Modellierung erfolgt eine Reduktion der Objektinformation auf für die jeweilige Domäne und Anwendung wesentliche Information.

## Semantik

Die Semantik als Lehre von den Bedeutungen hat in der Wissenschaft eine lange Tradition zunächst im Sinne von Bedeutungen sprachlicher Zeichen, in jüngerer Zeit aber zunehmend generell im Sinne der Abbildung der Bedeutung eines Bezeichneten durch ein (nicht notwendigerweise sprachliches) Zeichen, womit auch eine immer häufigere Verwendung dieses Begriffs in anderen Fachbereichen als der Linguistik einhergeht. Die einzelnen Verwendungsweisen sollen hier nicht weiter diskutiert werden. Im Rahmen der vorliegenden Arbeit genügt ein intuitives Verständnis des Begriffs „semantisch“ als die Bedeutung einer Sache im Verständnis des Menschen. Diese Bedeutung kann dabei immer nur personen- und kontextspezifisch festgelegt werden. Die bei Löbner (2003) für den Bereich der linguistischen Semantik erläuterten Begriffe „Äußerungskontext“ und „Äußerungsbedeutung“ entsprechen diesem Konzept. Der Äußerungskontext ist dabei „die Gesamtheit aller Gegebenheiten, die



sich auf die Referenz und Wahrheit einer Äußerung auswirken“ (Löbner (2003), S. 10), z. B. Sprecher, Adressat, Zeitpunkt, Ort oder gegebene relevante Fakten zum Zeitpunkt der Äußerung, die Äußerungsbedeutung ist dann definiert als „Bedeutung eines Ausdrucks, die sich aus seiner Verwendung und Interpretation in einem gegebenen Äußerungskontext ergibt“ (Löbner (2003), S. 11). Die Erweiterung des Semantikbegriffs auch auf nicht-sprachliche Ausdrücke ergibt dann direkt die oben angedeutete Personen- und Kontextabhängigkeit von Bedeutungen. Das menschliche Verständnis der Dinge wird als Grundlage für ihre Bedeutung hier besonders hervorgehoben, da ein Hauptziel dieser Arbeit die interaktive Modellierung von Objekten (gemeinsam mit dem Menschen) ist. Eine Semantik der Objekte hat sich daher am menschlichen Verständnis dieser Objekte zu orientieren, um hierfür einsetzbar und nützlich zu sein. Daher ist auch im Folgenden mit „Semantik“ nicht formale Semantik im informatischen Sinne gemeint, sondern die nicht notwendigerweise formalisierte Ebene der Bedeutungen im Alltagsverständnis.

### Semantische Objektmodellierung bzw. Objektrepräsentation

Auch der Begriff eines „semantischen Modells“ wird in der Literatur für unterschiedliche Inhalte verwendet. So bezeichnet beispielsweise beim Datenbankentwurf oder Softwareentwurf das semantische Modell den implementierungsabhängigen konzeptuellen Entwurf z. B. in Form eines Entity-Relationship-Diagramms. Im Rahmen der vorliegenden Arbeit wird der Begriff „Semantische Objektrepräsentation“ ausgehend von den obigen Überlegungen wie folgt verwendet:

**Definition 3.1 (Semantische Objektrepräsentation)** Eine semantische Objektrepräsentation erlaubt die strukturerhaltende Abbildung von Objekten mit ihrem für den Menschen in Abhängigkeit von ihrer Verwendung wesentlichen semantischen Gehalt unter Darstellung funktionaler Aspekte. Als abstrakte Informationsstruktur von semantischen Objektmodellen bildet sie somit die Grundlage für die Modellierung von Objekten in der Welt.

Als „Semantisches Objektmodell“ wird hier dementsprechend bezeichnet:

**Definition 3.2 (Semantisches Objektmodell)** Modell eines konkreten Objektes bzw. Objekttyps einer Domäne im Hinblick auf eine bestimmte Anwendung. Dem semantischen Objektmodell liegt strukturell eine semantische Objektrepräsentation zugrunde.

Im Folgenden ist der Einfachheit halber oft von „Objektmodell“ die Rede, hierunter wird im Rahmen dieser Arbeit stets ein semantisches Objektmodell verstanden, falls nichts anderes vermerkt ist. Die Begriffe „Semantische Objektmodellierung“ oder kurz „Objektmodellierung“ bezeichnen dann den Vorgang der Erstellung solcher Objektmodelle.

Die Beziehungen zwischen semantischer Objektrepräsentation und den eigentlichen Objektmodellen werden zur Verdeutlichung noch einmal in Abb. 3.1 dargestellt: Die semantische Objektrepräsentation, die im folgenden vorgestellt wird, dient als Grundstruktur für die Erstellung von semantischen Objektmodellen. Hierzu werden zuerst anwendungsspezifisch Modelle von Objektklassen auf der Basis dieser Repräsentation gebildet (für abstrakte Objektklassen wie „Geschirr“ oder konkrete Klassen wie ein Weinglas einer bestimmten Serie).

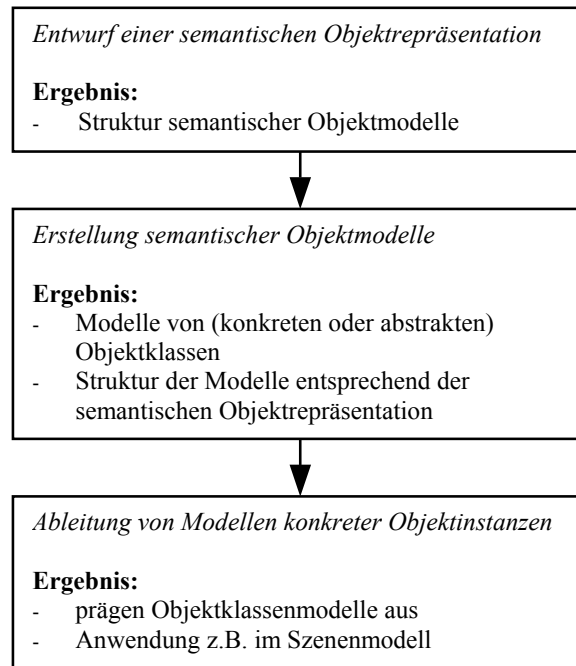


Abbildung 3.1: Ableitung von Modellen konkreter Objekte aus der Objektrepräsentation

Aus diesen Objektklassen werden dann nach Bedarf Modelle von Objektinstanzen ausgeprägt, die in der Welt existieren (vgl. die Erläuterungen in Kap. 4.4 zur Implementierung der Objektrepräsentation).

## 3.2 Anforderungen an die semantische Objektrepräsentation

Wie in Abschnitt 1.3 bereits diskutiert, ist es für Anwendungen wie den Einsatz eines Service-roboters in stark menschenzentrierten Umgebungen wie einem Haushalt unabdingbar, eine semantische Objektrepräsentation (nach Definition 3.1) zu verwenden. Insbesondere ergeben sich dabei folgende Anforderungen an die zu entwickelnde Objektrepräsentation:

### Verständlichkeit für den Menschen

Da in Alltagsumgebungen eine Kommunikation des Roboters vorwiegend mit nicht technisch vorgebildeten Benutzern stattfindet, sollten die Struktur und Inhalte der eingesetzten Objektmodelle für Menschen intuitiv erfassbar und verständlich sein, die Objektrepräsentation somit eine entsprechende menschenverständliche Darstellung unterstützen. Sinnvoll ist es somit, sich an menschlichen Kategorien und Repräsentationsformen zu orientieren, um den Benutzern einfach und schnell einen Einblick z. B. in den aktuellen Szenengraphen des Roboters zu ermöglichen oder Objektinformationen zur Korrektur und Ergänzung durch den Benutzer darzustellen. Dies erleichtert nicht nur die Kommunikation zwischen Mensch und Maschine und die Kommunizierbarkeit von Objektinformationen zwischen System und Benutzer, sondern auch die Erstellung von Benutzerschnittstellen, die Darstellung interner Zustände des Roboters etc.

### Abbildung funktionaler Objekteigenschaften

In einer Arbeitsumgebung wie dem Haushalt genügt es nicht, einige „statische“ Attribute von Objekten, z. B. deren Farbe, zu modellieren. Es ist vielmehr notwendig, Objekte in Kombination mit ihren potentiellen Rollen in Handlungen zu sehen, um Aufgaben kompetent lösen zu können. So kann eine Tasse ein Behälter sein, in dem Flüssigkeit transportiert werden kann. In Ermangelung einer geeigneten Gießkanne kann eine Tasse somit aber durchaus auch genutzt werden, um Blumen zu gießen. Eine einfache Zuweisung der Gießkanne zur Aufgabe „Blumen gießen“ genügt daher nicht, um mit den vielfältigen Anforderungen des menschlichen Alltags umzugehen. Vielmehr erwartet man von einem helfenden Robotersystem auch ein gewisses Mindestmaß an Problemlösungswissen, damit es auch in neuen oder ungewohnten Situationen und bei bisher noch unbekanntem Aufträgen kompetent handeln kann. Im obigen Beispiel ist es somit von Vorteil, grundsätzliche funktionale Eigenschaften von Gegenständen (z. B. „ist ein Behälter“) zu modellieren. Gerade in solchen Alltagssituationen ist eine Objektmodellierung, die anwendungsspezifisches funktionales Objektwissen nicht abbildet, kaum nutzbar. Die Modellierung von funktionalen Objekteigenschaften erlaubt dann auch eine Verknüpfung von Objekt- und Handlungswissen zur Handlungsplanung bei der Erledigung von Aufgaben.<sup>1</sup>

### Erweiterbarkeit und Flexibilität

In komplexen, dynamischen Umgebungen ist es nicht möglich, alle notwendigen Objektinformationen bereits im Vorhinein zu modellieren. Es werden sich im Gegenteil immer wieder Informationen ergeben, die neu in ein Objektmodell aufgenommen werden oder verändert werden müssen, oder neu zu modellierende Objekte werden in die Szene eingeführt. Um diesen Umständen Rechnung zu tragen, ist es wichtig, dass die Objektmodelle erweiterbar und flexibel sind. Insbesondere sollten die zu modellierenden Objektinformationen nicht unnötig eingeschränkt werden, und die Objektrepräsentation sollte möglichst unterschiedliche Arten von Informationen zulassen, um für verschiedenste Anforderungen nutzbar zu sein.

### Formalisierbarkeit

Da die resultierenden Objektmodelle maschinell verarbeitet und zur Objekterkennung und -detektion, Szenenanalyse oder Handlungsplanung genutzt werden, müssen sie von den eingesetzten Robotern bzw. anderen maschinellen Systemen verarbeitet werden können. Eine sinnvolle Formalisierbarkeit ist daher eine unabdingbare Voraussetzung für eine in technischen Systemen einsetzbare semantische Objektrepräsentation.

### Abbildung von Objektmodellen auf Sensordaten und umgekehrt

Das Problem einer Verknüpfung von Symbolen, hier insbesondere Repräsentationen von Objekten, mit Sensordaten (engl.: „anchoring symbols to sensor data“) als Teilaspekt des „symbol grounding problem“ (also der Frage, wie Symbole mit der Welt verknüpft werden und daher ihre Bedeutung erhalten) wird in der Forschung seit langem diskutiert, ist aber immer noch nicht umfassend gelöst (Ansätze hierzu finden

---

<sup>1</sup>Zum Stand der Forschung bezüglich funktionaler Objektaspekte vgl. Abschnitt 2.2, einen Versuch einer Festlegung des Begriffs „Funktion“ und seiner unterschiedlichen Ausprägungen findet sich beispielsweise bei Chandrasekaran und Josephson (1997), die zusätzlich auch einige interessante Hinweise auf weiterführende Literatur bieten.

sich unter anderem in Harnad (1990), in Coradeschi und Saffiotti (1999) mit einem Schwerpunkt auf autonomen Robotersystemen oder für sprachliche Symbole in Siskind (1995)). Für die hier zu entwickelnde Objektrepräsentation bedeutet dies, dass die Abbildung von Information aus Objektmodellen auf Information aus der sensorischen Verarbeitung möglichst einfach und intuitiv möglich sein sollte – eine allgemeine Lösung des Symbol Grounding Problems wird nicht angestrebt, da sie den Rahmen dieser Arbeit bei weitem überschreitet. Die gleiche Forderung gilt dann auch für den umgekehrten Weg, die Abbildung von Sensorinformationen auf Information in Objektmodellen. Für den Roboter bedeutet das insbesondere, dass einerseits Ergebnisse der sensorischen Verarbeitung in die Objektmodelle eingebunden werden können, andererseits aber auch die Objektmodelle herangezogen werden können, um die sensorische Verarbeitung und darauf aufbauende Verarbeitungsschritte zu unterstützen.

### 3.3 Entwurf einer semantischen Objektrepräsentation

Im Rahmen dieser Arbeit wird ein grundlegender Entwurf für eine semantische Objektrepräsentation erstellt, der unter anderem funktionale Aspekte der modellierten Objekte enthält und eine Darstellung von Objektinformationen in benutzerverständlichen Kategorien beinhaltet. Im Folgenden wird diese semantische Objektrepräsentation vorgestellt. Hierzu gibt zunächst Abschnitt 3.3.1 einen Überblick über die Struktur und die Komponenten der Repräsentation. Die einzelnen Komponenten werden dann detailliert in den darauf folgenden Abschnitten vorgestellt. Der Entwurf wird in Kap. 3.4 noch einmal zusammenfassend diskutiert.

#### 3.3.1 Struktur und Komponenten der Repräsentation

Die wesentlichen Bestandteile eines Objektmodells in der in dieser Arbeit vorgeschlagenen semantischen Objektrepräsentation sind Objekteigenschaften, Attribute und Attributwerte, Handlungen und Objektzustände.<sup>2</sup> Ein Objekt hat dabei verschiedene Eigenschaften, diese wiederum implizieren Handlungen, Zustände sowie unterschiedliche Attribute mit dazugehörigen Attributwerten. Eigenschaften stellen Information auf einer relativ abstrakten semantischen Ebene dar, z. B. dass es sich bei einem Objekt um einen Behälter handelt (**ist\_Behälter**) oder dass es gegriffen werden kann (**ist\_greifbar**). Alle Objekte mit einer solchen Eigenschaft gehören zu einer gemeinsamen Kategorie von Objekten, die eine bestimmte funktionale Bedeutung für Menschen impliziert, beispielsweise die Kategorie aller Behälter. Hierdurch erlauben es Objekteigenschaften unter anderem, Objekte mit Handlungen zu assoziieren. Zum Beispiel kann ein Objekt mit der Eigenschaft **ist\_transportierbar** von einem Ort zu einem anderen *bewegt* werden, ein Behälter kann *befüllt* oder *entleert* werden etc. Dieses Konzept der Eigenschaften von Objekten erlaubt somit eine intuitive Kommunikation zwischen dem menschlichen Benutzer und einem Robotersystem, da beide

---

<sup>2</sup>Um die einzelnen Bestandteile eines Objektmodells im Text eindeutig zu kennzeichnen, wird folgende Konvention verwendet: **Eigenschaften** werden fett gedruckt, *Attribute* und *Handlungen* kursiv, **Objektzustände** und **Objektinstanzen** gesperrt und **OBJEKTKLASSEN** in Kapitälchen (Die einzelnen Begriffe werden im Verlauf des Kapitels erläutert und definiert.).

in den gleichen menschenverständlichen Kategorien über Objekte denken und reden, und da zusätzlich Handlungsanweisungen des Benutzers sehr einfach auf die entsprechenden Objektkategorien abgebildet werden können.

Für ein maschinelles System sind Eigenschaften allein als Information über Objekte allerdings nicht detailliert genug. Wesentliche Informationen bei der Verarbeitung von Sensoreingaben, wie Farbinformation zur Objekterkennung, oder Information für die Aktorik wie Bewegungseinschränkungen, z. B. die maximal erlaubte Beschleunigung beim Tragen eines vollen Glases, können alleine durch Eigenschaften nicht dargestellt werden. Der zweite wichtige Bestandteil der entwickelten Objektrepräsentation sind daher Attribute und deren Attributwerte. Attribute beschreiben die notwendigen Informationen auf einer weniger abstrakten Ebene als Eigenschaften. Beispiele für Objektattribute sind *Farbe*, *Manipulationseinschränkungen*, *Bounding Box*, *3D-Modell* oder Sensorparameter für die Objekterkennung. Nicht alle Attribute haben für jedes Objekt eine Bedeutung. So ist die Angabe eines *Füllstands* sinnlos für ein Objekt, das nicht befüllbar ist. Deshalb folgen die Attribute von Objekten direkt aus deren Eigenschaften. Jede Eigenschaft impliziert dabei die Existenz verschiedener Objektattribute: So hat ein Behälter einen *Füllstand*, einen *maximalen Neigungswinkel* in Abhängigkeit vom *Füllstand* usw., Attribute, die für andere Objekte als Behälter nicht sinnvoll sind. Wenn die Eigenschaften eines Objektes bekannt sind, sind somit auch seine jeweiligen Attribute gegeben. Ein Attribut muss allerdings nicht ausschließlich nur einer Eigenschaft zugeordnet werden, sondern kann für mehrere Eigenschaften eines Objektes relevant sein. Ein Beispiel hierfür sind Manipulationseinschränkungen wie ein *maximaler Neigungswinkel* oder *maximale Beschleunigungen*, die von unterschiedlichen Eigenschaften beeinflusst werden können. In solchen Fällen wird ein Attribut so vielen Eigenschaften zugeordnet wie notwendig. Bei Änderungen wird der entsprechende Attributwert jeweils global für das gesamte Objekt verändert. Attribute beschreiben also quantitative Objektinformationen in detaillierterer Form. Die Art des Attributwerts, z. B. Boolescher Wert, Vektor, String, Trajektorie etc., wird dabei aus einer vorher festgelegten Menge von Werttypen gewählt.

Eigenschaften von Objekten implizieren nicht nur deren Attribute, sondern auch Handlungen, die mit diesen Objekten ausgeführt werden können. So kann ein befüllbares Objekt *befüllt* oder *entleert* werden, was für andere Objekte keinen Sinn ergibt, ein greifbares Objekt kann *gegriffen* und *losgelassen* werden etc. Vergleiche über Attributwerte können dabei als Vor- und Nachbedingungen von Handlungen genutzt werden. So kann ein Objekt nur *befüllt* werden, wenn sein *Füllstand* nicht bereits im Vorfeld zu hoch ist. Nach einer erfolgreichen Einfüllhandlung sollte der *Füllstand* dann entsprechend gestiegen sein. Um eine größtmögliche Anpassung an menschliche Denk- und Kommunikationsstrategien zu gewährleisten, wird zusätzlich das Konzept der Objektzustände eingeführt. Wenn Menschen über Objekte kommunizieren, benutzen sie in aller Regel nicht Ausdrücke wie „der Füllstand beträgt 90%“, um ein Objekt zu beschreiben, sondern Begriffe wie *voll*, *leer*, *heiß*, *kalt* etc., also unscharfe linguistische Variablen. Die Semantik dieser Begriffe kann dabei von Kontext zu Kontext und selbst von Sprecher zu Sprecher unterschiedlich sein. Übertragen auf das vorgeschlagene Objektmodell entsprechen diese Objektzustände Bezeichnungen für Kombinationen aus bestimmten Wertebereichen von Attributen. So kann ein Objekt, das als *voll* bezeichnet wird, in einem bestimmten Kontext einen *Füllstand* von über 90% haben. Hierbei sind auch Wertebereiche verschiedener Attribute kombinierbar, z. B. wenn ein Joghurt als *genießbar* bezeichnet wird, wenn es entweder das Mindesthaltbarkeitsdatum noch nicht

überschritten hat, oder wenn es dieses Datum überschritten hat, aber nicht verdorben ist. Die potentiellen Zustände eines Objektes ergeben sich ebenfalls aus dessen Eigenschaften.

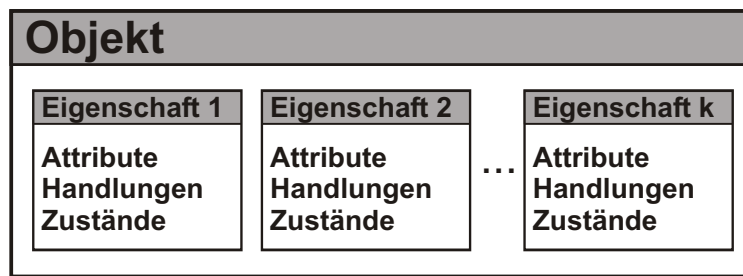


Abbildung 3.2: Grundstruktur der semantischen Objektrepräsentation

Abb. 3.2 stellt die Komponenten des Objektmodells im Überblick dar: Verschiedene Eigenschaften eines Objektes führen jeweils unterschiedliche Attribute, Handlungen und Objektzustände in das Objektmodell ein. Objekte als Ganzes existieren sowohl in Form von Objektklassen unterschiedlicher Allgemeinheit als auch in Form einzelner konkreter Objekte, die jeweils eine Instanz einer solchen Objektklasse darstellen. Objektklassen erben hierarchisch von anderen Objektklassen. Dies bedeutet insbesondere, dass Information über relevante Eigenschaften, Default-Attributwerte etc. vererbt wird. Eine Klasse der WEISSEN TASSEN eines bestimmten Geschirrs erbt z. B. Objektinformationen von einer allgemeineren Klasse von TASSEN (z. B. die Eigenschaft **ist\_befüllbar**). Gleichzeitig enthält die konkretere Objektklasse weitere Informationen wie die Farbe „weiß“, die entsprechende Höhe der Tassen, deren Durchmesser etc. Jede Instanz dieser WEISSEN TASSEN ist dann ebenfalls *befüllbar* usw. Mehrere konkrete Instanzen dieser Klasse können in der Welt existieren, von denen manche dann wiederum in einer konkreten Szene enthalten sind: beispielsweise zwei im Küchenschrank und eine in der Geschirrspülmaschine. Eigenschaften, Attribute und Attributwerte, Handlungen und Zustände werden also zwischen Objektklassen vererbt, und eine aus einer Objektklasse gebildete Objektinstanz prägt die Eigenschaften, Handlungen, Zustände und Attribute aus, die auch die entsprechende Objektklasse besitzt. Allerdings können die Attributwerte der Instanz sich von den Default-Attributwerten der Klasse unterscheiden und sind außerdem veränderlich über die Zeit. Zudem gibt es szenenrelevante Informationen, die nur Objektinstanzen in einer Szene zugeordnet werden. Hierzu gehören beispielsweise die Position einer Objektinstanz in einer konkreten Szene, aber auch räumliche oder funktionale Relationen zwischen Objekten einer Szene.

Die folgenden Abschnitte stellen die Objektrepräsentation formal dar und beschreiben ihre Komponenten im Detail. Die Implementierung dieses Entwurfs im Rahmen des interaktiven Objektmodellierungssystems IOMOS wird in Kap. 4.4 beschrieben. Eine Domänenspezifikation mit konkreten Beispielen für Eigenschaften, Attribute etc. für einen Serviceroboter in der Küche wird beispielhaft in Kap. 6 vorgestellt.

### 3.3.2 Attribute, Attributwerte und Werttypen

Wie oben bereits erläutert, wird Information über Objekte in der hier entwickelten Objektrepräsentation auf unterschiedlich detaillierten Ebenen modelliert. Auf der detailliertesten, konkretesten und sensornächsten Stufe der Objektrepräsentation wird Information in

Attributen modelliert, die dabei die kleinste verwendete Informationseinheit innerhalb des Objektmodells darstellen. Beispiele für Objektattribute sind *Füllstand*, *Farbgraph*, *Bounding Box* oder *Greifansatzpunkte* oder auch Informationen wie Sensorparameter zur Objekterfassung und bevorzugte Erkennungsmethoden, die beispielsweise für ein durchsichtiges Glas anders sein können als für eine nicht-durchsichtige Tasse. In bisherigen Objektmodellen wurden vorwiegend solche Attribute modelliert und nur selten semantische oder funktionale Information, wobei die Modelle sich oft auf geometrische Attribute oder Objektansichten beschränken (vgl. die Diskussion in Kap.2.2). Ein Attribut besteht aus Attributname und -wert sowie einer Liste der Attribute, von deren Attributwert sein Wert abhängt: beispielsweise hängen die Attributwerte von Manipulationseinschränkungen wie *maximale Beschleunigung* von den Werten anderer Attribute wie dem *aktuellen Füllstand* ab oder die *Bounding Box* des Objekts von seinem *3D-Modell*, das in Form einer 3D-Punktwolke vorliegen kann. In der vorliegenden Arbeit wird daher der Begriff „Attribut“ wie folgt verwendet:

**Definition 3.3 (Attribut)** Die Attribute eines Objektes stellen die kleinste Informationseinheit innerhalb des Objektmodells dar. Ein Attribut  $a$  ist ein Tupel aus Attributname  $N$ , Attributwert  $w_a$  und einer Liste  $L$  der Attribute, von deren Werten es abhängt:

$$a = (N, w_a, L). \quad (3.1)$$

Der Wert eines Attributes ist dann definiert als:

**Definition 3.4 (Attributwert)** Der Wert  $w_a$  eines Attributes  $a$  ist eine zum Attribut gehörige, über die Zeit veränderliche Information mit einem beliebigen im Voraus festgelegten Datentyp  $t_a \in T$ , dem so genannten „Werttyp“, wobei  $T = \{t_1, \dots, t_n\}$  die Menge aller möglichen Attributwerttypen bezeichnet.

Für Werttypen schließlich gilt:

**Definition 3.5 (Werttyp)** Ein Werttyp  $t$  legt eine Datenstruktur für Attributwerte sowie zur Datenstruktur gehörige Operatoren (wie Vergleichsoperatoren) fest. Er ist ein Tupel

$$t = (N, D^t, O^t), \quad (3.2)$$

wobei gilt

$N$	Name des Werttyps
$D^t$	Struktur des Datentyps
$O^t = \{o_1^t, \dots, o_n^t\}$	Menge der Operatoren für diese Datenstruktur

Somit ergibt sich die Struktur eines Attributes wie in Abb. 3.3 dargestellt. Die Menge der möglichen Werttypen  $T$  für Attribute ist dabei abhängig von der konkreten Anwendung und kann jederzeit durch zusätzliche Werttypen erweitert werden. Es ist dabei nur zu beachten, dass der Werttyp eines konkreten Attributes jeweils fest ist. Die Implementierung dieser

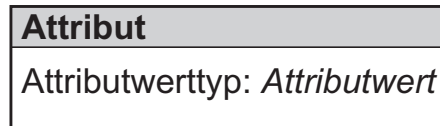


Abbildung 3.3: Schematischer Aufbau eines Objektattributes

Werttypen als „Wertklasse“, d. h. als tatsächliche C++-Klasse, die den Datentyp eines Wertes repräsentiert und darauf Vergleichsoperationen, Speicheroperationen etc. implementiert, wird in Kapitel 4.4 detailliert vorgestellt.

Per Konvention bezeichnet im Folgenden

- $A$  die Menge aller in der Welt existierenden Attribute,
- $a$  ein bestimmtes Attribut mit  $a \in A$ ,
- $w_a$  den Wert eines bestimmten Attributes  $a$ ,
- $T$  die Menge aller in der Welt existierenden Werttypen,
- $t_a$  den Werttyp des Attributwertes von  $a$  mit  $t_a \in T$ ,
- $O^t$  die Menge aller in der Welt existierenden Operatoren zum Werttyp  $t$ ,
- $o^t$  einen Operator zum Werttyp  $t$  mit  $o^t \in O^t$ , und insbesondere
- $VOP^t$  die Menge aller in der Welt existierenden Vergleichsoperatoren zum Werttyp  $t$  mit  $VOP^t \subset O^t$ ,
- $vop^t$  einen Vergleichsoperator zum Werttyp  $t$ , wobei gilt  $vop^t \in VOP^t$ .

Auch für die Menge  $A$  gilt dabei, dass ihre Elemente nur in Abhängigkeit von der Anwendung und Domäne sinnvoll festgelegt werden können.  $A$  kann im Verlauf der Anwendung durch weitere Attribute ergänzt werden, z. B. interaktiv durch den Benutzer (vgl. auch Kap. 5).

Um das Verständnis zu erleichtern, werden Attributwerte gelegentlich unter expliziter Angabe ihres Werttyps notiert, dann bezeichnet  $w_a[t]$  den Wert des Attributs  $a$  mit Werttyp  $t$ ,  $t \in T$ . Zum Beispiel gibt  $w_{Fuellstand}[\%] = 50$  den Wert des Attributs *Füllstand* mit 50 % an. Das Attribut selbst ist in diesem Fall das Tupel  $(Fuellstand, 50, \emptyset)$ .

Die Vergleichsoperatoren, die für jeden Werttyp definiert sind, können genutzt werden, um den Wertebereich eines Attributwertes einzuschränken. So kann ein Attributwert  $w_a[int]$  des Werttyps „ganze Zahl“ durch den Vergleichsoperator  $> (w_a[int], 100)$  auf den Wertebereich aller ganzen Zahlen über 100 eingeschränkt werden. Der Vergleichsoperator  $vop^t(w_a[t], v)$  vergleicht dabei den Attributwert  $w_a[t]$  mit einem vorgegebenen Referenzwert  $v$  und liefert bei Übereinstimmung der beiden Werte das Ergebnis 1, sonst 0:

$$vop^t : w_a[t] \rightarrow \{1; 0\} \quad (3.3)$$

Dabei können nur Vergleichsoperatoren auf Attributwerte angewendet werden, die für den Werttyp des Attributwertes definiert sind, und der Vergleichswert  $v$  muss nicht zwingend vom gleichen Datentyp sein wie der Attributwert. Insbesondere kann auch eine Verknüpfung von Vergleichsoperatoren, z. B. eine Konjunktion, genutzt werden, um den Wertebereich eines Attributwertes einzuschränken. Falls die Anforderungen dieser Vergleichsoperatoren



sich widersprechen, kann kein Attributwert in dem geforderten eingeschränkten Wertebereich liegen, die Anforderung ist nicht erfüllbar. Im Folgenden werden erfüllbare Einschränkungen von Attributwertbereichen vorausgesetzt, da nur diese für die praktische Anwendung von Objektmodellen von Relevanz sind.

### 3.3.3 Objektzustände

Die reine Anwendung von Vergleichsoperatoren auf Attributwerte liefert zunächst noch keine semantische Information, da sie noch keinen Bezug zwischen Attributwerten und Welt herstellt. Dies leisten in einem Objektmodell auf einer für den Menschen verständlichen Ebene die so genannten Objektzustände. Sie entsprechen den Zuordnungen, die von Menschen im Alltag ständig vorgenommen werden, wenn Objekte in der Kommunikation beispielsweise als **heiß**, **voll** oder **grün** bezeichnet werden. Solche Zuordnungen beschreiben jeweils einen bestimmten Wertebereich eines Attributwertes (bzw. eine konjunktive und/oder disjunktive Kombination aus den Wertebereichen mehrerer Attribute). Objektzustände bilden die Zuordnung zwischen Attributwerten und menschenverständlichen Objektbeschreibungen. Sie ermöglichen es damit einerseits, Objektinformation (z. B. aus den Sensorsystemen) für den Menschen verständlich und übersichtlich zu präsentieren, andererseits erlauben sie die Zuordnung von menschlichen Äußerungen und Anweisungen auf für ein maschinelles System erfassbare, messbare und berechenbare Größen.

Formal lässt sich ein Objektzustand wie folgt darstellen:

**Definition 3.6 (Objektzustand)** Ein Zustand  $z$  ordnet bestimmten Wertebereichen der Werte  $w_{a_1}, \dots, w_{a_k}$  eines oder mehrerer Attribute  $a_1, \dots, a_k$  ein Symbol zu. Er ist ein Tupel

$$z = \left( N, \bigodot_{h=1}^k \left( \text{vop}_{j_h}^{t_{a_h}} (w_{a_h} [t_{a_h}], v_h) \right) \right) \quad (3.4)$$

wobei gilt

$N$	Name des Zustands
$\text{vop}_{j_h}^{t_{a_h}} (w_{a_h} [t_{a_h}], v_h)$	$j_h$ -ter Vergleichsoperator zum Werttyp $t_{a_h}$ , angewendet auf den Attributwert $w_{a_h}$ und den Referenzwert $v_h$ , mit $\text{vop}_{j_h}^{t_{a_h}} \in \text{VOP}^{t_a}$ beliebig
$\bigodot$	beliebige Verkettung aus Konjunktionen und Disjunktionen

Per Konvention bezeichnet im Folgenden

$Z$	die Menge aller in der Welt existierenden Zustände,
$z$	einen bestimmten Zustand mit $z \in Z$ .

Haushaltstypische Beispiele für Zustände sind **voll** oder **leer** für Objekte mit der Eigenschaft **ist\_Behälter** in Abhängigkeit von deren *Füllstand*, Begriffe wie **rot** oder **blau** für Farbbereiche oder Zuordnungen wie **heiß** und **kalt** für Temperaturen. Ein Zustand wie

lauwarm kann beispielsweise festgelegt werden durch folgende Verkettung von Vergleichsoperatoren:  $(Temperatur > 25^\circ C) \wedge (Temperatur < 35^\circ C)$ , ein Lebensmittel könnte als **noch essbar** bezeichnet werden, wenn folgende Verkettung von Vergleichsoperatoren wahr ist:  $(Verfallsdatum \geq aktuellesDatum) \wedge (verschimmelt = nein)$ .

Die Semantik von Objektzuständen ist kontext- und anwendungsspezifisch; so kann im Kontext „Gasthaus“ der Zustand **voll** bedeuten „mindestens bis zum Eichstrich gefüllt“. In einem Haushalt hingegen wird ein Glas vielleicht als **voll** bezeichnet, wenn es zu mindestens 90 % gefüllt ist.<sup>3</sup> Daher können die Zustände ebenso wie alle anderen Informationen über Objekte immer nur im Zusammenhang mit einer konkreten Anwendung sinnvoll gewählt werden, d. h. die den Objektmodellen zugrunde liegende Ontologie kann nicht allgemeingültig für alle denkbaren Situationen oder Kontexte sein, sondern wird sich auf eine bestimmte Situation wie die des Serviceroboters im Haushalt beziehen müssen.

Ebenso wie für Handlungen und Attribute gilt auch für Zustände, dass nur Objekte mit einer bestimmten Eigenschaft einen bestimmten Zustand einnehmen können. Ein Zustand tritt also nur in Abhängigkeit von einer bestimmten Eigenschaft auf. Die Attribute, auf die sich der Zustand bezieht, müssen ebenfalls dieser Eigenschaft zugeordnet sein.

### 3.3.4 Handlungen

Die Eigenschaften eines Objektes bringen nicht nur seine Attribute mit sich und die Zustände, die es einnehmen kann, sondern implizieren auch Handlungen, die mit diesem Objekt ausgeführt werden können. Im Rahmen der vorliegenden Arbeit soll hierbei kein vollständiges Handlungsmodell entworfen werden, sondern eine Schnittstelle zwischen Handlungs- und Objektmodellierung definiert werden als Grundlage für Handlungslernen oder -planung. Es werden deshalb hier nur die handlungsbezogenen Aspekte von Objekten betrachtet, nicht aber die Details eines umfassenden Handlungsmodells. Eine Ansatz zur Integration von Bewegungsmodellen und Objektmodellen sowie Möglichkeiten, diese kombinierten Informationen beispielsweise zur Handlungsplanung oder Bewegungsprädiktion einzusetzen, stellen Becher et al. (2006) vor. Eine Handlung ist dann im Rahmen der vorliegenden Arbeit definiert als:

**Definition 3.7 (Handlung)** Eine Handlung  $h$  als Bestandteil der Objektrepräsentation ist ein Tupel

$$h = (N, V, N, P), \quad (3.5)$$

wobei gilt

---

<sup>3</sup>Interessanterweise benutzen Menschen Begriffe für Objektzustände nicht nur stark situationspezifisch und auch standpunktabhängig (wie in dem Scherz über das „halbvolle vs. halbleere Glas“), sondern auch stark in Abhängigkeit von ihren körperlichen Empfindungen, wie sich in einer einfachen Versuchsanordnung demonstrieren lässt: Es werden drei Schüsseln aufgestellt, die linke gefüllt mit kaltem, die mittlere mit lauwarmem und die rechte mit heißem Wasser. Wenn ein Mensch nun die linke Hand zunächst in die linke Schüssel mit kaltem Wasser hält und dann in die mittlere, wird er das lauwarmer Wasser in der Regel als „heiß“ bezeichnen. Wenn gleichzeitig die rechte Hand zunächst in das heiße Wasser gehalten wird und dann ebenfalls in die mittlere Schüssel mit lauwarmem Wasser, wird diese Empfindung dann als „kalt“ bezeichnet, obwohl natürlich klar ist, dass das lauwarmer Wasser in der mittleren Schüssel nicht gleichzeitig heiß und kalt sein kann. Trotzdem fällt es Menschen schwer, sich in ihren Objektbeschreibungen von solchen körperlichen Empfindungen zu lösen.

*N* Name der Handlung  
*V* objektbezogene Vorbedingungen der Handlung  
*N* objektbezogene Nachbedingungen der Handlung  
*P* evtl. weitere objektbezogene Parameter

Über die Vor- und Nachbedingungen können Handlungen mit Attributen verknüpft werden. So kann die Handlung *befüllen* bei einem Objekt mit der Eigenschaft **ist\_Behälter** in einem bestimmten Kontext nur durchgeführt werden, wenn der *Füllstand*  $\leq 50\%$  ist (Vorbedingung), und nach Beendigung der Handlung gilt dann: *Füllstand*  $\geq 80\%$  (Nachbedingung). Die weiterhin möglichen objektbezogenen Parameter können Hardware- oder Sensorikinformationen sein, die für bestimmte Handlungen objektspezifisch abzulegen sind.

Wie auch Attribute und Zustände können Handlungen immer nur in Abhängigkeit von einer gewählten Domäne und gegebenen Anforderungen sinnvoll festgelegt werden. Handlungen treten stets in Abhängigkeit von einer Eigenschaft auf, d. h. dass Objekte mit einer bestimmten Eigenschaft für diejenigen Handlungen verwendet werden können, die mit dieser Eigenschaft assoziiert sind. Falls eine Handlung über Vor- oder Nachbedingungen mit Attributen verknüpft ist, müssen diese Attribute der gleichen Eigenschaft zugehörig sein wie die jeweilige Handlung. Per Konvention bezeichnet im Folgenden

*H* die Menge aller in der Welt existierenden Handlungen,  
*h* eine bestimmte Handlung mit  $h \in H$ .

### 3.3.5 Eigenschaften

Objekteigenschaften sind ein wesentliches Element einer semantischen Objektrepräsentation nach Definition 3.1, da sie sowohl semantische Information im allgemeinen als auch speziell funktionale Information in das Objektmodell einführen. Eigenschaften gliedern die vorliegenden Informationen über Objekte in sinnvolle Einheiten, die einem intuitiven menschlichen Verständnis der Welt entsprechen (vgl. die Überlegungen in Abschnitt 3.2), und erlauben die temporäre Zuordnung von beschreibenden Alltagsbegriffen wie *kalt*, *voll* etc. zu Objekten. Gleichzeitig dienen sie als Anknüpfungspunkt für Handlungen, die mit dem Gegenstand ausgeführt werden können, so dass das Objektmodell sinnvoll in die Repräsentation und Planung von Handlungen und Aufgaben eingebunden werden kann.

Gegenstände mit einer gemeinsamen Eigenschaft gehören einer gemeinsamen *Kategorie* von Objekten im Sinne einer Begriffsbildung an.<sup>4</sup> So sind z. B. alle Objekte mit der Eigenschaft **ist\_flüssig** Mitglieder der Kategorie der Flüssigkeiten. Im Rahmen der vorgeschlagenen Objektrepräsentation wird der Begriff „Eigenschaft“ daher wie folgt verwendet:

<sup>4</sup>Die Problematik der Konzeptbildung und Kategorisierung wird in der Wissenschaft kontrovers und detailliert diskutiert, und es existiert eine Fülle an interessanter Literatur unterschiedlichster Fachgebiete aus den letzten Jahrzehnten. Da die Kategorienbildung beim Menschen nicht Gegenstand dieser Arbeit ist, soll diese Diskussion hier nicht weiter vertieft werden. Es wurde allerdings bewusst ein allgemeines Verständnis des Begriffs „Kategorie“ verwendet, dass sich entsprechend mit den wesentlichen Theorien zur Kategorienbildung beim Menschen vereinen lässt. Einen guten Überblick über die unterschiedlichen Standpunkte bieten Komatsu (1992), Smith und Medin (1981) oder Strube et al. (1996). Wichtige Theorien der Kategorienbildung sind die Theorie der Familienähnlichkeit, vertreten beispielsweise durch Rosch und Mervis (1975), sowie erklärungs-basierte Ansätze, vertreten unter anderem durch Lakoff (vgl. Lakoff (1987)).

**Definition 3.8 (Eigenschaft)** Eine Eigenschaft  $e$  ist ein Tupel

$$e = (N, \mathcal{A}, \mathcal{H}, \mathcal{Z}), \quad (3.6)$$

wobei

$N$	Name der Eigenschaft
$\mathcal{A} = \{a_1, \dots, a_l\}, \mathcal{A} \subseteq A$	Menge aller von dieser Eigenschaft implizierten Attribute
$\mathcal{H} = \{h_1, \dots, h_m\}, \mathcal{H} \subseteq H$	Menge aller von dieser Eigenschaft implizierten Handlungen
$\mathcal{Z} = \{z_1, \dots, z_n\}, \mathcal{Z} \subseteq Z$	Menge aller von dieser Eigenschaft implizierten Zustände

Eine Eigenschaft verknüpft somit jeweils mehrere inhaltlich zusammengehörige Attribute von Objekten. Diese Attribute können voneinander abhängen, so ist der maximal erlaubte Neigungswinkel eines Behälters wesentlich von dessen Füllstand abhängig. Beiden gemeinsam ist aber, dass sie nur für solche Gegenstände sinnvoll sind, die über die jeweilige gemeinsame Eigenschaft verfügen. Zu beachten ist, dass innerhalb dieser Repräsentation ein Attribut durchaus mehreren Eigenschaften zugeordnet sein kann, und daher der Wert eines Attributes in einem Objektmodell durch mehrere Eigenschaften beeinflusst werden kann. Beispielsweise sind ohne weiteres mehrere Eigenschaften denkbar, die als Attribut Manipulationseinschränkungen wie einen *maximalen Neigungswinkel* oder eine *maximale Beschleunigung* einführen. Es handelt sich hierbei aber in Bezug auf ein bestimmtes Objekt jeweils um dasselbe Attribut, d.h. der Attributwert ist auch für alle Eigenschaften jeweils gleich.

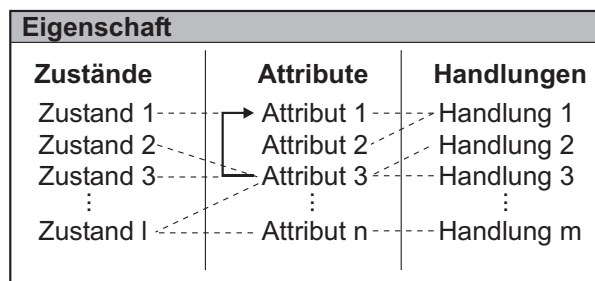


Abbildung 3.4: Schematische Darstellung einer Eigenschaft mit ihren Bestandteilen

Objekteigenschaften implizieren aber nicht nur Attribute, sondern auch Zustände und auf einer funktionalen Ebene Handlungen, die mit diesem Objekt ausgeführt werden können. So können Objekte mit der Eigenschaft **ist\_Behälter** *gefüllt* oder *entleert* werden oder die entsprechenden Zustände wie *voll* oder *leer* einnehmen. Generell gilt, dass die Objekte einer bestimmten Kategorie (d. h. Objekte mit einer bestimmten Eigenschaft) zu bestimmten Handlungen verwendet werden können und zumindest zeitweise in einer Szene bestimmte Zustände einnehmen können. Diese Vorgehensweise ist nicht nur für den menschlichen Benutzer intuitiv verständlich, sondern erleichtert gleichzeitig auch die Verarbeitung von Daten aus der Interaktion mit dem Benutzer (z. B. von Anweisungen) und die Handlungsplanung, weil Handlungen und Zustandsbeschreibungen direkt auf einzelne Objekte umgesetzt werden können. Bei der Anweisung, Kaffee einzuschenken, kommen als potentiell zu benutzende Objekte nur Behälter in Frage, so dass beispielsweise die Suche nach geeigneten Objekten mit diesen Randbedingungen deutlich schneller und robuster vonstatten gehen kann. Abb. 3.4

stellt die Bestandteile einer Eigenschaft und ihre Beziehungen untereinander noch einmal zusammenfassend dar.

### 3.3.6 Objektklassen

Das in der hier vorgeschlagenen semantischen Objektrepräsentation entwickelte Schema wird zur Erzeugung von Objektmodellen auf die Objektklassen einer bestimmten Domäne angewendet (exemplarisch geschieht dies in den folgenden Kapiteln für die Domäne „Küche“). Bei den Modellen von Objektklassen und -instanzen handelt es sich dann um Objektmodelle im Sinne von Def. 3.2, d. h. um Darstellungen konkreter Objekte bzw. Objekttypen einer Domäne im Hinblick auf eine bestimmte Anwendung. Der Begriff „Objektklasse“ wird in dieser Arbeit wie folgt verwendet:

**Definition 3.9 (Objektklasse)** Eine Objektklasse ist die abstrakte Repräsentation aller Objekte, die einem bestimmten, konkreten oder abstrakten, Objekttyp angehören. Die Objektklasse enthält alle objektspezifischen Informationen in Form von Eigenschaften, Attributen und Attributwerten, Zuständen und Handlungen, soweit sie für alle Objekte dieses Objekttyps gültig sind. Objektklassen können hierarchisch von anderen Klassen erben.

Zu beachten ist, dass die Begriffe „Objektklasse“ und „Objektinstanz“ in der hier vorgestellten Objektrepräsentation zwar in Analogie an die programmiertechnischen Begriffe „Klasse“ und „Instanz“ gewählt sind, sich aber nicht streng im Sinne einer objektorientierten Programmierung verstehen. Insbesondere die Behandlung von mehrfach auftretenden Attributwerten wurde spezifisch für die hier entwickelte Objektrepräsentation gelöst. Abb. 3.6 stellt noch einmal ein Objektmodell als Ganzes schematisch dar. Jeder Objektklasse werden dabei passende Eigenschaften zugewiesen, die wiederum adäquate Attribute, Zustände und Handlungen für dieses Objekt implizieren.

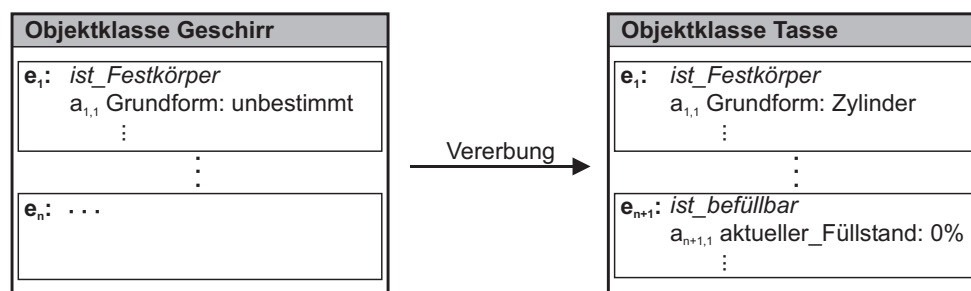


Abbildung 3.5: Beispielhafte Darstellung von Vererbung zwischen Objektklassen

Wie bereits in Def. 3.9 erwähnt, können Objektklassen hierarchisch voneinander erben. Eine solche Objekthierarchie für die Domäne „Küche“ ist beispielhaft dargestellt in Kap. 6.3.1. Diese Objekthierarchien können von abstrakten Objektklassen wie GESCHIRR bis hin zu konkreten Objektklassen wie einer bestimmten Klasse von TASSEN reichen. Generell gilt dabei, dass Tochterklassen die Eigenschaften und korrespondierenden Attribute, Zustände und Handlungen ihrer Mutterklasse erben. Die Default-Attributwerte, Vor- und Nachbedingungen von Handlungen etc. der Mutterklasse können dabei entweder übernommen oder in der Tochterklasse konkretisiert und damit überschrieben werden (vgl. Abb. 3.5). Gleichzeitig kann die Tochterklasse auch selbst noch weitere Eigenschaften mit sich bringen.

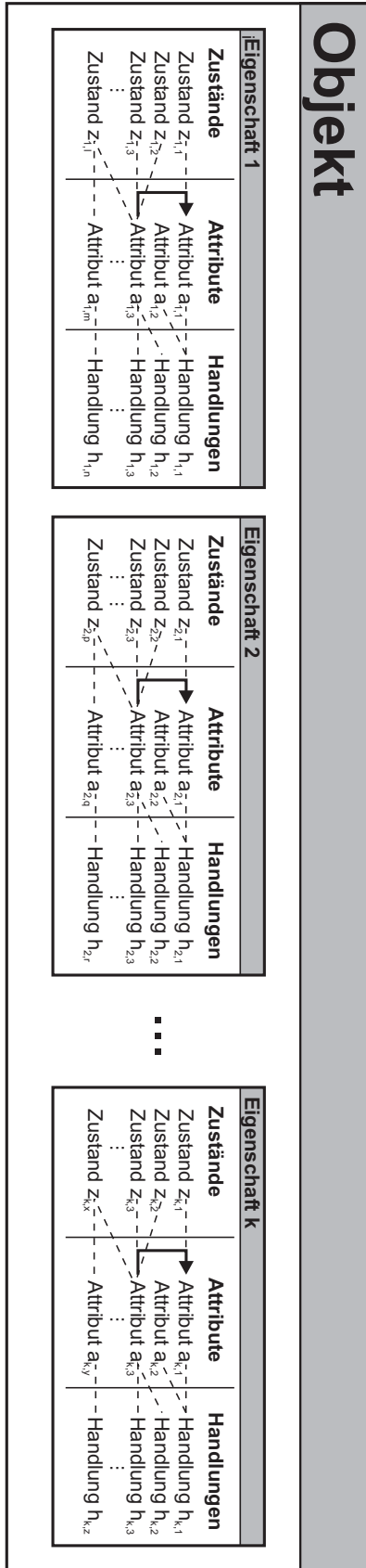


Abbildung 3.6: Komponenten eines Objektmodells im Detail

### 3.3.7 Objektinstanzen

Objektklassen beschreiben nur abstrakte Typen von Objekten, ein tatsächlich in der Welt vorhandenes Objekt hingegen wird durch eine Objektinstanz dargestellt. Der Begriff „Objektinstanz“ (oder kurz „Instanz“) wird dabei im Rahmen dieser Arbeit wie folgt verwendet:

**Definition 3.10 (Objektinstanz)** Eine Objektinstanz stellt das Modell eines in der Welt existierenden Objektes dar und ist insofern eine Ausprägung einer Objektklasse. Es können gleichzeitig mehrere Instanzen zu einer Objektklasse existieren. Die Ausprägungen der Attributwerte einer Instanz (und damit auch ihrer Zustände) ändern sich mit den Veränderungen des dargestellten Objektes.

Instanzen sind stets von einer Objektklasse abgeleitet. Eine Instanz enthält dabei diejenigen Eigenschaften, Attribute, Handlungen und Zustände, die auch die Objektklasse enthält. Die Instanz kann deren Default-Attributwerte übernehmen, kann aber auch (bei der Instanziierung oder später) Attributwerte mit eigenen, aktuellen Werten überschreiben (vgl. Abb. 3.7).

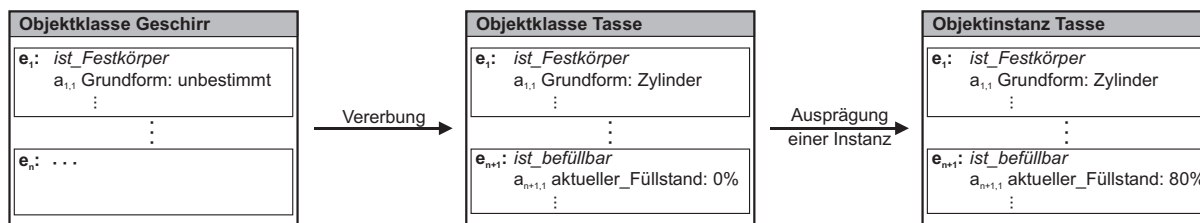


Abbildung 3.7: Vererbung zwischen Objektklassen und -instanzen und Ausprägung von Attributwerten konkreter Objekte

Objektinstanzen repräsentieren konkrete Objekte in der Welt. Einige dieser Objekte können in einem bestimmten Szenenausschnitt zu einer bestimmten Zeit vorhanden sein, so dass dann entsprechend auch die Objektinstanzen im jeweiligen Szenenmodell vorhanden sind. Die Ausprägungen der Objektattribute, wie oben bereits erläutert, können dabei von Instanz zu Instanz unterschiedlich sein und auch über die Zeit variieren: beispielsweise hat eine *Tasse* zu einem bestimmten Zeitpunkt einen *Füllstand* von 50 %, eine andere einen *Füllstand* von 0 %.

## 3.4 Diskussion

In diesem Kapitel wurde eine semantische Objektrepräsentation im Sinne von Def. 3.1 vorgestellt, die für unterschiedlichste Anwendungen flexibel eingesetzt werden kann. In Kap. 6 wird eine Taxonomie von Begrifflichkeiten der Domäne „Küche“ vorgestellt, die auf dieser Objektrepräsentation beruht, in Kap. 4 ein System zur interaktiven Objektmodellierung auf dieser Basis. Die im Rahmen dieser Arbeit entwickelten Komponenten können somit beispielsweise verwendet werden, um einen Serviceroboter im Haushalt mit einem Objektmodell zu versehen, auf dessen Grundlage Manipulation, Objekterkennung, Planung etc. ausgeführt werden können. Als weitere Anwendungen auf der Basis dieser Objektrepräsentation mit jeweils an die spezifischen Anforderungen angepassten Domänenspezifikationen

und Objektmodellen sind so unterschiedliche Bereiche wie der Einsatz von Servicerobotern in anderen Umgebungen (Hotels, Bahnhofshallen, Flughäfen, Museen, ...), autonome Fahrzeuge bzw. Fahrerassistenzsysteme im Straßenverkehr oder autonome Transportplattformen z. B. in Krankenhäusern denkbar. All diesen Einsatzgebieten ist gemein, dass es sich um hochdynamische und stark menschenzentrierte Umgebungen handelt, und dass die betreffenden Systeme über ein domänenspezifisches Objektmodell verfügen müssen, dass ihnen eine Verknüpfung von Objektwissen mit Sensordaten sowie die Ausführung sinnvoller Aktionen gestattet.

In diesem Zusammenhang als grundlegend erweist sich das Konzept der Objekteigenschaften, die nicht nur die in bisherigen Objektmodellen üblichen Attribute, sondern auch Objektzustände und Handlungen mit sich bringen. Die Eigenschaften erlauben damit einerseits eine direkte Modellierung von semantischer Objektinformation und andererseits die Darstellung funktionaler Objektaspekte in Form der Zustände und vor allem der Handlungen, die eine direkte Verbindung von der Objekt- zur Handlungsmodellierung darstellen und somit unter anderem eine sinnvolle Aufgaben- und Handlungsplanung, Szenenerfassung sowie Bewegungserfassung und -planung ermöglichen. Es wird somit der Tatsache Rechnung getragen, dass Objekte nur sinnvoll unter Berücksichtigung ihrer Nutzungsmöglichkeiten modelliert werden können. Andererseits ergibt sich durch die Attribute von Objekten und insbesondere durch die Möglichkeit, auch Informationen wie Sensorparameter oder bevorzugte Erkennungsmethoden im Objektmodell abzulegen, die gewünschte Anbindung an die Sensorik und Aktorik der Robotersysteme.

Zusätzlich ist die vorgeschlagene Repräsentation flexibel und dynamisch einsetzbar. Dies bedeutet nicht nur, dass unterschiedlichste Gegebenheiten und Anwendungsszenarien auf diese Art modelliert werden können, sondern auch dass die Objektmodelle im Laufe der Anwendung beliebig erweitert oder korrigiert werden können. Als weiterer Vorteil stellt sich der deutliche Bezug der vorgeschlagenen Objektrepräsentation zur Repräsentation von Objekten beim Menschen dar. Damit wird einerseits dem Roboter oder anderen maschinellen System eine vereinfachte Erfassung einer menschenzentrierten Umgebung mit all ihren Besonderheiten ermöglicht, andererseits werden aber auch die Kommunikation zwischen Mensch und Roboter und das Verständnis des Menschen für die internen Verarbeitungsvorgänge des Roboters erleichtert, da Mensch und Maschine in den gleichen Kategorien und Begriffen über Objekte denken und reden.

### 3.5 Zusammenfassung

In diesem Kapitel wurde zunächst die Verwendung der Begriffe „Modell“, „Repräsentation“ und „Semantik“ sowie darauf aufbauend „semantische Objektrepräsentation“ bzw. „semantisches Objektmodell“ geklärt. Es wurden dann, aufbauend auf der bereits in Kap. 1 diskutierten Problemstellung und dem vorgeschlagenen Lösungsansatz, die Anforderungen an die semantische Objektrepräsentation spezifiziert. Der Zusammenhang zwischen den einzelnen Komponenten der Objektrepräsentation wurde dann zunächst im Überblick dargestellt, danach wurden sie im Detail formal dargestellt und erläutert. Die vorgestellte Objektrepräsentation bildet die Grundlage für die Entwicklungen, die in den nächsten Kapiteln vorgestellt werden. Das Kapitel schließt mit einer Diskussion dieser Repräsentation, insbesonde-



re bezüglich ihrer potentiellen Anwendungen, ihrer semantischen und funktionalen Aspekte sowie ihrer Bezüge zur Repräsentation von Objekten beim Menschen.



# Kapitel 4

## Das interaktive Objektmodellierungssystem IOMOS

Nachdem im vorhergehenden Kapitel ein grundlegender Entwurf für die Objektrepräsentation vorgestellt wurde, werden in diesem und dem hierauf folgenden Kapitel die interaktive Modellierung von Objekten mittels des interaktiven Objektmodellierungssystems IOMOS vorgestellt. Hierzu werden zunächst die Anforderungen an das Modellierungssystem und die Grundstruktur des Systems erläutert sowie Details der Implementierung der Objektrepräsentation und der Speicherung und Darstellung von Objekten. Die Objektrepräsentation und Datenhaltung sind dabei wesentliche Bestandteile des interaktiven Modellierungssystems IOMOS. Auf dieser Grundlage werden dann im darauf folgenden Kapitel der Aufbau der interaktiven Objektmodellierungsumgebung, die multimodalen Interaktionskomponenten des Systems sowie die Komponenten zur semi-autonomen Ermittlung von Objektattributwerten vorgestellt.

### 4.1 Anforderungen an das Modellierungssystem

Die Anforderungen an das zu entwickelnde interaktive Modellierungssystem orientieren sich am Szenario eines Serviceroboters im Haushalt. Die Ergebnisse lassen sich aufgrund der Eigenschaften der gewählten Anwendung allerdings auch auf andere Bereiche der Objektmodellierung und Robotik übertragen. Insbesondere ergeben sich aus dem gewählten Szenario folgende Anforderungen:

#### **Grad der Interaktivität**

Wie bereits in Kapitel 1.3 dargelegt, lässt sich eine sinnvolle Objektmodellierung in diesem und vergleichbar komplexen Szenarien nur in Interaktion mit dem Benutzer durchführen. Gleichzeitig soll vermieden werden, dass Benutzer durch Routineaufgaben unnötig belastet werden, die das zu entwickelnde Modellierungssystem bis zu einem gewissen Grad autonom durchführen kann. Auch bei den autonom durchführbaren Teilfunktionen der Objektmodellierung sollten allerdings Kontrolle, Korrektur und Ergänzung durch den menschlichen Benutzer zu jeder Zeit möglich sein.

### **Intuitive Verständlichkeit auch für ungeschulte Benutzer**

Durch das gewählte Szenario ist auf längere Sicht die Nutzung von Robotersystemen auch für Nicht-Experten denkbar und erwünscht. Daher ist in allen Bereichen der Interaktion mit dem menschlichen Benutzer, hier insbesondere auch bei der Objektmodellierung, ein möglichst hoher Grad an intuitiver Verständlichkeit auch für Nicht-Experten zu wahren und die Bedienbarkeit des Systems möglichst ohne Programmierkenntnisse oder einen hohen Grad an technischem Vorwissen anzustreben. Insbesondere sollte somit die Verwendung einfacher, intuitiver Interaktionsmodi bevorzugt werden.

### **Ergonomische Kriterien und Bedienbarkeit**

Sowohl für ungeschulte Benutzer als auch für Experten wird die Bedienbarkeit des Modellierungssystems wesentlich erleichtert durch die Einhaltung ergonomischer Grundkriterien, die je nach verwendetem Interaktionsmodus die Bedienung möglichst einfach und wenig umständlich machen (vgl. hierzu die Überlegungen in den Abschnitten 5.1 und 5.2).

### **Flexibilität und Erweiterbarkeit**

Ebenfalls aus dem gewählten Szenario ergibt sich die Anforderung, die Objektmodelle flexibel und erweiterbar zu halten, da nur durch anpassbare Modelle einer menschenzentrierten, hochdynamischen Umgebung entsprochen werden kann. Die Grundlage hierfür bildet die flexible semantische Objektrepräsentation. Die Erfüllung dieser Anforderungen erlaubt den Einsatz der Modellierungsumgebung auch für andere Anwendungen nicht nur aus dem Bereich der Servicerobotik.

### **Qualität der Ergebnisse**

Um qualitativ möglichst hochwertige Objektmodelle zu erhalten, ist eine entsprechende Sensorik notwendig. Leider ist es derzeit noch schwierig, hochauflösende Sensoren auf einem Roboter zu integrieren, unter anderem wegen der Abmessungen und des Gewichts der derzeit zur Verfügung stehenden Sensorsysteme. Es ist daher sinnvoll, die Objektmodelle in einer speziellen Modellierungsumgebung anzulegen, das mit entsprechend hochwertigen Sensorsystemen ausgerüstet ist. Da die technische Entwicklung in diesem Bereich aber schnell voranschreitet, ist davon auszugehen, dass bereits in wenigen Jahren gleichwertige Systeme direkt auf Servicerobotern integriert werden können, so dass die Objektmodelle unmittelbar in Zusammenarbeit zwischen Mensch und Roboter im Verlauf ihrer täglichen Zusammenarbeit erstellt werden können. Das hier entworfene System bleibt hiervon unberührt, da es in diesem Fall als Systemkomponente in einen Roboter integriert werden kann, und nicht nur wie im Moment über eine Schnittstelle zur Szenenmodellierung und anderen Roboterkomponenten.

## **4.2 Systementwurf und Ablauf der Objektmodellierung**

Der Ablauf der interaktiven Erstellung eines Objektmodells ist in Abb. 4.1 im Überblick dargestellt. Vom Benutzer wird zunächst die Erstellung eines neuen Objektmodells initialisiert. Hierbei steuert der Benutzer bereits einige grundlegende Informationen wie den Objektnamen oder wichtige Eigenschaften bei. Auch Attributwerte können bereits in diesem Stadium präzisiert werden. In einem zweiten Schritt, der ebenfalls vom Benutzer angestoßen wird,

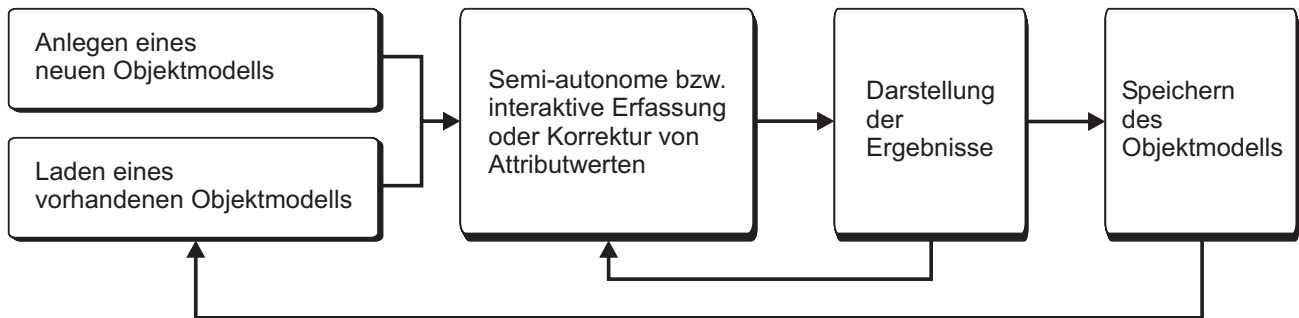


Abbildung 4.1: Ablaufschema der Erstellung eines Objektmodells im interaktiven Objektmodellierungssystem IOMOS

erfasst das Modellierungssystem dann die Attributwerte des Objektes, soweit diese autonom erfassbar sind. Zur Bestimmung einzelner Attributwerte kann dabei auch ein interaktiver Dialog notwendig sein. Erfasste Objektinformationen können dann interaktiv vom Benutzer korrigiert bzw. ergänzt werden. Zuletzt wird das neue Objektmodell in einer Objektmodelldatenbank abgelegt, woraus es später jederzeit zur weiteren Ergänzung und Korrektur wieder in das System eingeladen werden kann. Eine solche Anpassung bzw. Erweiterung von Objektmodellen läuft dann analog zur Neuerstellung ab. Abb. 4.2 zeigt die Erstellung eines Objektmodells noch einmal im Detail.

Abb. 4.3 stellt die Architektur des interaktiven Objektmodellierungssystems IOMOS dar, das sich in folgende Komponenten gliedern lässt:

### Implementierung der Objektrepräsentation

Die Implementierung der in Kap. 3 vorgestellten Objektrepräsentation dient als Grundlage für die interaktive Modellierung von Objekten in IOMOS. Details dieser Implementierung werden in Abschnitt 4.4 vorgestellt.

### Datenverwaltung/Objektmodellverwaltung

Die zentrale Datenverwaltungskomponente erlaubt das Neuanlegen, Verändern, Löschen, Speichern und Laden von Objektmodellen und Teilen von Objektmodellen (vgl. Abschnitt 4.3.1). Die Datenverwaltung ist in der Implementierung direkt an eine graphische Benutzerschnittstelle angegliedert, die dem Benutzer den Zugriff auf alle in IOMOS implementierten Funktionalitäten ermöglicht.

### Interaktionskomponenten

Die Systemkomponenten zur Benutzerinteraktion sind an eine zentrale graphische Oberfläche angegliedert. Sie stellt die wesentlichen Funktionalitäten zur Modellerstellung und -verwaltung zur Verfügung und bildet gleichzeitig die Schnittstelle zu anderen Interaktionsmodi (vgl. die Darstellung der Interaktion zwischen System und Benutzer sowie der zur Verfügung stehenden Interaktionsmodi in Kapitel 5) und erlaubt den Zugriff auf die Komponenten zur semi-autonomen und interaktiven Erfassung von Objektattributwerten. Eine solche Erfassung von Objektinformationen wird aus der graphischen Schnittstelle heraus vom Benutzer initialisiert. Die Ergebnisse des Verarbeitungsprozesses werden für den Benutzer dann wieder in der GUI dargestellt.

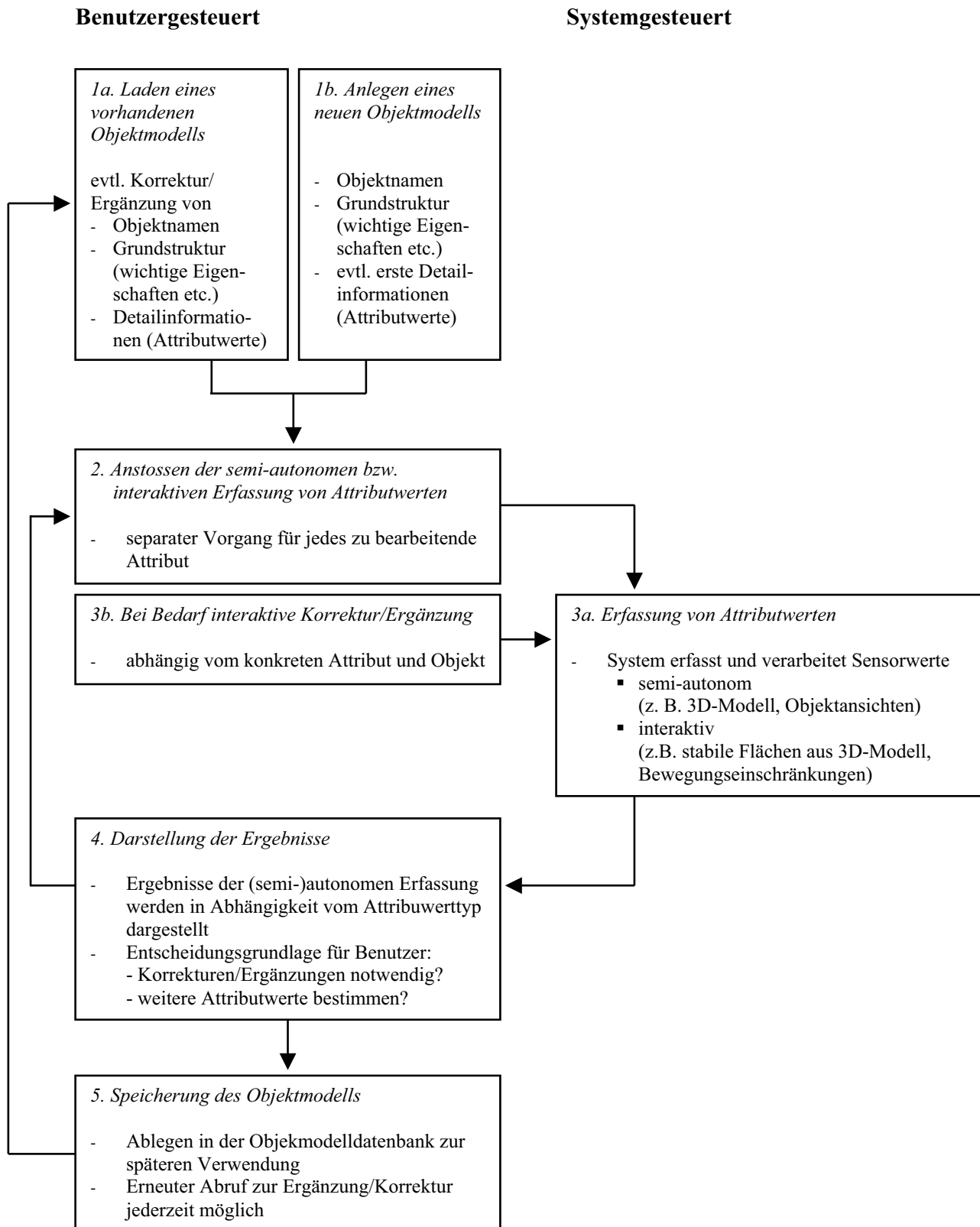


Abbildung 4.2: Details der Prozesskette zur Erstellung eines Objektmodells

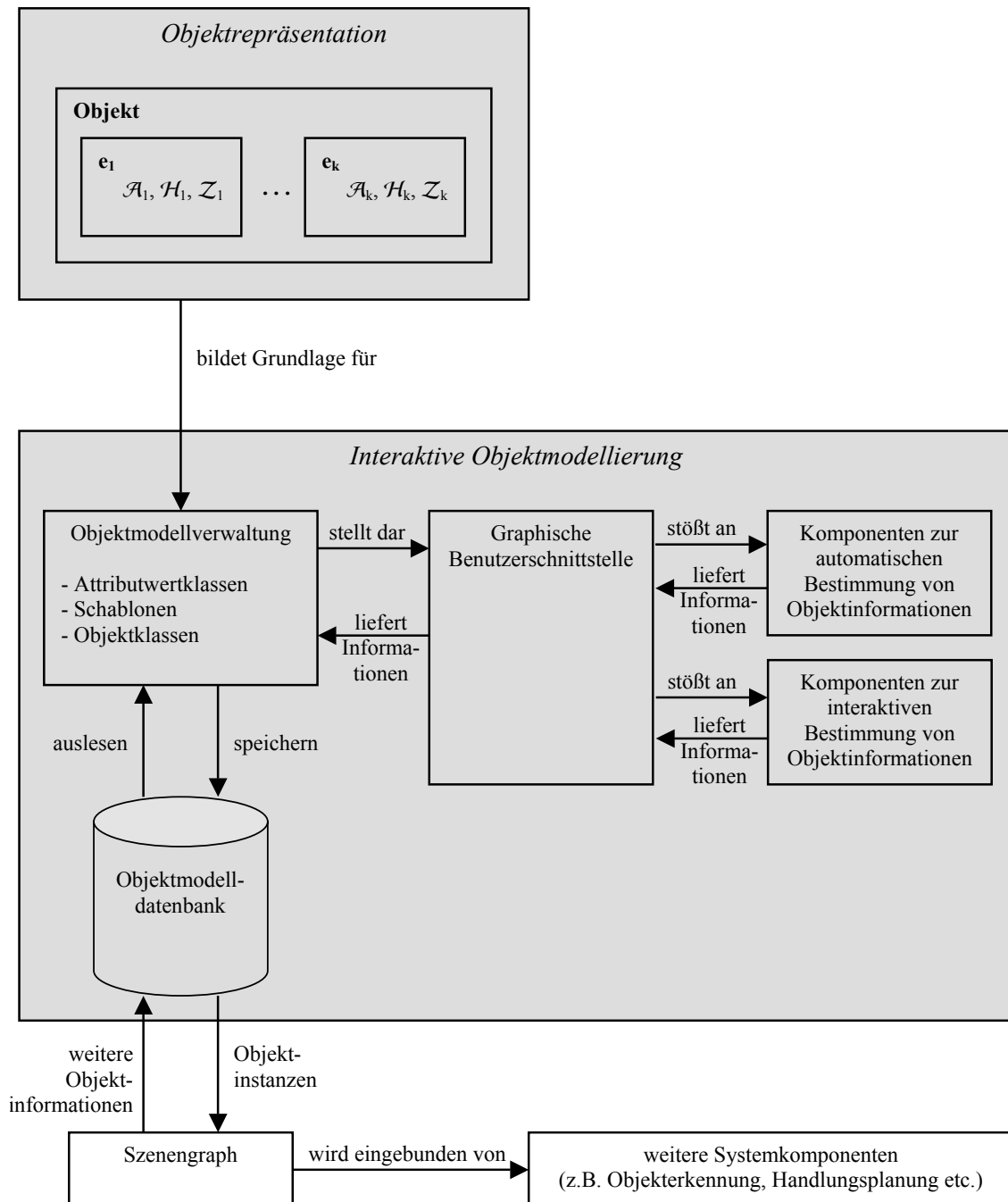


Abbildung 4.3: Architektur des interaktiven Objektmodellierungssystems IOMOS. Die dunkel hinterlegten Komponenten wurden in dieser Arbeit entwickelt und implementiert.

### Komponenten zur autonomen Erfassung von Objektattributwerten

Diese Bestandteile des interaktiven Objektmodellierungssystems dienen zur Erfassung von Objektinformationen, für die keine Interaktion mit dem Benutzer notwendig ist. Dabei bleibt der Ablauf der Modellierung in der Hand des Benutzers, der die Bestimmung einzelner Objektattribute veranlasst und die solcherart erhaltenen Informationen interaktiv korrigieren, ergänzen oder verwerfen kann. Kap. 5 stellt die Komponenten, die Sensorik und den Ablauf der autonomen Objekterfassung im Detail dar.

### Objektmodelldatenbank

Die Objektmodelldatenbank dient der dauerhaften Speicherung erstellter Objektmodelle sowie der Schablonen für Eigenschaften, Attribute etc. Ihre Struktur wird in Abschnitt 4.3.2 vorgestellt. Die dort abgelegten Objektinformationen können dann zur Laufzeit ausgelesen und beispielsweise im Szenengraphen verwendet werden. Umgekehrt ist die Neuanlage, Veränderung und Ergänzung von Objektinformationen über das interaktive Objektmodellierungssystem möglich, aber auch eine Ergänzung von Informationen aus der sensorischen Verarbeitung auf einem Robotersystem, soweit dabei dauerhaft gültige Objektinformationen erlangt werden können. Nur temporär bzw. für einzelne Objektinstanzen in einer bestimmten Szene relevante Informationen wie der aktuelle, veränderliche Füllstand einer bestimmten Tasse, werden in der Regel nicht dauerhaft in der Objektmodelldatenbank abgelegt.

### Szenengraph und weitere Roboterkomponenten

Eine typische Anwendung, die auf die gelernten Objektmodelle zugreift, ist ein auf dem Roboter integrierter Szenengraph, der als Schnittstelle zu anderen Roboterkomponenten dient. In ihm werden jeweils diejenigen Objektinstanzen eingeladen und vorgehalten (vgl. auch die Erläuterungen in Abschnitt 4.3.3), die in der aktuellen Szene zu finden sind. Dies kann die Umgebung eines Serviceroboters sein, in der sich eine gewisse Anzahl an Haushaltsgegenständen befindet. Die Art und Anzahl der Objekte in der Szene kann zur Laufzeit beliebig variiert werden, wenn sich die reale Umwelt bzw. ihre Wahrnehmung durch den Roboter entsprechend ändert. Über den Szenengraphen sind die erstellten Objektmodelle dann anderen Roboterkomponenten zugänglich, z. B. den Komponenten zur Sprachverarbeitung, Objekterkennung, Handlungsplanung, Pfadplanung etc. Diese können auch die Modelle der Objektinstanzen im Szenengraphen bei Bedarf verändern bzw. durch weitere Informationen ergänzen, beispielsweise wenn durch Beobachtung festgestellt wird, dass der Benutzer eine Tasse befüllt. Im Szenengraphen wird dann auch zusätzliche szenenspezifische Objektinformation wie die Position der Objekte in der Szene oder Relationen zwischen Objekten vorgehalten. Einige mögliche räumliche und semantische Relationen sind Abschnitt 6.3.2 dargestellt. Der Zusammenhang zwischen der Objektmodelldatenbank, dem Szenenmodell und weiteren Bestandteilen der Roboterarchitektur wird in Abschnitt 7.3.2 ausführlich diskutiert.

## 4.3 Datenverwaltung, Speicherung und Darstellung von Objektmodellen

Neben der interaktiven Erstellung und Verwaltung der gewünschten Objektmodelle und Schablonen (vgl. Abschnitt 4.4.1) muss das interaktive Objektmodellierungssystem auch den Zugriff verschiedenster anderer Systemkomponenten gewährleisten sowie unterschiedliche Methoden zur Speicherung bzw. Vorhaltung der interaktiv erstellten Objektmodelle und Modellbestandteile für die Verwendung auf einem Serviceroboter realisieren. Zum einen ist hierbei eine dauerhafte Speicherung der Daten erwünscht, da die Modelle nicht bei jedem Systemstart neu erstellt werden sollen und können. Zum anderen müssen die Objektmodelle für einen Online-Zugriff anderer Roboterkomponenten in geeigneter Weise verfügbar



gemacht werden. Dies erfordert echtzeitfähige Zugriffsmöglichkeiten für Anwendungen im laufenden Roboterbetrieb. Und schließlich ist eine Möglichkeit erforderlich, die Inhalte der Objektmodelle darzustellen, um sie dem menschlichen Benutzer zu präsentieren.

### 4.3.1 Verwaltung der Objektinformationen

Um eine sinnvolle Nutzung von Objektdaten zu gewährleisten, sind folgende interaktive Aktionen zur Datenverwaltung erforderlich:

- Anlegen von neuen Objektmodellen
- Löschen von vorhandenen Objektmodellen
- Verändern von Teilinformationen in Objektmodellen
- Hinzufügen und Entfernen von Teilinformationen
- Aussagekräftige Anzeigemodalitäten für komplette Objektmodelle und Teile daraus
- Dauerhaftes Speichern und Abrufen von Objektmodellen

Zusätzlich ist für gegebene Domänen eine Schablonen-Verwaltung von Objektmodellbestandteilen hilfreich, wie sie in Abschnitt 4.4.1 beschrieben wird, um den Erstellungs- und Verwaltungsaufwand von Objektmodellen zu minimieren. Auch die in Abschnitt 4.4.2 beschriebenen a priori festgelegten Wertklassen für Attributwerte tragen hierzu bei, indem beispielsweise besondere Anzeigemodi für bestimmte Wertklassen gezielt implementiert werden können, wie eine 3D-Ansicht für Punktwolken. Das Speichern der Objektmodelle im interaktiven Modellierungssystem IOMOS wird im nächsten Abschnitt detaillierter beschrieben. Die interaktive Verwaltung der Objektdaten erfolgt in IOMOS hauptsächlich über eine graphische Benutzerschnittstelle, aus der heraus einzelne Module gezielt aufgerufen werden können. Die Schnittstelle ist in Abschnitt 5.4.1 detailliert beschrieben. Details der Datenverwaltung werden in Abschnitt 4.4 über die Implementierung der Objektrepräsentation im Detail dargestellt.

### 4.3.2 Objektmodelldatenbank und Speicherung von Objektmodellen

Die dauerhafte Speicherung der Objektmodelle und Schablonen geschieht in einer so genannten Objektmodelldatenbank. Hierbei ist zum einen der Datenaustausch innerhalb des Systems zu berücksichtigen, zum anderen aber auch der Datentransfer zu und von anderen Komponenten des Robotersystems. Um eine einfache und flexible Speicherung sowie einen effizienten Zugriff anderer Komponenten auf die gespeicherten Modelle zu ermöglichen, wurden zwei unterschiedliche Systeme implementiert und getestet. Es handelt sich hierbei zum einen um eine MySQL-Datenbank (MySQL AB (2007)) mit Corba-Schnittstelle (Object Management Group (OMG) (2007)), zum anderen um die Speicherung der Objektdaten als

Dateien in einem Verzeichnisbaum. Es zeigte sich im Gebrauch, dass die erste Lösung mit expliziter Datenbank den Zugriff der internen Systemkomponenten, wie beispielsweise der in Abschnitt 5.5 beschriebenen Applikationen zur interaktiven Modellierung von Objektattributen), unnötig verlangsamte. Im Vergleich zur Speicherung als Dateibaum brachte sie keine wesentlichen Vorteile, da die Stärken einer Datenbank in diesem Systementwurf nicht ausgenutzt werden. In das System wurde daher die zweite Lösung, eine Speicherung von Objektinformationen als Dateien in einem Verzeichnisbaum, integriert.

Zur internen Verwaltung der Objektdaten greifen die Komponenten von IOMOS dabei direkt auf den jeweiligen Objektverzeichnisbaum zu. Da beispielsweise bei der Erfassung von Objektattributen teilweise große Datenmengen anfallen, hat diese Lösung den Vorteil, dass das Senden der entsprechenden Daten zwischen den einzelnen Systemkomponenten entfällt. Hierdurch wird die Verarbeitungsgeschwindigkeit deutlich verbessert, da jede Komponente nur die Daten ausliest, die auch von ihr benötigt werden. Als Schnittstelle zu anderen Systemkomponenten des Roboters dient eine CORBA-Schnittstelle, die allen anderen Modulen einen weitgehend plattform- und programmiersprachenunabhängigen Zugriff mit gekapselter Funktionalität ermöglicht. Die Objektmodelldatenbank und die Schnittstelle sind in Abb. 4.4 noch einmal im Überblick dargestellt.

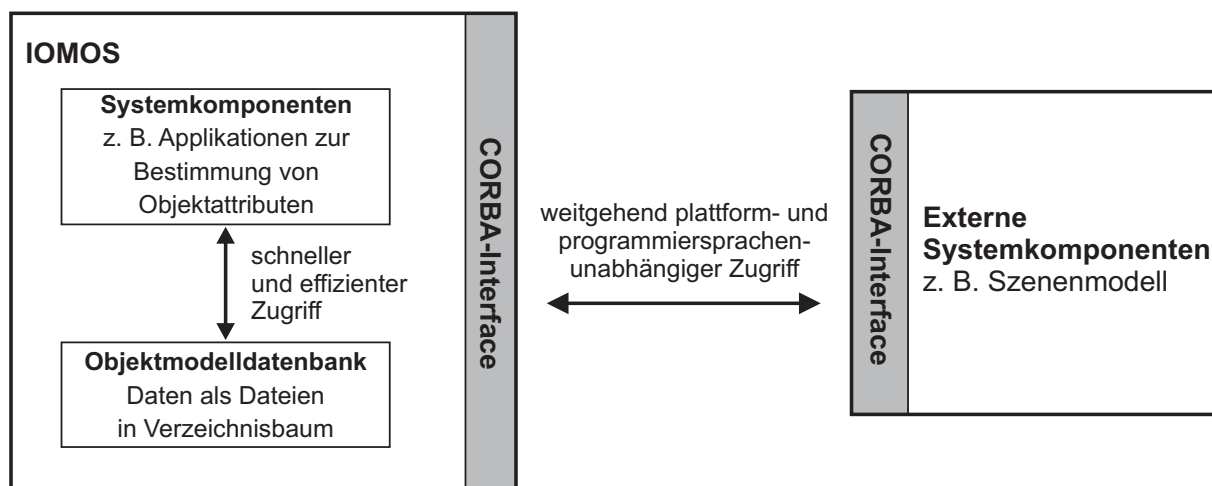


Abbildung 4.4: Schematische Darstellung der Datenablage in der Objektmodelldatenbank und des externen Zugriffs mittels Corba-Schnittstelle

Zusätzlich zur Speicherung ist eine geeignete Darstellung der Objektinformationen erforderlich, und zwar einerseits im Hinblick auf eine effiziente Speicherung und flexible Datenverwaltung, andererseits aber auch unter Berücksichtigung der Interaktivität des Systems, die eine übersichtliche und intuitive Darstellung von Informationen für den menschlichen Benutzer verlangt. Hierzu wurde eine Repräsentation der Objektmodelle in einem einfachen XML-Format gewählt. Diese Darstellung bietet folgende Vorteile: Flexibilität, Erweiterbarkeit, eine einfache Verarbeitung mit gängiger Software sowie die Möglichkeit, adäquate Darstellungsformen für unterschiedliche Inhalte zu wählen. Ein Ausschnitt aus einem solchen als XML-Datensatz gespeicherten Objektmodell ist in Abb. 4.5 beispielhaft wiedergegeben.

Eine Beschreibung der Objektrepräsentation in einer Meta-Sprache wie UML wurde im

Rahmen der vorliegenden Arbeit nicht vorgenommen. Das Hauptaugenmerk lag hier auf dem Entwurf einer Repräsentation, die die gegebenen Anforderungen möglichst gut erfüllt, insbesondere auch die Anlehnung an menschliche Formen der Objektrepräsentation. Bei einer Integration in ein übergeordnetes Modellierungskonzept wie UML besteht hingegen die Gefahr, dass konzeptionelle Modellierungsentscheidungen vornehmlich im Hinblick auf die Möglichkeiten getroffen werden, die das übergeordnete Modell zur Verfügung stellt. Dies sollte hier bewusst vermieden werden, um den gestellten Anforderungen möglichst gut gerecht zu werden. Hinzu kommt, dass speziell UML sehr stark darauf ausgerichtet ist, Konzepte für möglichst viele Modellierungsprobleme zur Verfügung zu stellen. Demzufolge hat es sich zu einer relativ komplexen und teilweise redundanten und ästhetisch inkonsistenten Modellierungssprache entwickelt. Durch Konzentration auf das Wesentliche, in diesem Fall die für die gegebene Aufgabe zu modellierenden Konzepte, konnte im Rahmen der vorliegenden Arbeit eine übersichtliche, konsistente und für den Menschen verständliche Objektrepräsentation geschaffen werden, die die Anforderungen der Zielanwendungen voll erfüllt.

### 4.3.3 Objektinstanzen zur Laufzeit

Zur Laufzeit eines Serviceroboters oder eines anderen auf die Objektdaten zugreifenden Systems sind Zugriffe auf die Objektmodelldatenbank bei Anfragen zu Objektmodellen aus Performanzgründen aufgrund der großen Datenmengen und der nicht garantierten Echtzeitfähigkeit von Corba nicht sinnvoll. Gefordert ist daher ein schneller echtzeitfähiger Weg zur Einbindung und Weiterverarbeitung der Daten, wie sie unter anderem im Szenengraphen notwendig wird. Da die externen Systemkomponenten überwiegend in C++ implementiert sind, können die Modelle der Objektinstanzen, die zur Laufzeit des Roboters benötigt werden, entsprechend aus den vorliegenden XML-Strukturen heraus als C++-Objekte aufgebaut werden. Dies erfolgt entweder zu Beginn der externen Anwendung durch einmaligen Corba-Zugriff auf die Objektmodelldatenbank, oder, falls Objektinstanzen nachträglich benötigt werden, zu einem beliebigen späteren Zeitpunkt. Für diese Konvertierung muss von der externen Anwendung eine Bibliothek eingebunden werden, die Informationen über die benötigten Wertklassen der Objekte enthält (vgl. Erläuterungen in Abschnitt 4.4.2).

Die solcherart aufgebauten C++-Objekte stehen den verarbeitenden Komponenten dann zur Verfügung und können von diesen bei Bedarf verändert oder ergänzt werden, falls neue Informationen über die Objektinstanzen vorliegen. Falls sich hierbei neue Langzeitinformationen über eine Objektinstanz ergeben, können diese nach einer Rückkonvertierung in das XML-Speicherformat zurück in die Objektmodelldatenbank geschrieben werden. Die Online-Objektinstanzen können beispielsweise in das Szenenmodell integriert werden. Sie stehen dann dort für andere Roboterkomponenten zur Verfügung und können für den menschlichen Benutzer als Bestandteil einer Szene visualisiert werden.

## 4.4 Implementierung der Objektrepräsentation

Bei der Implementierung der in Kap. 3 beschriebenen Objektrepräsentation ergeben sich einige interessante Details, die in den folgenden Abschnitte näher erläutert werden. Insbesondere

```

<ObjectClass>
  <Name>Tasse</Name>
  <isInstance>>false</isInstance>
  <Features>
    <Feature>
      <Name>ist_befuellbar</Name>
      <Attributes>
        <Attribute>
          <Name>aktuellerFuellstand</Name>
          <Reliability>0.900000</Reliability>
          <ValueClass>
            <Type>4</Type>
            <Specifics>
              <Name>Prozent</Name>
              <Timestamp>2007-03-01 15:47:22</Timestamp>
              <isDefault>no</isDefault>
              <Value>65</Value>
            </Specifics>
          </ValueClass>
        </Attribute>
        <Attribute>
          <Name>maximalerFuellstand</Name>
          <Reliability>1.000000</Reliability>
          <ValueClass>
            <Type>4</Type>
            <Specifics>
              <Name>Prozent</Name>
              <Timestamp>2007-02-16 13:51:26</Timestamp>
              <isDefault>yes</isDefault>
              <Value>95</Value>
            </Specifics>
          </ValueClass>
        </Attribute>
        ...
      </Attributes>
      <Actions>
        <Action>
          <Name>befuellen</Name>
          <RequiredAttributes>
            <AttributeName>aktuellerFuellstand</AttributeName>
            <AttributeName>maximalerFuellstand</AttributeName>
          </RequiredAttributes>
          <Conditions>
            <AlternativePreconditions>
              <Precondition>aktuellerFuellstand < maximalerFuellstand</Precondition>
            </AlternativePreconditions>
            <Postconditions>
              <Postcondition>aktuellerFuellstand <= maximalerFuellstand</Postcondition>
              <Postcondition>aktuellerFuellstand >= 80%</Postcondition>
            </Postconditions>
          </Conditions>
        </Action>
        ...
      </Actions>
      <States>
        <State>
          <Name>voll</Name>
          <RequiredAttributes>
            <AttributeName>aktuellerFuellstand</AttributeName>
          </RequiredAttributes>
          <Condition>(aktuellerFuellstand > 80%)</Condition>
        </State>
        ...
      </States>
    </Feature>
    ...
  </Features>
</ObjectClass>

```

Abbildung 4.5: Beispielhafter Ausschnitt aus einem in Form eines XML-Datensatzes abgelegten Objektmodell

haben diese Implementierungsentscheidungen Auswirkungen auf die Performanz und Benutzerfreundlichkeit des Systems, auch wenn sie nicht der Objektrepräsentation selbst inhärent sind.

#### 4.4.1 Schablonen

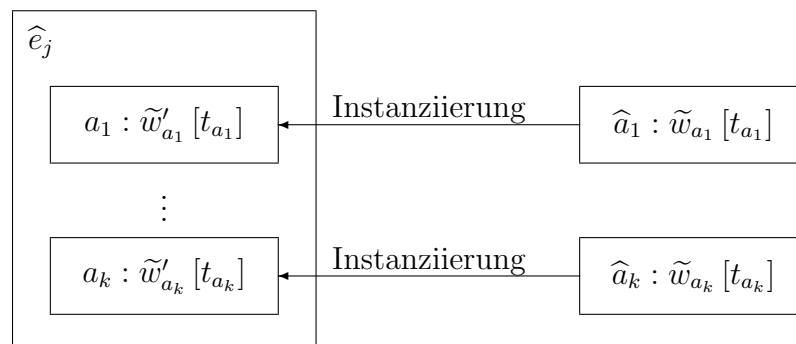
Um die interaktive Erstellung von Objektmodellen zu erleichtern und eine Anpassung an die vorgegebene Domäne zu ermöglichen, werden in IOMOS so genannte „Schablonen“ eingesetzt. Es handelt sich hierbei um Vorlagen für bestimmte Attribute, Eigenschaften, Zustände und Handlungen. Diese Schablonen können dann jeweils in Eigenschaften bzw. Objektmodellen eingebunden und mit konkreten Attributwerten etc. versehen werden. Beispielsweise kann für die Domäne „Haushalt“ eine Schablone für das Attribut *Füllstand* angelegt und gespeichert werden. Bei Anlegen der Eigenschaft **ist\_befüllbar** wird dann eine Instanz der Attributschablone *Füllstand* dieser Eigenschaft hinzugefügt und bei Bedarf mit einem anderen Attributwert versehen als in der ursprünglichen Schablone. Bei der Eigenschaft **ist\_befüllbar** selbst handelt es sich dann auch wieder um eine Schablone, von der Instanzen einzelner Objektklassen hinzugefügt werden können. Während dieses Prozesses können ebenfalls wieder die Attributwerte etc. der Schabloneninstanzen nach Bedarf angepasst werden. Die Schritte der Schablonenerzeugung und -verwendung als kaskadierte Instanziierung werden im Folgenden schematisch dargestellt:

1. Anlegen von Schablonen  $\hat{a}$ ,  $\hat{z}$  und  $\hat{h}$  für Attribute, Zustände und Handlungen, z. B. :

$$\hat{a}_k : \tilde{w}_{a_k} [t_{a_k}]$$

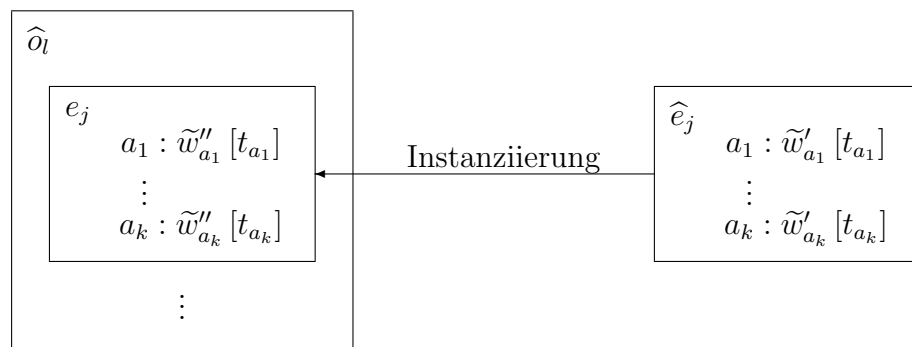
wobei  $\tilde{w}_a$  der Defaultwert des Attributs a ist.

2. Anlegen von Eigenschaftsschablonen  $\hat{e}$  und Einfügen von Instanzen der Attributs-, Zustands- und Handlungsschablonen in die  $\hat{e}$ , dabei gegebenenfalls Anpassung der Standardattributwerte von  $\tilde{w}_a$  in  $\tilde{w}'_a$ , z. B. :



wobei  $\tilde{w}'_a$  der neue Default-Attributwert des Attributes bei Verwendung in der entsprechenden Eigenschaft ist.

3. Erstellung von Objektklassen  $\hat{o}$  unter Verwendung von Instanzen der Eigenschaftsschablonen, dabei gegebenenfalls Anpassung der Default-Attributwerte  $\tilde{w}'_a$  in  $\tilde{w}''_a$ , z. B. :



wobei  $\tilde{w}''_a$  der neue Default-Attributwert des Attributes bei Verwendung in dieser Objektklasse ist. Bei Instanziierung der Objektklasse gilt dann dieser Default-Attributwert  $\tilde{w}''_a$  solange, bis der Attributwert der Objektinstanz zum ersten Mal zur Laufzeit umgesetzt wird. Ab diesem Zeitpunkt bezeichnet  $w_a$  den jeweils aktuellen Wert dieses Attributes in Bezug auf die Objektinstanz.

Diese Schablonen dienen zum einen der Benutzerfreundlichkeit und Übersichtlichkeit, da je nach Domäne häufig verwendete Eigenschaften, Attribute etc. nur einmal angelegt werden müssen und dann, entsprechend angepasst, jederzeit verwendet werden können. Zum anderen sind sie so implementiert, dass eine einmal angelegte Schabloneninstanz völlig von der ursprünglichen Schablone getrennt ist. Änderungen an Schablonen führen daher nicht zu Änderungen an bereits erstellten Objektmodellen, in denen diese Schablonen eingesetzt wurden, da ansonsten unkontrollierbare und für den Benutzer nicht mehr überschaubare Nebeneffekte auftreten könnten.

#### 4.4.2 Wertklassen

Zur Verarbeitung von Attributwerten werden in IOMOS so genannte „Wertklassen“ angelegt. Es handelt sich hierbei um in C++ implementierte Werttypen entsprechend der vorgestellten Objektrepräsentation. Auf diesen für jede Anwendung oder Domäne separat festzulegenden Datentypen werden zusätzlich Operatoren wie Speicheroperationen (die es unter anderem jeder Klasse ermöglichen, sich als XML-Datensatz zu speichern), Vergleichsoperationen (wie der Vergleich der Länge verschiedener Trajektorien) und Anzeigeoperationen implementiert (z. B. könnte für die Anzeige einer Punktwolken-Wertklasse eine dreidimensionale Darstellung gewählt werden). Die Art der Werttypen muss vor Einsatz des Systems festgelegt werden, da sonst weder eine Speicherung der Objektmodelle als XML-Datensatz noch die erneute Erstellung von C++-Objekten aus diesen XML-Datensätzen möglich ist, da das hierzu notwendige Grundwissen über die Datentypen in den Wertklassen abgelegt ist. Falls Datentypen verwendet werden, für die keine separate Wertklasse eingeführt werden soll, verwenden diese standardmäßig die Wertklasse „String“.

Die anzulegenden Wertklassen und insbesondere ihre Operatoren sind dabei anwendungsspezifisch. Unter Verwendung von Wertklassenoperatoren kann so beispielsweise im Hinblick auf

eine bestimmte Domäne bestimmt werden, welche von zwei Trajektorien die Längere ist, ob zwei Farben gleich sind, und dies falls gewünscht auch unter Bezug auf unterschiedliche Datentypen wie das RGB- und das HLS-Farbsystem, sofern entsprechende Vergleichsoperatoren implementiert sind.

### 4.4.3 Besonderheiten von Attributen

Attribute unterscheiden sich in drei Punkten von den anderen Objektbestandteilen wie Eigenschaften, Zuständen etc.: zum einen ist ihnen jeweils ein fester Attributwerttyp zugewiesen, zum zweiten können Attributwerte von den Attributwerten anderer Attribute abhängen (beispielsweise eine Bounding Box von der Ausprägung des 3D-Modells), und schließlich können Attribute mehrfach in einem Objekt vorkommen, und zwar sowohl als eigenständiges Attribut als auch unterschiedlichen Eigenschaften zugeordnet. Die Besonderheiten von Werttypen wurden im vorhergehenden Abschnitt bereits erläutert, hier werden nur noch die Implementierungsentscheidungen im Zusammenhang mit mehrfach auftretenden Attributen und Abhängigkeiten zwischen Attributen vorgestellt.

Zur Verwaltung von mehrfach auftretenden Attributen werden die Attribute jedes Objektes (sowohl für Objektklassen als auch für Instanzen) in einer von den Eigenschaften getrennten Attributliste gespeichert. Auch bei mehrfachem Auftreten wird jedes Attribut darin nur einmal abgelegt. Dies bedeutet insbesondere, dass gleiche Attribute innerhalb eines Objektes auch den gleichen Attributwert haben. Dies ist deshalb sinnvoll, weil eine Veränderung des Attributwertes (z. B. des maximalen Kippwinkels eines Behälters, weil er befüllt wurde) jeweils für das ganze Objekt in Effekt treten und frühere Einstellungen überschreiben sollte. Wenn ein Attribut mehrfach in ein Objekt eingefügt wird, ist somit ein definiertes Vorgehen zur Auflösung von Konflikten zwischen unterschiedlichen Attributwerten notwendig. Bei Einfügen eines Attributes in ein Objektmodell wird das Attribut, falls nicht bereits vorhanden, in der zentralen Attributliste abgelegt und der entsprechende Zähler für dieses Attribut inkrementiert. Der Attributwert wird dabei im Konfliktfall, d. h. bei mehrfachem Auftreten eines Attributes, nach folgenden Regeln bestimmt:

#### **Gleicher Attributwert**

Der Attributwert wird übernommen. Dies trifft auch zu, wenn beide Attribute noch den Default-Attributwert aus der Attribut-Schablone haben.

#### **Default-Attributwert vs. bereits gesetzter Attributwert**

Falls eins der Attribute noch den Default-Attributwert hat, der andere Attributwert aber bereits gesetzt wurde, wird der gesetzte Attributwert übernommen.

#### **Alle anderen Fälle von unterschiedlichen Attributwerten**

Der Attributwert wird auf „unbestimmt“ gesetzt, da das System nicht für jeden Fall verlässlich bestimmen kann, welcher Attributwert der Korrekte ist.

Bei Veränderung eines Attributwerts in einem Objektmodell wird er entsprechend dann auch für alle Instanzen dieses Attributs zentral verändert. Wenn ein Attribut gelöscht wird, wird

dessen Zähler dekrementiert, bei Löschung der letzten Instanz wird dann das zentral in der Attributliste gespeicherte Attribut endgültig entfernt.

Bei Abhängigkeiten zwischen Attributen, wie der einer Bounding Box von der 3D-Punktwolke, wird die entsprechende Abhängigkeit bei den Attributen gespeichert. Wird dann ein Attribut einer Eigenschaft zugeordnet oder in ein Objektmodell eingefügt, müssen auch diese Abhängigkeiten berücksichtigt und die jeweils zugrunde liegenden Attribute mit eingefügt werden.

## 4.5 Zusammenfassung

In diesem Kapitel wurden zunächst die Anforderungen an ein interaktives Modellierungssystem zur Lösung der gegebenen Problemstellung näher untersucht. Die Struktur des im Hinblick auf diese Anforderungen entwickelten interaktiven Objektmodellierungssystems IOMOS wurde danach detailliert erläutert. Die weiteren Abschnitte dieses Kapitels befassten sich einerseits mit der Datenverwaltung und der Speicherung und Darstellung von Objektmodellen in IOMOS, andererseits mit der Implementierung der im vorhergehenden Kapitel vorgestellten Objektrepräsentation. Es wurden jeweils die Besonderheiten des Systementwurfs und der Implementierung im Hinblick auf die gegebene Aufgabe vorgestellt. Eine detaillierte Erläuterung der einzelnen Komponenten zur Benutzerinteraktion und zur Modellierung von Objektattributen im Rahmen von IOMOS erfolgt dann im nächsten Kapitel.



## Kapitel 5

# Sensorgestützte interaktive Objektmodellierung

Dieses Kapitel stellt die im Rahmen der vorliegenden Arbeit entstandenen Komponenten zur interaktiven Objektmodellierung im Detail vor. Hierzu werden zunächst wesentliche Grundlagen einer effizienten Mensch-Maschine-Interaktion kurz dargestellt und die speziellen Anforderungen an das Interaktionssystem durch die gegebene Aufgabe erläutert. Im darauf folgenden Abschnitt wird die entwickelte interaktive Modellierungsumgebung mit ihrer speziellen Ausstattung zur Interaktion und Objektmodellierung vorgestellt. Die Abschnitte 5.4 und 5.5 schließlich präsentieren zunächst die neu entwickelten und in IOMOS integrierten Interaktionsmöglichkeiten und dann Implementierungen beispielhafter semi-autonomer und interaktiver Modellierungsverfahren. Eine kurze Darstellung der Integration der Komponenten zu einem Gesamtsystem schließt das Kapitel ab.

### 5.1 Grundlagen der Mensch-Maschine-Interaktion

Im Bereich der Mensch-Maschine-Interaktion existiert eine Fülle von Arbeiten zu unterschiedlichsten Aspekten dieser Interaktion. Oft werden dabei graphische Benutzerschnittstellen betrachtet (vgl. auch Referenzen in Abschnitt 2.4.1), aber auch verschiedene andere Problemfelder wurden bereits eingehend untersucht. An dieser Stelle soll daher kein vollständiger Überblick über dieses Forschungsgebiet gegeben werden. Stattdessen werden kurz die wesentlichen Aspekte einer effizienten Mensch-Maschine-Interaktion dargestellt, soweit sie für die interaktive Objektmodellierung von Bedeutung sind. Für einen Einstieg in dieses umfangreiche Themengebiet bieten sich beispielsweise Preim (1999), Heinecke (2004) Dix et al. (2006) oder Dahm (2006) an.

Für graphische Benutzerschnittstellen existiert eine Fülle an spezieller Literatur zur Gestaltung und zu Entwurfsprinzipien, z. B. Galitz (1997), Zetie (1995) oder speziell für Windows-Oberflächen Wessel (1998). Dabei werden jedoch in der Regel nur die bereits für den allgemeineren Fall festgelegten Prinzipien auf die spezielle Darstellungsform übertragen und Gestaltungsprinzipien bis hin zur Icon-Gestaltung oder farblichen Gestaltungselementen vorgeschlagen. Andere Arbeiten betrachten hingegen sehr schnittstellenspezifische Probleme wie

die Optimierung der Bedienzeit von mausgesteuerten graphischen Schnittstellen wie in Oel (2002).

Ein wesentliches Grundprinzip einer gelungenen Mensch-Maschine-Schnittstelle ist es, die Interaktion für den Menschen so einfach, intuitiv und angenehm wie möglich zu machen. Dies bedeutet, dass auch interne maschinelle Repräsentationen an eine menschengerechte Darstellung angepasst werden müssen, um eine bestmögliche Verständlichkeit und übersichtliche Darstellung für den Benutzer zu erreichen. Eine gelungene Mensch-Maschine-Schnittstelle hat somit Auswirkungen auf das maschinelle System, die über die eigentliche Funktionalität der Schnittstelle hinausreichen (vgl. die Überlegungen zum Entwurf der semantischen Objektrepräsentation in Abschnitt 3.2). Speziell beim Entwurf von interaktiven Systemen bzw. von deren Schnittstellen werden unter anderem folgende Punkte als wesentlich erachtet:

- Die Benutzerschnittstelle soll sich an den Fähigkeiten und Besonderheiten der zu erwartenden Benutzer orientieren.
- Der Mensch soll nicht mit Routineaufgaben belastet werden.
- Die Bedienmöglichkeiten sollen möglichst ergonomisch angeordnet und gestaltet sein.
- Das System soll möglichst tolerant gegenüber Anwenderfehlern sein und in einem sinnvollen Umfang auch das Rückgängigmachen ausgeführter Teilschritte ermöglichen.
- Fehler sollen verständlich dargestellt werden.
- Soweit das im Rahmen der gegebenen Aufgabe sinnvoll ist, sollen nach Möglichkeit unterschiedliche Interaktionsmodi kombiniert werden.
- Information sollen nach inhaltlichen Gesichtspunkten klar gegliedert sein. Der Benutzer soll nach Möglichkeit nicht mit zu viel Information überfrachtet werden.

Diese und weitere Anforderungen an eine bestmögliche Mensch-Maschine-Schnittstelle werden in der Literatur in unterschiedlichen Formulierungen immer wieder genannt. Vereinfacht ausgedrückt lassen sie sich in der Anforderung zusammenfassen, dass das System zu jeder Zeit auf den Benutzer, seine Verarbeitungskapazitäten und seine bevorzugten Vorgehensweisen Rücksicht nehmen sollte. In der Praxis wird diese Anforderung allerdings oft nur rudimentär umgesetzt oder überhaupt nicht beachtet. Beim Entwurf des im Folgenden dargestellten Systems wurde auf Bedienerfreundlichkeit, übersichtliche Darstellung und intuitive Schnittstellen geachtet, insbesondere da die Eingabe von komplexen Objektinformationen bei ungeeigneter Schnittstellengestaltung nicht nur ungeübte Benutzer stark fordern kann. Im nächsten Abschnitt werden daher zunächst einige grundsätzliche Überlegungen zu den Anforderungen an das interaktive Objektmodellierungssystem IOMOS im Hinblick auf die Benutzerinteraktion dargelegt. Relevante Entwurfsentscheidungen und Besonderheiten der Benutzerinteraktion werden dann im weiteren Verlauf des Kapitels jeweils bei der Darstellung der Interaktionsmodi bzw. der zu erhebenden Objektinformation erläutert.

## 5.2 Anforderungen an die interaktiven Komponenten des Modellierungssystems

Die Anforderungen an das zu entwickelnde interaktive Modellierungssystem wurden durch die spezielle Anwendung vorgegeben. Insbesondere sollten die resultierenden Objektmodelle darauf ausgerichtet sein, von einem Serviceroboter in einem Haushalt verwendet zu werden. Grundlage der Anwendung war daher der Einsatz in einer hochdynamischen, menschenzentrierten Umgebung, in der sich vornehmlich nicht Spezialisten, sondern durchschnittlich vorgebildete Personen bewegen. Zu realisieren war somit zum einen eine unkomplizierte, intuitive Schnittstelle entsprechend der Ausführungen im vorhergehenden Abschnitt, zum anderen sollte aber der für eine solche Anwendung notwendigen Komplexität und Flexibilität der Objektmodelle Rechnung getragen werden (vgl. Ausführungen in Abschnitt 4.1). Aus diesen Überlegungen heraus wurden folgende Anforderungen an die interaktiven Komponenten des Modellierungssystems ausgearbeitet:

Es wurde davon ausgegangen, dass das interaktive Modellierungssystem IOMOS derzeit noch von Benutzern mit technischen Vorkenntnissen bedient wird, die aber nicht notwendigerweise an der Entwicklung des Systems oder der Objektrepräsentation beteiligt waren. Dies bedeutet, dass die Interaktionskomponenten für einen technisch vorgebildeten Benutzer ohne spezielle Systemkenntnisse intuitiv und gut bedienbar sein sollten. Auf längere Sicht (z. B. bei einem System, das zukünftig auf Servicerobotern im Haushalt integriert sein wird) sollte dann auch von durchschnittlich erfahrenen Benutzern in Haushaltsumgebungen ausgegangen werden. Auch dann wird allerdings immer noch eine gewisse Affinität zur Bedienung technischer Geräte vorausgesetzt werden müssen. Nur so kann gewährleistet werden, dass die Anwender mit einem solchen komplexen System mit unterschiedlichen technischen Komponenten umgehen können und auch das Potential einschätzen können, das ein Serviceroboter in der Anwendung der interaktiv erstellten Objektmodelle aufweist.

Entsprechend wurde für die Interaktionskomponenten von IOMOS im derzeitigen Systemstand eine freie Benutzerführung gewählt, da von technischen Grundkenntnissen ausgegangen werden konnte. Beim Transfer eines solchen interaktiven Modellierungssystems in Alltagsumgebungen, in denen die Anwender kein oder nur wenig Vorwissen über den Aufbau der Objektrepräsentation und die Möglichkeiten und Einschränkungen der technischen Geräte mitbringen, ist auch eine stärkere Führung des Benutzers durch einzelne Modellierungsschritte denkbar, so wie manche anderen Softwaresysteme heute bereits einen „Expertenmodus“ und einen stark geführten Modus für „Laien“ anbietet. Andererseits sollte das System aber auch nicht durch zu weite Vereinfachungen die notwendige Komplexität und den Detailreichtum der Objektmodelle zu sehr einschränken oder die für eine sinnvolle Handhabung notwendige Flexibilität der Modelle durch zu starre Vorgaben beschneiden. Der Komplexität der zu modellierenden Information wird daher auch in diesem Fall Rechnung zu tragen sein. Derzeit ist die Interaktion zwischen Benutzer und Modellierungssystem deshalb vorwiegend benutzergeführt realisiert, da davon ausgegangen werden konnte, dass die Anwender den hierfür notwendigen Überblick besitzen.

Ein weiterer wesentlicher Punkt ist die Flexibilität des Systems im Hinblick auf das Hinzufügen weiterer Interaktions- und Modellierungskomponenten. IOMOS wurde bereits in der Planung auf mögliche Erweiterungen hin ausgelegt. Die Interaktionskomponenten sind auf

die graphische Benutzerschnittstelle zugeschnitten, so dass bei Einbindung zusätzlicher zu modellierender Objektattribute hierzu passende Interaktionskomponenten dann in die entsprechenden Applikationen der graphischen Benutzerschnittstelle integriert werden können. Bei der Datenübergabe zwischen der Hauptapplikation und den Unterapplikationen wurde ebenfalls auf eine entsprechende Flexibilität geachtet.

## 5.3 Aufbau der Modellierungsumgebung

Zur interaktiven Objektmodellierung wurde eine von den Anwendungssystemen getrennte Modellierungsumgebung entworfen und aufgesetzt (erstmalig vorgeschlagen in Becher et al. (2003); vgl. auch Eisenberg (2006) zum grundsätzlichen Aufbau der Modellierungsumgebung sowie Eisenberg (2006) und Kasper (2007) zu einzelnen Systembestandteilen). Die Nutzung dieser Modellierungsumgebung bietet folgende Vorteile:

### Hochgenaue Sensorik

In einer stationären Modellierungsumgebung kann hochgenaue Sensorik integriert werden, die derzeit aufgrund ihres Preises, ihres Gewichts und ihrer Abmessungen noch nicht zur Verwendung auf einem Robotersystem geeignet ist.

### Störungsfreie Objekterfassung

Eine anfängliche Erstellung von qualitativ hochwertigen Objektmodellen zur Referenz für spätere Anwendungen wird durch die Unabhängigkeit der Modellierungsumgebung von den Umgebungsbedingungen eines Roboters und von den Störungen in dessen Umgebung ermöglicht, beispielsweise durch die Unabhängigkeit von äußeren Beleuchtungsbedingungen.

### A priori-Erstellung von Objektmodellen für verschiedene Anwendungsbereiche

In solch einer Modellierungsumgebung können gleichzeitig Objektmodelle für unterschiedliche Anwendungsbereiche und verschiedene Robotersysteme erstellt werden, die dann nicht spezifisch für die Sensorik und Verarbeitungskapazitäten eines bestimmten Robotersystems ausgelegt, sondern universell einsetzbar sind.

Die Modellierungsumgebung enthält verschiedene Sensorkomponenten, die teilweise auch zur interaktiven Modellierung genutzt werden können. Die im Rahmen dieser Arbeit vorgestellte Methode ist aber nicht von der Verwendung dieser Modellierungsumgebung abhängig, sondern die vorgeschlagenen Objektmodelle sind so flexibel und erweiterbar gehalten, dass auch zur Laufzeit des Roboters jederzeit neue Modelle erstellt und bereits vorhandene Modelle um wichtige Informationen ergänzt werden können. Zukünftig ist auch eine Integration der interaktiven Objektmodellierung direkt auf einem Serviceroboter denkbar, da die zur Verfügung stehenden Sensoren mit entsprechender Größe und entsprechendem Gewicht immer hochauflösender und auch preislich attraktiver werden. Mittelfristig wird es daher durchaus möglich sein, die Aufgabe der interaktiven Objektmodellierung direkt auf Robotern, in intelligenten Räumen oder in ähnlichen maschinellen Systemen mit entsprechender Hardware-Ausstattung zu integrieren.

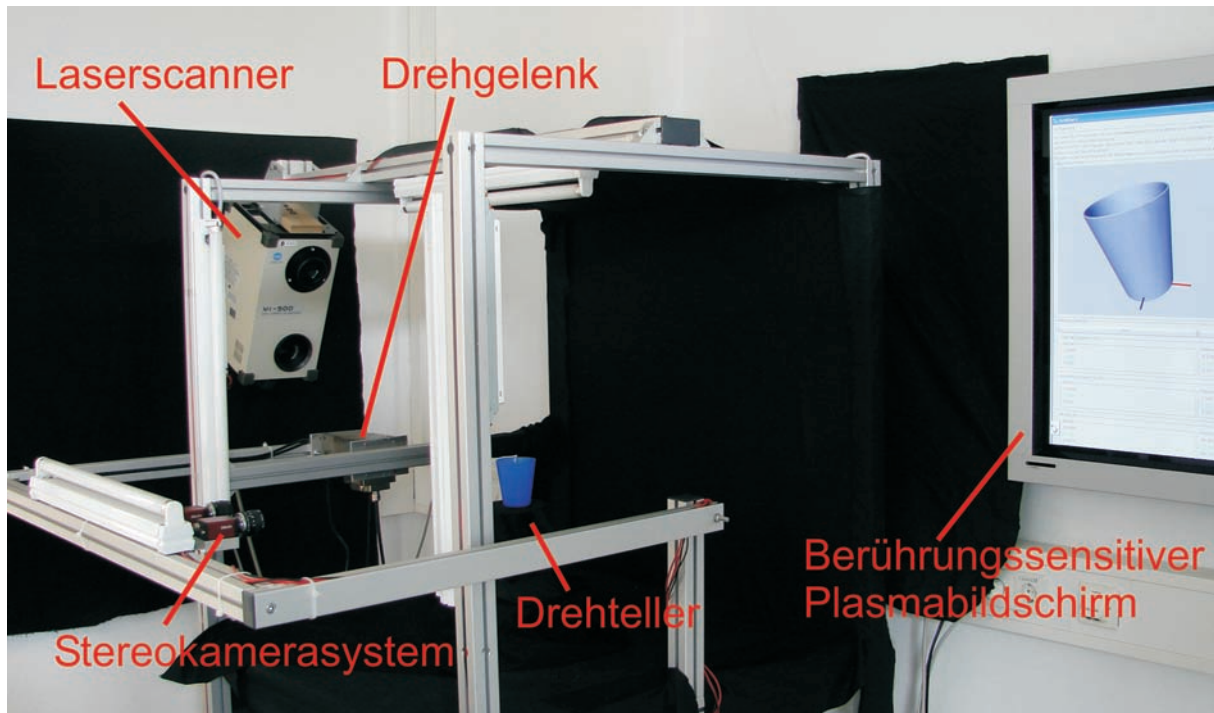


Abbildung 5.1: Gesamtansicht der interaktiven Objektmodellierungsumgebung

Abb. 5.1 zeigt die Modellierungsumgebung im Überblick. Im Folgenden werden ihr Aufbau und ihre Bestandteile beschrieben. Die technischen Daten der verwendeten Geräte sind in Anhang A wiedergegeben.

### Erfassungsumgebung

Die Modellierungsumgebung definiert ein abgegrenztes Raumvolumen, innerhalb dessen Objekte sensorisch erfasst werden (vgl. Abb. 5.2). Benutzeraktionen zur Erstellung von Objektmodellen finden teilweise innerhalb dieses Raumes statt (wie beim Abstellen eines Objektes zur sensorischen Erfassung), können teilweise aber auch außerhalb stattfinden (sprachliche Benutzeraktionen, Benutzereingaben an der graphischen Schnittstelle etc.), je nach Sensorik und Interaktionsmodi, die für die aktuelle Aufgabe relevant sind. Um möglichst störungsfreie Sensoraufnahmen zu gewährleisten, sind sowohl die Erfassungsfläche, auf der Objekte zur Modellierung abgestellt werden können, als auch der Hintergrund der Modellierungsumgebung mit schwarzem Stoff überzogen. Dies erhöht die Qualität der aufgenommenen Punktwolken und gestattet eine einfachere Weiterverarbeitung von Farbkameraaufnahmen. Die Abbildungen 5.3 und 5.4 zeigen schematisch den Aufbau der Erfassungsumgebung mit den integrierten Sensoren frontal und von oben.

### Hochauflösendes Farb-Stereokamerasystem auf schwenkbarer Halterung

Zur Erfassung von Objektansichten, Farb- und Texturinformationen wird ein Farb-Stereokamerasystem bestehend aus zwei Kameras vom Typ *Marlin 145C2* der Firma Allied Vision Technologies eingesetzt mit einer Auflösung von jeweils  $1392 \times 1040$  Bildpunkten (Allied Vision Technologies (2007)). Es ist auf einer schwenkbaren Halterung befestigt, so dass Objekte aus unterschiedlichen Höhenstellungen erfasst werden können (vgl. schematische Darstellung in Abb. 5.5). Diese Halterung besteht aus einem

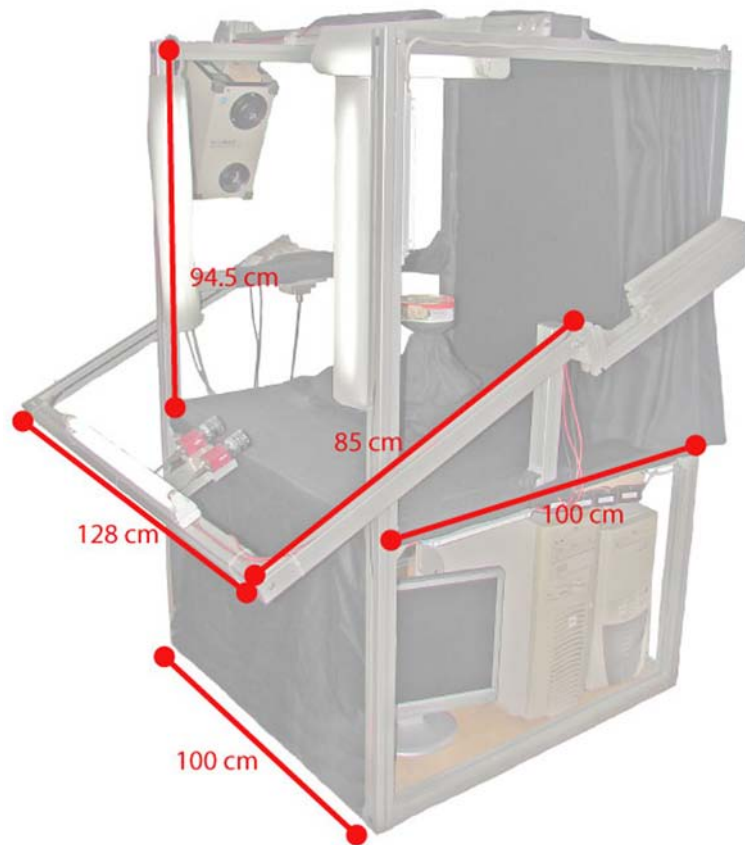


Abbildung 5.2: Abmessungen der Modellierungsumgebung

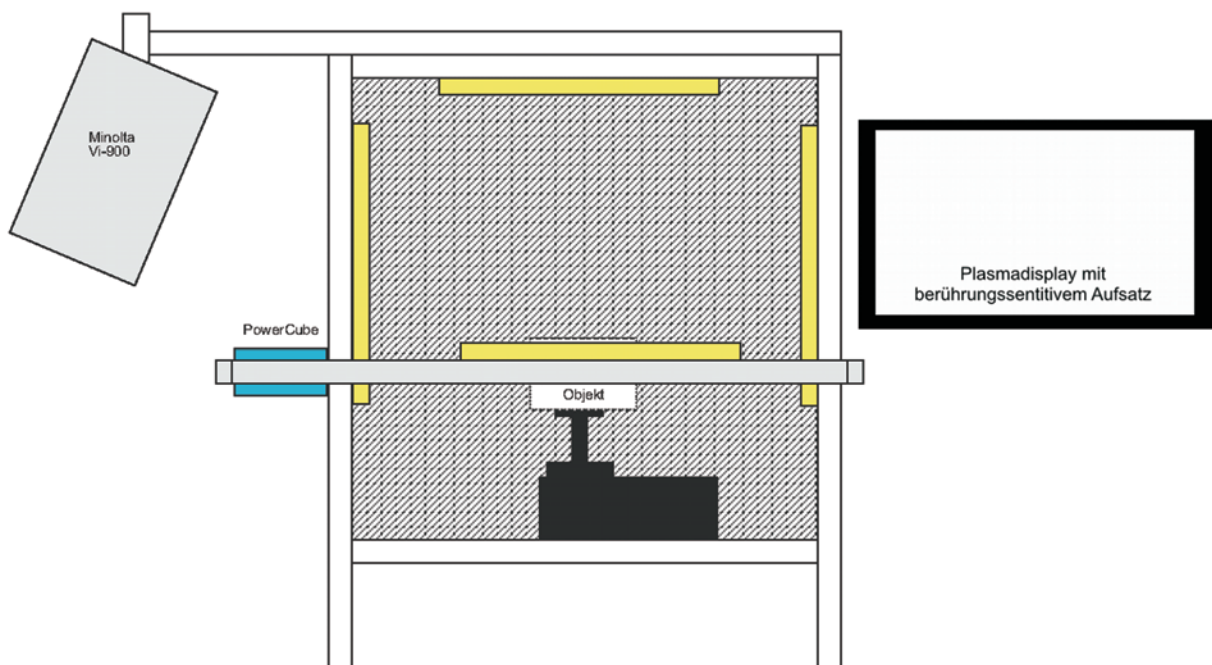


Abbildung 5.3: Schematische Darstellung der Erfassungsumgebung mit integrierter Sensorik von vorne, nach: Eisenberg (2006)

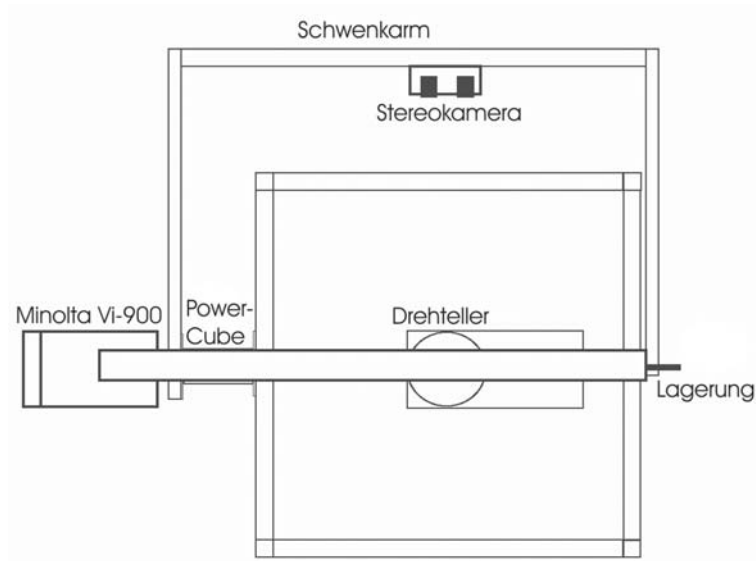


Abbildung 5.4: Schematische Darstellung der Erfassungsumgebung mit integrierter Sensorik von oben, aus: Eisenberg (2006)

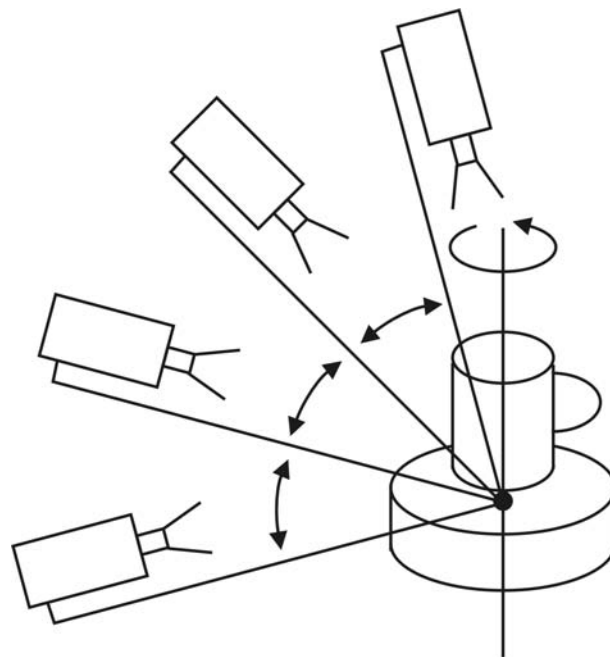


Abbildung 5.5: Schwenken des Stereokamerasystems zur Objektaufnahme aus unterschiedlichen Sichten

drehbaren Bügel aus Leichtbauprofilen, der auf der einen Seite an einem Drehgelenk befestigt und auf der anderen Seite drehbar gelagert ist. Er kann mittels des Drehgelenkes um bis zu ca.  $80^\circ$  nach oben und ca.  $25^\circ$  nach unten, relativ zur Nullstellung bei waagrechttem Bügel, gedreht werden. Somit sind beispielsweise auch Blicke von oben in eine Tasse oder andere Behälter hinein möglich, ohne dass das Objekt dafür auf die Seite gelegt werden muss. In Kombination mit dem integrierten Drehteller (siehe unten) ist dabei eine nahezu ebenso flexible Wahl der Aufnahme­richtung für Objektansichten möglich wie beispielsweise bei der in Abschnitt 2.3 vorgestellten Arbeit von



Fautz (2002) (vgl. die in Abb. 5.6 beispielhaft wiedergegebenen unterschiedlichen Kameraperspektiven). Allerdings ist in der vorliegenden Arbeit ein wesentlich geringerer technischer Aufwand notwendig, da kein Roboterarm benötigt wird. Bei dem Drehgelenk handelt es sich um einen Amtec *PowerCube PR 090* (Amtec Robotics (2007)), der über eine CAN-Schnittstelle angesteuert wird.

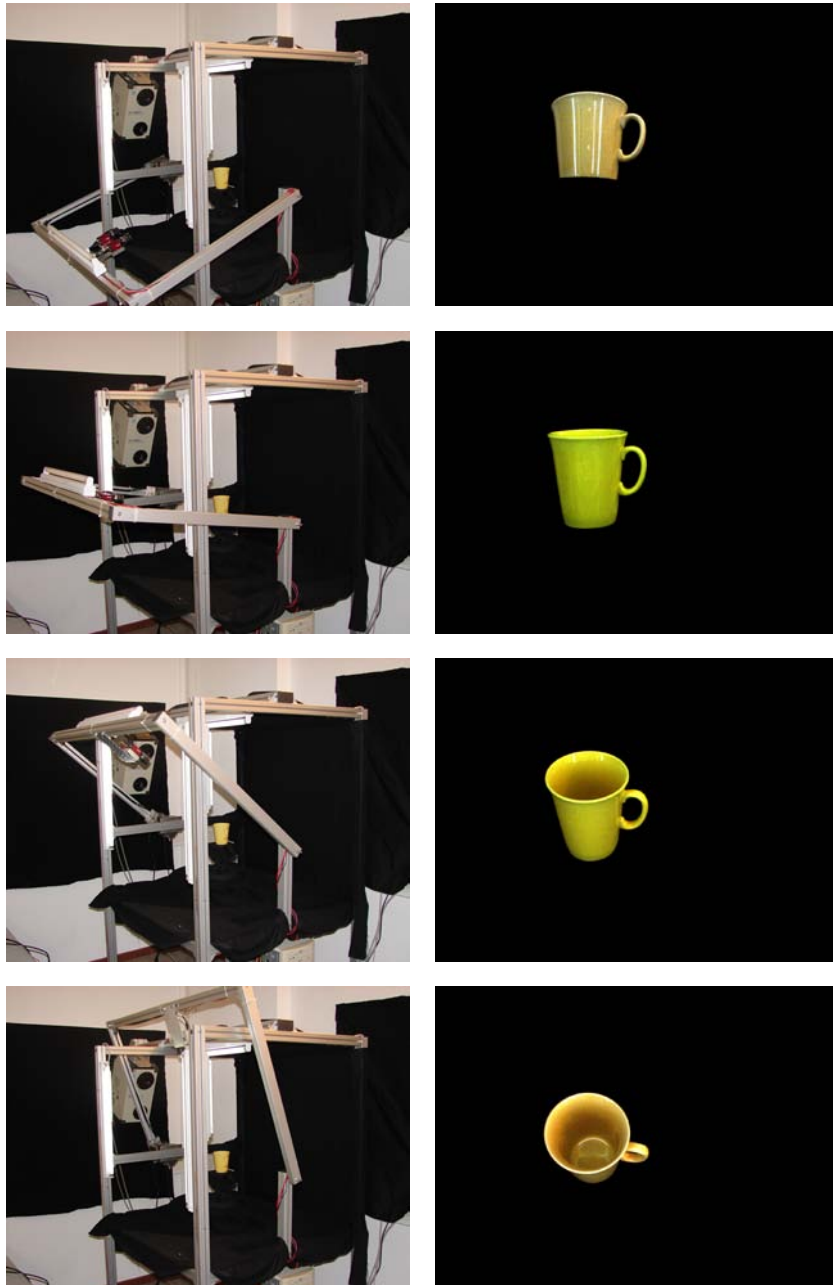


Abbildung 5.6: Unterschiedliche Kameraperspektiven mit jeweils korrespondierenden Objektansichten

Als problematisch bei dieser drehbaren Konstruktion erwies sich das Schwingen des Drehbügels, das wiederum zu einem leichten Schwingen der Kamera während der Aufnahmen führte oder alternativ, bei angemessenen Wartezeiten bis zur Beruhigung des Drehbügels, zu relativ langen Aufnahmezeiten. Die drehbare Lagerung auf der nicht aktiv gedrehten Seite des Bügels erwies sich dabei als nicht ausreichend, um das Schwin-



gen genügend zu reduzieren. Zur Schwingungsunterdrückung wurden die Regelungsparameter des PowerCubes modifiziert sowie die Drehgeschwindigkeit reduziert.

### Laserscanner

In einer Ausrichtung von  $90^\circ$  zur Stereokamera ist ein hochauflösender Laserscanner *VI-900* der Firma Minolta angebracht (Konica Minolta (2007)), der mittels des Verfahrens der aktiven Triangulation 3D-Punktwolken von Objekten mit Genauigkeiten im Submillimeterbereich erzeugen kann. Hierbei wird eine Laserlichtlinie auf das Objekt projiziert, und durch Bestimmung der Verzerrung der Linie auf dem Objekt werden Punkte auf der Objektoberfläche bestimmt. Dieser Sensor bestimmt dabei in 2,5 Sekunden eine Punktwolke der Objektoberfläche mit einer Auflösung von  $640 \times 480$  Bildpunkten mit einer maximalen Genauigkeit von bis zu 0,22 mm in Abhängigkeit von der Raumrichtung. Erfasst werden können je nach angebrachtem Objektiv Objekte bis zu einer Größe von ca. 1 m in einer Entfernung von ca. 0,6 bis 1,2 m.

Der Laserscanner ist derart in der Modellierungsumgebung eingestellt, dass sein Erfassungsbereich gängige Haushaltsgegenstände wie Geschirr, Lebensmittelverpackungen und andere Objekte dieser Größe abdeckt, die auf dem Drehteller platziert werden. Um auch Aufnahmen von oben beispielsweise in Behälter zu ermöglichen, wurde der Scanner leicht erhöht angeordnet, die Messungen erfolgen somit mit einer Neigung von ca.  $30^\circ$  nach unten (vgl. Abb. 5.3).

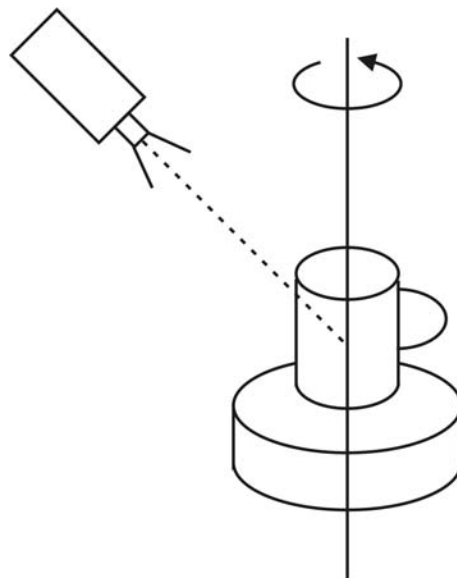


Abbildung 5.7: Schematische Darstellung des Drehtellers zur Rundumerfassung von Objekten

### Drehteller

Objekte können in der Modellierungsumgebung auf einem Drehteller platziert werden, so dass sie softwaregesteuert aus allen Richtungen von Laserscanner und Stereokamerasystem erfasst werden können (vgl. Abb. 5.7). Bei diesem Drehteller handelt es sich um einen über eine serielle Schnittstelle angesprochenen Isel *Turntable RF-1* (Isel Automation (2007)), der auf ihm abgestellte Objekte um seine senkrechte Rotationsachse dreht. Der gewählte Drehteller hat eine Positioniergenauigkeit von 2 Bogenminuten und wird über eine RS232-Schnittstelle angesteuert.

Anstatt wie bei Fautz (2002) die Sensoren um das zu modellierende Objekt herum zu bewegen, wurde in der vorliegenden Arbeit mit dem Drehteller der umgekehrte Weg gewählt: Die Sensoren sind fest installiert, wohingegen die Objekte bewegt werden. Dies bringt durch die Verwendung eines schnellen und hochpräzisen Drehtellers zum einen den Vorteil, dass ein gleichermaßen schnelles und genaues Bewegen der Sensoren relativ aufwändig wäre. Zum anderen ist es vorteilhaft, nur das zu modellierende Objekt zu bewegen anstatt mehrerer Sensorsysteme. Der technische und konzeptionelle Aufwand wird hierdurch wesentlich verkleinert.

### Rechnergesteuerte Beleuchtung

Die Beleuchtung der Modellierungsumgebung besteht aus vier Tageslicht-Leuchtstoffröhren mit einer Leistung von jeweils 18 W mit je einem Vorschaltgerät Philips HF-R 118 TLD zur Dimmung und zusätzlich einer Digital-/Analog-Wandlerkarte zur Erzeugung der Steuerungsspannung. Die Lampen sind so angebracht, dass sie die Erfassungsumgebung aus vier unterschiedlichen Richtungen beleuchten. Die Helligkeit der vier Lampen kann unabhängig voneinander per Software eingestellt werden. Durch Zu- und Abschalten einzelner Röhren kann auch die Einfallsrichtung des Lichts je nach Aufgabenstellung verändert werden. Die Leuchtstoffröhren sind austauschbar, so dass auch andere Beleuchtungssituationen realisiert werden können. Die Vorschaltgeräte haben eine Ausgangsfrequenz von ca. 42 kHz. Dadurch kann eine gleichmäßige Beleuchtung des Objekts ohne störende Helligkeitsschwankungen während der Objektaufnahmen sichergestellt werden.

Die Lampen sind wie folgt positioniert (vgl. Abb. 5.3): Je eine senkrechte Lampe ist schräg links bzw. rechts vor dem Drehteller an den Schienen der Modellierungsumgebung angebracht, eine weitere Lampe über dem Drehteller und eine Lampe am Drehbügel hinter dem Stereokamerasystem, um eine Beleuchtung aus der Blickperspektive eines Roboters simulieren zu können. Eine Beleuchtung direkt von vorne auf die Kameras ist bei der Erstellung von Objektmodellen nicht sinnvoll, da im Gegenlicht die Qualität der resultierenden Aufnahmen nicht ausreichend ist. Die Abbildungen 5.8 und 5.9 zeigen eine Sequenz unterschiedlicher Beleuchtungssituationen bei der Modellierung eines Objektes.

### Datenhandschuh

Um Fingerstellungen und Griffe des Benutzers im Umgang mit Objekten zu erfassen, wird ein Datenhandschuh vom Typ *CyberGlove* der Firma Immersion mit 22 Freiheitsgraden verwendet (Immersion (2004)). Der Benutzer kann mittels dieses Datenhandschuhs typische Handhabungen für bestimmte Objekte vorführen. Dadurch müssen abstrakte Werte wie Greifkräfte nicht abgeschätzt und textuell eingegeben werden, was insbesondere nicht speziell geschulten Benutzern in Alltagsumgebungen schwer fallen würde. Für den Einsatz des Datenhandschuhs ist aufgrund der Anordnung der Geräte die Nähe zur Modellierungsumgebung vorteilhaft. Der Benutzer muss das Objekt aber nicht zwingend innerhalb der Modellierungsumgebung halten, sondern kann auch daneben stehen und simultan mit der freien Hand die graphische Schnittstelle bedienen.

### Magnetfeldbasierter Positionssensor

Am Datenhandschuh ist ein magnetfeldbasierter Positionssensor *Fastrack* von Polhe-



Abbildung 5.8: Unterschiedliche Beleuchtungssituationen in einer Objektmodellierungssequenz (1). *Oben*: dunkelste Einstellung, *Mitte*: hellste Einstellung, *unten*: Lichteinfall von links

mus (Polhemus (2007)) zur Erfassung von Geschwindigkeiten und Beschleunigungen der Benutzerhand angebracht. Aufgrund des magnetfeldbasierten Messprinzips führen metallische Gegenstände, wie sie in der Modellierungsumgebung vorkommen, zu Feldverschiebungen. Zur Ermittlung von korrekten absoluten Positionswerten müsste das System daher sehr häufig nachkalibriert werden. Dies ist allerdings in der vorliegenden Anwendung nicht relevant, da hier zur Objektmodellierung keine absoluten Positionen erfasst werden. Vielmehr sollen nur relative Geschwindigkeiten und Beschleunigungen erfasst werden, damit der Benutzer beispielsweise komfortabel eine maximale Beschleunigung für einen gefüllten Behälter erfassen kann.

### Hochauflösender Plasmabildschirm mit berührungssensitivem Aufsatz

Eine wichtige Komponente der Interaktion zwischen Modellierungssystem und Benutzer ist die graphische Benutzerschnittstelle, die auf einem hochauflösenden FullHD-Plasmabildschirm mit berührungssensitivem Aufsatz dargestellt wird. Es handelt sich hierbei um einen Plasmabildschirm des Typs *TH-50PF9EK* von Panasonic mit einer Auflösung von  $1920 \times 1080$  Bildpunkten (Panasonic (2007)). Der zugehörige berührungssensitive Aufsatz *TY-TP50P8-S* des gleichen Herstellers arbeitet auf der Basis von Infrarot-Lichtschranken, die seitlich im Aufsatz befestigt sind. Die eigentlich



Abbildung 5.9: Unterschiedliche Beleuchtungssituationen in einer Objektmodellierungssequenz (2). *Oben*: Lichteinfall von links und vorne, *Mitte*: von oben, *unten*: von rechts

Berührung des Bildschirms ist somit nicht notwendig, es genügt, den Finger bis kurz vor den Bildschirm zu bewegen. Für die Berührungen wird mit diesem System eine Auflösung von  $2209 \times 1241$  Bildpunkten erreicht. Angegeben ist eine minimale Stiftgröße von  $6 \times 6$  mm, in der Praxis erwies sich bei Bedienung mit den Fingern eine Abmessung von ca.  $1,5 \text{ cm} \times 1 \text{ cm}$  als die minimale Größe von anzuklickenden Elementen in einer graphischen Oberfläche.

### Mikrofon

Zur Erfassung von Spracheingaben des Benutzers wird ein handelsübliches tragbares Mikrofon verwendet. Hier sind für die Zukunft auch aufwendigere Lösungen (z. B. kabellose Mikrophone oder Richtmikrophone) denkbar, um den Bedienkomfort noch weiter zu erhöhen.

### Inertial-Joystick

Die Interaktionsmodi, die dem Benutzer zur Verfügung stehen, werden ergänzt durch den Einsatz eines Inertialsensors *InertiaCube2* der Firma Intersense (InterSense (2007)) mit drei Freiheitsgraden, der Rotationen bis  $360^\circ$  alle Achsen im Raum erfassen kann. Das Referenzkoordinatensystem wird dabei durch das Magnetfeld definiert, das von einem internen Kompass bestimmt wird, der gleichzeitig auch zur Reduktion der Mess-

drift verwendet werden kann. Dieser Sensor wurde in einen handelsüblichen Joystick eingebaut, so dass der Benutzer mittels eines ihm vertrauten, ergonomisch gebauten Eingabegeräts intuitiv z. B. die Darstellung eines Objektmodells drehen kann. Zusätzlich können die Knöpfe des Joysticks dazu verwendet werden, Benutzerkommandos an das System zu übertragen. Dieses Eingabegerät kann somit auch dazu verwendet werden, dreidimensionale Modelle ergonomisch und intuitiv auf dem Bildschirm zu drehen, die ausgewählte Zielposition per Knopfdruck zu bestätigen und sie dann mit dem Drehteller auf das reale Objekt in der Modellierungsumgebung zu übertragen.

## 5.4 Benutzerinteraktion zur Objektmodellierung

IOMOS erlaubt unterschiedliche Interaktionsmodi zwischen Modellierungssystem und Benutzer. In diesem Abschnitt werden die integrierten Interaktionsmodi im Detail erläutert und ihre jeweiligen Nutzungsmöglichkeiten aufgezeigt. Die interaktive bzw. semi-autonome Ermittlung von Attributwerten unter Verwendung dieser Interaktionsmodi wird dann im nächsten Abschnitt an ausgewählten Beispielen vorgestellt.

### 5.4.1 Graphische Benutzerschnittstelle auf berührungssensitivem Bildschirm

Kernstück der Interaktion zwischen dem Benutzer und dem Modellierungssystem IOMOS ist die graphische Benutzerschnittstelle (engl. „graphical user interface“, GUI) des Systems. Die GUI bildet die Schnittstelle zwischen dem Benutzer und den restlichen Systemkomponenten, sowohl zwischen dem Benutzer und den nicht-interaktiven Komponenten wie der Implementierung der Objektrepräsentation oder der Objektmodelldatenbank, als auch zwischen dem Benutzer und den anderen Interaktionskomponenten, die in den folgenden Abschnitten beschrieben sind. Wie in Abb. 4.2 bereits dargestellt, werden die einzelnen Schritte des Modellierungsprozesses stets vom menschlichen Benutzer angestoßen. Dies geschieht über die GUI bzw. über die an die GUI angebundene sprachliche Eingabeschnittstelle. Weitere Interaktionsmodi und Formen der autonomen bzw. semi-interaktiven Erfassung von Objektinformation sind dem Benutzer gegenüber somit stets hinter der GUI verborgen, wie auch in Abb. 4.3 zu sehen ist. Das gleiche gilt für bereits modellierte und gespeicherte Objekte, die für Systemkomponenten zwar über die Objektmodelldatenbank und den Szenengraphen zugänglich sind, für den Benutzer aber stets in der GUI dargestellt werden. Dort kann er dann auch gewünschte Änderungen vornehmen und neu erstellte oder veränderte Objekte speichern und löschen.

Diese zentrale Stellung wurde der GUI deshalb zugewiesen, weil eine Bedienung über die graphische Oberfläche für große Teile des Objektmodellierungsprozesses intuitiv gestaltet werden kann und die vorhandenen Objektmodellinformationen strukturiert und übersichtlich dargestellt werden können. Zudem ist sie für den Benutzer einfach zu handhaben, und für die notwendige Zusammenführung aller Informationen in einer Modalität kommt aus Gründen der Übersicht ebenfalls nur eine graphische Benutzerschnittstelle in Frage.

Die GUI als zentrale Interaktionskomponente dient aber nicht nur zum Anstoßen der anderen autonomen und semi-autonomen Modellierungsmodule, sondern ist selbst auch ein wesentliches Mittel für den Benutzer, Objektinformationen an das System zu übermitteln. Dies sind insbesondere:

- Grundlegende Informationen wie der Name eines neuen Objektes oder seine Eigenschaften, die dann wiederum Attribute, Handlungen und Zustände mit sich bringen.
- Informationen, die das System nicht autonom ermitteln kann. Hierzu gehören unter anderem benutzerspezifische oder stark situationsspezifische Informationen wie das von einem Benutzer bevorzugte Geschirr oder die Bedeutung von Zuständen wie VOLL oder HALBVOLL innerhalb der gewünschten Domäne.
- Beiträge des Benutzers zu semi-autonom ermittelbaren Attributwerten wie die Auswahl von geeigneten Objektansichten aus einer größeren Menge ermittelter Ansichten.
- Korrekturen oder Ergänzungen zu autonom ermittelten Attributwerten, z. B. wenn vom System aufgrund eines Messfehlers ein offensichtlich falscher oder ungenauer Wert festgelegt wurde.

Um der Bedeutung der GUI als zentraler Interaktionskomponente Rechnung zu tragen, und um die Interaktion für den Benutzer so bequem wie möglich zu gestalten, wird die graphische Benutzerschnittstelle auf einem hochauflösenden Plasmabildschirm mit berührungssensitivem Aufsatz dargestellt, das in unmittelbarer Nähe der Modellierungsumgebung an der Wand befestigt ist, wie in Abb. 5.10 zu sehen ist.

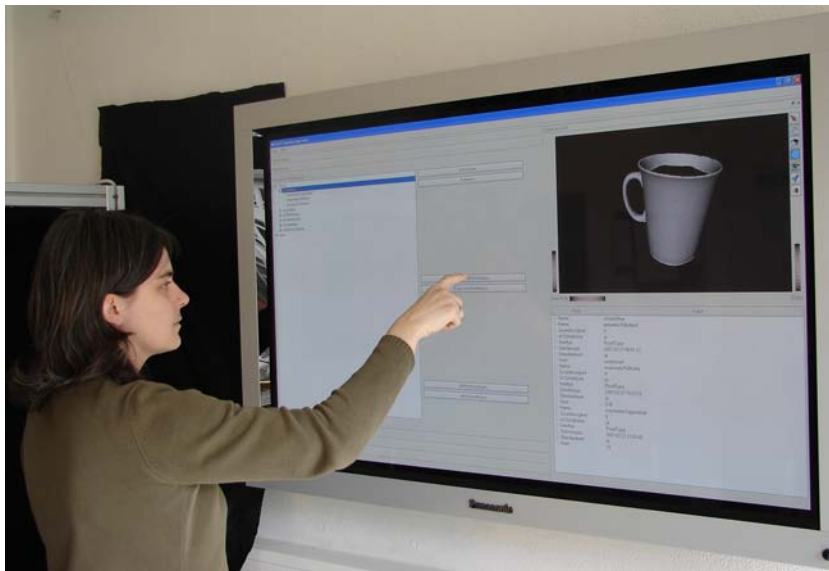


Abbildung 5.10: Bedienung der graphischen Benutzerschnittstelle auf dem berührungssensitiven hochauflösenden Plasmabildschirm

Die GUI wurde unter Verwendung von Qt 4.2 realisiert (Trolltech (2007)). Sie besteht aus einem Hauptfenster mit verschiedenen Reitern, in denen Objektklassen und Objektinstanzen sowie die unterschiedlichen Schablonen für Eigenschaften, Attribute etc. angelegt und

vom Benutzer mit semantischem Gehalt gefüllt werden können. Zur interaktiven oder semi-autonomen Ermittlung bestimmter Attributwerte, wie 3D-Punktwolken, Objektansichten, stabilen Flächen oder Manipulationseinschränkungen, werden dann jeweils Unterapplikationen geöffnet, die speziell auf die Modellierung dieser Attribute abgestimmt sind. Zur Datenübergabe zwischen Haupt- und Unterapplikationen, beispielsweise zur Übergabe der 3D-Punktwolke an das Modul zur Berechnung der Bounding Box, werden die Daten von jeder Applikation an einer definierten Stelle zwischengespeichert, so dass das jeweilige Empfängermodul die Daten dann bei Bedarf auslesen kann. Hierzu wird das in Abschnitt 4.3.2 kurz beschriebene XML-Datenformat verwendet. Abb. 5.11 zeigt einen Ausschnitt aus der Hauptapplikation der GUI.

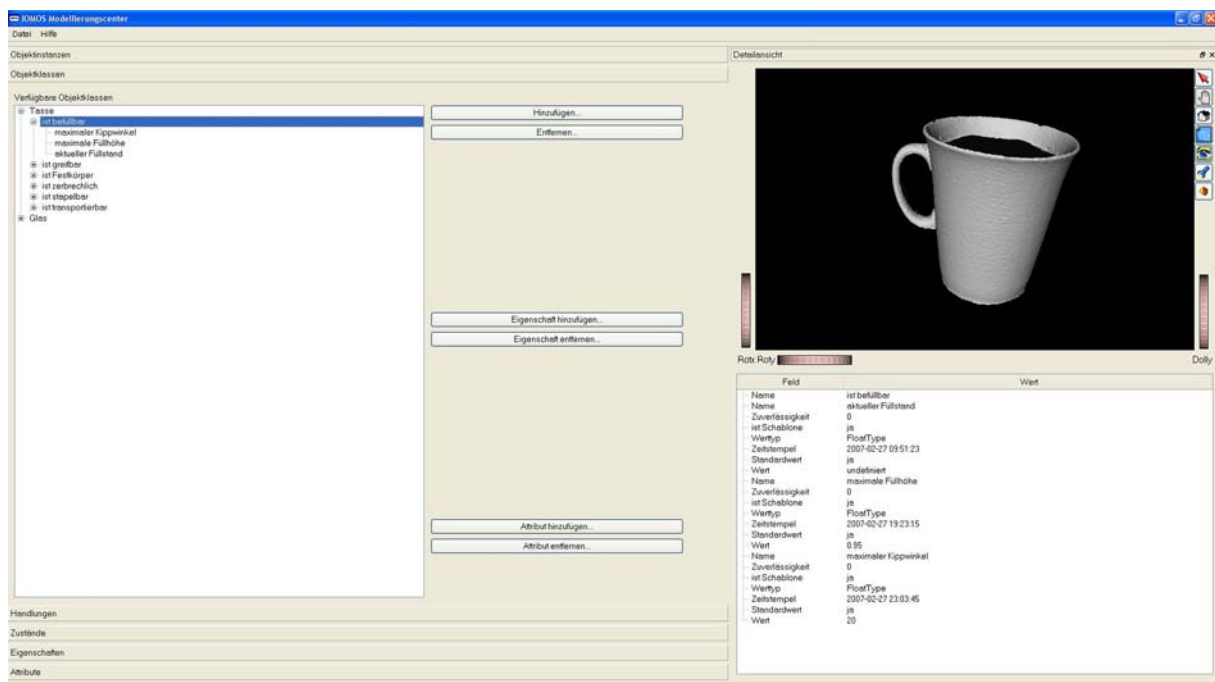


Abbildung 5.11: Ausschnitt aus dem Hauptfenster der graphischen Benutzerschnittstelle: Modellierung von Objektklassen

## 5.4.2 Spracheingabe

Mittels sprachlicher Kommunikation und Interaktion lässt sich für den Menschen eine der natürlichsten, intuitivsten und einfachsten Formen des Informationsaustausches überhaupt realisieren. Ein weiterer Vorteil dieser Interaktionsform ist, dass hierzu keine Hände oder andere Körperteile benötigt werden, und dass der Benutzer keine Bewegungen ausführen muss. Um die Bedienung von IOMOS für den Benutzer so angenehm und intuitiv wie möglich zu machen, wurde daher der deutsche Spracherkenner des Interactive Systems Laboratories (ISL, an der Carnegie Mellon University, USA, und der Universität Karlsruhe (TH)) zur Realisierung einer einfachen Form der sprachlichen Interaktion an die graphische Benutzerschnittstelle angebunden (vgl. Soltau et al. (2001) und Fügen et al. (2006)). Es wurde hierfür eine Einzelworterkennung realisiert mit einem Vokabular, das jeweils spezifisch für das gerade aktive Fenster der GUI ist. Hierdurch wird eine sehr gute Erkennungsqualität gewährleistet, da nicht ausführbare Befehle vom System auch nicht als solche erkannt werden.

Die sprachlichen Kommandos, die vom Spracherkennung über ein handelsübliches tragbares Mikrofon entgegengenommen werden, sind mit bestimmten Mauskommandos in der GUI verknüpft. So kann der Benutzer beispielsweise eine Taste wie „Speichern“ auf dem berührungssensitiven Bildschirm drücken, er kann aber auch per Spracheingabe das Kommando „speichern“ geben. Dies erlaubt es dem Benutzer, die Hände für andere Modellierungsaufgaben z. B. mit dem Datenhandschuh freizuhalten und vermeidet gleichzeitig ein zu manchen Zeiten ungelegenes Drücken der jeweils gewünschten Befehle auf dem großen Plasma-Bildschirm. Hierdurch werden Interaktionsgeschwindigkeit und -komfort erhöht.

Ein komplexeres Spracherkennungsmodul, möglicherweise auch mit Dialogmanager, ist zwar denkbar, aber in diesem Kontext nicht unbedingt notwendig, da nur eine einfache GUI-Steuerung erforderlich ist. Die rein oder überwiegend sprachliche Eingabe von komplexen Objektinformationen ist für den Benutzer mühsam und erfordert eine hohe Konzentration und einen guten Überblick über das gesamte Objekt, wohingegen die gewählte visuell-zentrierte Darstellungs- und Eingabeform auch für ungeübte Benutzer leicht zu bewältigen ist. Aus diesem Grund würde eine komplexere Sprachverarbeitung keinen wesentlichen Gewinn erwarten lassen, aber gleichzeitig das Risiko mit sich bringen, dass der Benutzer durch die Dialogführung zusätzlich belastet oder abgelenkt wird (vgl. hierzu auch Shneiderman (2000)). Zudem würde sich die Verarbeitungszeit wesentlich erhöhen, und auch das Risiko von fehlerhaften Erkennungen steigt deutlich mit der Komplexität der Verarbeitung. Daher beschränkt sich die vorliegende Arbeit auf eine einfache Spracheingabe auf Einzelwortbasis.

### 5.4.3 Joystick mit Inertialsensor

Wie bereits erläutert handelt es sich bei diesem Gerät um einen handelsüblichen Joystick, an dessen unterem Ende ein Inertialsensor des Typs *InertiaCube2* von Intersense befestigt wurde (vgl. Abb. 5.12 sowie Eisenberg (2006) für technische Details des Aufbaus). Im Gegensatz zu Gieslers „Zauberstab“ (Giesler et al. (2004)) handelt es sich hierbei nicht um ein mit optischen Markern bestücktes Zeigelement, dessen genaue Position und Ausrichtung dann kamerabasiert bestimmt werden. Stattdessen werden die relativen Änderungen der Orientierung hier direkt über den Inertialsensor bestimmt. Der Aufbau ist eher vergleichbar mit Wilson und Shafer's „XWand“ (Wilson und Shafer (2003)), wobei dieser für andere Anwendungen gedachte Sensorstab technisch deutlich komplexer gehalten ist und daher in seinen Möglichkeiten für einen ungeübten Benutzer intuitiv schwieriger zu überblicken und anzuwenden ist. Falls in IOMOS zusätzlich zur Orientierung auch die Position der Benutzerhand bestimmt werden soll, wird hierfür der magnetfeldbasierte Positionssensor verwendet, der in Abschnitt 5.4.5 beschrieben ist.

Mögliche Einsatzgebiete des Inertial-Joysticks sind zum einen die einfache und bequeme Drehung von dreidimensionalen Objektmodellen in der graphischen Oberfläche durch eine entsprechende Rotation dieses Eingabegeräts. Zum zweiten kann nach dieser Drehung unter Verwendung der Joystick-Knöpfe die gewünschte Zielposition des Objektes bestätigt werden, so dass dann das reale Objekt in der Modellierungsumgebung (mittels des Drehtellers) bzw. das Stereokamerasystem (mittels des Drehbügels) entsprechend für weitere Aufnahmen positioniert werden können. Eine Anwendung dieser Interaktionsmodalität zur Modellierung stabiler Objektflächen ist in Abschnitt 5.5.4 beschrieben. Die Drehung von Objektmodellen in der graphischen Oberfläche mittels des Inertial-Joysticks ist in Abb. 5.13 dargestellt.





Abbildung 5.12: Joystick mit integriertem Inertialsensor

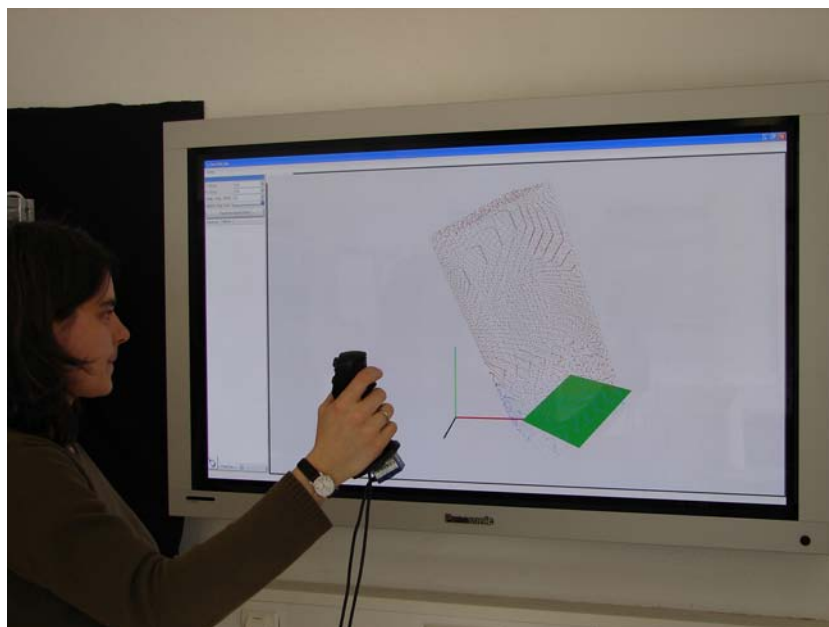


Abbildung 5.13: Einsatz des Inertial-Joysticks zur Rotation von dreidimensionalen Objektmodellen

Entsprechend ergeben sich bei der Einbindung dieses Gerätes in die interaktive Modellierungsumgebung folgende Teilprobleme: die Kommunikation und Datenübertragung zwischen GUI und Inertial-Joystick und die Anwendung der gemessenen Benutzerbewegungen auf das dreidimensionale Modell in der GUI, sowie die Umsetzung der als Quaternionen ausgelesenen Sensordaten des Inertialsensors auf die Position von Drehteller und Drehbügel.

#### 5.4.4 Datenhandschuh mit Kraftsensorik

Das interaktive Modellierungssystem wird ergänzt durch einen Datenhandschuh *CyberGlove* von Immersion mit 22 Freiheitsgraden, der zusätzlich noch mit 12 Kraftsensoren auf den Fingern und der Handfläche bestückt ist, wie sie im Rahmen der Arbeiten von Zöllner (vgl.



Abbildung 5.14: Datenhandschuh mit integrierter Kraftsensorik (Zöllner (2005)) und magnetfeldbasiertem Positionssensor

Zöllner (2005)) entwickelt wurden (Abb. 5.14). Diese Sensorkombination bildet die Grundlage dafür, dass zukünftig sowohl adäquate Griffotypen bzw. Greifansatzpunkte als auch mögliche Greifkräfte für Objekte interaktiv gelernt werden können. Im Bereich *Lernen von Greifinformationen* wurden u. a. einige Arbeiten von Zöllner et al. durchgeführt (einen guten Überblick gibt auch hier Zöllner (2005)), in denen diese Objektattribute erfolgreich bestimmt werden konnten. Ein Datenhandschuh mit einer entsprechenden Kraftsensorik erlaubt es dabei auch einem ungeübten Benutzer, objektspezifische Greifinformationen schnell und intuitiv an ein maschinelles System zu übermitteln. Eine Einbindung dieser Interaktionsmodalität in IOMOS ermöglicht zukünftig die einfache und hinreichend genaue Bestimmung der wesentlichen greifspezifischen Informationen für zu modellierende Objekte und ist deshalb für Anwendungen wie einen Serviceroboter im Haushalt empfehlenswert. Der Datenhandschuh ist kombiniert mit einem magnetfeldbasierten Positionssensor, der als Interaktionsmöglichkeit im nächsten Abschnitt näher erläutert wird. Derzeit wird der Datenhandschuh vorwiegend in dieser Sensorkombination eingesetzt, um somit einen definierten Befestigungspunkt für den magnetfeldbasierten Positionssensor auf dem Handrücken zu haben. Die eigentlichen Arbeiten zur Erfassung von Greifinformationen sind nicht im Rahmen der vorliegenden Arbeit entstanden, sondern sollen hier zukünftig eingebunden werden. Details zu diesen Anwendungen sind in der oben angegebenen Literatur ausführlich dargestellt.

#### 5.4.5 Magnetfeldbasierter Positionssensor

Am Handrücken des Datenhandschuhs ist ein magnetfeldbasierter Positionssensor *Fastrack* von Polhemus befestigt, wie in Abb. 5.14 zu sehen ist. Der Sensor wird in IOMOS zur interaktiven Modellierung von Bewegungseinschränkungen bei Manipulationen verwendet, insbesondere von maximalen Geschwindigkeiten, maximalen Beschleunigungen und maximalen Kippwinkeln von Objekten. Diese Informationen sind beispielsweise dann relevant, wenn ein gefüllter Behälter bewegt werden soll, und sind daher bei haushaltstypischen Verrichtungen häufig in Betracht zu ziehen. Da hier, wie oben bereits erwähnt, nur relative Bewegungen untersucht werden, ist die Störanfälligkeit dieses Messprinzips z. B. bei metallischen Gegenständen in der Umgebung unkritisch, da eine systematische Verschiebung der absoluten Positionswerte das Ergebnis nicht beeinflusst. Es ist einzig die relative Rotation zwischen

Weltkoordinatensystem und Objektkoordinatensystem zu betrachten, die initial zu Beginn einer Modellierungsphase bestimmt wird und eine Umrechnung der vom Sensor gelieferten globalen Eulerwinkel in das lokale Koordinatensystem vorzunehmen (vgl. Kasper (2007)). Die Ermittlung der relevanten Manipulationseinschränkungen wird in Abschnitt 5.5.6 im Detail erläutert. Abb. 5.15 zeigt den Einsatz des magnetfeldbasierten Positionssensors zur intuitiven Modellierung von Bewegungseinschränkungen.

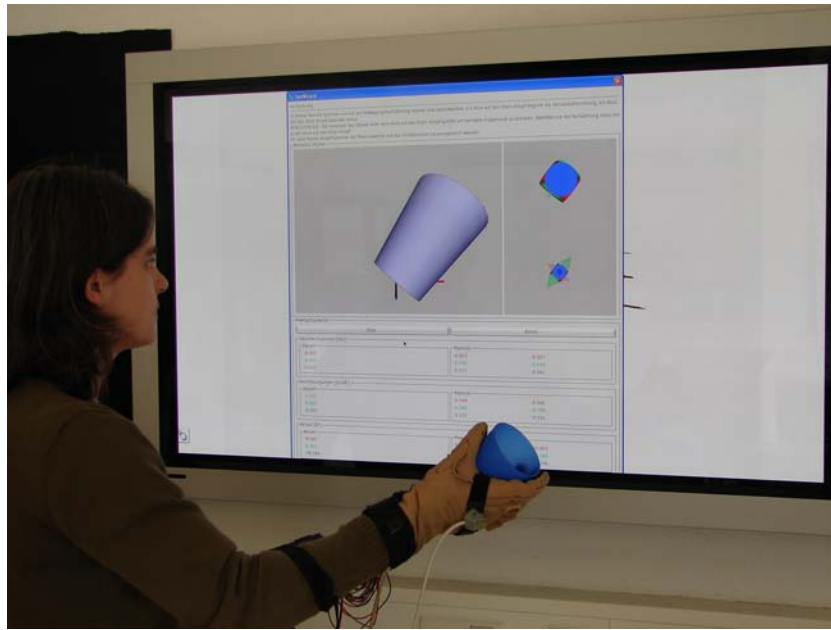


Abbildung 5.15: Anwendung des magnetfeldbasierten Positionssensors zur intuitiven Objektmodellierung

## 5.5 Interaktive bzw. semi-autonome Ermittlung von Objektattributen

Im Rahmen der vorliegenden Arbeit wurden einige Module zur interaktiven und zur semi-autonomen Bestimmung von Objektattributen in Verbindung mit der Modellierungsumgebung entwickelt und implementiert, die die obigen Interaktionsmodi einsetzen. Im Folgenden wird die Vorgehensweise zur Ermittlung einiger dieser Attribute detailliert beschrieben. Weitere Attribute können textuell über die graphische Benutzerschnittstelle erfasst werden, diese werden hier nicht näher erläutert. Auch die zugrunde liegende systematische Untersuchung und Klassifikation von relevanten Attributen für das Aufgabenfeld „Küche“ kann hier aus Platzgründen nicht im Detail dargestellt werden (siehe Abschnitt 6.3.4, für eine ausführliche Darstellung vgl. Kasper (2007)). Ziel dieser Arbeit war es dabei, wie bereits in Kap. 1 erläutert, ein schlüssiges Konzept zur interaktiven Modellierung zu erarbeiten und eine adäquate Objektrepräsentation zu entwerfen, wobei auch bereits bekannte Modellierungs- und Interaktionsmethoden sinnvoll integriert werden sollten. Die im Folgenden dargestellten Module zur Modellierung von Objektattributen wurden daher exemplarisch implementiert, um die Möglichkeiten des Systems aufzuzeigen und besonders interessante neue Interak-

tionsmöglichkeiten zu untersuchen. Sie dienen als Grundlage für die Systemevaluation in Kap. 7.

### 5.5.1 3D-Geometrie

Für jede Art von Manipulation, Bahnplanung und viele weitere Aufgaben eines Serviceroboters oder anderen maschinellen Systems ist ein geometrisches Modell der Umgebung und insbesondere der vorhandenen Gegenstände unerlässlich. Eine wichtige Aufgabe in IOMOS ist daher die zuverlässige, möglichst genaue Erfassung der 3D-Geometrie von Objekten sowie die Abspeicherung, die Anzeige und die Bereitstellung dieser Information für den Benutzer und für andere Systemkomponenten in einem geeigneten Format. Hierbei sind auch Performanzaspekte zu beachten, da teilweise große Datenmengen zu verarbeiten sind.

Die 3D-Geometrie von Objekten wird in IOMOS in Form einer Punktwolke mittels aktiver Triangulation durch einen hochauflösenden Laserscanner ermittelt, der stationär in der Modellierungsumgebung integriert ist (vgl. Abschnitt 5.3 sowie Anhang A für die technischen Daten des Geräts). Zur graphischen Darstellung für den Benutzer sowie für weitere Berechnungen kann diese Punktwolke bei Bedarf in ein Dreiecksnetz umgewandelt werden (vgl. Abb. 5.16). Zur dauerhaften Speicherung der Geometriedaten werden die Punktwolken dann als XML-Datensatz abgelegt wie in Abschnitt 4.3.2 beschrieben.

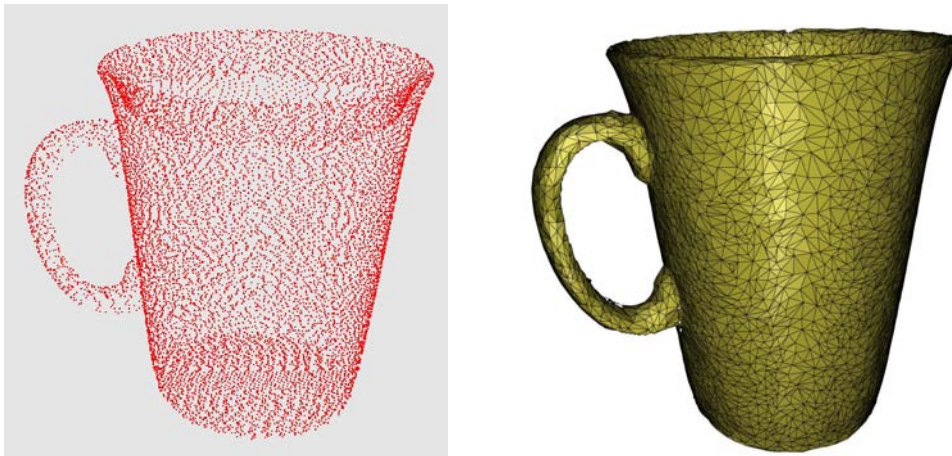


Abbildung 5.16: Punktwolke einer Tasse und zugehöriges Dreiecksnetz

Um ein vollständiges Modell einer Objektoberfläche zu erhalten, sind mehrere Aufnahmen aus verschiedenen Blickwinkeln notwendig. Der Benutzer kann zwar jede einzelne dieser Aufnahmen von Hand anstoßen, sinnvollerweise wird hierzu aber zunächst eine automatische Serie von Aufnahmen in festen Winkelabständen durchgeführt (vgl. Ausschnitt aus der graphischen Oberfläche zur Einstellung der Parameter in Abb. 5.17). Die einzelnen Punktwolken werden dann zu einer Gesamt-Punktwolke des Objektes fusioniert und dem Benutzer angezeigt. Je nach Ergebnis kann der Benutzer dann bei Bedarf eine weitere Detail-Aufnahmeserie in kleineren Winkelabständen starten oder einzelne zusätzliche Aufnahmen initiieren, beispielsweise wenn eine konkaver Teil der Objektoberfläche nicht vollständig erfasst wurde oder wenn im weiteren Verlauf der Anwendung eine besonders genaue Erfassung des Objekts aus einer bestimmten Sicht angezeigt erscheint. Ebenso können nachträglich einzelne Aufnahmen gelöscht werden, falls diese sich als überflüssig erweisen.

### 5.5.2 Objektansichten

Zur Aufnahme von Objektansichten wird das oben beschriebene Farb-Stereokamerasystem verwendet, das so auf einem schwenkbaren Arm befestigt ist, dass auch Objektaufnahmen von unten und von oben aus frei einstellbaren Winkelstellungen zwischen  $-25^\circ$  und  $80^\circ$  möglich sind. Zusätzlich kann das zu erfassende Objekt mittels des Drehtellers aus allen Richtungen aufgenommen werden. Objektansichten aus unterschiedlichen Aufnahmerichtungen sind eine wichtige Grundlage für eine ansichtsbasierte Objekterkennung, wie sie beispielsweise in Azad et al. (2006) vorgenommen wird. Aus den Stereoaufnahmen kann dabei bei der weiteren Verarbeitung der Objektmodelle auch eine Tiefenbestimmung erfolgen. Durch die frei einstellbare Intensität und Richtung der Beleuchtung können zudem bereits bei den Aufnahmen der Objektansichten unterschiedliche Beleuchtungssituationen simuliert werden, was eine spätere Anwendung der gewonnenen Objektinformationen in realen Alltagssituationen erleichtert.

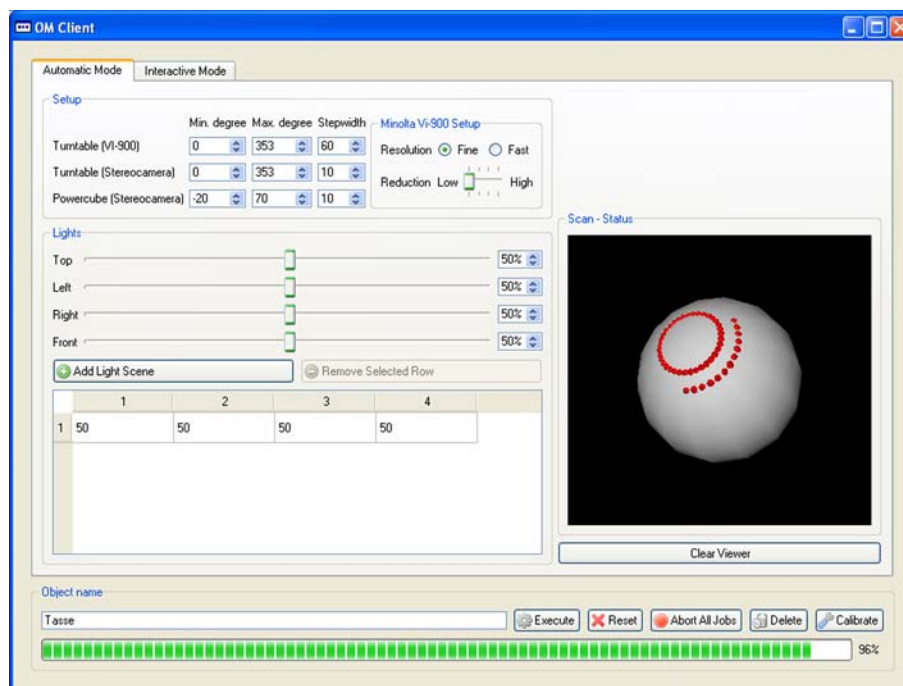


Abbildung 5.17: Graphische Bedienoberfläche zur Einstellung der Aufnahmeparameter von 3D-Geometrie und Objektansichten

Wie auch bei der Bestimmung der 3D-Objektgeometrie kann der Benutzer auch hier vorgeben, aus welchen Stellungen von Drehteller und Schwenkarm eine Aufnahmenserie erstellt werden soll. Möglich ist auch hier das nachträgliche Hinzufügen von Aufnahmen aus anderen Blickwinkeln bzw. das Löschen von nicht benötigten Aufnahmen. Beispielhafte Objektaufnahmen mit zugehöriger Kamerastellung sind in Abb. 5.6 dargestellt, unterschiedliche Beleuchtungssituationen bei der Aufnahme in Abb. 5.8 und 5.9. Abb. 5.17 zeigt die grafische Oberfläche zur Einstellung der Aufnahmeparameter von 3D-Geometrie und Objektansichten.

### 5.5.3 Silhouetten

Als „Silhouette“ wird in der Bildverarbeitung der Umriss eines Objektes bezeichnet. Diese Informationen werden beispielsweise zur Objekterkennung verwendet. Grundsätzlich sind unterschiedliche Verfahren zur Bestimmung von Objektsilhouetten anwendbar, insbesondere eine Berechnung auf der Basis von Punktwolken oder aus Kameraaufnahmen mittels Blob-suche. In der vorliegenden Arbeit wurde der zweite Weg gewählt, da die zur Verfügung stehenden Kamerabilder von hoher Auflösung sind und zusätzlich durch den vorhandenen schwarzen Hintergrund in der Modellierungsumgebung die Extraktion der relevanten Objektblobs im Kamerabild mittels eines Region Growing-Verfahrens erleichtert wird. Als „Blob“ werden in der Bildverarbeitung zusammenhängende Bildpunktregionen innerhalb eines Bildes bezeichnet, deren Zustand sich vom Rest des Bildes, dem so genannten „Hintergrund“, abgrenzen lässt. Im Folgenden handelt es sich bei den gesuchten Blobs immer um diejenigen Bildpunkte, die den modellierten Objekten zugehörig sind. Der Hintergrund ist aufgrund des Aufbaus der Modellierungsumgebung schwarz. Die Silhouette des Objektes ist dann der Umriss des das Objekt beschreibenden Blobs (vgl. Abb. 5.18).



Abbildung 5.18: Kameraaufnahme eines Objektes, daraus extrahierter Objektblob und resultierende Silhouette

Beim Region Growing-Verfahren handelt es sich um ein Bildsegmentierungsverfahren, bei dem ausgehend von initialen Zellen oder Bildpunktregionen möglichst große zusammenhängende Regionen in einem Bild gesucht werden. Hierzu werden gleichartige Elemente zu größeren zusammenhängenden Regionen verschmolzen, indem der bereits ausgewählten Bildregion ihre Nachbarbildpunkte hinzugefügt werden, falls diese der Region ähnlich genug sind. Solcherart wird das ganze Bild durchsucht, bis alle zusammenhängenden Regionen identifiziert werden konnten. Die Objektsilhouette kann somit autonom aus einem vorliegenden Kamerabild extrahiert werden und muss durch den Benutzer nur noch gegebenenfalls kontrolliert werden.

Das eingesetzte Verfahren ist in Alg. 5.1 dargestellt. Nach einer Konvertierung in ein Schwarz-Weiß-Bild und einer Binarisierung werden mittels einer Verkettung von morphologischen Öffnungs- und Schließungs-Operatoren kleinere Ungenauigkeiten und fehlerhafte Bildpunkte im Bild entfernt. Die verbleibenden Blobs werden bestimmt und Blobs unterhalb einer bestimmten Größe gelöscht, um kleine Artefakte aus der Bildverarbeitung und den Kameraaufnahmen zu entfernen. In der Praxis hat sich hier eine minimale Blobgröße von ca. zehn Bildpunkten bewährt. Berechnet werden jeweils die Silhouette mit Größe des Objektes, Anzahl der Silhouettenpunkte, Koordinaten der Punkte und Menge der Konturen pro Objekt (äußere begrenzende Kontur und innenliegende Konturen um Löcher im Objekt herum, wie beim Henkel einer Kanne).



<b>Eingabe:</b> $I_c$	/*Farbbild des Objektes*/
<b>Ausgabe:</b> $S = \{S_i\}$	/*Silhouetten*/
1: $I_b \leftarrow \emptyset$	
2: $I_b \leftarrow inGWumwandeln(I_c)$	/*Grauwertbild des Objektes erstellen*/
3: $I_b \leftarrow binarisiere(I_b)$	/*Grauwertbild binarisieren*/
4: $I_b \leftarrow \text{öffnenUndSchließen}(I_b)$	/*Fehler entfernen*/
5: $L_B \leftarrow \emptyset$	/*Liste der Blobs anlegen*/
6: $L_B \leftarrow blobsBerechnen(I_b)$	/*Blobs berechnen*/
7: <b>for all</b> $B_i \in L_B$ <b>do</b>	
8: <b>if</b> $B_i < \text{Schwellwert}$ <b>then</b>	
9: $löschen(B_i)$	/*Sehr kleine Blobs löschen*/
10: <b>else</b>	
11: $S_i \leftarrow silhouette(B_i)$	
12: <b>end if</b>	
13: <b>end for</b>	
14: <b>return</b> $S$	

Algorithmus 5.1: Ablauf der Silhouettenextraktion

### 5.5.4 Stabile Flächen

Stabile Flächen sind Flächen, auf denen ein Objekt in einem stabilen Gleichgewicht auf einer ebenen, ausreichend großen Unterlage abgestellt werden kann. Ein Beispiel für die stabilen Flächen eines haushaltstypischen Objektes ist in Abb. 5.19 dargestellt, wobei die zweite Fläche entlang des Griffes der Tasse zur besseren Übersichtlichkeit nicht dargestellt wurde. Die Information über stabile Flächen eines Objektes ist für weitere Systemkomponenten wie z. B. Handlungsplanung oder Greifplanung unerlässlich. Es ist dabei allerdings zu beachten, dass nicht alle stabilen Flächen auch tatsächlich in jeder Situation als Abstellfläche in Betracht kommen bzw. erwünscht sind. So kann beispielsweise der obere Rand einer Tasse als stabile Fläche betrachtet werden, ein Abstellen der Tasse mit der Öffnung nach unten auf einem Tisch ist möglich. Die Nutzung dieser stabilen Fläche kann jedoch in einer bestimmten Situation (Tasse gefüllt mit heißem Kaffee) unerwünscht sein, in einer anderen Situation jedoch sogar die einzige Abstellmöglichkeit sein, die aus dem Kontext heraus in Frage kommt (Abstellen der verschmutzten Tasse in der Geschirrspülmaschine). Die eigentliche Auswahl geeigneter stabiler Flächen ist jeweils auch vor dem Hintergrund des aktuellen Szenen- und Handlungskontextes zu sehen, die im Rahmen der interaktiven Modellierung gelernten stabilen Flächen sind somit nur als potentielle Abstellflächen zu betrachten.

Die stabilen Flächen eines Objektes werden unter Einbeziehung des Benutzers interaktiv aus der dreidimensionalen Objektgeometrie abgeleitet, indem jeweils eine geeignete Untergruppe der Punktwolke selektiert und daraus eine Ausgleichsebene berechnet wird. Deren Ortsvektor und Flächennormale repräsentieren dann die jeweilige stabile Fläche (vgl. Kasper (2007)). Dabei wird der oben beschriebene Inertial-Joystick (Abschnitt 5.4.3) eingesetzt, um die Darstellung der 3D-Geometrie in der graphischen Oberfläche in eine Position zu drehen, bei der die zu modellierende stabile Fläche sich unten am Objekt befindet, das Objekt sich also in der Stellung befindet, in der es üblicherweise auf dieser Fläche abgestellt würde. Diese Position wird dann bestätigt und für die untersten Punkten der Punktwolke eine Ausgleichsebene berechnet. Diese wird als stabile Fläche gespeichert. Der Vorgang wird vom



Abbildung 5.19: Potentielle stabile Flächen einer Tasse, aus: Kasper (2007)

Benutzer so lange wiederholt, bis alle gewünschten stabilen Flächen extrahiert sind. Alg. 5.2 zeigt das Vorgehen bei der interaktiven Bestimmung stabiler Flächen Schritt für Schritt.

<b>Eingabe:</b> $P$	<i>/*3D-Punktwolke des Objektes*/</i>
<b>Ausgabe:</b> $F$	<i>/*Liste der stabilen Flächen <math>F_1, \dots, F_n</math>.*/</i>
1: $anzeigen(P)$	<i>/*Punktwolke <math>P</math> einlesen und darstellen*/</i>
2: $F \leftarrow \emptyset$	
3: <b>for all</b> $F_i$ mit $F_i = \text{StabileFläche}$ <b>do</b>	
4:	<i>/*Stabile Fläche nach unten drehen*/</i>
5: <b>repeat</b>	
6: $drehen(P)$	
7: <b>until</b> $F_i$ ist unten	
8: $\hat{P} \leftarrow \emptyset$	<i>/*Untere Objektpunkte bestimmen*/</i>
9: <b>for all</b> $p \in P, p = (p_x, p_y, p_z)$ <b>do</b>	
10: <b>if</b> $p_z < \text{Schwelle}$ <b>then</b>	
11: $hinzufügen(p, \hat{P})$	
12: <b>end if</b>	
13: <b>end for</b>	
14: $F_i \leftarrow \text{berechneAusgleichsebene}(\hat{P})$	<i>/*Stabile Fläche berechnen (Alg. 5.3)*/</i>
15:	<i>/*Fläche darstellen, gegebenenfalls korrigieren*/</i>
16: $anzeigen(F_i)$	
17: <b>if</b> $F_i = \text{StabileFläche}$ <b>then</b>	
18: $hinzufügen(F_i, F)$	
19: <b>else</b>	
20:     GOTO Line 5	
21: <b>end if</b>	
22: <b>end for</b>	
23: <b>return</b> $F$	

**Algorithmus 5.2:** Ablauf der interaktiven Modellierung stabiler Flächen

Zur Berechnung der stabilen Fläche werden dabei diejenigen Punkte  $\vec{p} \in P, \vec{p} = (p_x, p_y, p_z)$  der Objekt-Punkt wolke  $P$  verwendet, die sich in der graphischen Visualisierung unten am Objekt befinden und somit die kleinsten  $z$ -Werte aufweisen. Zur Auswahl dieser für die



Flächenberechnung relevanten Punkte wird zunächst eine Hilfsebene  $E_H$  eingeführt, die parallel zur  $xy$ -Ebene unterhalb der Punktwolke liegt:

$$E_H : \left( \vec{x} - \begin{pmatrix} 0 \\ 0 \\ z_H \end{pmatrix} \right) \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0 \quad (5.1)$$

und

$$p_z > z_H. \quad (5.2)$$

Für alle Punkte  $\vec{p} \in P$  wird jetzt der jeweilige Abstand  $a_p$  zu  $E_H$  bestimmt.  $\bar{a}$  ist dann der mittlere Abstand aller Punkte  $\vec{p}$  zu  $E_H$ . Zur Bestimmung der stabilen Fläche werden alle Punkte  $p_u \in P$  ausgewählt, für die gilt

$$a_{p_u} \leq f \cdot \bar{a}, f \in [0 \dots 1] \quad (5.3)$$

$f$  ist dabei ein frei wählbarer Faktor zwischen 0 und 1, der über die graphische Benutzerschnittstelle eingestellt werden kann, um mehr oder weniger Punkte zur Berechnung der Ausgleichsebene zu selektieren. Eine kleinere Anzahl von Punkten macht dabei die Berechnung genauer, erfordert aber auch eine größere Präzision beim Platzieren des Objektmodells mit dem Inertial-Joystick.

Die eigentliche Berechnung einer stabilen Fläche  $F_i$  erfolgt mittels eines Gradientenabstiegsverfahrens und wird vom Benutzer erst bei hinreichend guter Auswahl der der Berechnung zugrunde liegenden Objektpunkte angestoßen. Sie ist in Alg. 5.3 zusammenfassend dargestellt. Die solcherart ermittelte Ebene beschreibt innerhalb der Objektgrenzen die stabile Fläche und wird daher im Folgenden „stabile Ebene“ genannt.

Die Gütefunktion zur Bewertung der potentiellen stabilen Ebenen im Rahmen von Alg. 5.3 lautet dabei

$$b = \sum_{i=1}^n (k \cdot \text{dist}(E, p_i))^2 \quad (5.4)$$

wobei  $p_i \in \hat{P}$  die ausgewählten Punkte sind,  $E$  die jeweils zu bewertende Ebene und  $k \in \mathbb{R}$  ein Faktor zur Gewichtung der  $p_i$ .  $k$  erlaubt es, unterhalb der aktuellen Ebene liegende Punkte stärker zu gewichten und so zu vermeiden, dass die stabile Fläche innerhalb des Objektes zu liegen kommt. Der Faktor  $k$  kann vom Benutzer in der graphischen Oberfläche nach Bedarf angepasst werden.

Die Komponente zur interaktiven Modellierung stabiler Flächen ist eine Unterapplikation der Objektmodellierungs-GUI. In ihr werden dem Benutzer nicht nur die Möglichkeit gegeben, alle oben angesprochenen Parameter zur Ermittlung einer stabilen Fläche einzustellen, sondern es werden auch die aktuell zur Ermittlung ausgewählten unteren Punkte der Punktwolke farbig gekennzeichnet sowie bereits ermittelte stabile Flächen dargestellt. Der Benutzer

```

Eingabe:  $\hat{P}$  /*Menge der zur Berechnung ausgewählten Objektpunkte*/
Ausgabe:  $E_{opt}$  /*stabile Ebene*/
1:  $E_{opt}, E, E_{neu} \leftarrow \emptyset$ 
2:  $b, b_{neu} \leftarrow 0$ 
3:  $\vec{p}_S \leftarrow \frac{1}{n} \cdot \sum_{i=1}^n \vec{p}_i, \forall \vec{p}_i \in \hat{P}$  /*Berechne Schwerpunkt von  $\hat{P}$ */
4:  $E_{Start} : (\vec{x} - \vec{p}_S) \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = 0$  /*Wähle Ausgangsebene*/
5:  $b_{neu} \leftarrow bewerte(E_{Start})$  /*Bewertungsfunktion (vgl. Gleichung 5.4)*/
6:  $b = b_{neu} + 1, E_{neu} = E_{Start}$ 
7: /*Gradientenabstieg*/
8: while  $b_{neu} < b$  do
9:    $b = b_{neu}, E = E_{neu}$ 
10:   $\{E_{neu}^1, \dots, E_{neu}^6\} \leftarrow variiere(E)$  /*Zulässige Variationen sind:*/
      /*1), 2) Verschiebung in positiver und negativer Normalenrichtung*/
      /*3) - 6) Positive und negative Rotation um beide Achsen*/
      /*des Ebenenkoordinatensystems*/
11:  for all  $E_{neu}^k$  do
12:     $b_{neu}^k \leftarrow bewerte(E_{neu}^k)$  /*Bewerte neue Ebenen (vgl. Gleichung 5.4)*/
13:  end for
14:   $E_{neu}, b_{neu} \leftarrow min(b_{neu}^1, \dots, b_{neu}^6)$  /*Bestimme beste neue Ebene*/
15: end while
16:  $E_{opt} \leftarrow E$ 
17: return  $E_{opt}$ 

```

**Algorithmus 5.3:** Berechnung einer stabilen Fläche aus ausgewählten Objektpunkten

hat zudem jederzeit die Möglichkeit, eine bereits bestimmte stabile Fläche zu löschen oder zu korrigieren.

Abb. 5.20 stellt die graphische Oberfläche der Anwendung dar. Es sind drei orthographische Ansichten sowie eine perspektivische (rechts unten) enthalten, für die jeweils mit der Maus die Kameraposition, der Vergrößerungsfaktor und bei der perspektivischen Ansicht zusätzlich die Rotation eingestellt werden können. Eine beliebige dieser Ansichten kann auch als vergrößerte Einzelansicht ausgewählt werden. Die blau eingefärbten Punkte werden zur Ermittlung der stabilen Fläche berücksichtigt, wobei diese Auswahl jeweils sofort bei Drehung der Punktwolke mit dem Inertial-Joystick aktualisiert wird.

Abb. 5.21 zeigt eine Modellierungssequenz mit eingeblendeten stabilen Flächen: (1) Zunächst wird die erste Fläche ausgewählt, indem die gewünschte Objektseite nach unten gedreht wird. (2) Aus den blau eingefärbten Punkten wird dann eine potentielle stabile Fläche berechnet, deren Lage durch das grüne Rechteck dargestellt wird. (3) Zur Kontrolle des Ergebnisses wird in eine perspektivische Einzelansicht umgeschaltet. Auch hier kann das Objekt beliebig gedreht werden. (4) Die Fläche wurde übernommen, entsprechend ist das Rechteck jetzt weiß eingefärbt, und der Benutzer dreht bereits die nächste stabile Fläche nach unten. (5) Auch die zweite stabile Fläche wurde berechnet und kann jetzt übernommen oder falls notwendig noch korrigiert werden. (6) Erneut in der perspektivischen Einzelansicht zu sehen: Zwei stabile Flächen sind bereits bestätigt, eine dritte potentielle Fläche wurde soeben berechnet. (7) Nun sind alle Flächen berechnet, die letzte Fläche muss noch bestätigt werden. (8) Zur besseren

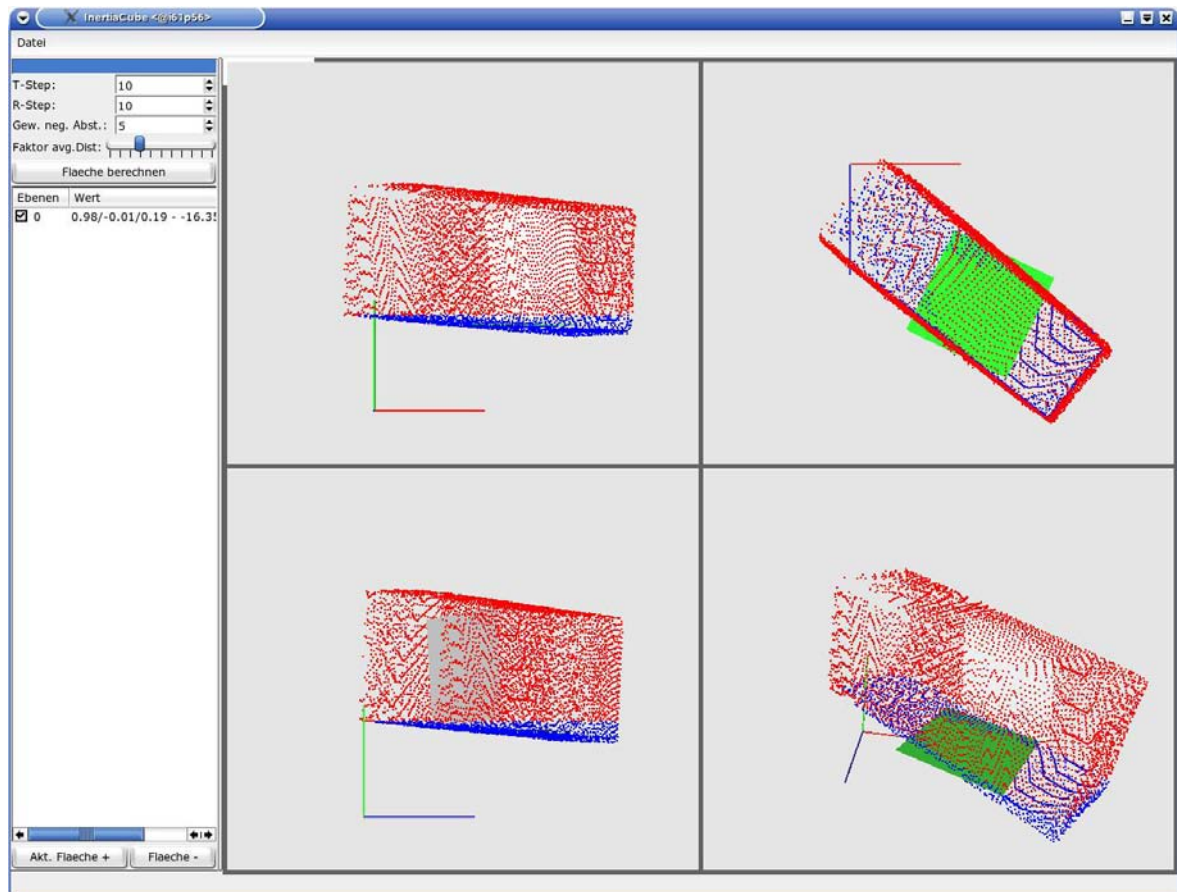


Abbildung 5.20: Unterschiedliche Objektansichten in der graphischen Oberfläche zur interaktiven Modellierung stabiler Flächen

Übersicht und Endkontrolle werden die stabilen Flächen teilweise ausgeblendet, so dass die verbleibenden Flächen besser zu sehen sind. (9) und (10) Hier ist der Einfluss des Faktors  $f$  noch einmal im Detail zu sehen. Je größer dieser Faktor ist, desto mehr Punkte werden zur Berechnung der Fläche herangezogen, und dementsprechend besser ist die Fläche zu sehen und einzustellen, allerdings leidet auch die Genauigkeit der Berechnung. In der graphischen Oberfläche ist  $f$  über einen Schieberegler einstellbar.

In Abschnitt. 7.2.2 findet sich eine Evaluation dieses Vorgehens und insbesondere die Einschätzung von Probanden bezüglich der Intuitivität und Bequemlichkeit der Modellierung.

### 5.5.5 Hauptachsen

Als *Hauptachsen* eines Objektes werden im Rahmen dieser Arbeit seine bevorzugten Koordinatenachsen bezeichnet. Es kann sich hierbei um die Hauptträgheitsachsen handeln, wobei der Ursprung des Hauptachsenkoordinatensystems dann im Schwerpunkt des Objektes liegt. Allerdings kann es für manche Anwendungen nützlich sein, auch andere bevorzugte Achsen als Hauptachsen auszuzeichnen, wobei der Ursprung des Koordinatensystems dann je nach Anwendung sinnvoll zu wählen ist. Im Rahmen der vorliegenden Arbeit werden für ein Objekt je drei aufeinander senkrecht stehende Achsen als Hauptachsen ausgewählt, wobei in

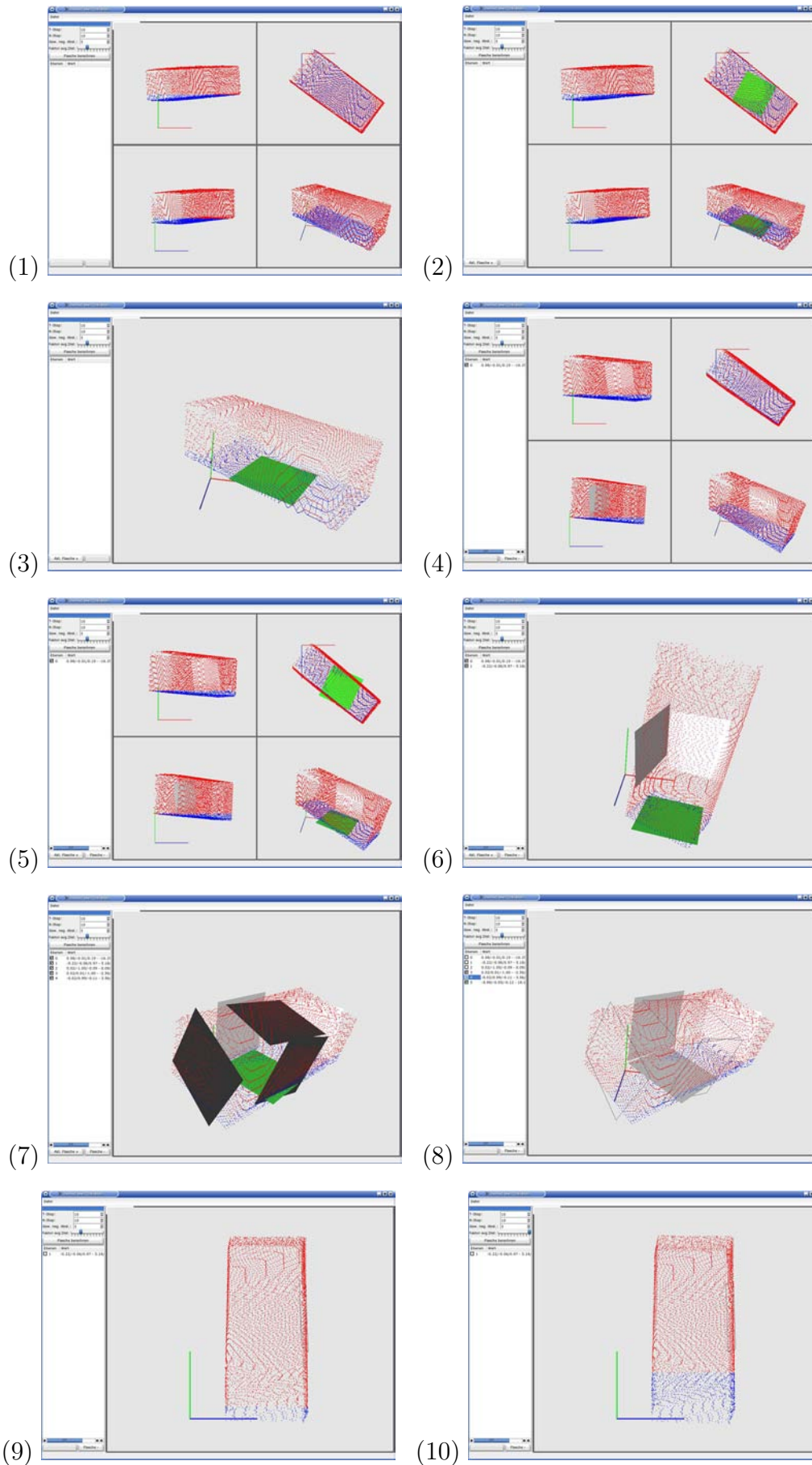


Abbildung 5.21: Sequenz zur Modellierung der stabilen Flächen einer quaderförmigen Lebensmittelverpackung

der Regel bei der Festlegung der Hauptachsen potentiell vorhandene Symmetrien berücksichtigt werden. Bei einem zylindrischen Objekt wie einem Glas verläuft dann eine Hauptachse entlang der Symmetrieachse, die beiden weiteren sind beliebig (vgl. auch das Beispiel in Abb. 5.22). Wissen über die Lage und Richtung der Hauptachsen eines Objektes kann unter anderem bei der Objektmanipulation ausgenutzt werden.

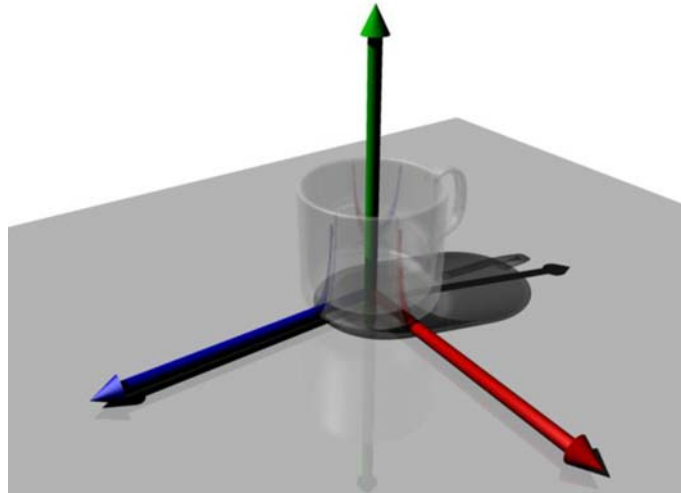


Abbildung 5.22: Hauptachsen einer Tasse, aus: Kasper (2007)

Die Bestimmung der Hauptachsen erfolgt interaktiv mit dem Benutzer. Hierzu wird die 3D-Geometrie des betreffenden Objektes dargestellt (vgl. Abb. 5.23), die vom Benutzer je nach Bedarf gedreht bzw. verschoben werden kann, um eine günstige Ansicht zu erreichen. Der Benutzer kann dann zunächst in der dreidimensionalen Darstellung durch eine translatorische Bewegung den Ursprung des Hauptachsen-Koordinatensystems in Bezug auf das Objekt festlegen. Danach wird durch rotatorische Bewegungen die Ausrichtung des vorgegebenen Hauptachsen-Koordinatensystems festgelegt. Repräsentiert werden die Hauptachsen durch den Ortsvektor des Hauptachsen-Koordinatensystems im lokalen Objektkoordinatensystem und durch die entsprechende Richtung der Hauptachsen im lokalen Koordinatensystem.

Auch diese Unterapplikation zur interaktiven Modellierung von Hauptachsen wird von der Hauptapplikation der graphischen Oberfläche aus aufgerufen und ist ähnlich wie die Oberfläche zur Ermittlung der stabilen Flächen aufgebaut. Zusätzlich zur 3D-Punktwolke werden die bereits ausgewählten Hauptachsen farbig angezeigt, und der Benutzer kann wählen zwischen translatorischen und rotatorischen Bewegungen, jeweils bezogen auf das 3D-Modell des Objektes (um eine andere Ansicht zu erhalten) oder auf die eingeblendeten Hauptachsen (um deren Position bzw. Orientierung anzupassen). Die oben bereits erwähnte vierfache Ansicht mit drei orthographischen und einer perspektivischen Ansicht erleichtern es dem Benutzer, die Lage der Achsen im dreidimensionalen Raum festzulegen.

### 5.5.6 Bewegungseinschränkungen bei der Manipulation

Bei der Handhabung von Objekten, insbesondere wenn sie bewegt werden, sind unterschiedliche Bewegungseinschränkungen denkbar, die entsprechend modelliert werden müssen. Im

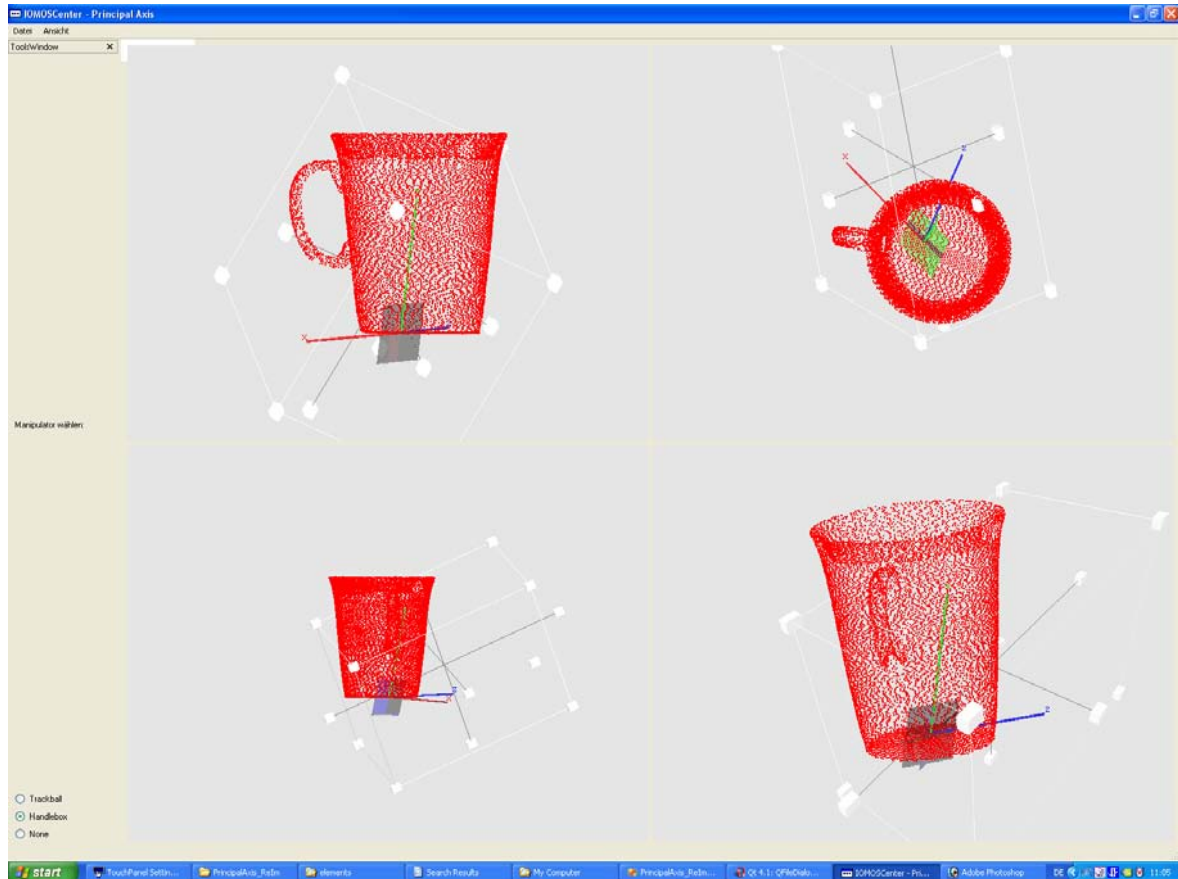


Abbildung 5.23: Graphische Oberfläche zur interaktiven Bestimmung der Hauptachsen von Objekten

Rahmen dieser Arbeit wurde beispielhaft die Modellierung von maximalen Bewegungsgeschwindigkeiten, maximalen Bewegungsbeschleunigungen und maximalen Kippwinkeln untersucht (vgl. Kasper (2007)), wie sie z. B. beim Transport von gefüllten Behältern relevant sind. Eine Erweiterung dieser Modelle im Hinblick auf die Einbindung von Bewegungskontexten („volle Tasse tragen“ vs. „leere Tasse tragen“) wird in Becher et al. (2006) vorgeschlagen. In diesem Bereich sind zukünftig weiterführende Arbeiten, auch zur Integration von Bewegungsmodellen mit Objektmodellen, denkbar und sinnvoll.

Die max. Geschwindigkeit, max. Beschleunigung und der max. Kippwinkel eines Objektes werden in einer gemeinsamen Unterapplikation der graphischen Benutzerschnittstelle unter Verwendung des magnetfeldbasierten Positionssensors (vgl. Abschnitt 5.4.5) interaktiv modelliert. Hierzu entscheidet der Benutzer zunächst anhand einer 3D-Darstellung der Objektpunktwolke mit eingeblendeten Koordinatenachsen, auf welche Koordinatenachsen des Objektes sich die zu ermittelnden Einschränkungen beziehen. Anhand des real vorliegenden Objektes kann der Benutzer dann mit einer natürlichen Bewegung die gewünschten Manipulationseinschränkungen für dieses Objekt demonstrieren. Der Ablauf dieser Modellierung ist in Alg. 5.4 dargestellt.

Zur Bestimmung der gewünschten Werte muss zunächst in den aufgezeichneten Vorführungsdaten die Initialrotation des Sensors gegenüber dem Weltkoordinatensystem berücksichtigt werden. Zudem müssen die Sensorwerte aus dem globalen Weltkoordinatensystem in das

<b>Eingabe:</b> $P$	/*3D-Punktwolke des Objektes*/
<b>Ausgabe:</b> $\vec{V}_{max} = \{v_{x,max}, v_{y,max}, v_{z,max}\}$	/*Max. Geschwindigkeiten*/
1: $\vec{A}_{max} = \{a_{x,max}, a_{y,max}, a_{z,max}\}$	/*Max. Beschleunigungen*/
2: $\vec{W}_{max} = \{\Psi_{max}, \Theta_{max}, \Phi_{max}\}$	/*Max. Kippwinkel*/
3:	/*Auswahl der gesuchten Maximalwerte*/
4: $\vec{V}'_{max} \leftarrow \text{auswählen}(\vec{V}_{max})$	
5: $\vec{A}'_{max} \leftarrow \text{auswählen}(\vec{A}_{max})$	
6: $\vec{W}'_{max} \leftarrow \text{auswählen}(\vec{W}_{max})$	
7: $\text{anzeigen}(P)$	/*Punktwolke $P$ einlesen und darstellen*/
8: $D \leftarrow \text{bewege}(\text{Objekt})$	/*Bewegungsvorführung $D(P, R, t)$ aufnehmen*/
9:	/*Bestimmung der tatsächlichen Relativrotation mit Korrektur der Initialrotation*/
10: $R_{Korrektur} = R_{Start}^{-1}$	
11: $R' = R_{Korrektur} \cdot R$	
12:	/*max. Geschwindigkeiten bestimmen (Alg. 5.5)*/
13: $\vec{V}_{max} \leftarrow \text{berechneMaxGeschwindigkeit}(\vec{V}'_{max}, R', D)$	
14:	/*max. Beschleunigungen bestimmen (Alg. 5.6)*/
15: $\vec{A}_{max} \leftarrow \text{berechneMaxBeschleunigung}(\vec{A}'_{max}, \vec{V}_{max}, D)$	
16:	/*max. Kippwinkel bestimmen*/
17: $\vec{W}_{max} \leftarrow \text{berechneMaxKippwinkel}(\vec{W}'_{max}, D)$	
18: <b>return</b> $\vec{V}_{max}, \vec{A}_{max}, \vec{W}_{max}$	

**Algorithmus 5.4:** Ablauf der interaktiven Modellierung von Manipulationseinschränkungen

<b>Eingabe:</b> $D_{Start}(P_{Start}, R_{Start}, t_{Start})$ ,	/*Sensordaten der Vorführung*/
1: $D_{Ende}(P_{Ende}, R_{Ende}, t_{Start})$	
<b>Ausgabe:</b> $\vec{V}_{max} = \{v_{x,max}, v_{y,max}, v_{z,max}\}$	/*Max. Geschwindigkeiten*/
2:	/*Translation in lokalen Koordinaten*/
3: $\vec{d} = \vec{P}_{Ende} - \vec{P}_{Start}$	
4: $\vec{d}_l = \vec{R} \cdot \vec{d}$	
5:	
6: $\vec{V}_{max} = \frac{\vec{d}_l}{\Delta_t}$	
7: <b>return</b> $\vec{V}_{max}$	

**Algorithmus 5.5:** Berechnung der gesuchten maximalen Geschwindigkeiten

<b>Eingabe:</b> $\vec{V}_{max} = \{v_{x,max}, v_{y,max}, v_{z,max}\}$	/*Max. Geschwindigkeiten*/
1: $\Delta_t$	/*Zeitdifferenz*/
<b>Ausgabe:</b> $\vec{A}_{max} = \{a_{x,max}, a_{y,max}, a_{z,max}\}$	/*Max. Beschleunigungen*/
2:	
3: $\vec{A}_{max} = \frac{\vec{V}_{max}}{\Delta_t}$	
4: <b>return</b> $\vec{A}_{max}$	

**Algorithmus 5.6:** Berechnung der gesuchten maximalen Beschleunigungen



lokale Objektkoordinatensystem umgerechnet werden. Danach werden die gesuchten maximalen Geschwindigkeiten (Alg. 5.5) und hieraus die maximalen Beschleunigungen bestimmt (Alg. 5.6) sowie die gesuchten maximalen Kippwinkel berechnet. Da der verwendete magnetfeldbasierte Rotationssensor die Ergebnisse in Eulerwinkeln nach Luftfahrtkonvention bezüglich des globalen Weltkoordinatensystems angibt, muss für die Bestimmung der maximalen Kippwinkel eine Umrechnung in das lokale Koordinatensystem erfolgen, wobei die Rotationskorrektur bezüglich der Initialrotation aus Alg. 5.4 zu berücksichtigen ist.

Nach der Luftfahrtnorm (DIN 9300) ist die Transformation vom Weltkoordinatensystem in das lokale Koordinatensystem zunächst um den Gierwinkel  $\Psi$ , dann um den Nickwinkel  $\Theta$  und schließlich um den Rollwinkel  $\Phi$  definiert. Dabei liegt in der Flugsteuerung die  $x_l$ -Achse des lokalen Koordinatensystem entlang der Längsachse des Flugzeuges, die  $y_l$ -Achse entlang der Querachse und die  $z_l$ -Achse zeigt senkrecht zu den beiden anderen Achsen nach unten. Die Drehung um den Gierwinkel  $\Psi$  entspricht dabei einer Drehung um die  $z_l$ -Achse, die zweite Drehung um den Nickwinkel  $\Theta$  einer Drehung um die neue  $y'_l$ -Achse, und die letzte Drehung um den Rollwinkel  $\Phi$  entspricht einer Drehung um die neue  $x''_l$ -Achse.

$$\begin{aligned}
 R' &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \Phi & \sin \Phi \\ 0 & -\sin \Phi & \cos \Phi \end{pmatrix} \begin{pmatrix} \cos \Theta & 0 & -\sin \Theta \\ 0 & 1 & 0 \\ \sin \Theta & 0 & \cos \Theta \end{pmatrix} \begin{pmatrix} \cos \Psi & \sin \Psi & 0 \\ -\sin \Psi & \cos \Psi & 0 \\ 0 & 0 & 1 \end{pmatrix} & (5.5) \\
 &= \begin{pmatrix} \cos \Theta \cos \Psi & \cos \Theta \sin \Psi & -\sin \Theta \\ \sin \Phi \sin \Theta \cos \Psi - \cos \Phi \sin \Psi & \sin \Phi \sin \Theta \sin \Psi + \cos \Phi \cos \Psi & \sin \Phi \cos \Theta \\ \cos \Phi \sin \Theta \cos \Psi + \sin \Phi \sin \Psi & \cos \Phi \sin \Theta \sin \Psi - \sin \Phi \cos \Psi & \cos \Phi \cos \Theta \end{pmatrix} & (5.6)
 \end{aligned}$$

Danach erfolgt wie in Alg. 5.4 angegeben die Einbeziehung der Initialrotation des Sensors:

$$R = R_{Korrektur} \cdot R'. \quad (5.7)$$

Hieraus ergeben sich die gesuchten relativen maximalen Kippwinkel  $\Theta_{max}$ ,  $\Psi_{max}$  und  $\Phi_{max}$ :

$$\Theta_{max} = \arcsin(-R_{(1,3)}) \quad (5.8)$$

$$\Psi_{max} = \arcsin\left(\frac{R_{(1,2)}}{\cos(\Theta_{max})}\right) \quad (5.9)$$

$$\Phi_{max} = \arcsin\left(\frac{R_{(2,3)}}{\cos(\Theta_{max})}\right) \quad (5.10)$$

Die Berechnung dieser Manipulationseinschränkungen wird durch den Benutzer in einer Unterapplikation der graphischen Benutzeroberfläche gestartet, die ihn als Assistent in vier Schritten durch die Datenaufnahme und Ergebnisbestimmung führt:



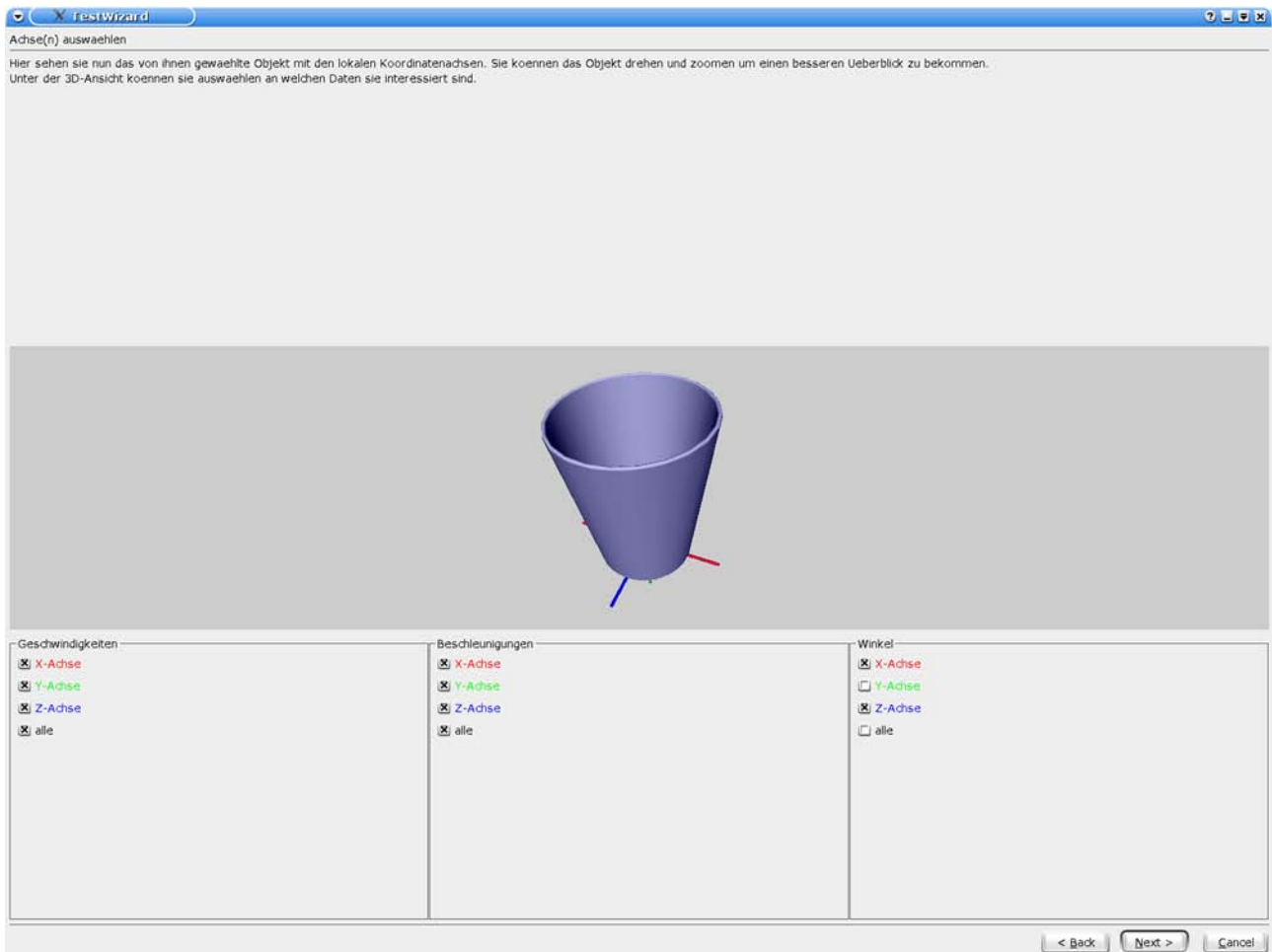


Abbildung 5.24: Assistent zur interaktiven Modellierung von Manipulationseinschränkungen, Schritt 2: Auswahl der zu bestimmenden Manipulationseinschränkungen in Bezug auf die angezeigten Objektkoordinatenachsen

1. Auswahl der dem Objekt zugehörigen Punktwolke
2. Auswahl der für dieses Objekt zu bestimmenden Manipulationseinschränkungen in Bezug auf die angezeigten Objektkoordinatenachsen durch Anklicken der jeweils gewünschten Achsen für maximale Geschwindigkeiten, maximale Beschleunigungen und maximale Kippwinkel (vgl. Abb. 5.24)
3. Vorführung der Bewegungen, wobei die jeweils ermittelten Ergebnisse textuell und für maximale Geschwindigkeiten und Beschleunigungen auch graphisch dargestellt und im Verlauf der Vorführung in Echtzeit aktualisiert werden (vgl. Abb. 5.25)
4. Anzeige und Speichern der Ergebnisse

Mit dem verwendeten magnetfeldbasierten Positionssensor wurde in dieser Anwendung eine Abtastfrequenz von ca. 4 Hz realisiert, was für die gegebenen Anforderungen ausreichend war. Eine Evaluation dieses Moduls zur interaktiven Modellierung von Manipulationseinschränkungen erfolgt in Abschnitt 7.2.3.

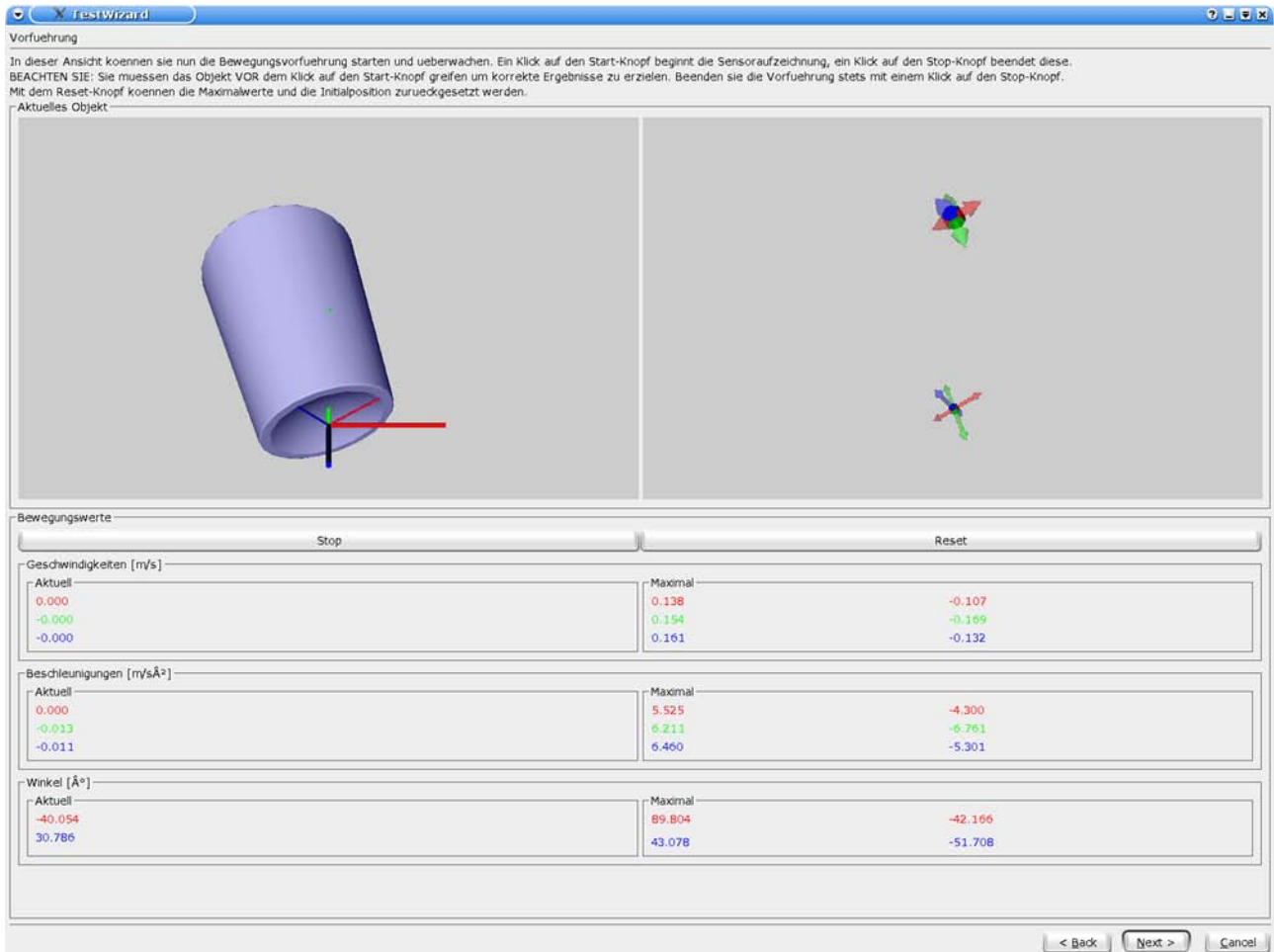


Abbildung 5.25: Assistent zur interaktiven Modellierung von Manipulationseinschränkungen, Schritt 3: Vorfuehrung der Bewegungen mit textueller und graphischer Darstellung der ermittelten Ergebnisse in Echtzeit

## 5.6 Systemintegration

Die Integration der einzelnen Modellierungsverfahren zu einem Gesamtsystem wurde bereits in den vorhergehenden Abschnitten, insbesondere in Abschnitt 5.4.1, teilweise erläutert. Hier werden daher nur noch einige kurze weiterführende Informationen gegeben. Da alle Informationen und die Interaktion mit dem Benutzer im Hauptfenster der graphischen Oberfläche zusammenlaufen, und die anderen Applikationen nur bei Bedarf aufgerufen werden, werden auch Daten zwischen den Applikationen nur bei Bedarf ausgetauscht. Hierzu werden die Daten an einer definierten Stelle gespeichert (vgl. Erläuterungen in Abschnitt 4.3.2) und können von der betreffenden Empfänger-Applikation dann abgerufen werden. Hierdurch werden unnötige Datentransfers vermieden, die bei den teilweise anfallenden Datenmengen das System unnötig verlangsamen würden.

Realisiert wurde das interaktive Modellierungssystem als verteiltes System auf verschiedenen Rechnern mit den Betriebssystemen Windows und Linux, da die notwendigen Gerätetreiber teilweise nur unter bestimmten Betriebssystemen zur Verfügung standen. Für die Daten wurde aufgrund der großen Datenmengen eine separate externe Festplatte in Betrieb genommen,

die über einen Netzwerkzugang systemweit erreichbar ist. Durch die vorhandene Netzwerkanbindung können auch andere Systemkomponenten auf die gelernten Objektmodelle zugreifen und bei Bedarf Objektinstanzen erstellen oder Informationen ergänzen und verändern.

## 5.7 Zusammenfassung

In diesem Kapitel wurden die sensorgestützte interaktive Modellierung in IOMOS und ihre Komponenten ausführlich vorgestellt und diskutiert. Hierzu wurden zunächst die wichtigsten Grundlagen der Mensch-Maschine-Interaktion aus der Literatur zusammengefasst und die Anforderungen an eine solche Interaktion dargestellt, wie sie sich aus der konkreten Aufgabenstellung der vorliegenden Arbeit ergeben. Danach wurde die im Rahmen der Arbeit entwickelte Modellierungsumgebung vorgestellt und ihre Bestandteile erläutert. Nach einer Darstellung der unterschiedlichen möglichen Interaktionsmodi wurde die Verwendung dieser Interaktionsmöglichkeiten exemplarisch an unterschiedlichen zu bestimmenden Objektattributen vorgestellt und dabei die jeweilige Vorgehensweise und das Verfahren im Detail erläutert. Das Kapitel schließt mit einigen grundlegenden Informationen zur Systemintegration.



## Kapitel 6

# Die Domäne „Küche“ als Anwendungsszenario

Als Anwendungsbeispiel und Grundlage für die Evaluation wurde im Rahmen dieser Arbeit beispielhaft eine Taxonomie von Begrifflichkeiten für die Domäne *Küche* entworfen.<sup>1</sup> Es handelt sich hierbei um eine dynamische, menschenzentrierte Alltagsumgebung, mit variierenden Anforderungen und Verwendungsmöglichkeiten für Objekte. Bei einer solch komplexen Domäne kann eine Domänenspezifikation naturgemäß nicht vollständig sein, sondern wird bestimmte Bereiche intensiver abdecken als andere. Der Schwerpunkt bei dieser vorliegenden Spezifikation lag auf der Anwendung für einen Serviceroboter, der dem Menschen im Haushalt, speziell in der Küche, zur Hand gehen soll. Daher wurden besonders Tätigkeiten aufgenommen, die derzeitige Serviceroboter realistisch ausführen können, und entsprechende Objekte berücksichtigt. Die entwickelte Taxonomie von Begrifflichkeiten ist beispielhaft auf diese Aufgabenstellung zugeschnitten, um die Nutzung der hier entwickelten Objektrepräsentation in einer realen Anwendung zu demonstrieren.

In diesem Kapitel werden zunächst einige grundlegende Begriffe und Vorgehensweisen des Ontologieentwurfs vorgestellt. Abschnitt 6.2 gibt danach einen Überblick über die Besonderheiten der Domäne *Küche* im Bezug auf die gewählte Anwendung und diskutiert einige Aspekte, die beim Entwurf der geforderten Domänenspezifikation zu berücksichtigen sind, während Abschnitt 6.3 das Ergebnis dieses Entwurfs vorstellt. Das Kapitel schließt mit einigen beispielhaften Objektmodellausschnitten. Die Grundlagen zum Ontologieentwurf werden nur kurz behandelt, soweit sie im Rahmen der gegebenen Aufgabenstellung von Interesse sind. Die Domänenspezifikation selbst dient im Rahmen dieser Arbeit zur Verdeutlichung des Modellierungsentwurfs, und um eine mögliche Anwendung zu demonstrieren. Durch das ganze Kapitel hindurch werden die Bestandteile der Objektrepräsentation notiert wie in Fußnote 2 auf Seite 48 angegeben.

---

<sup>1</sup>Die vorliegende Arbeit fand statt im Rahmen des DFG-geförderten Sonderforschungsbereichs 588 „Humanoide Roboter – Lernende und kooperierende multimodale Roboter“. Einen Überblick über die Ziele und bisher erreichten Ergebnisse dieses Projekts geben Becher et al. (2004), Steinhaus et al. (2004) und Dillmann et al. (2004). In diesem Projekt soll insbesondere ein Haushaltsroboter entwickelt werden, der dem Menschen Tätigkeiten in der Küche und im Haushalt abnimmt. Die im Rahmen der vorliegenden Arbeit entworfene und zur Darstellung und Evaluierung der Arbeiten genutzte Domänenspezifikation knüpft daher direkt an die Aufgabenstellung des Sonderforschungsbereichs an.

## 6.1 Grundlagen des Ontologieentwurfs und Entwurfsentscheidungen

Nach Mädche et al. (2001) sind Ontologien „formale Modelle einer Anwendungsdomäne, die dazu dienen den Austausch und das Teilen von Wissen zu erleichtern“ (Mädche et al. (2001), S. 1). Entscheidend ist dabei, dass zum einen eine Menge von Begriffen (ein Vokabular) existiert, zum anderen aber auch eine Menge von Regeln über die Beziehungen zwischen diesen Begriffen, also eine Beschreibung der Semantik des Anwendungsbereiches. Ontologien werden derzeit häufig verwendet in Bereichen wie dem Semantic Web, automatisierten Transaktionen mit großen Produktdatenbanken oder auch Wissensmanagement- und Wissensakquise-Systemen. In der Literatur wird dabei immer wieder betont, dass Ontologien einen bestimmten Ausschnitt aus der Welt formal beschreiben, und dass Ontologien als gemeinsame Festlegungen von Begrifflichkeiten und deren Bedeutung der Kommunikation und dem Verständnis verschiedener (menschlicher und maschineller) Interaktionspartner untereinander dienen:

„A body of formally represented knowledge is based on a *conceptualization*: the objects, concepts, and other entities that are presumed to exist in some area of interest and the relationships that hold them [...]. A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. [...] An *ontology* is an explicit specification of a conceptualization.“ (Gruber (1993), S. 1).

„Ontologien dienen der Verbesserung der Kommunikation zwischen menschlichen und maschinellen Akteuren.“ (Mädche et al. (2001), S. 1).

Eine Ontologie kann dabei niemals die komplette Wirklichkeit abdecken, sondern dient immer zur Beschreibung eines Ausschnitts der Wirklichkeit bzw. zur Formalisierung eines konkreten Problems oder des notwendigen Wissens in Bezug auf ein bestimmte Anwendung.

Es existiert eine Fülle an Literatur zu unterschiedlichsten Aspekten von Ontologien, darunter auch einige empfehlenswerte Arbeiten mit Hinweisen zum Entwurf von Ontologien, wie beispielsweise Noy und Hafner (1997) mit einer Klassifizierung unterschiedlicher bekannter Ontologien nach festgelegten Kriterien, Chandrasekaran et al. (1999) mit einer ausführlichen Bibliographie oder Noy und McGuinness (2001). Noy und McGuinness (2001) schlagen die folgenden Schritte zum Entwurf einer Ontologie vor:

1. Domäne und Umfang der Ontologie festlegen
2. Falls möglich existierende Ontologien wiederverwenden
3. Wichtige Begriffe in der Ontologie auflisten
4. Klassen und Klassenhierarchie definieren
5. Merkmale (slots) von Klassen definieren
6. Aspekte dieser Merkmale definieren
7. Instanzen kreieren

Hierbei ist allerdings zu beachten, dass Noy und McGuinness (2001) von einem sehr allgemein gehaltenen Objektmodell ausgehen. Da bei der im Folgenden zu entwerfenden Domänenspezifikation die im Rahmen dieser Arbeit entwickelte Objektrepräsentation verwendet werden sollte, müssen einige dieser Schritte leicht abgewandelt werden: außer den Objektklassen sind hier auch noch Eigenschaften, Attribute und Werttypen, Handlungen und Zustände zu definieren sowie die Beziehungen zwischen diesen. Das Wiederverwenden einer existierenden Ontologie (Schritt 2) war hier nicht möglich, da keine frei erhältliche Ontologie vorlag, die den Anforderungen entsprach und die insbesondere auch kompatibel zu dieser Objektrepräsentation war. Daher wurde in Anlehnung an Noy und McGuinness (2001) die in Abb. 6.1 dargestellte Vorgehensweise gewählt.

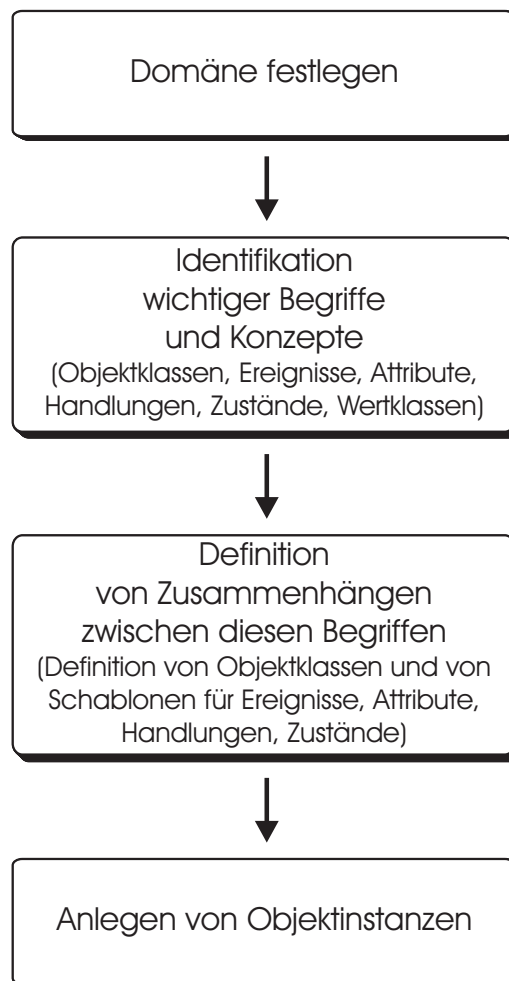


Abbildung 6.1: Entwurfsschritte zur Erstellung der Domänenspezifikation

Zur Repräsentation von Ontologien werden unterschiedliche Sprachen verwendet. Einen Überblick über die wichtigsten Repräsentations- und Anfragesprachen geben Mädche und Motik (2003). Eine Formalisierung einer Ontologie kann allerdings meist auf unterschiedliche Arten erfolgen, so dass die tatsächliche Auswahl eines Repräsentationsformalismus oft eher von praktischen Gesichtspunkten als von grundsätzlichen Entscheidungen abhängt. Eine weit verbreitete Repräsentationssprache ist die sogenannte Web Ontology Language OWL, die auf der RDF-Syntax basiert. OWL stellt drei Beschreibungsebenen unterschiedlicher Mächtigkeit zur Verfügung, OWL Lite, OWL DL und OWL Full. Während OWL DL durch Reduktion auf

Beschreibungslogik entscheidbar ist, entfällt diese Restriktion und damit die Entscheidbarkeit bei OWL Full. OWL bietet damit je nach den spezifischen Anforderungen die Möglichkeit, die in ihrer Komplexität adäquate Sprachebene zu verwenden. Die Modellierung von Ontologien erfolgt entweder mit einem Top-Down- oder mit einem Bottom-Up-Ansatz, ersteres über die Einbindung einer Top-Level-Ontologie wie der IEEE Standard Upper Ontology (Pease und Niles (2002)) oder DOLCE (Gangemi et al. (2002)), und dann immer weitergehendere Spezialisierung, letzteres indem zunächst die zu repräsentierenden Details festgelegt werden und dann hieraus immer weiter abstrahiert wird wie in van der Vet und Mars (1998). Im Rahmen dieser Arbeit wurde keine Anbindung an eine Top-Level-Ontologie vorgenommen. Der Entwurf ist aber hinreichend allgemein, so dass eine Einbindung in unterschiedliche Top-Level-Ontologien möglich wäre. Für die Zwecke der vorliegenden Arbeit erschien eine Festlegung sowohl hinsichtlich einer geeigneten Top-Level-Ontologie als auch hinsichtlich des Repräsentationsformalismus nicht notwendig, da die Entwicklung einer Ontologie hier nicht selbst Gegenstand der wissenschaftlichen Untersuchung war, sondern Grundlage für die Anwendung und Evaluation des interaktiven Objektmodellierungssystems.

Naheliegender ist die Darstellung von Ontologien in einem Markup-Language-Format wie XML oder UML, da hierdurch die Vorteile einer flexiblen Sprache mit standardisierter Syntax genutzt werden können. Beispiele hierfür finden sich unter anderem in Cranefield et al. (2001) und Chella et al. (2002). Letztere befassen sich gleichzeitig mit dem Entwurf einer Ontologie für in Gebäuden befindliche Umgebungen, die für Roboter geeignet sein soll. Allerdings ist ihr Entwurf relativ allgemein gehalten und befasst sich nicht mit den Detailproblemen einer Ontologie, wie sie für einen Serviceroboter zur Erledigung von Aufgaben im Haushalt notwendig ist. Hinzu kommt, dass Chella et al. sich nicht mit dem Problem des Lernens von Objektmodellen befassen, sondern von einer a priori-Modellierung der Ontologie und aller semantischen Inhalte ausgehen. Dies ist aber für eine reale Anwendung untauglich, bei der auch dynamische Umgebungen, unbekannte Objekte oder neue Aufgaben von einem Roboter erfasst werden müssen.

## 6.2 Besonderheiten der Domäne *Küche*

Sowohl bei der Festlegung des Modellbegriffs (vgl. Diskussion in Abschnitt 3.1) als auch im Rahmen von grundlegenden Überlegungen zu Ontologien (wie in Abschnitt 6.1) wird in der Literatur stets die Notwendigkeit betont, die Modelle bzw. Ontologien anwendungsbezogen zu erstellen, oder, wie Stachowiak (1973) es ausdrückt, nicht nur zu berücksichtigen, „*wovon* etwas Modell ist, sondern auch, *für wen*, *wann* und *wozu* [...]“ (Stachowiak (1973), S. 133). Im Folgenden sollen daher die wichtigsten Anforderungen und Besonderheiten der Domäne *Küche* kurz untersucht werden.

Da die Küche eine in hohem Grade menschenzentrierte Anwendungsumgebung darstellt und die zu erwartenden Aufgaben für einen Serviceroboter sich an menschlichen Handlungen und Vorgehensweisen orientieren, sollte eine Taxonomie von Begrifflichkeiten dieser Domäne möglichst menschliche Kategorien und Vorgehensweisen abbilden. Dies ermöglicht zum einen eine möglichst anwendungsorientierte Verwendung der Domänenspezifikation für Dinge wie Handlungsplanung etc., zum anderen aber auch eine möglichst einfache und effiziente Kommunikation mit den menschlichen Benutzern, die in dieser Domäne von großer Bedeutung



ist. Gleichzeitig ist semantisches Wissen über die verwendeten Objekte in dieser Domäne von hoher Bedeutung. Die in gängigen Objektmodellen oftmals ausschließlich repräsentierten Informationen zur Objekterkennung oder einfache geometrische Informationen genügen hier nicht. Für typische Küchenanwendungen und -aufgaben reicht es nicht aus, Objekte zu finden und zu erkennen, sondern ein maschinelles System muss sich auch über potentielle Verwendungen der Objekte im klaren sein sowie darüber, in welchem Zustand sich ein Objekt aktuell befindet und was die Bedeutung dieser Informationen ist.

Zusätzlich zu diesen grundsätzlichen Erwägungen ergeben sich in dieser Domäne auch einige praktische Anforderungen an eine Domänenspezifikation. Zum einen sind hier funktionale Objektaspekte wichtig, da eine hohe Verzahnung von Objekten und mit ihnen auszuführenden Handlungen gegeben ist. Zum zweiten sind diese Objekte teilweise stark auf bestimmte Anwendungen spezialisiert (wie ein Korkenzieher), andererseits können viele Aufgaben in dieser Domäne aber auch mit vordergründig sehr unterschiedlichen Objekten gelöst werden (wie das Gießen von Blumen mit fast jedem Behälter). Weiterhin handelt es sich bei Aufgaben in dieser Domäne oft um Routineaufgaben, die aber andererseits einen hohen Variationsgrad aufweisen können. So ist die Aufgabe, den Tisch für eine Mahlzeit zu decken, zwar mehrmals täglich aktuell, die eigentliche Ausführung variiert aber stark mit den konkreten Umständen wie Personenzahl, Anlass, Anzahl der Gänge, Art der Speisen, etc., oder das Benutzen der Mikrowelle läuft zwar im Prinzip immer gleich ab, die einzustellenden Parameter wie Dauer und Leistung können aber von vielen Umständen abhängen. Hinzu kommt schließlich, dass in dieser Domäne von manchen Objektklassen viele Instanzen existieren, die meist nicht voneinander unterscheidbar sind (beispielsweise von Geschirrtellen). Eine Unterscheidung zwischen einzelnen Instanzen kann sich dann auch erst im Lauf der Benutzung ergeben, z. B. durch Benutzungsspuren oder den aktuellen Inhalt eines Behälters.

### 6.3 Entwurf einer Spezifikation der Domäne *Küche*

Die vorliegende Arbeit gliedert sich in einen größeren Projektzusammenhang ein, innerhalb dessen ein Serviceroboter entwickelt werden soll, der dem Menschen in der Küche und anderen Haushaltsumgebungen bei alltäglichen Verrichtungen zur Hand geht. Als Grundlage für die Evaluation und Darstellung der entwickelten Objektrepräsentation und der Ergebnisse aus der interaktiven Modellierungsumgebung wurde daher eine Spezifikation der Domäne *Küche* entwickelt, die direkt als Grundlage für andere Systemkomponenten auf einem solchen Robotersystem eingesetzt werden kann. Die inhaltliche Ausrichtung dieser Taxonomie von Begrifflichkeiten trägt somit den Anforderungen der konkreten Anwendung Rechnung. Dies beschränkt allerdings nicht die allgemeinere Anwendbarkeit der hier erzielten Ergebnisse zur interaktiven Objektmodellerstellung, da diese nicht von der konkret eingesetzten Ontologie abhängen. Insbesondere sind auch Erweiterungen dieser Domänenspezifikation auf andere menschenzentrierte Einsatzbereiche (z. B. auf andere Aspekte des Einsatzbereichs „Haushalt“) und Spezialisierungen (z. B. auf den Bereich „Kochen“) denkbar. An dieser Stelle kann nur ein Überblick über einige interessante Details der Spezifikation gegeben werden. Eine im Rahmen der vorliegenden Arbeit erfolgte ausführlichere Untersuchung der zu modellierenden Attribute, ihrer Ableitung aus wichtigen Küchenhandlungen, ihrer möglichen Repräsentationen sowie der hierfür notwendigen Werttypen ist in Kasper (2007) wieder-

gegeben. Dort sind auch die darauf aufbauenden detaillierten Überlegungen zu sinnvollen Interaktionsformen, die eine interaktive Modellierung dieser Attribute erlauben, dargestellt.

### 6.3.1 Objekte und Objekthierarchie

Um sowohl die Modellierung zu vereinfachen als auch um die Domänenspezifikation für den Benutzer verständlich zu gestalten, wurden die zu modellierenden Küchenobjekte nicht nur erfasst, sondern auch eine Klassenhierarchie dieser Objekte erstellt. Dies bedeutet auch, dass Informationen wie Objekteigenschaften, Standard-Attributwerte etc. soweit als möglich zwischen Klassen vererbt werden können. Die Abbildungen 6.2 bis 6.6 zeigen einige Ausschnitte aus dieser Hierarchie. Für menschenzentrierte Domänen wie eine Küche typisch ist dabei insbesondere die Kategorie FUNKTIONALE ELEMENTE, die diejenigen Objekte bzw. Objektbestandteile in einer Küche umfasst, die überwiegend funktionale Bedeutung haben, z. B. TÜRKLINKEN, DREHKNÖPFE von Elektrogeräten, SCHALTER etc. Hier spiegelt sich in der gewählten Objekthierarchie bereits die Konzentration auf funktionale und semantische Objektbestandteile wider. Je nach konkreter Anwendung, für die die Hierarchie verwendet werden soll, können dabei natürlich andere Kategorien gewählt bzw. andere Objekte in die Hierarchie integriert werden.

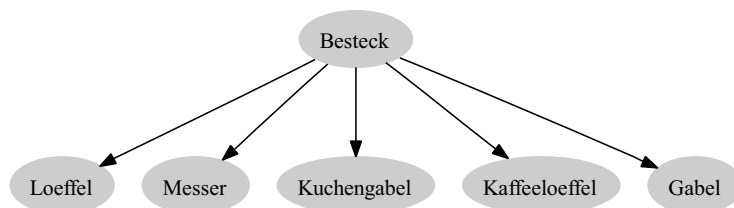


Abbildung 6.2: Ausschnitt aus der Hierarchie gebräuchlicher Küchenobjekte (1): Besteck

### 6.3.2 Relationen zwischen Objekten

Bei Relationen zwischen Objekten ist zunächst an räumliche Relationen zu denken wie *in* oder *auf*, aber auch an semantische Beziehungen wie die zwischen TOPF und DECKEL oder zwischen TASSE und UNTERTASSE, die häufig zusammen verwendet werden. Auch *ist-Bestandteil-von* (wie im Fall von SCHRANKTÜR und SCHRANK) und spezielle räumliche Relationen wie bewegliche Verbindungen (wie die Verbindung zwischen SCHRANKTÜR und SCHRANKKORPUS) treten in typischen Haushaltsumgebungen, und im speziellen in Küchen, an unterschiedlichen Stellen auf. Als grundlegende Relationen zwischen Objekten für den Einsatz eines Serviceroboters in einer Küche wurden deshalb die in Tab. 6.1 dargestellten ermittelt. Diese Relationen können im Rahmen einer Szenenmodellierung als grundlegende Beziehungen zwischen Objekten einer Szene auftreten (vgl. Abschnitt 4.2 zur Verwendung von Objektinstanzen zur Szenenmodellierung).

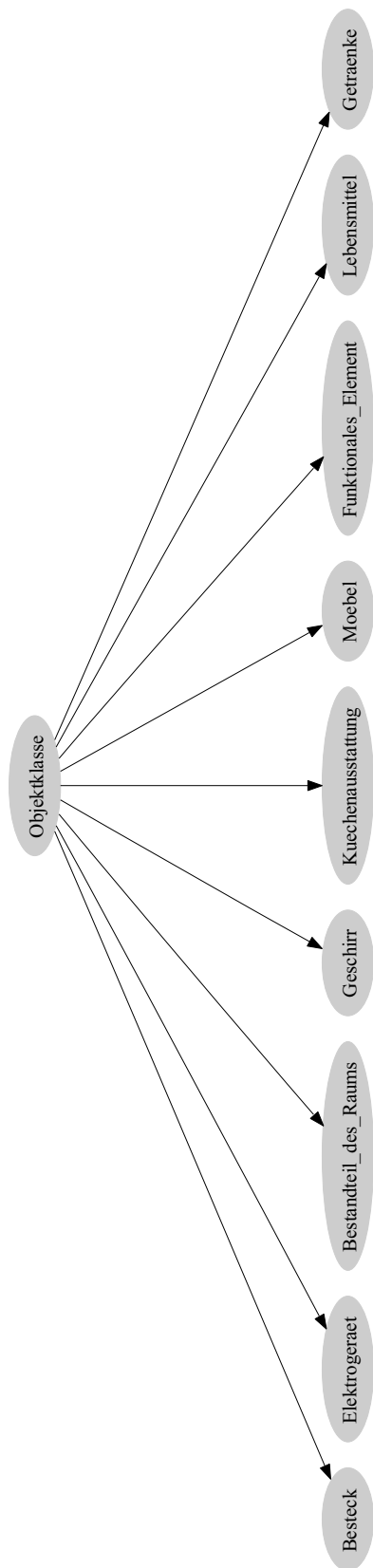


Abbildung 6.3: Ausschnitt aus der Hierarchie gebräuchlicher Küchenobjekte (2): oberste Ebene der Hierarchie

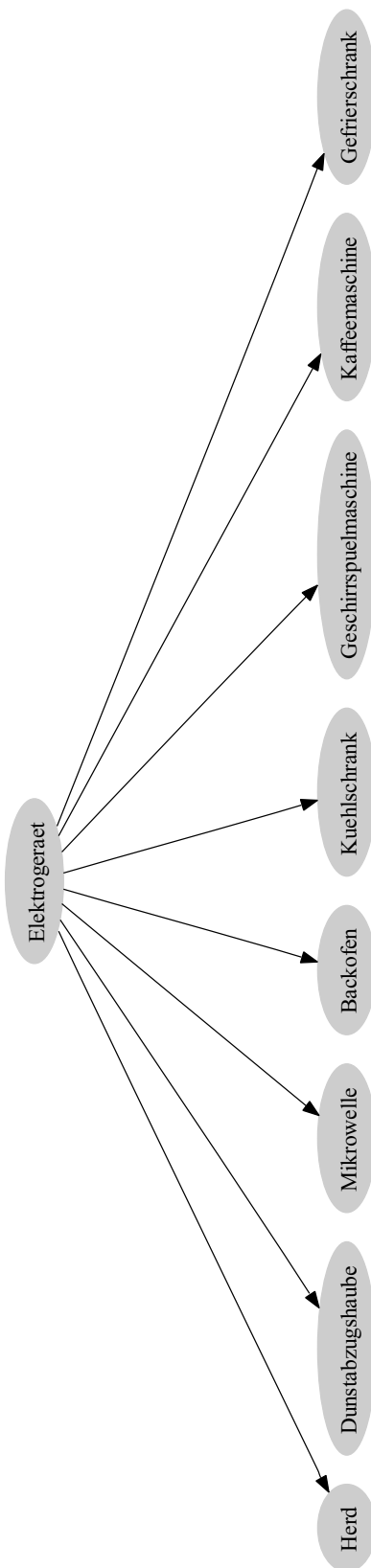


Abbildung 6.4: Ausschnitte aus der Hierarchie gebräuchlicher Küchenobjekte (3): Elektrogeräte

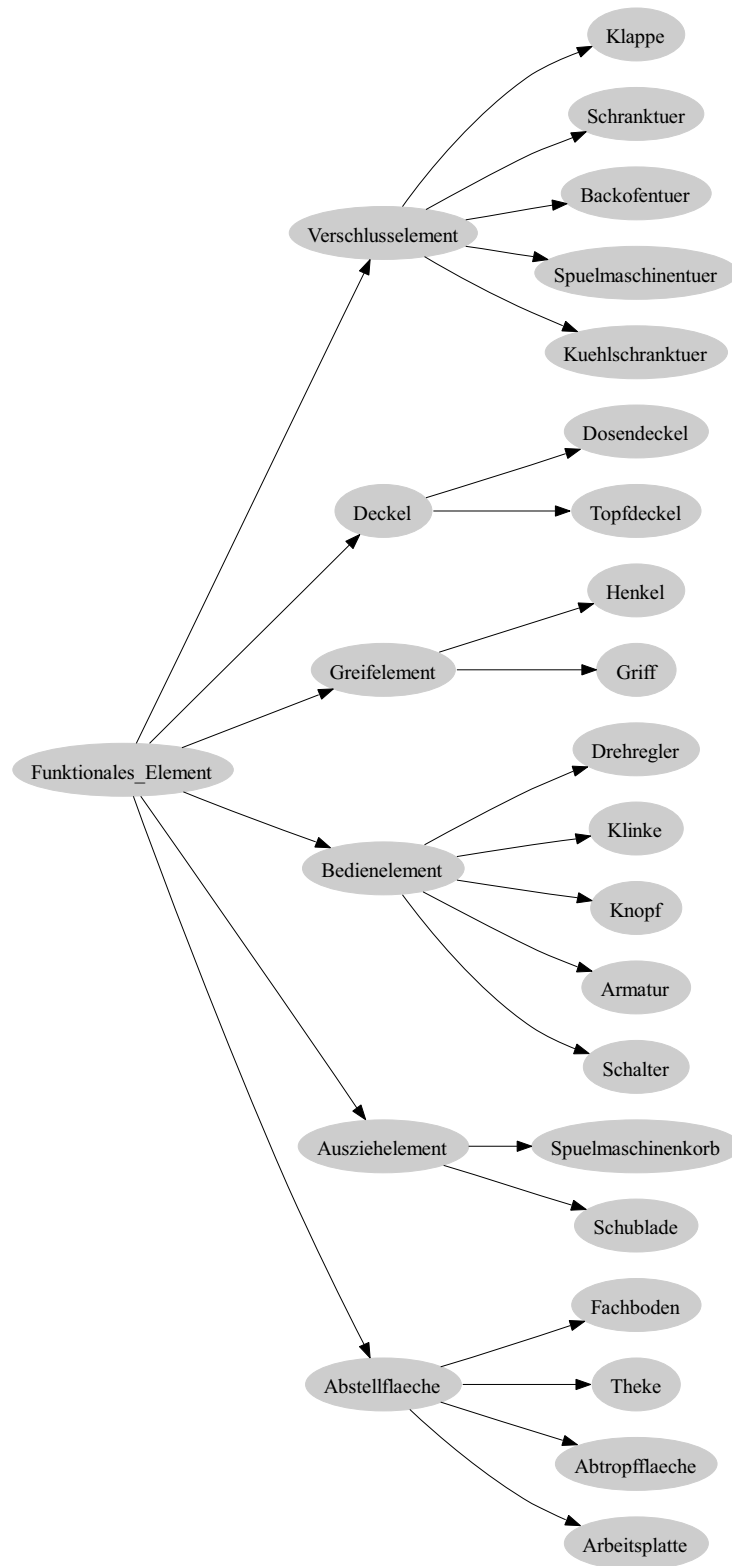


Abbildung 6.5: Ausschnitt aus der Hierarchie gebräuchlicher Küchenobjekte (4): Funktionale Elemente

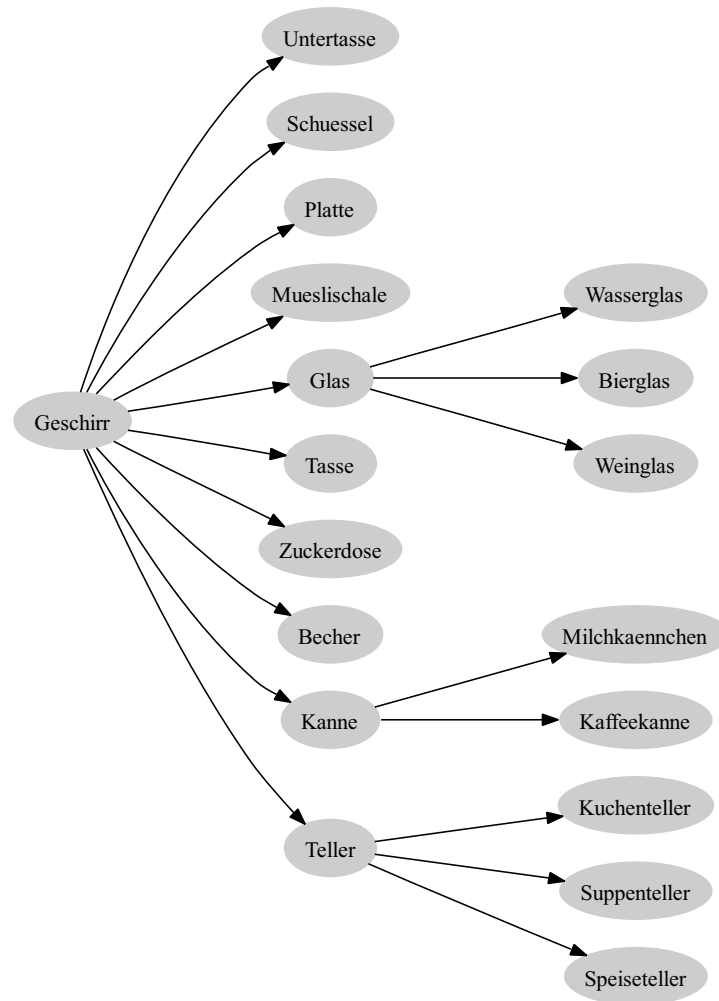


Abbildung 6.6: Ausschnitt aus der Hierarchie gebräuchlicher Küchenobjekte (5): Geschirr

Name der Objektrelation	Art der Relation
in	räumliche Relation
auf	räumliche Relation
unter	räumliche Relation
neben	räumliche Relation
berührt	räumliche Relation
wird-verwendet-mit	semantische Relation
ist-Bestandteil-von	semantische Relation
ist-verbunden-mit	räumliche Relation
ist-beweglich-verbunden-mit	räumliche Relation

Tabelle 6.1: Wichtige Relationen zwischen Objekten in der Domäne *Küche*

### 6.3.3 Eigenschaften

Als das semantische Gerüst der Objektrepräsentation bilden Eigenschaften die Verbindung zwischen Objektklassen bzw. -instanzen und ihren Attributen sowie den mit ihnen verknüpften Handlungen und Zuständen. Tab. 6.2 und 6.3 zeigen daher wichtige Objekteigenschaften der Domäne *Küche* zusammen mit Beispielen von mit ihnen verbundenen Objekten, Attributen, Handlungen und Zuständen. Die ausgewählten Eigenschaften zeigen dabei auch deutlich die funktionalen Anteile der zu modellierenden Objekte und erlauben gleichzeitig die Anwendung dieser Objekte unter unterschiedlichen Anforderungen. Die Verbindung zwischen Objekten und ihren Eigenschaften wird auch deutlich an den Beispielen in Abschnitt 6.4.

### 6.3.4 Attribute

Attribute und ihre Werte sind diejenigen Bausteine eines Objektmodells, die seine Anwendung erst ermöglichen. Sie bilden den Bezug zu den eingesetzten Sensor- und Aktorsystemen, indem die dort akquirierten bzw. verwendeten Daten direkt in Attributwerte umgerechnet werden können, wie bei Kameraansichten. In vielen gängigen Objektmodellen sind daher auch ausschließlich oder fast ausschließlich Informationen auf Attributebene enthalten. Da je nach den geplanten Anwendungen viele und unterschiedliche Attribute relevant sind, ist eine vollständige Auflistung hier weder möglich noch sinnvoll. Einige Beispiele sind in den Tabellen 6.2 und 6.3 enthalten, und in Kap. 5 ist ausführlich die interaktive Bestimmung der Werte einiger wichtiger Attribute vorgestellt. Im Rahmen der vorliegenden Arbeit erfolgte eine ausführliche Untersuchung der notwendigen Attribute für einen Serviceroboter in der Küche, ihre Herleitung aus den durchzuführenden Aufgaben sowie Überlegungen zu ihrer Repräsentation, Darstellung und Modellierung (eine ausführliche Darstellung findet sich in Kasper (2007)). Einen Überblick über die aus wichtigen Handlungen in Küchenumgebungen abgeleiteten Objektattribute ist in Abb. 6.7 dargestellt. Im Rahmen dieser Untersuchung konnte auch abgeleitet werden, dass die zu erfassenden Attributwerte sich auf wenige grundsätzliche Werttypen zurückführen lassen, die im nächsten Abschnitt dargestellt sind.

### 6.3.5 Werttypen

Als grundlegende Werttypen in diesem Kontext wurden die in Tab. 6.4 dargestellten Repräsentationsformen identifiziert (vgl. hierzu auch Kasper (2007)). Für alle Werttypen sind dabei jeweils passende Operatoren für ihre Darstellung (z. B. eine drehbare dreidimensionale Darstellung für die Punktwolken), für das Abspeichern im festgelegten XML-Format (vgl. Abschnitt 4.3.2) und für Vergleiche (z. B. für Längenvergleiche zwischen Trajektorien) festzulegen und zu implementieren.

### 6.3.6 Handlungen

Den Eigenschaften zugeordnete Handlungen sind ein weiterer grundlegender Bestandteil der vorgeschlagenen Objektrepräsentation, da sie eine direkte Verknüpfung zwischen Objekt-

Eigenschaft	Beispiele für Objektklassen	Beispiele für Attribute	Beispiele für Handlungen	Beispiele für Zustände
<b>ist_transportierbar</b>	TASSE, MESSER	<i>Manipulationseinschränkungen, stabile Flächen</i>	<i>tragen, abstellen</i>	wird_transportiert
<b>ist_greifbar</b>	TASSE, DOSE	<i>Grifftyp, Greifansatzpunkte, max. Greifkraft</i>	<i>greifen, loslassen</i>	gegriffen
<b>ist_befüllbar</b>	TASSE, KAFFEEKANNE	<i>aktueller Füllstand, Manipulationseinschränkungen, max. Füllhöhe, aktueller Inhalt</i>	<i>füllen, entleeren</i>	voll, halbvoll, leer
<b>ist_beladbar</b>	SCHRANK, REGAL	<i>Anzahl Fachböden, Ver-schließbarkeit</i>	<i>hineinstellen, heraus-nehmen</i>	leer, teilweise gefüllt, voll
<b>ist_erhitzbar</b>	MILCH, WASSER	<i>aktuelle Temperatur, max. Temperatur, bevorzugte Er-wärmungsmethode</i>	<i>erhitzen</i>	kalt, lauwarm, heiß
<b>ist_flüssig</b>	MILCH, KAFFEE	<i>Zähflüssigkeit, Menge</i>	<i>ausschütten</i>	verschüttet, in_Behälter
<b>ist_Festkörper</b>	TASSE, SCHRANK	<i>Bounding Box, Hauptach-sen, 3D-Punktwolke, An-sichten</i>	<i>anstoßen</i>	starr, elastisch
<b>ist_pulvrig</b>	SALZ, MEHL	<i>Grobkörnigkeit</i>	<i>schütten</i>	verschüttet, in_Behälter
<b>ist_deformierbar</b>	SCHWAMM, LAPPEN	<i>aktuelle Bounding Box, Ori-ginalkörper in Ruhezustand</i>	<i>zerknautschen</i>	verformt, nicht_verformt
<b>ist_essbar</b>	APFEL	<i>bevorzugte Serviermethode, Verzehrttemperatur</i>	<i>anrichten, zubereiten</i>	zubereitet

Tabelle 6.2: Typische Objekteigenschaften in der Domäne Küche (1)

Eigenschaft	Beispiele für Objektklassen	Beispiele für Attribute	Beispiele für Handlungen	Beispiele für Zustände
ist_trinkbar	KAFFEE, COLA	bevorzugtes Serviergefäß, Verzehrtemperatur	servieren, eingießen	eingeschenkt
ist_drückbar	BEDIENKNOPF, LICHTSCHALTER	Schalterstand, min. Druckkraft, max. Druckkraft	drücken, loslassen	gedrückt, nicht gedrückt
ist_drehbar	DREHREGLER	Reglerstand, mögliche Drehrichtung	drehen	in Nullstellung
ist_ausziehbar	SCHUBLADE, APO-THEKERSCHRANK	aktueller Öffnungsgrad, min. Öffnungskraft	ausziehen, schließen	ausgezogen, teilweise ausgezogen, nicht ausgezogen
ist_Griff	SCHUBLADENGRIFF	Grifftyp, max. Greifkraft	greifen, loslassen	gegriffen
ist_Deckel	TOPFDECKEL	Öffnungsmethode, Dichtigkeit	öffnen, verschließen	offen, verschlossen
ist_Verschluss	SCHRANKTÜR	Öffnungsgrad	öffnen, schließen	teilweise offen, geschlossen
ist_Abstellfläche	TISCH, ARBEITSPLATTE	freie Fläche	abstellen	frei, teilweise frei, voll
ist_durchsichtig	GLAS	Erkennungsmethode, Erkennparameter	hindurchsehen	klar, matt
ist_zerbrechlich	GLAS	max. Druckkraft	zerbrechen	unbeschädigt, beschädigt, zerbrochen
ist_stapelbar	TELLER	max. Stapelhöhe	stapeln	oberstes Objekt in Stapel, gestapelt, nicht gestapelt

Tabelle 6.3: Typische Objekteigenschaften in der Domäne Küche (2)



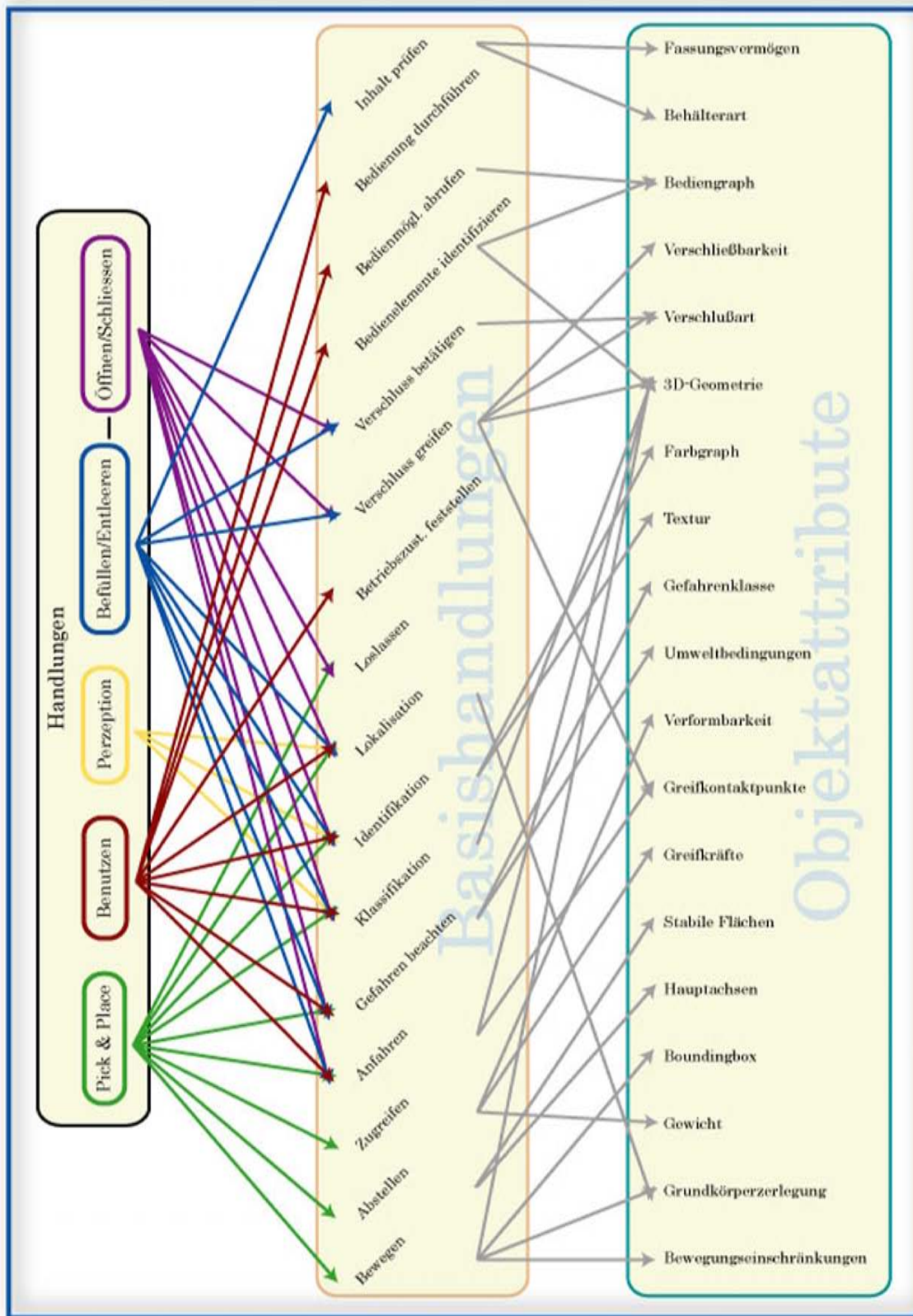


Abbildung 6.7: Ableitung von wichtigen Objektattributen aus gebräuchlichen Handlungen in der Küche, (aus: Kasper (2007), S. 25)

Werttyp	Verwendungsbeispiele
übliche Zahlentypen, Zeichen, Mengen, Listen, Zeichenketten	grundlegende Repräsentationsformen wie in gängigen Programmiersprachen
Graphen	Farbgraphen, Bediengraphen für Elektrogeräte
Vektoren	Hauptachsen, maximale Kippwinkel, Geschwindigkeiten und Beschleunigungen
Matrizen	Bilder, affine Transformationen
Text	Beschreibungen
Strukturen	stabile Flächen
Auswahlliste	Verformbarkeit, Gefahrenklasse
Punktwolken	3D-Geometrie, Bounding Box
Trajektorien	An- und Abrückbewegungen
Polygone	Dreiecke für Dreiecksnetze

Tabelle 6.4: Wichtige Werttypen in der Domäne *Küche*

und Handlungswissen gestatten und so beispielsweise eine schnelle und effiziente Handlungsplanung ermöglichen. Auf die eigentliche Repräsentation von Handlungen und das Lernen von Handlungswissen wird im Rahmen dieser Arbeit nicht eingegangen. Weiterführende Informationen zu diesem Thema finden sich in Rogalla (2003) oder Zöllner (2005). Auch eine Unterscheidung zwischen verschiedenen Abstraktionsstufen des Handlungswissens erfolgt im Rahmen dieser Arbeit nicht. Die Objektmodelle sind aber überwiegend mit einer mittleren Abstraktionsstufe von Handlungen verknüpft, also weder mit fein unterteilten, aktorabhängigen Elementaroperatoren (vgl. die Definition bei Zöllner (2005), S. 48) noch mit komplexen Handlungsfolgen wie „Tisch decken“. Letztere können bei Bedarf aus den hier gegebenen Basishandlungen zusammengesetzt werden. Eine Untersuchung unterschiedlicher Abstraktionsstufen von Alltagshandlungen sowie ein Entwurf einer semantischen Handlungsrepräsentation findet sich bei Henrich (2005).

In Abb. 6.8 sind Beispiele für mit Küchenobjekten durchführbare Handlungen dargestellt, jeweils mit ausgewählten Attributwerten für Vor- und Nachbedingungen. Weitere Handlungen zu den aufgeführten Eigenschaften sind beispielhaft in den Tabellen 6.2 und 6.3 angegeben.

### 6.3.7 Zustände

Wie in Abschnitt 3.3.3 bereits beschrieben, sind Zustände eine Bezeichnung für bestimmte Attributwertbereiche. Sie erlauben es unter anderem, eine effizientere und mehr an menschlichen Kommunikationsstrategien ausgerichtete Interaktion zu führen, indem beispielsweise für

<i>füllen</i>	[ist_befüllbar]
Vorbed.: <i>aktueller Füllstand</i> $\leq 50\%$	
Nachbed.: <i>aktueller Füllstand</i> $\geq 90\%$ <i>aktueller Füllstand</i> $\leq \text{max. Füllstand}$	

<i>abstellen</i>	[ist_transportierbar]
Vorbed.: <i>stabile Fläche</i> ist unten	
Nachbed.: <i>stabile Fläche</i> auf Abstellfläche	

<i>stapeln</i>	[ist_stapelbar]
Vorbed.: <i>akt. Stapelhöhe</i> $< \text{max. Stapelhöhe}$	
Nachbed.: <i>akt. Stapelhöhe</i> $> \text{vorh. Stapelhöhe}$	

Abbildung 6.8: Beispiele für Handlungen in der Domäne Küche

voll	[ist_befüllbar]
VOP: <i>aktueller Füllstand</i> $\geq 90\%$	

lauwarm	[ist_erhitzbar]
VOP: <i>akt. Temperatur</i> $\geq 25^\circ\text{C} \wedge \text{akt. Temperatur} \leq 35^\circ\text{C}$	

Abbildung 6.9: Beispiele für Zustände in der Domäne Küche

die Domäne *Küche* Zustände wie **voll**, **heiß** oder **offen** in Abhängigkeit von bestimmten Attributwerten definiert werden. Somit ist sowohl die Wahl der zu modellierenden Zustände als auch deren genauer Definitionsbereich stark anwendungsspezifisch. Die Tabellen 6.2 und 6.3 stellen beispielhaft auch einige mögliche Zustände in dieser Domäne und ihre Zugehörigkeit zu bestimmten Eigenschaften dar. Abb. 6.9 zeigt exemplarisch ausgearbeitete Zustände für Anwendungen in der Küche mitsamt möglicher zugehöriger Wertebereiche sowie ihre Zuordnung zu Eigenschaften.

## 6.4 Exemplarische Darstellung einer Objektklasse

Im Folgenden werden beispielhaft für die entwickelten Objektmodelle einige Ausschnitte aus einem Modell der Objektklasse TASSE dargestellt. Die interaktive Erstellung eines solchen Modells wird dann in Abschnitt 7.4 diskutiert. Hierzu werden zunächst die Eigenschaften des Objektes festgelegt, wie in Abb. 6.10 dargestellt.



Abbildung 6.10: Eigenschaften der Objektklasse TASSE

Diesen Eigenschaften werden dann zugehörige Attribute, Zustände und Handlungen zugewiesen. In der praktischen Umsetzung in IOMOS kann dieser Schritt auch vor der eigentlichen Modellierung der Objektklasse erfolgen, indem Schablonen für Eigenschaften und ihre Bestandteile angelegt werden.

Exemplarisch für die Ausarbeitung der Eigenschaften der Objektklasse TASSE ist in Abb. 6.11 ein Ausschnitt aus der Eigenschaft **ist\_befüllbar** dieser Objektklasse dargestellt. Angegeben ist eine kleine Auswahl der zugehörigen Attribute, Handlungen und Zustände sowie für die Attribute die jeweiligen Standardattributwerte. Die Eigenschaft **ist\_befüllbar** ist dabei in diesem Fall spezifisch für diese spezielle Objektklasse (angedeutet durch das „[TASSE]“ rechts oben im Modell) und differiert möglicherweise von der Eigenschaftsschablone **ist\_befüllbar**, indem einzelne Attributwerte etc. an die Erfordernisse von Objekten mit dieser speziellen Eigenschaft angepasst sind. Weitere Spezifizierungen von Attributwerten können dann bei der Ableitung von Objektinstanzen aus dem Objektklassenmodell erfolgen oder während der Anwendung dieser Objektinstanzen, beispielsweise wenn sich der aktuelle Füllstand einer Tasseninstanz ändert.

Abb. 6.12 zeigt die Handlungsbeispiele aus Abb. 6.8, die nun in die Objektklasse TASSE eingefügt wurden. Während des Instanzierens der Handlungen beim Einsetzen in das Objektklassenmodell können die Vor- und Nachbedingungen bei Bedarf an die neuen, spezielleren Erfordernisse angepasst werden. Im Beispiel wurde als eine Nachbedingung zu *füllen* der neue Mindestfüllstand hier etwas verringert, da zu heißen Getränken oft noch Milch hinzugefügt wird und daher teilweise ein etwas niedrigerer Füllstand zum Abschluss einer erfolgreichen Füllhandlung ausreicht. Außerdem zu beachten ist die Tatsache, dass in der Nachbedingung zu *füllen* auch hier noch der Verweis auf den Attributwert des Attributes

<b>ist_befüllbar</b>	[TASSE]
<i>aktueller Füllstand</i> = undefiniert <i>maximale Füllhöhe</i> = 95% <i>maximaler Kippwinkel</i> = 15° ...	Attribute
<i>füllen</i> <i>entleeren</i> ...	Handlungen
voll halbvoll leer ...	Zustände

Abbildung 6.11: Eigenschaft **ist\_befüllbar** der Objektklasse TASSE

*maximaler Füllstand* zu sehen ist. Dieser Attributwert wurde auch im Objektmodell nicht als feste Zahl in die Handlung *füllen* eingefügt, weil er über die Zeit veränderlich ist, und hier jeweils der aktuelle Wert dieses Attributes als Maßstab anzuwenden ist.

Einige der Objektklasse TASSE zugehörige Zustände sind in Abb. 6.13 dargestellt. Auch hier können bei Bedarf die zur Zustandsbestimmung angewandten Vergleichsoperatoren oder deren Verknüpfungen gegenüber den Zustandsschablonen angepasst werden, wenn ein Zustand einer Eigenschaft zugeordnet wird oder auch wenn im nächsten Schritt über das Einfügen einer Eigenschaft in ein Modell eines Objektes ein Zustand einem Objekt zugeordnet wird.

## 6.5 Zusammenfassung

In diesem Kapitel wurde die im Rahmen der vorliegenden Arbeit entwickelte Spezifikation der Domäne *Küche* vorgestellt, die auch zur praktischen Anwendung und Evaluation der Arbeit herangezogen wird. Hierzu wurden zunächst einige grundlegende Begriffe des Ontologieentwurfs geklärt und Entwurfsentscheidungen erläutert. Nach einigen grundsätzlichen Überlegungen zu den Besonderheiten der Domäne wurden beispielhaft einige interessante Ausschnitte aus der eigentlichen Taxonomie von Begrifflichkeiten vorgestellt, gegliedert nach den Bestandteilen der oben vorgestellten semantischen Objektrepräsentation. Das Kapitel

<i>füllen</i>	[ist_befüllbar, TASSE]
Vorbed.: <i>aktueller Füllstand</i> $\leq$ 50%	
Nachbed.: <i>aktueller Füllstand</i> $\geq$ 80%	
	<i>aktueller Füllstand</i> $\leq$ <i>max. Füllstand</i>

<i>abstellen</i>	[ist_transportierbar, TASSE]
Vorbed.: <i>stabile Fläche</i> ist unten	
Nachbed.: <i>stabile Fläche</i> auf Abstellfläche	

<i>stapeln</i>	[ist_stapelbar, TASSE]
Vorbed.: <i>akt. Stapelhöhe</i> $<$ <i>max. Stapelhöhe</i>	
Nachbed.: <i>akt. Stapelhöhe</i> $>$ <i>vorh. Stapelhöhe</i>	

Abbildung 6.12: *füllen*, *abstellen* und *stapeln* als Beispiele für Handlungen der Objektklasse TASSE

voll	[ist_befüllbar, TASSE]
VOP: <i>aktueller Füllstand</i> $\geq$ 90%	

halbvoll	[ist_befüllbar, TASSE]
VOP: <i>akt. Füllstand</i> $\geq$ 40% $\wedge$ <i>akt. Füllstand</i> $\leq$ 60%	

leer	[ist_befüllbar, TASSE]
VOP: <i>aktueller Füllstand</i> = 0%	

Abbildung 6.13: *voll*, *halbvoll* und *leer* als Beispiele für Zustände der Objektklasse TASSE

schließt mit einer beispielhaften Darstellung von Ausschnitten aus dem Objektmodell der Objektklasse TASSE.

# Kapitel 7

## Evaluation und Bewertung

Eine Evaluation und Bewertung der vorgestellten Arbeiten erfolgt aufgrund des Systemscharakters der Arbeit sowie der nicht durch Messungen vergleichbaren Ergebnisse überwiegend qualitativ bzw. durch eine Auswertung von Benutzereinschätzungen. Zur Bewertung heranzuziehende Kriterien sind somit zum einen die Qualität der Objektrepräsentation bezüglich Flexibilität, Praxistauglichkeit, Anwendbarkeit auf andere Kontexte etc. (vgl. Diskussion der Anforderungen an die Objektrepräsentation in Abschnitt 3.2), zum anderen die Qualität der interaktiven Modellierung und dabei vor allem deren Benutzerfreundlichkeit, Intuitivität, die Qualität der erstellten Modelle und die Kombination aus autonomen und interaktiven Modellierungsschritten (vgl. hierzu auch die Ausführungen in den Abschnitten 4.1 und 5.2). Bei der Evaluation und Bewertung der Ergebnisse der vorliegenden Arbeit ist insbesondere zu berücksichtigen, ob die Stichhaltigkeit der in Abschnitt 1.3 vorgestellten Thesen in dieser Arbeit überzeugend belegt werden konnte.

### 7.1 Evaluation der semantischen Objektrepräsentation

Eine Untersuchung und Bewertung der semantischen Objektrepräsentation kann nicht quantitativ durchgeführt werden, sondern ist nur qualitativ anhand potentieller Einsatzgebiete möglich. Eine Betrachtung der in Abschnitt 3.2 herausgearbeiteten Anforderungen an die semantische Objektrepräsentation ergibt folgende Ergebnisse:

#### **Verständlichkeit für den Menschen**

Dieser Punkt hängt auch von den gewählten Interaktionsmodi ab (vgl. Diskussion unten), da diese die Objektrepräsentation und somit die grundlegende Struktur von Objektmodellen vermitteln. Es lässt sich allerdings beobachten, dass selbst nicht fachlich vorgebildete Benutzer nach einer kurzen Erläuterung das Konzept der semantischen Strukturierung von Objekten durch deren Eigenschaften schnell erfassen und eigenständig auf andere Objekte und Eigenschaften übertragen können. Für das menschliche Verständnis zu erstellender Objektmodelle ist es daher vorteilhaft, dass die Struktur gegenüber der weit verbreiteten Attribut-Wert-Paar-Struktur vordergründig komplexer gestaltet wurde, um somit auch semantische Informationen und funktionale Verwendungsmöglichkeiten der Objekte einfach und übersichtlich zu erfassen.

### **Abbildung funktionaler Objekteigenschaften**

Die gewünschte Abbildung funktionaler Objektinformationen in der Objektrepräsentation ergibt sich direkt aus den gewählten Objekteigenschaften mit ihren funktionalen Bestandteilen sowie aus der Verbindung zur Handlungsmodellierung über die Integration von Wissen über ausführbare Handlungen.

### **Erweiterbarkeit und Flexibilität**

Die Erweiterbarkeit und Flexibilität der resultierenden Objektmodelle wurde im Verlauf der interaktiven Modellierung hinreichend bestätigt, indem alle Objektinformationen, die für einen Serviceroboter im Haushalt wichtig waren, im Rahmen verschiedener Projekte mit der hier vorgestellten Objektrepräsentation modelliert, nach Bedarf erweitert und abgerufen werden konnten.

### **Formalisierbarkeit**

Eine abstrakte Formalisierung der semantischen Objektrepräsentation wurde bereits in Abschnitt 3.3 dargestellt. Die Wahl des Schlussfolgerungssystems ist dabei relativ frei, möglich wäre z. B. die Verwendung von Beschreibungslogiken (engl.: description logics, DL) bei entsprechender Formulierung der Inhalte.

### **Abbildung von Objektmodellen auf Sensordaten und umgekehrt**

Die gestellte Anforderung, dass sowohl Ergebnisse der sensorischen Verarbeitung einfach in die Objektmodelle integriert werden können, als auch die Objektmodelle zur Unterstützung der sensorischen Verarbeitung verwendet werden können, wurde im Verlauf der interaktiven Ermittlung von Objektattributen in Kap. 5 belegt, wo einerseits unterschiedlichste Formen von Sensorinformationen in die Objektmodelle aufgenommen werden, andererseits aber auch Objektinformationen wie Wissen über die Eigenschaften eines Objekts zur Bestimmung der adäquaten sensorischen Verarbeitungsschritte herangezogen wird. Die Einbindung der auf der Grundlage dieser Objektrepräsentation erstellten Objektmodelle in eine Roboterarchitektur und damit auch die Verbindung von sensorischer Verarbeitung und Objektmodellen wird ausführlich auch in Abschnitt 7.3.2 diskutiert.

### **Übertragbarkeit auf andere Kontexte**

Die Übertragbarkeit der semantischen Objektrepräsentation auf andere Kontexte ist aufgrund der Flexibilität der Objektrepräsentation und ihrer Anpassungsfähigkeit (indem beispielsweise kontextadäquate Werttypen und eine passende Ontologie ausgewählt werden) möglich. Durch die menschenverständliche Repräsentation und die Erfassung von semantischen Informationen sind in allen menschenzentrierten Kontexten und Anwendungen Vorteile zu erwarten.

Zusammenfassend lässt sich festhalten, dass die gewählte Objektrepräsentation mit ihren semantischen und funktionalen Bestandteilen Möglichkeiten bietet, die durch eine rein geometrische oder auf andere einfache Objektattribute beschränkte Repräsentation, wie sie in anderen Arbeiten häufig verwendet wird, nicht realisiert werden können. Neben der besseren Verständlichkeit für den Menschen, die die Kommunikation in realen Umgebungen erleichtert, wird auch eine Integration von Robotersystemen in eine menschenzentrierte Umgebung erleichtert, da die Bedeutungen der dort vorhandenen Objekte und ihre Einsatzmöglichkeiten



erfasst werden können. Dies ist eine Grundvoraussetzung z. B. für eine gelungene Handlungsplanung, die auf nicht semantisch angereicherten Modellen nur eingeschränkt möglich ist. Die hier vorgeschlagene semantische Objektrepräsentation führt somit zu Anwendungsmodellen von Objekten, die deutlich erweiterte Möglichkeiten bieten.

## 7.2 Evaluation des multimodalen interaktiven Modellierungssystems

Um das interaktive Modellierungssystem IOMOS zu bewerten, erfolgen hier zunächst einige grundsätzliche Überlegungen zur Realisierung des Systems, danach werden exemplarisch zwei Interaktionsmodi zur Modellierung von Objektattributen evaluiert.

### 7.2.1 Grundsätzliche Überlegungen

Zur Bewertung des multimodalen interaktiven Modellierungssystems IOMOS ist zum einen zu untersuchen, ob die in Abschnitt 4.1 ermittelten Anforderungen an das interaktive Modellierungssystem umgesetzt, zum anderen ob im Verlauf der Arbeit Evidenz für die in Abschnitt 1.3 formulierten Aussagen aufgezeigt werden konnte. Dies sind insbesondere die Annahmen, dass eine semantische Objektmodellierung in menschenzentrierten Umgebungen nur in Interaktion mit dem Benutzer erreichbar ist; dass die Mensch-Maschine-Schnittstelle hierbei so gestaltet werden kann, dass interaktive Objektmodellierung auch für ungeübte Benutzer möglich ist; und dass durch bestmögliche automatische Ausnutzung von sensoriiell gewonnener Information der Benutzer weitestgehend vor Routineaufgaben bei der Objektmodellierung bewahrt werden kann, so dass er nur noch überwachend tätig wird oder um dem System semantische Informationen anzubieten, die nur mit dem Hintergrundwissen des Benutzers zu erschließen sind. Als Anforderungen an das System ergaben sich aus dem gewählten Szenario:

#### **Grad der Interaktivität**

Die Objektmodellierung in IOMOS erfolgt benutzergesteuert, d. h. der Benutzer entscheidet darüber, welche Objektteile modelliert werden. Das System schlägt dabei sinnvolle Schritte vor, wie das gleichzeitige Einbinden voneinander abhängiger Attribute in eine Eigenschaft. Attributwerte, die voraussichtlich korrekt und fehlerfrei autonom bestimmt werden können, werden durch das System auch autonom erfasst. Allerdings erfolgt der Anstoß zu diesem Modellierungsschritt ebenfalls durch den Benutzer, und die erzielten Ergebnisse werden dem Benutzer zur Kontrolle bzw. gegebenenfalls zur Ergänzung dargestellt. Hierdurch wird erreicht, dass der Mensch sich auf diejenigen Schritte des Modellierungsprozesses konzentrieren kann, die seiner besonderen Aufmerksamkeit bedürfen bzw. wo seine Kenntnisse und sein Wissen notwendig sind.

#### **Intuitive Verständlichkeit auch für ungeschulte Benutzer**

Zu berücksichtigen waren hier insbesondere die Zielgruppe und die technischen und inhaltlichen Vorkenntnisse dieser Benutzer. Da im späteren Einsatzgebiet ein hohes Maß

an Interaktion mit Laien zu erwarten ist, wurden die Interaktionsmöglichkeiten intuitiv gestaltet, um auch dieser Zielgruppe eine einfache und schnell erlernbare Benutzung des Modellierungssystems zu ermöglichen (vgl. auch die detaillierteren Untersuchungen in den Abschnitten 7.2.2 und 7.2.3).

### **Ergonomische Kriterien und Bedienbarkeit**

Bei der Entwicklung der Interaktionsschnittstelle wurde darauf geachtet, diese möglichst an allgemein anerkannten Grundkriterien der Ergonomie auszurichten, wie sie in Abschnitt 5.1 zusammenfassend dargestellt werden, und somit eine möglichst hohe Bedienbarkeit zu erreichen.

### **Flexibilität und Erweiterbarkeit der Objektmodelle**

Dieser Anforderung wurde durch die Verwendung der eigens entwickelten semantischen Objektrepräsentation als Grundstruktur für die Objektmodelle Rechnung getragen.

### **Qualität der Ergebnisse**

Durch die Verwendung hochwertiger Sensorsysteme mit entsprechend hoher Genauigkeit und Auflösung konnte eine entsprechende Qualität bei den sensoruell erfassten Objektdaten erreicht werden.

### **Flexibilität und Einbindung weiterer Interaktionsmodi**

Die Möglichkeit, jederzeit weitere Interaktionsmodi in IOMOS einzubinden, wird durch eine flexible und erweiterbare Grundstruktur und entsprechende Verfahren zum Datenaustausch gewährleistet. Für weitere interaktiv zu ermittelnde Objektattribute müssen nur entsprechende neue Unterapplikationen erstellt und an das Hauptfenster der GUI angebunden werden. Die flexible Objektrepräsentation ermöglicht ein problemloses Einfügen weiterer Objektinformationen.

In IOMOS werden innovative, intuitive Interaktionsformen eingesetzt sowie Kombinationen von intuitiven Interaktionsmodi, die spezifisch auf die zu modellierenden Objektattribute abgestimmt sind. Nach erfolgter Modellierung werden die Attributwerte als Information für den Benutzer wertklassenspezifisch dargestellt. Eine rein autonome Modellierung, wie sie in manchen anderen Systemen angestrebt wird, bringt demgegenüber keinen Vorteil, da entscheidende Teile der semantischen Objektmodelle vom Benutzer beigesteuert werden, insbesondere Wissen über Eigenschaften und funktionales Objektwissen. Zusätzlich ist die semi-autonome, benutzergesteuerte Modellierung einzelner Objektattribute an vielen Stellen einem rein autonomen Ansatz überlegen, da auch hier durch Einsatz des Hintergrundwissens des Benutzers viele Verarbeitungsschritte eingespart werden können, wie bei der Erstellung einer vollständigen 3D-Punktwolke bei unregelmäßig geformten Objekten. Auch erlaubt es die enge Interaktion mit dem Benutzer, speziell diejenigen Objektinformationen zu erfassen, die für eine konkrete Anwendung in einer realen Umgebung auch benötigt werden. In diesem Sinne können die zu Beginn der Arbeit formulierten Behauptungen im Rahmen dieser Arbeit empirisch begründet werden.

### 7.2.2 Interaktive Modellierung stabiler Flächen mit dem Inertial-Joystick

Die Evaluierung des Moduls zur interaktiven Modellierung stabiler Flächen unter Verwendung des Inertial-Joysticks umfasst zum einen den Inertial-Joystick als Interaktionsmodus an sich, zum anderen seine Nutzung zur Modellierung stabiler Flächen (vgl. auch Kasper (2007)). Zur ersten Frage zeigte sich bei Verwendung des Gerätes, dass zwar der Inertialsensor an sich ausreichend schnell und präzise für die gewünschte Anwendung ist, dass aber die Befestigung dieses Sensors an einem Joystick zu praktischen Problemen führt. Insbesondere fiel manchen Versuchspersonen die präzise Steuerung des Joysticks und seine Drehung um die jeweils gewünschte Achse schwer. Zudem führte ein Drücken der Knöpfe am Joystick, z.B. um eine ausgewählte Orientierung zu bestätigen, immer wieder zu einem Verziehen des Geräts und damit zur Auswahl einer gegenüber der Zielorientierung leicht verschobenen Orientierung. Diese Probleme verschwanden, wenn der Inertialsensor separat verwendet wurde und die Auswahl der gewünschten Orientierung per Mausklick erfolgte. Für zukünftige Anwendungen ist daher eine Verbesserung des Gerätes geplant, indem der Inertialsensor möglichst kabellos in ein entsprechend verbessertes, beispielsweise kugelförmiges, Gehäuse eingebunden wird.

Zur interaktiven Modellierung stabiler Flächen mit dem Inertialsensor wurde eine Evaluierung mit Versuchspersonen durchgeführt, bei der ausschließlich diese Unterapplikation untersucht und bewertet wurde. Eine beispielhafte Modellierungssequenz der stabilen Flächen eines Objektes wurde bereits in Abb. 5.21 dargestellt. Zur Evaluation wurden insgesamt acht Probanden gebeten, die stabilen Flächen einer quaderförmigen Lebensmittelpackung (also insgesamt sechs Flächen) interaktiv zu modellieren, wobei aufgrund der kleinen Zahl an Versuchspersonen statistisch nur Tendenzen aufgezeigt werden können. Alle Versuchsteilnehmer verfügten über Erfahrungen im Umgang mit Rechnern und zusätzlich über unterschiedlich umfangreiche Kenntnisse auf dem Gebiet der Objektmodellierung. Die Probanden wurden in zwei Gruppen aufgeteilt. Beiden Gruppen wurde die Problemstellung erläutert, eine der beiden Gruppen erhielt zusätzlich eine Einführung in die graphische Oberfläche zur Modellierung stabiler Flächen. Diese Subapplikation wurde für den Versuch getrennt von der Haupt-GUI betrieben, um diesen Interaktionsmodus ohne störende Nebeneffekte zu untersuchen. Jeder Proband modellierte die stabilen Flächen des Objektes jeweils drei Mal mit dem Inertialsensor. Aufgezeichnet wurden jeweils die zur Modellierung benötigte Zeit und die ermittelten stabilen Flächen. Zusätzlich wurde jede Versuchsperson nach Ablauf der Modellierungsphase mit einem Fragebogen nach ihrer Einschätzung zu den folgenden Punkten befragt:

- Inhalt der graphischen Oberfläche: Vollständigkeit, Verständlichkeit der Präsentation
- Strukturierung der graphischen Oberfläche: Übersichtlichkeit, gelungene Aufteilung
- Ergonomie der verwendeten Geräte, unabhängig von der gestellten Aufgabe
- Einsatz der Geräte für diese Aufgabe: prinzipielle Eignung, gelungene tatsächliche Einbindung
- Intuitivität der graphischen Oberfläche
- Intuitivität der Geräte

Die Bewertung war dabei in Schulnoten von 1 bis 6 abzugeben. Zusätzlich wurden frei formulierte Kommentare und Verbesserungsvorschläge abgefragt.

Bei der Versuchsdurchführung ergaben sich gravierende Unterschiede zwischen den beiden Gruppen nur hinsichtlich des Ausmaßes, in dem die Möglichkeiten der graphischen Oberfläche zur Unterstützung eingesetzt wurden. Probanden ohne Einführung in die Bedienung der graphischen Oberfläche waren oft nicht in der Lage, sich die unterschiedlichen Anzeige- und Nutzungsvarianten der graphischen Oberfläche zu erschließen. Zusätzlich gab es einzelne Probanden, die große Probleme mit der Erfassung einer dreidimensionalen Objektdarstellung und der Zuordnung der angezeigten stabilen Flächen hatten. Hierbei handelt es sich vermutlich um grundsätzliche Defizite in der räumlichen Orientierung. Eine einfache Lösung hierfür in Form einer geeigneten verständlichen Darstellung dreidimensionaler Daten existiert dabei nicht, durch häufige Benutzung solcher Darstellungen treten aber Gewöhnungseffekte auf, die die Benutzung erleichtern.

Bearbeitungszeit [min]	1. Modellierungs durchlauf	2. Modellierungs durchlauf	3. Modellierungs durchlauf
Durchschnitt	7,9	5,15	3,15
Standardabweichung	5,33	2,88	2,00

Tabelle 7.1: Bearbeitungszeiten für die Modellierung der stabilen Flächen

Bei den Bearbeitungszeiten für die Modellierungsaufgabe (vgl. Tab. 7.1) zeigten sich keine offensichtlichen Unterschiede zwischen den beiden Versuchspersonengruppen. Eine Person benötigte auffällig lange für die Bearbeitung, dies erklärt auch die relativ hohen Standardabweichungen. Hier wäre eine größere Zahl von Versuchspersonen notwendig, um eine verlässliche Statistik zu erhalten. Da alle anderen Versuchspersonen aber relativ ähnliche Bearbeitungszeiten benötigten, ist davon auszugehen, dass der Durchschnitt insgesamt aussagekräftig ist und es sich bei dieser Person um einen vernachlässigbaren Ausreißer handelt. Deutlich zu sehen ist, wie die durchschnittliche Bearbeitungszeit mit der Zahl an Durchläufen abnimmt. Hier zeigt sich somit ein hoher Lerneffekt bei den Versuchspersonen und eine Gewöhnung an das Interaktionsgerät. Gleichzeitig nimmt die Standardabweichung mit der Zahl der Durchläufe stark ab. Dies lässt darauf schließen, dass Versuchspersonen, die am Anfang besonders lang zur vollständigen Modellierung der stabilen Flächen benötigten, besonders stark von einem Lern- und Gewöhnungseffekt profitierten. Falls diese Beobachtung sich bei einer größeren Zahl von Probanden bestätigen sollte, würde dies besonders für die Eignung des Gerätes bei Verwendung durch wenig technisch vorgebildete Benutzer sprechen, da hier entsprechend ein großer Lerneffekt zu erwarten wäre. Diese Benutzer sollten in diesem Fall innerhalb weniger Modellierungsdurchläufe so geübt werden, dass sie in etwa den Stand von besser vorgebildeten Benutzern erreichen. Die gemessenen Bearbeitungszeiten können allerdings nur mit Vorsicht als absolute Zeitdauern interpretiert werden, da die Versuchspersonen aufgrund der fehlenden Vorgaben für die zu erreichende Zielgenauigkeit der Modellierung Unterschiede in der angestrebten Bearbeitungsgenauigkeit und damit auch der Bearbeitungszeit aufwiesen.

Bei der Beurteilung der Korrektheit der ermittelten stabilen Flächen ist zum einen die Richtung der ermittelten Normalenvektoren zu betrachten, die aufgrund der Quaderform des

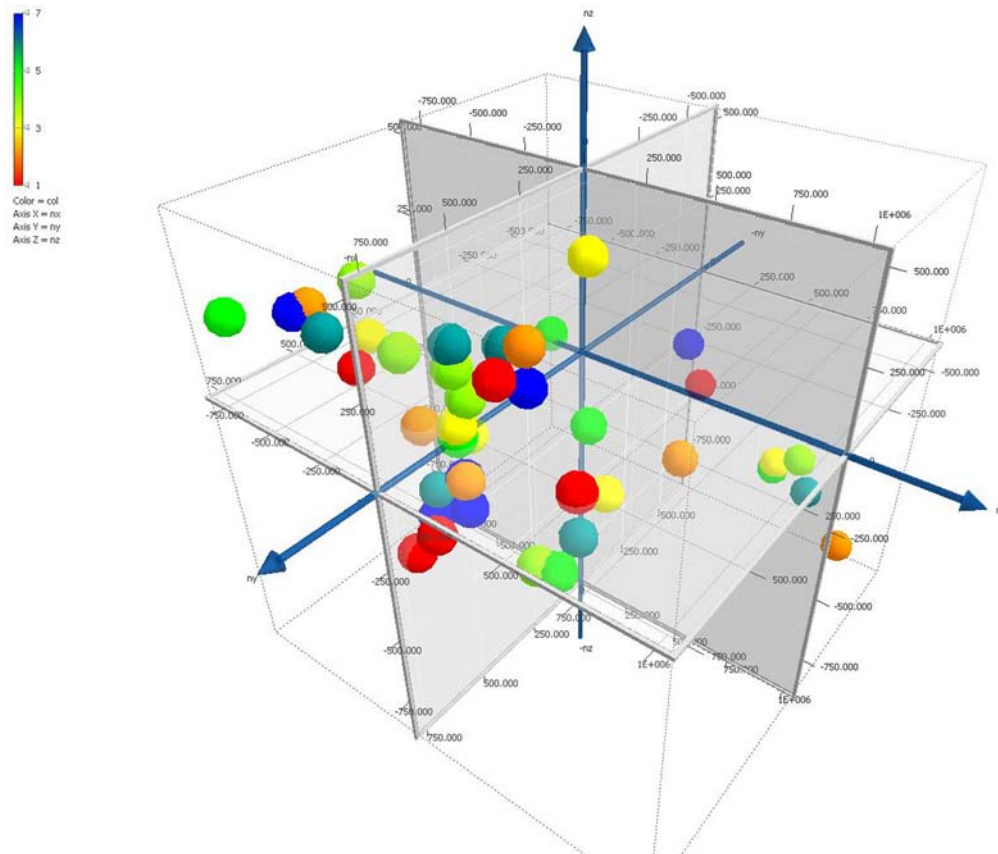


Abbildung 7.1: Bezugspunkte der von den Probanden ermittelten stabilen Ebenen, farblich getrennt nach Probanden aufgetragen (aus: Kasper (2007), S. 79)

Objektes im Idealfall jeweils parallel zu den Achsen bzw. negativen Achsenrichtungen liegen müssen. Zum zweiten können die Bezugspunkte der Ebenen zwischen den Versuchspersonen verglichen werden. Hier ergibt sich eine relativ große Variation der Bezugspunkte (vgl. Abb. 7.1). Dies bedeutet, dass die zur Ermittlung der stabilen Flächen herangezogenen Punktemenge sich stark von Versuchsperson zu Versuchsperson unterschieden hat. Ob dies die Qualität der Ergebnisse in nennenswertem Maße beeinträchtigt, müsste durch weitere Untersuchungen geklärt werden. Die durchschnittliche Abweichung der ermittelten Normalenvektoren von den Achsenrichtungen betrug  $7,66^\circ$  (Standardabweichung  $4,69$ ) gemittelt über alle Versuchspersonen (vgl. Darstellung der ermittelten Ebenennormalen pro Versuchsperson in Abb. 7.2). Dies ist als Grundlage für praktische Anwendungen ein gutes Ergebnis, insbesondere da die Streuung über die Versuchspersonen relativ gering war, d. h. es gelang allen Probanden, eine ausreichend gute Modellierung der stabilen Flächen zu erreichen. Auffällig war, dass die Modellierung der Stirnseiten des Quaders insgesamt deutlich bessere Ergebnisse brachte als die Modellierung der anderen Seiten. Dies war zum Teil der Ausgangsstellung des dreidimensionalen Quadermodells zuzuschreiben, die die Erfassung einer der beiden Stirnseiten besonders vereinfachte. Zum anderen begünstigte die Form des Interaktionsgerätes die Modellierung dieser beiden Seiten und zudem waren die kleineren Stirnseiten für die Versuchspersonen leichter zu halten als die vergleichsweise größeren anderen Seiten. Mit einer verbesserten Version des Interaktionsgerätes sollten solche Unterschiede geringer werden.

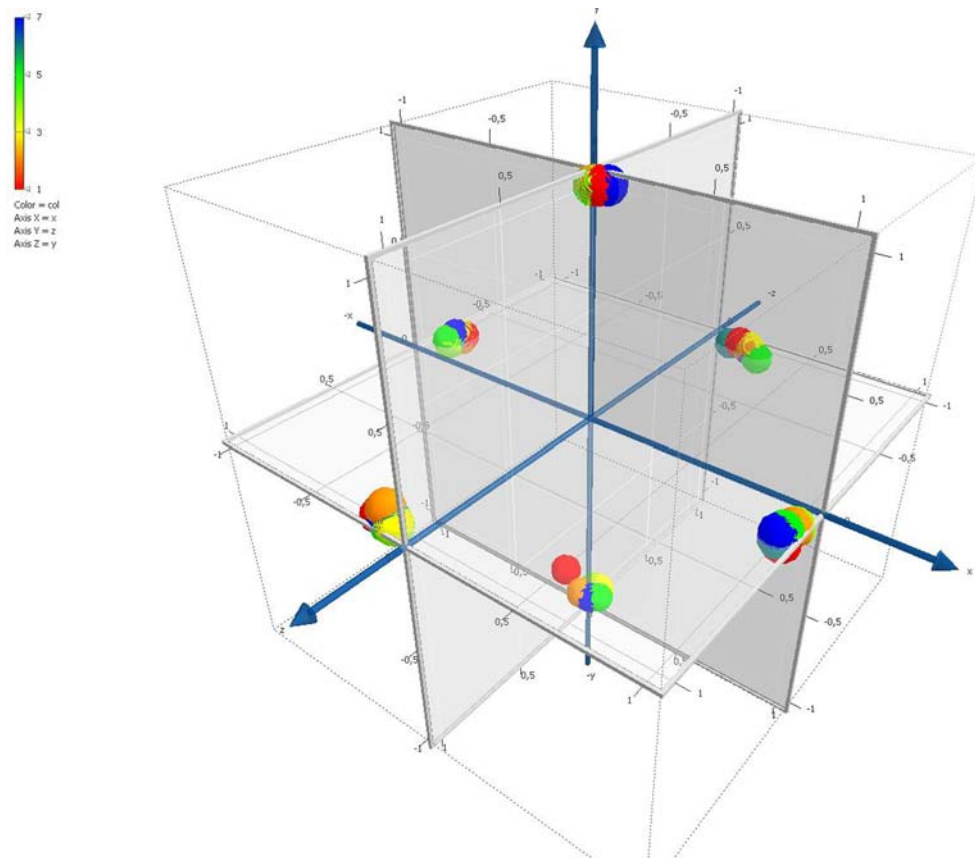


Abbildung 7.2: Normalenvektoren der von den Probanden ermittelten stabilen Ebenen, farblich getrennt nach Probanden aufgetragen (aus: Kasper (2007), S. 79)

Bewertung in Schulnoten	Graph. Oberfläche		Interaktionsgerät		Intuitivität	
	Inhalt	Strukturierung	Ergonomie	Einbindung	Graph. Oberfläche	Interaktionsgerät
Durchschnitt	2	1,875	2,875	2,25	1,625	1,5
Standardabweichung	0,53	0,83	1,36	1,39	0,74	0,53

Tabelle 7.2: Bewertung der Modellierung der stabilen Flächen durch die Versuchspersonen

Die Bewertung dieses Modellierungsverfahrens durch die Versuchspersonen ist divergent (vgl. Darstellung der Ergebnisse in Tab. 7.2). Einerseits wurde die Intuitivität der Modellierung mit dem Inertialsensor gelobt und in Freitext-Aussagen auch sein Einsatz zur Modellierung von stabilen Flächen als intuitiv hervorgehoben. Andererseits wurde diese Form der Interaktion auch wiederholt kritisiert, da es vielen Versuchspersonen schwer fiel, den Inertialsensor so ruhig in der Hand zu halten, wie dies für eine genaue Modellierung der stabilen Flächen notwendig ist. Dies spiegelt sich auch in der vergleichsweise schlechten Durchschnittsnote für die Ergonomie des Interaktionsgerätes. Hier besteht also wie oben bereits angemerkt der Bedarf, die Interaktion mit dem Inertialsensor zu verbessern und in eine geeignetere Hardware einzubinden. Insgesamt waren die Kommentare der Versuchspersonen bezüglich der Verwendung eines Inertialsensors zur Modellierung stabiler Flächen jedoch so ermutigend,

dass diese Interaktionsmöglichkeit mit den notwendigen Verbesserungen unbedingt weiter angeboten werden sollte. Obwohl das Gerät als solches noch Raum für Verbesserungen lässt, wurde sein Einsatz zur Bedienung und Modellierung der stabilen Flächen doch als einleuchtend und intuitiv hervorgehoben. Die graphische Oberfläche dieses Moduls wurde insgesamt positiv bewertet, auch hier nannten die Probanden jedoch Verbesserungsvorschläge wie das Anzeigen der Objekte als Dreiecksnetz anstatt als Punktwolke zur besseren Übersichtlichkeit, oder die Anpassung der Form und Größe der eingeblendeten stabilen Flächen an das tatsächliche Objekt.

### 7.2.3 Interaktive Modellierung von Manipulationseinschränkungen mit dem magnetfeldbasierten Positionssensor

Die Evaluation der interaktiven Modellierung von Manipulationseinschränkungen mit dem auf dem Handrücken befestigten magnetfeldbasierten Positionssensor wurde analog zur Evaluation der Modellierung von stabilen Flächen durchgeführt, wie sie im vorhergehenden Abschnitt beschrieben wurde (vgl. auch Kasper (2007)). Hier konnten allerdings die erzielten Ergebnisse nicht bewertet werden, da die Anwendung eine freie Wahl der vorgeführten Manipulationseinschränkungen zulässt. Im Gegensatz zur Modellierung der stabilen Flächen konnten hier bei der Vorführung keine Unterschiede zwischen den Versuchspersonengruppen festgestellt werden: Bei der Auswertung der Bearbeitungszeiten und bei der Beurteilung des Systems durch die Versuchspersonen ergaben sich keine offensichtlichen Unterschiede zwischen den beiden Gruppen von Versuchspersonen, so dass im Folgenden nicht mehr zwischen ihnen unterschieden wird.

Bearbeitungszeit [min]	1. Modellierungs durchlauf	2. Modellierungs durchlauf	3. Modellierungs durchlauf
Durchschnitt	3,75	1	1
Standardabweichung	0,89	0	0

Tabelle 7.3: Bearbeitungszeiten für die Modellierung von Manipulationseinschränkungen

Die durchschnittlichen Bearbeitungszeiten pro Durchlauf und deren Standardabweichungen sind in Tab. 7.3 wiedergegeben. Es zeigt sich hierbei ein relativ schneller und großer Lerneffekt, bereits nach dem ersten Durchlauf benötigen alle Versuchspersonen nur noch ca. 1 Minute, um die gewünschten Manipulationseinschränkungen zu modellieren. Dieser Lerneffekt ist zum einen dadurch bedingt, dass die Versuchspersonen nach dem ersten Durchlauf den Versuchsablauf bereits kannten, zum anderen durch einen geübteren Umgang mit der interaktiven Schnittstelle zur Datenaufnahme.

Bei der Bewertung durch die Versuchspersonen schnitt das Modellierungsverfahren durchweg sehr gut ab. Die Ergebnisse der Befragung der Versuchspersonen sind in Tab. 7.4 abgebildet. Hervorzuheben sind hier vor allem die Intuitivität der Hardware sowie ihre Verwendung zur Modellierung der Manipulationseinschränkungen – beide Aspekte wurden von den Versuchspersonen äußerst positiv bewertet. Auch die graphische Oberfläche zur Modellierung von Manipulationseinschränkungen wird als intuitiv und gut strukturiert empfunden. Dies

Bewertung in Schulnoten	Graph. Oberfläche		Interaktionsgerät		Intuitivität	
	Inhalt	Strukturierung	Ergonomie	Einbindung	Graph. Oberfläche	Interaktionsgerät
Durchschnitt	1,5	1,375	1,5	1,375	1,25	1,125
Standardabweichung	0,53	0,52	0,76	0,74	0,46	0,35

Tabelle 7.4: Bewertung der Modellierung von Manipulationseinschränkungen durch die Versuchspersonen

ist möglicherweise mit auf die Assistentenform zurückzuführen, die die Nutzung der graphischen Oberfläche auch ohne explizite Einführung erleichtert. Dies würde auch erklären, warum bei diesem Versuch sowohl die Personengruppe mit Einführung in die Benutzeroberfläche als auch die Gruppe ohne Einführung etwa mit den gleichen Ergebnissen abschnitten. Die Versuchspersonen merkten außerdem an, dass klarere Vorgaben bezüglich der durchzuführenden Bewegungen möglicherweise die Modellierung erleichtern könnten. Angesichts der Tatsache, dass der Aufbau ansonsten als intuitiv und verständlich gelobt wurde, genügt hier vermutlich die Erfahrung von wenigen Durchläufen pro Benutzer, um diese Unsicherheit zu beseitigen. Insgesamt lässt sich festhalten, dass der gewählte Interaktionsmodus und die graphische Benutzerschnittstelle intuitiv und gut verständlich sind, und dass die Wahl dieser Interaktionsmöglichkeit zur Modellierung von Manipulationseinschränkungen sich als sinnvoll und vielversprechend erwiesen hat.

## 7.3 Evaluation der resultierenden Objektmodelle

Um die Qualität der hier erstellten Objektmodelle zu bewerten, muss zunächst die Frage nach einem geeigneten Qualitätsmaß beantwortet werden. Erst wenn dieses bestimmt ist, können Aussagen über die erreichten Ergebnisse vorgenommen werden.

### 7.3.1 Bewertung der Qualität der Objektmodelle

Generell bieten sich zur Bewertung von Objektmodellen unterschiedliche Qualitätsmaße an, die je nach Hintergrund und Zielsetzung der Modellierung mehr oder weniger aussagekräftige Ergebnisse liefern. Mögliche Bewertungskriterien können dabei sein:

#### Genauigkeit

Die Genauigkeit von Objektmodellen bezieht sich in der Regel auf einzelne Objektattribute, z. B. die Auflösung eines 3D-Modells. Eine Bewertung nach diesem Kriterium ist also in hohem Maße spezifisch für die verwendeten Sensorsysteme und Methoden der Sensordatenverarbeitung und kann sich somit nur auf einzelne Details eines Objektmodells beziehen, aber nicht auf ein Objektmodell bzw. eine Objektrepräsentation als Ganzes. Somit kommt dieses Kriterium vor allem dann zur Verwendung, wenn



bestimmte Sensorsysteme oder innovative Vorgehensweisen zur Bestimmung von Objektattributen bewertet werden sollen. Die Evaluation von komplexen semantischen Objektmodellen, wie sie in der vorliegenden Arbeit erstellt wurden, anhand dieses Gütekriteriums ist daher nicht sinnvoll.

### **Zugriffszahlen**

Eine weiteres mögliches Bewertungskriterium für Objektmodelle, die für Forschungsprojekte zur Verfügung gestellt werden, sind die Zugriffs- bzw. Verwendungszahlen der Modelle. Diese sind jedoch höchstens dann aussagekräftig, wenn die Verwendung direkt quantitativ gemessen werden kann, und nur zum Vergleich verschiedener Modelle in der gleichen Ausgangssituation. Bei unterschiedlichen Ausgangssituationen oder Einsatzzwecken bzw. zur grundsätzlichen Bewertung eines Typs von Objektmodellen ist dieses Kriterium nicht geeignet, da hier die Zugriffszahlen nicht in geeigneter Relation betrachtet werden können.

### **Einsatzmöglichkeiten und qualitative Verwendung**

Weiterhin können als Maß für die Güte von Objektmodellen deren Einsatzmöglichkeiten herangezogen werden. Je mehr unterschiedlichen anderen Komponenten beispielsweise in einem Robotersystem die Objektmodelle wichtige Informationen liefern und von je mehr anderen Komponenten die Modelle mit Informationen angereichert werden, umso vielfältiger und flexibler sind die Objektmodelle. Es handelt sich hierbei allerdings um ein rein qualitatives Bewertungskriterium, das insbesondere im Hinblick auf das jeweilige Ziel der Modellierung gesehen werden muss. Es sind durchaus Einsatzgebiete vorstellbar, in denen möglichst einfache Objektmodelle von Nutzen sind, die nicht mehr Information enthalten als für eine sehr spezielle Aufgabe benötigt wird. Im Hinblick auf den Einsatz in menschenzentrierten, komplexen Umgebungen können allerdings solche Objektmodelle als positiv bewertet werden, die möglichst vielfältige Informationen enthalten und daher für möglichst viele andere Systembestandteile relevant sind.

Im Hinblick auf die Ziele der vorliegenden Arbeit, d. h. insbesondere auf die propagierte interaktive semantische Modellierung, und auf die angestrebte Verwendung der Objektmodelle in einem Robotersystem in einer menschenzentrierten Umgebung, ist das erste Bewertungskriterium hier nicht aussagekräftig. Die Genauigkeit der Sensordaten, z. B. die Auflösung des 3D-Modells, kann nicht zur Bewertung herangezogen werden, da die eingesetzten Sensoren hier nur exemplarisch verwendet wurden und jederzeit durch andere Sensorsysteme ersetzt werden können. Für die semantischen und interaktiv angelegten Anteile der Objektmodelle ist eine Bewertung nach quantitativen Maßstäben ebenfalls nicht möglich, da hier der Modellierungsvorgang an sich im Mittelpunkt der Arbeit stand (vgl. auch die Untersuchung der interaktiven Modellierungsmodi in den Abschnitten 7.2.2 und 7.2.3). Die Ergebnisse dieses Modellierungsvorgangs sind wiederum sehr stark abhängig vom Benutzer, vom Kontext und von der gewünschten Anwendung, so dass hier eine qualitative Evaluation wenig sinnvoll ist.

Eine Bewertung der Objektmodelle nach den Zugriffszahlen auf die Objektmodelle ist im Rahmen der vorliegenden Arbeit ebenfalls wenig aussagekräftig, da im Rahmen der übergeordneten Roboterarchitektur nicht direkt auf die Objektmodelldatenbank zugegriffen wurde, sondern indirekt über das aktuelle Szenenmodell (vgl. Darstellung der Architektur in

Abb. 7.3). Gleichzeitig werden die erzeugten Objektmodelle auch über eine Internetseite anderen Forschungsgruppen zur Verfügung gestellt. Auch hier erscheint die Angabe von Zugriffszahlen relativ willkürlich, da diese z. B. durch gezielte Werbung stark beeinflusst werden können.

Somit bleibt als aussagekräftige Evaluationsmöglichkeit eine qualitative Bewertung der erzielten Objektmodelle im Hinblick auf ihre Verwendung und Nutzungsmöglichkeiten. Dies erfolgt im nächsten Abschnitt sowohl in Bezug auf die konkrete eingesetzte Roboterarchitektur als auch im Hinblick auf eine Verwendung der Objektmodelle, wie sie beispielsweise über das Internet zur Verfügung gestellt werden, in anderen Arbeiten.

### 7.3.2 Verwendungsmöglichkeiten und Nutzung der erstellten Objektmodelle

In Kap. 4 wurden bereits die Details der IOMOS-Architektur und des Datenflusses zu und von anderen Systemkomponenten eines Robotersystems erläutert. Hier soll daher insbesondere die Verwendung der erstellten semantischen Objektmodelle innerhalb eines komplexen Robotersystems und damit deren Bedeutung innerhalb eines solchen Systems dargestellt werden. Die im Rahmen dieser Arbeit erstellten Objektmodelle wurden in zwei Projekten aus dem Bereich Humanoide Robotik bzw. Servicerobotik eingesetzt, der „Deutschen Servicerobotik Initiative“ DESIRE (gefördert vom Bundesministerium für Bildung und Forschung) und dem Sonderforschungsbereich 588 „Humanoide Roboter – Lernende und kooperierende multimodale Roboter“ (gefördert von der Deutschen Forschungsgemeinschaft DFG). Während IOMOS im Rahmen von DESIRE hauptsächlich einzelne Objektattribute lieferte, z. B. Objektansichten zur Erzeugung von SIFT-Features für die Objekterkennung, wird im SFB Humanoide Roboter ein komplexeres Objektmodell eingesetzt, das für viele anderen Systemkomponenten Bedeutung hat. Im folgenden wird daher die Verwendung der erzeugten semantischen Objektmodelle im SFB Humanoide Roboter detailliert betrachtet.

Wie in Abb. 7.3 dargestellt, ist in der zugrundeliegenden Prozesskette des Robotersystems das Szenenmodell das Bindeglied zwischen den in IOMOS angelegten Objektklassen und -instanzen und den darauf zugreifenden anderen Systemkomponenten. Im Szenenmodell wird ein Modell der jeweils aktuellen Szene aufgebaut und vorgehalten, das auch die in der Szene vorhandenen Objekte beinhaltet. Diese werden vom Szenenmodell aus der Objektmodell-datenbank ausgelesen und mittels einer eindeutigen Objekt-Identifikationskennung in die aktuelle Szene eingefügt. Alle anderen Komponenten können dann bei Bedarf die gewünschten Objektinformationen über das Szenenmodell und die Objekt-Kennung auslesen. Hierbei ist zu beachten, dass nicht alle Komponenten auf alle Objektinformationen zugreifen. Wie in der Grafik dargestellt, werden im Gegenteil sehr unterschiedliche Informationen von den einzelnen Komponenten berücksichtigt. Die Vielfältigkeit, Flexibilität und Erweiterbarkeit der Objektrepräsentation erlaubt dabei die Modellierung dieser und weiterer sehr verschiedener Objektinformationen. Gleichzeitig können andere Systemkomponenten auch Ergänzungen an den Objektinstanzen in der aktuellen Szene vornehmen, beispielsweise wenn eine Komponente detektiert hat, dass sich der aktuelle Füllstand eines Objektes verändert hat (vgl. auch Darstellung der Datenverwaltung in den Abschnitten 4.3.2 und 4.3.3). Wesentliche Systemkomponenten, die auf die Objektmodelle zugreifen, sind:

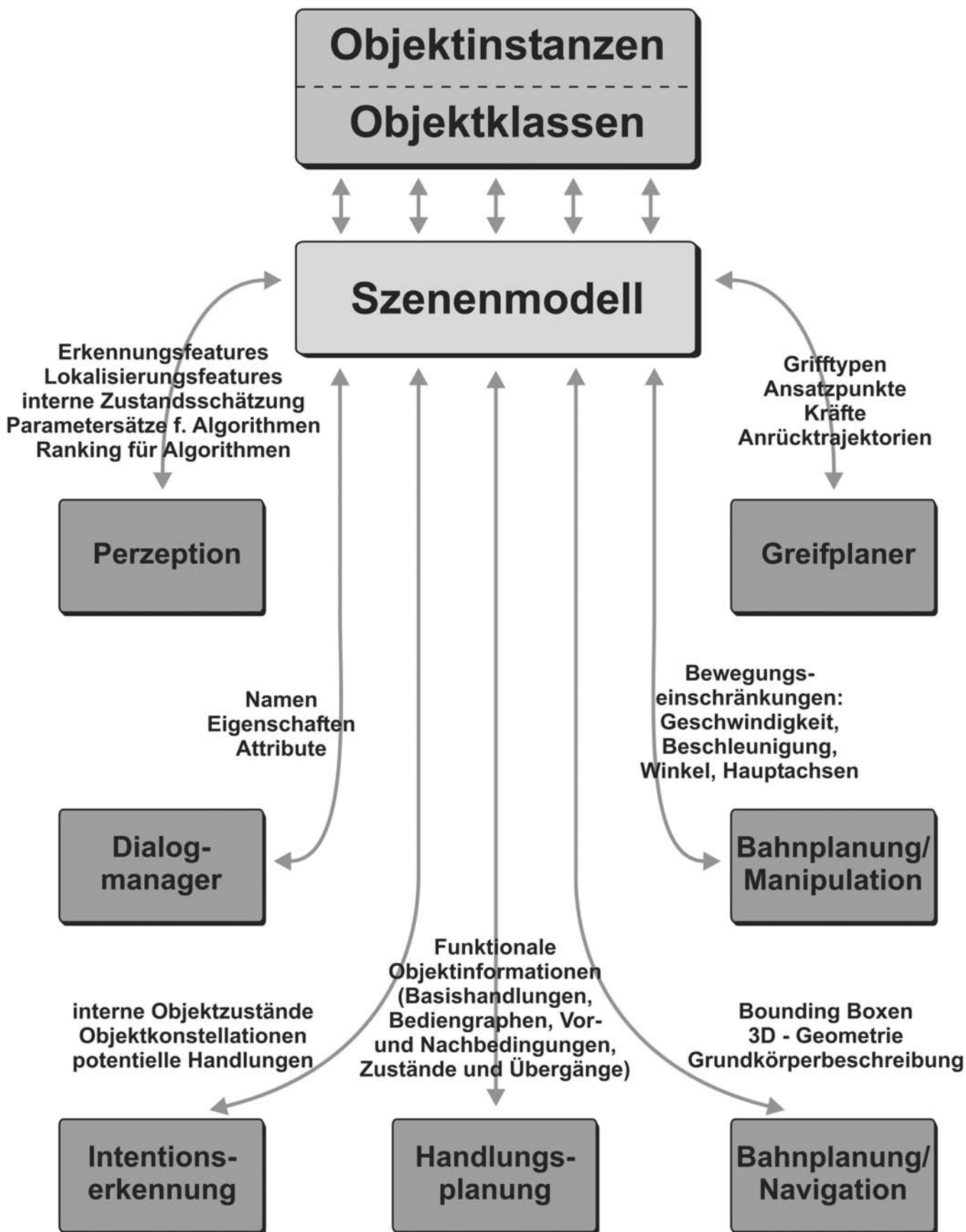


Abbildung 7.3: Einbindung der Objektmodelle in die Gesamtarchitektur eines Robotersystems im Sonderforschungsbereich SFB 588 „Humanoide Roboter“

## Perzeption

Die Perzeption eines Robotersystems ist eng mit seinen Objektmodellen verknüpft. Detaillierte Objektinformationen wie die in DESIRE aus den Objektansichten abgeleiteten SIFT-Features sind beispielsweise für die Objekterkennung unabdingbar. Von den Perzeptionskomponenten benötigte objektspezifische Detailinformationen wie Parametersätze oder Rankings für Algorithmen können von diesen Komponenten wiederum bei den einzelnen Objektklassen und -instanzen nach Bedarf ergänzt und später mit diesen Objektmodellen wieder abgerufen werden.

## Dialogmanager

Im Dialogmanager sind vornehmlich diejenigen Objektinformationen von Bedeutung, die eine effiziente und erfolgreiche Kommunikation mit dem menschlichen Benutzer ermöglichen. Da beim Entwurf der Objektrepräsentation besonderes Augenmerk darauf gelegt wurde, die Struktur der Objektmodelle möglichst verständlich für den Menschen zu halten, bieten sich die vorliegenden Objektmodelle hierfür geradezu an. Wesentliche Informationen sind in diesem Zusammenhang die Namen, Eigenschaften und Attribute von Objekten.

## Intentionserkennung

Um die Intention des menschlichen Benutzers zu erkennen und adäquat zu reagieren, muss das Robotersystem Zugriff auf Informationen wie interne Objektzustände und potentiell mit Objekten ausführbare Handlungen haben, wie sie im Rahmen der Objektmodelle zur Verfügung gestellt werden.

## Handlungsplanung

Für die Planung der Roboterhandlungen von besonderer Bedeutung sind funktionale Informationen über die beteiligten Objekte wie z. B. mit Objekten ausführbare Handlungen mit ihren Vor- und Nachbedingungen in Bezug auf Objektattribute.

## Bahnplanung/Navigation

Eine erfolgreiche Bahnplanung zur Navigation ist nur möglich, wenn eine gewissen Menge an geometrischen Informationen und Informationen über die Objektform vorliegt. Je nach Situation und eingesetztem Bahnplanungsalgorithmus werden hier beispielsweise Bounding Boxen oder 3D-Geometrien aus den Objektmodellen verwendet.

## Bahnplanung/Manipulation

Die Bahnplanung zur Manipulation beinhaltet insbesondere die Planung von Armbewegungen z. B. zum Greifen. Hierzu stellen die Objektmodelle Informationen zur Verfügung wie Bewegungseinschränkungen für Objekte, d. h. maximale Geschwindigkeiten, maximale Beschleunigungen und maximale Kippwinkel.

## Greifplaner

Zur detaillierten objektspezifischen Greifplanung werden neben in Frage kommenden Griffotypen auch Greif-Ansatzpunkte, Greifkräfte und Anrücktrajektorien benötigt. Diese wurden nicht im Rahmen der vorliegenden Arbeit bestimmt, da hierzu bereits einige Arbeiten existierten. Die in anderen Komponenten des Robotersystems ermittelten diesbezüglichen Ergebnisse können über die oben beschriebenen Mechanismen ergänzend bei den Objektmodellen abgelegt werden. Zusätzlich wurde ein Datenhandschuh bereits in die Objektmodellierungsumgebung aufgenommen und somit

die Grundlage für die Einbindung von Methoden zum Lernen von Greifinformationen gelegt.

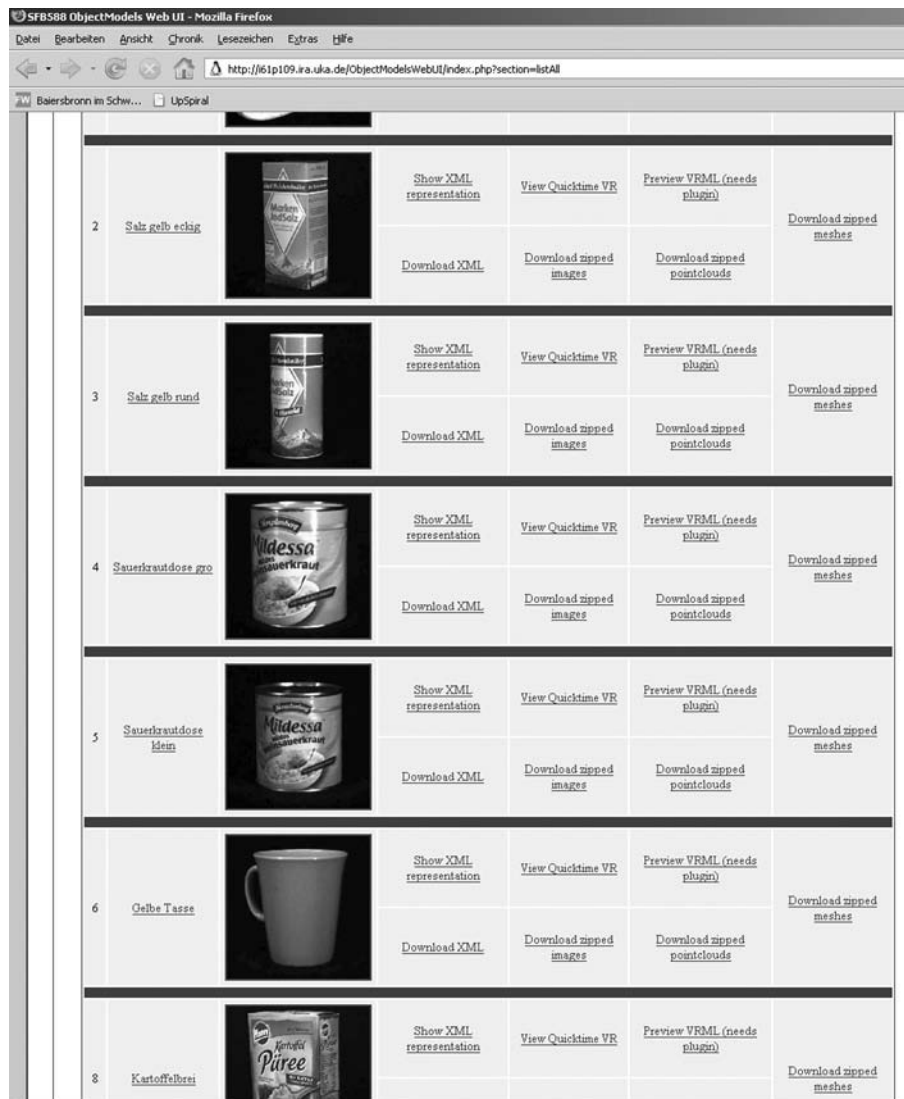


Abbildung 7.4: Im Internet zur Verfügung gestellte Objektmodelle mit 3D-Vorschau, Homepage erstellt von A. Kasper

Um die im Rahmen dieser Arbeit erzielten Ergebnisse einer möglichst breiten Forschungsöffentlichkeit zugänglich zu machen, werden die erstellten Objektmodelle anderen Forschungsprojekten auch auf einer Homepage zur Verfügung gestellt.<sup>1</sup> Auf dieser Homepage finden sich neben Informationen zu der verwendeten Sensorik und der Modellierungsumgebung die Objektmodelle in unterschiedlicher Form. Zum Speichern bzw. zur Ansicht zur Verfügung stehen die erzielten XML-Repräsentationen, Objektansichten, Punktwolken und Dreiecksnetze sowie eine Vorschau auf das dreidimensionale Objekt als Quicktime VR (aus Kameransichten zusammengesetzt) und als VRML-Darstellung. Abb. 7.4 zeigt einen Ausschnitt aus der Homepage mit den zur Verfügung stehenden Formaten.

<sup>1</sup><http://www.iaim.ira.uka.de/ObjectModels>, Homepage erstellt von Alexander Kasper.

## 7.4 Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE

Im Folgenden wird der Ablauf einer vollständigen Modellierungssequenz in IOMOS anhand einiger interessanter Ausschnitte dargestellt und erläutert (vgl. Abbildungen 7.5 bis 7.9). Erstellt wird hier ein Objektmodell für die Objektklasse TASSE. Hierzu werden vom Benutzer zunächst Schablonen für die relevanten Attribute, Handlungen und Zustände in der graphischen Oberfläche erstellt. Diese dienen wiederum als Grundlage für das Erstellen von Eigenschaftsschablonen, in die die jeweils passenden Attribute, Handlungen und Zustände eingebunden werden. Dabei werden die Standardattributwerte falls notwendig angepasst. Danach wird in der graphischen Benutzeroberfläche die Objektklasse TASSE mit grundlegenden Informationen wie ihrem Namen angelegt. Dies kann bequem direkt neben der Modellierungsumgebung auf dem Plasmabildschirm mit berührungssensitivem Aufsatz geschehen. Der Objektklasse werden nun ebenfalls im Hauptfenster der GUI ihre Eigenschaften hinzugefügt. Hierdurch werden ihr automatisch auch die Attribute, Handlungen und Zustände zugewiesen, die diese Eigenschaften mit sich bringen. Dabei können noch Korrekturen an den Standard-Attributwerten erfolgen, soweit dies notwendig ist.

Anschließend wird das zu modellierende Objekt in der Modellierungsumgebung auf dem Drehteller platziert, damit es von den dort vorhandenen Sensoren erfasst werden kann. Grundsätzlich wird, wie oben beschrieben, diese sensorielle Erfassung immer vom Menschen angestoßen, der auch entscheidet, welche Attributwerte in welcher Reihenfolge zu erfassen sind. In der Praxis bewährt es sich allerdings, einige grundlegende Attributwerte wie die dreidimensionale Punktwolke und Objektansichten zuerst zu bestimmen, da aus ihnen viele andere Attributwerte berechnet werden. Der Benutzer legt in einem Unterfenster der GUI die Parameter zur Aufnahme der dreidimensionalen Objektgeometrie und der Objektansichten fest. Dies sind hier insbesondere die Winkelabstände des Drehtellers und des Kamerabügels, in denen Aufnahmen gemacht werden sollen, sowie die gewünschten Beleuchtungssituationen. Nach Beendigung der Aufnahmen kann das Ergebnis vom Benutzer korrigiert und falls erforderlich noch weitere Aufnahmen aus anderen Rotationswinkeln gemacht werden.

Auch die Modellierung aller weiteren Attributwerte wird zunächst im Hauptfenster der GUI vom Benutzer angestoßen. Teilweise können die Werte direkt textuell in diesem Hauptfenster eingegeben werden, wie dies oben bereits für den Objektklassennamen erfolgte. Bei komplexeren Attributwerten öffnen sich jeweils eigene Unterdialoge, in denen eine Modellierung erfolgt. Dies ist beispielsweise der Fall bei der Modellierung der Hauptachsen des Objektes (vgl. Abschnitt 5.5.5) sowie der Modellierung der stabilen Flächen, die wie in Abschnitt 5.5.4 beschrieben abläuft. Dieser Modellierungsschritt erfolgt mit dem Inertial-Joystick, einem neuartigen, intuitiven Interaktionsgerät. Auch die Modellierung der Manipulationseinschränkungen wurde oben bereits detailliert beschrieben. Hier erfolgt nach der Auswahl der zu betrachtenden Objektachsen eine Modellierung unter Verwendung eines magnetfeldbasierten Positionssensors. Nach Abschluss aller gewünschten Modellierungsschritte liegt das Objekt dann vollständig im Hauptfenster der GUI vor, wo die Attributwerte soweit als möglich direkt angezeigt werden. Das Objektklassenmodell und die Schablonen können jederzeit in der Objektmodelldatenbank gespeichert werden und aus dieser auch später wieder abgerufen werden, um in der GUI erneut angezeigt, verändert oder ergänzt zu werden.

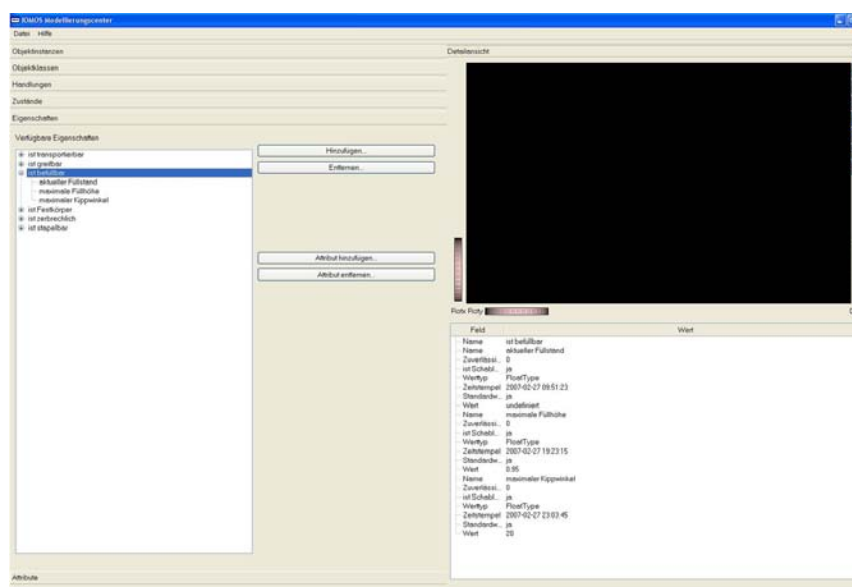
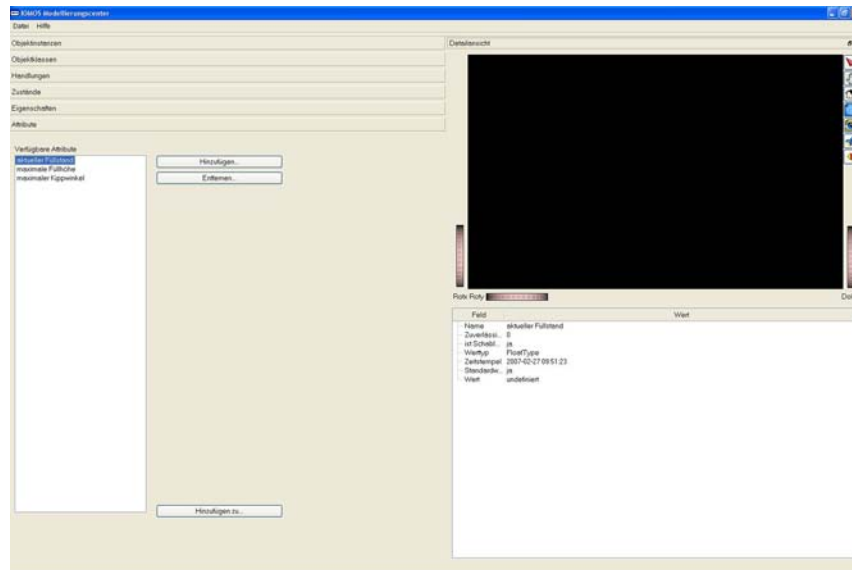


Abbildung 7.5: Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (1). *Oben*: Erstellen von Attributschablonen, *Mitte*: Erstellen von Eigenschaftsschablonen, und *unten*: Benutzerin beim Anlegen der Objektklasse auf dem berührungssensitiven Bildschirm

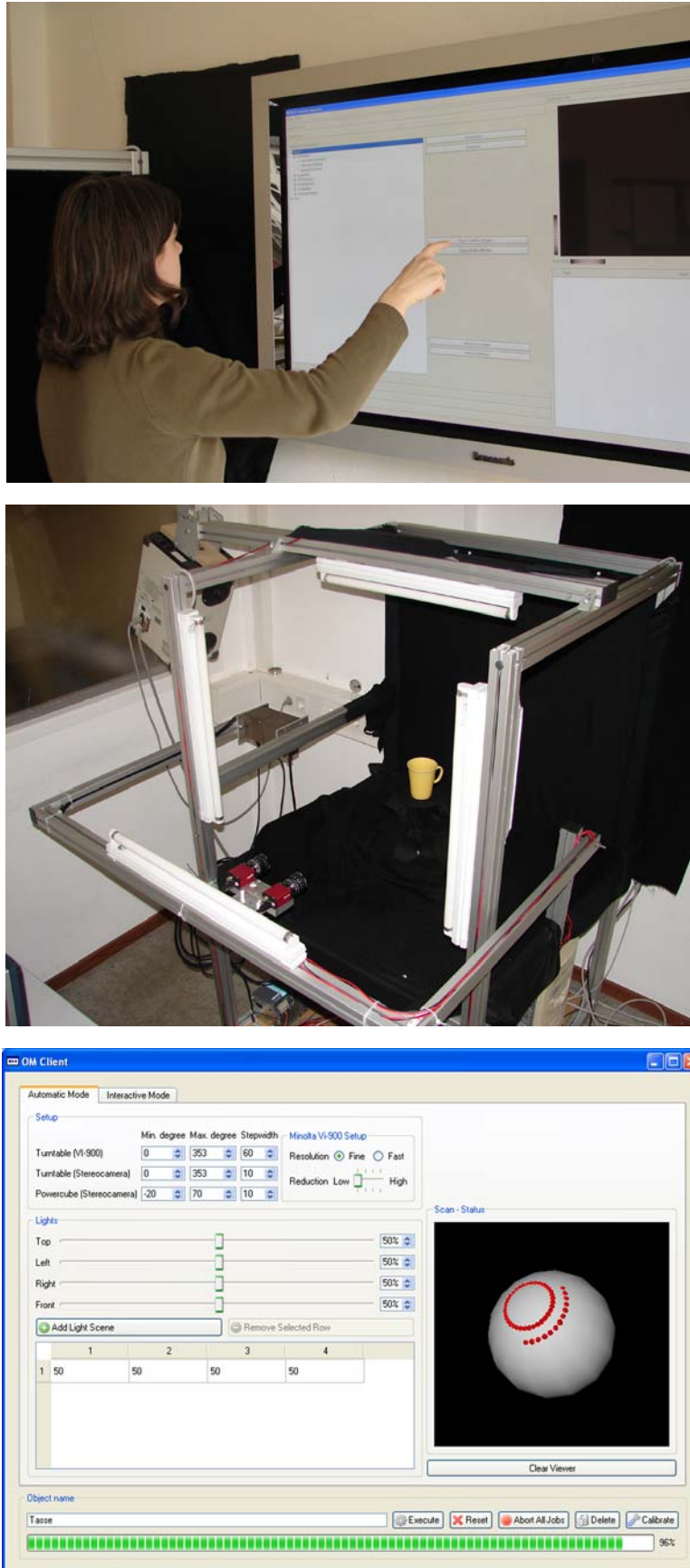


Abbildung 7.6: Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (2). *Ober:* Hinzufügen von Eigenschaften zur Objektklasse, *Mitte:* Platzieren des Objektes in der Modellierungsumgebung und *unten:* Auswählen der Parameter zur Erfassung der dreidimensionalen Objektgeometrie und der Objektansichten und -silhouetten



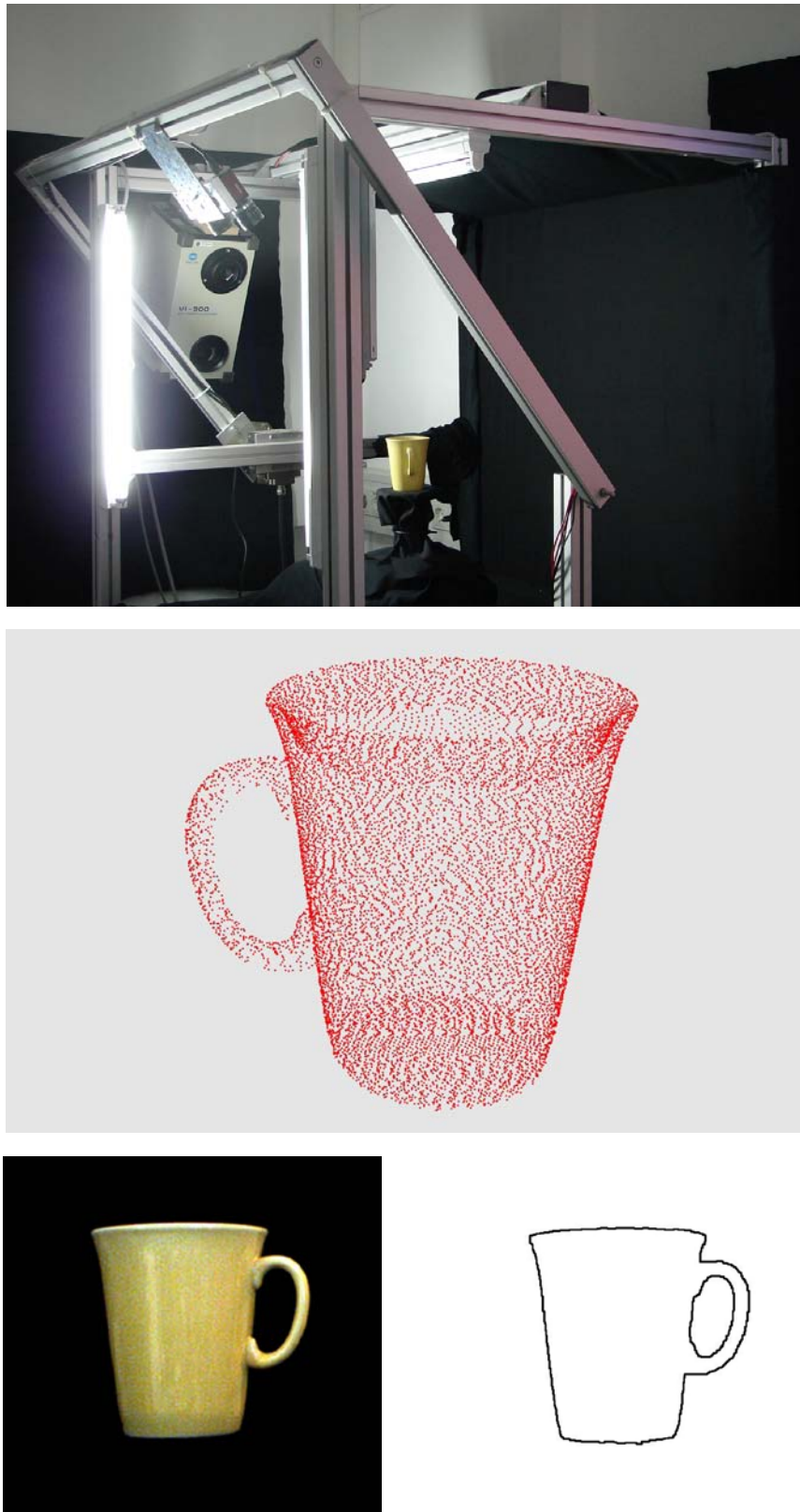


Abbildung 7.7: Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (3).  
*Oben:* Einsatz von Drehbühgel, Drehteller und Beleuchtungsvariationen zur Objektmodellierung, *Mitte:* Resultierende Punktwolke und *unten:* Beispiel für erfasste Objektansicht und -silhouette

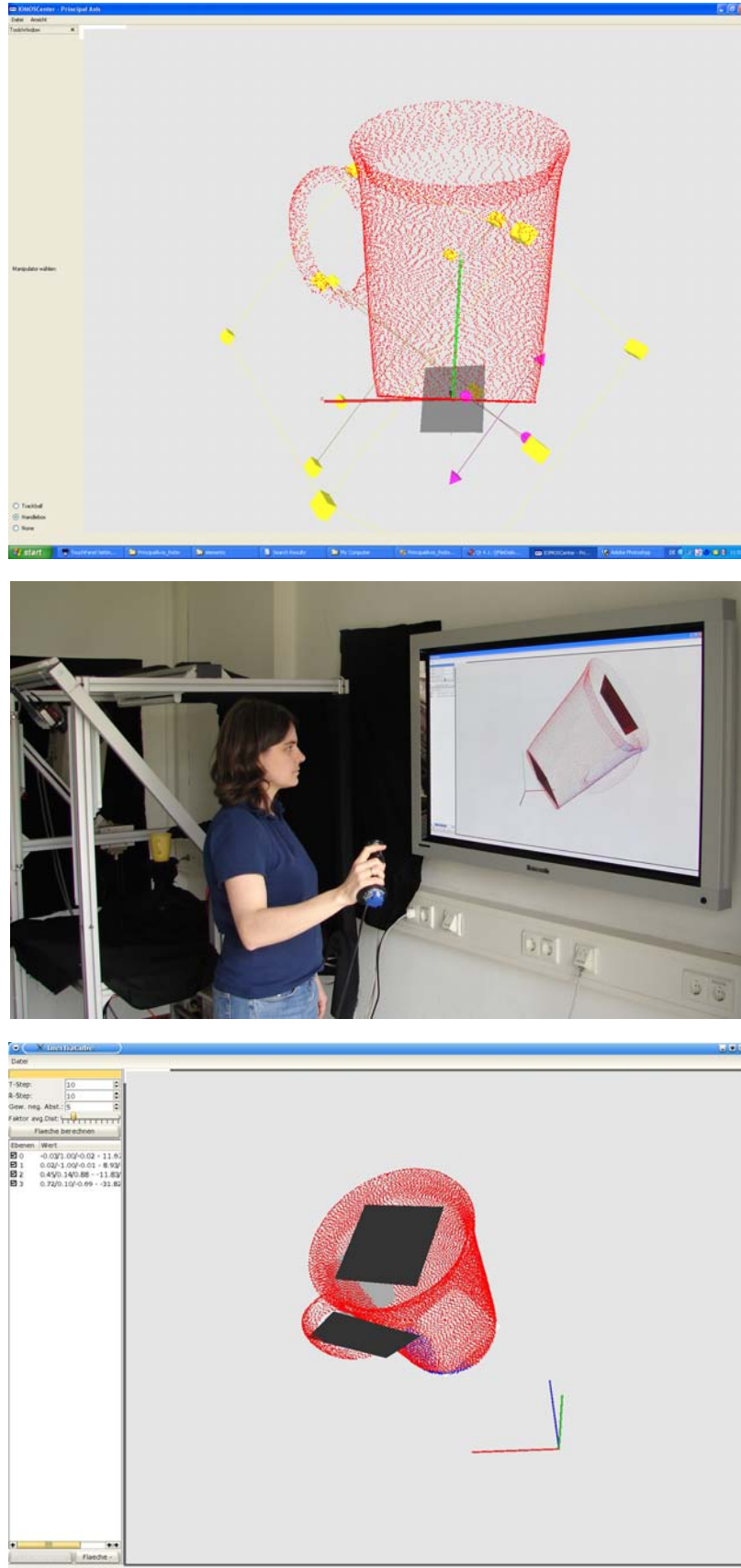


Abbildung 7.8: Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (4). *Oben:* Interaktive Modellierung der Hauptachsen, *Mitte:* Modellierung der stabilen Flächen durch eine Benutzerin und *unten:* Resultierende stabile Flächen

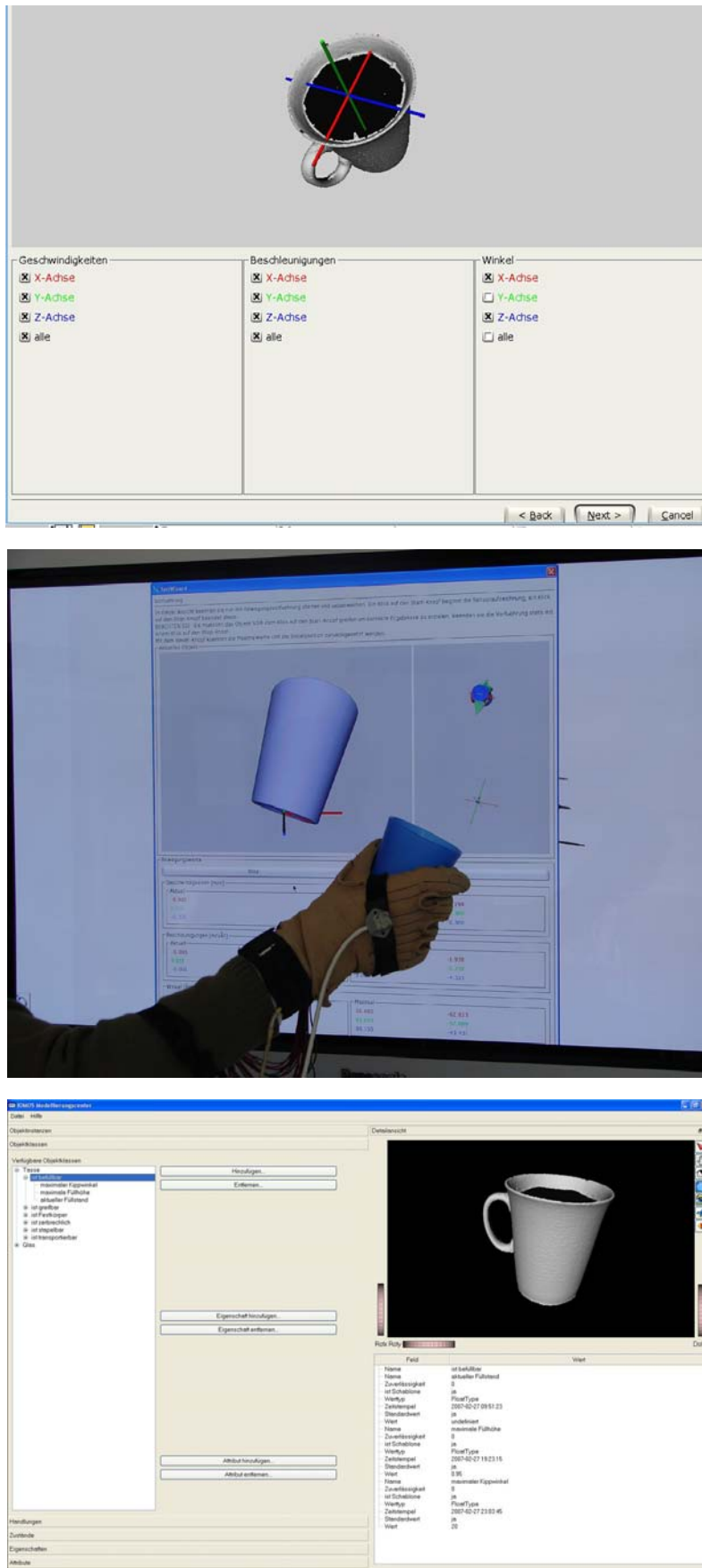


Abbildung 7.9: Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (5). *Oben:* Auswahl der zu modellierenden Manipulationseinschränkungen, *Mitte:* Modellierung dieser Attribute durch eine Benutzerin und *unten:* Anzeige des Objektmodells nach Abschluss der Modellierung

## 7.5 Bewertung der Ergebnisse

Zusammenfassend lässt sich aus den obigen Evaluierungen folgern, dass sowohl die hier entwickelte semantische Objektrepräsentation als auch das interaktive Objektmodellierungssystem IOMOS den an sie gestellten Erwartungen gerecht werden und gute Ergebnisse zeigen. Zudem gelang es, die zu Beginn der Arbeit aufgestellten Thesen durch die hier erzielten Ergebnisse zu untermauern. Die semantische Objektrepräsentation eignet sich nicht nur für Anwendungen im Rahmen von IOMOS, sondern auch als Grundlage für andere Objektmodellierungssysteme, bei denen die Repräsentation semantischer Information erforderlich ist oder die für menschenzentrierte Umgebungen verwendet werden sollen. Mit IOMOS gelang es nicht nur, ein vollständiges System zur interaktiven Objektmodellierung zu entwerfen und aufzusetzen, es konnte auch die hierfür gut geeignete semantische Objektrepräsentation integriert werden. Zudem wurden neue, intuitive Interaktionsmodi entwickelt und im Rahmen von IOMOS mit vielversprechenden Ergebnissen eingesetzt und evaluiert. Das System erlaubt eine äußerst flexible Objektmodellierung für unterschiedlichste Kontexte, und die hierin erstellten Objektmodelle werden im Rahmen unterschiedlicher Projekte im Bereich Servicerobotik, insbesondere für Anwendungen von Servicerobotern im Haushalt, eingesetzt. Die sich aus der Evaluation und der praktischen Anwendung des Systems ergebenden weiterführenden Verbesserungs- und Erweiterungsmöglichkeiten werden in Abschnitt 8.3 noch einmal zusammenfassend dargestellt. Insgesamt aber lassen die hier erzielten Ergebnisse erfolgreiche Einsatzmöglichkeiten für die interaktive Objektmodellierung nicht nur im Bereich der Servicerobotik erhoffen.

## 7.6 Zusammenfassung

Im Hinblick auf die in den vorhergehenden Kapiteln aufgestellten Kriterien und aufgezeigten Anforderungen an die Arbeit wurden in diesem Kapitel sowohl die semantische Objektrepräsentation als auch das interaktive Objektmodellierungssystem IOMOS untersucht und bewertet. Hierbei ergab sich, dass die gestellten Anforderungen im Rahmen der Arbeit erfüllt werden konnten und es zudem gelang, Evidenz für die zu Beginn der Arbeit aufgestellten Thesen zu schaffen. Besonders detailliert wurden zwei intuitive, teilweise neuartige, Interaktionsmodi im Rahmen ihrer Verwendung zur interaktiven Objektmodellierung untersucht. Auch hierbei ergaben sich hervorragende Ergebnisse und sehr ermutigende Rückmeldungen durch die Versuchspersonen. In einer abschließenden Bewertung konnte festgehalten werden, dass sowohl die semantische Objektrepräsentation als auch das interaktive Modellierungssystem IOMOS die gestellten Anforderungen voll erfüllen.

# Kapitel 8

## Abschließende Betrachtungen

### 8.1 Zusammenfassung

Bei der Modellierung von menschenzentrierten Umgebungen beispielsweise im Bereich Servicerobotik stellen sich besondere Probleme, die mit den Anforderungen, wie sie z. B. an die Modelle eines Industrieroboters gestellt werden, nicht vergleichbar sind. Die vorliegende Arbeit geht davon aus, dass die Objekte in einer solchen Umgebung nur dann sinnvoll modelliert werden können, wenn die Modelle auch semantische Objektinformationen enthalten. Ein zweiter Grundgedanke der Arbeit ist es, dass semantische Objektmodelle nur in Interaktion mit einem menschlichen Benutzer erstellt werden können. Ziel der Arbeit war es somit zum einen, eine adäquate semantische Objektrepräsentation zu entwerfen, zum anderen ein interaktives Modellierungssystem zu konzipieren und aufzusetzen, in dem Mensch und Maschine gemeinsam Objekte auch semantisch modellieren können. Hierbei ergeben sich dann auch spezifische Anforderungen. So soll der Benutzer unter anderem nicht mit unnötigen Routineaufgaben überfrachtet werden, sondern sein Wissen nur dort einbringen müssen, wo dies auch sinnvoll oder notwendig ist.

Um diese Ziele zu erreichen, wurde zunächst der Stand der Forschung in den Bereichen semantische und interaktive Objektmodellierung sowie beim Lernen von Objektmodellen betrachtet. Hierbei konnte festgestellt werden, dass keines der bereits existierenden Systeme die sich hier stellenden Anforderungen erfüllt. Im Gegenteil werden häufig nur stark reduzierte, anwendungsspezifische Objektmodelle eingesetzt, beispielsweise rein geometrische Objektmodelle. Bei der Untersuchung der gängigen Arbeiten stellte sich heraus, dass kein umfassender Ansatz existiert, der alle notwendigen Bereiche im gleichen Maße berücksichtigt, insbesondere keine Arbeit zur interaktiven semantischen Objektmodellierung, die vergleichbar umfassend ist wie der hier verfolgte Ansatz.

Ausgehend von den grundlegenden Überlegungen in Kapitel 1 wurde danach eine semantische Objektrepräsentation entworfen, die eine flexible und für den Menschen intuitiv verständliche Objektmodellierung unter Einbeziehung funktionaler Aspekte erlaubt. Ein Kernbestandteil dieser Repräsentation sind die so genannten *Eigenschaften* von Objekten, die dem menschlichen Verständnis von Objekten und ihren Nutzungsweisen entsprechen. Objekte im Haushalt können hierdurch mit Eigenschaften wie **ist\_befüllbar**, **ist\_stapelbar** oder **ist\_transportierbar** belegt werden. Aus den Eigenschaften eines Objektes ergeben sich

unmittelbar die ihm zugeordneten Attribute, Zustände und Handlungen. *Attribute* sind dabei die kleinste Modellierungseinheit, in der die gängigen Objektbeschreibungen wie 3D-Geometrie, Bounding Box oder Ansichten modelliert werden. *Zustände* sind kontextabhängige natürlichsprachliche Beschreibungen von Attributwertbereichen. So kann der Zustand **heiß** eine Objekttemperatur von über 50°C bezeichnen. *Handlungen* schließlich bilden die Verbindung zwischen Objektmodellen und Handlungsmodellierung, indem den Objekten potentiell mit ihnen ausführbare Handlungen zugewiesen sowie deren Vor- und Nachbedingungen auf Attributbasis definiert werden.

Diese semantische Objektrepräsentation bildet die Grundlage für das im Rahmen dieser Arbeit entwickelte **Interaktive ObjektMOdellierungsSystem** IOMOS. Nach Klärung der Anforderungen an ein solches System, wie sie sich aus den vorhergehenden Überlegungen ergeben, wurde zunächst die IOMOS zugrunde liegende Struktur erläutert, bevor die Besonderheiten der Datenhaltung in IOMOS und die Implementierung der semantischen Objektrepräsentation im Detail vorgestellt wurden.

Danach wurde die sensorgestützte interaktive Objektmodellierung in IOMOS als eine weitere wesentliche Errungenschaft der Arbeit ausführlich vorgestellt. Hierzu wurden zunächst einige wichtige allgemeine Grundlagen der Mensch-Maschine-Interaktion dargelegt und hieraus Anforderungen an die interaktiven Komponenten in IOMOS abgeleitet. Eine Beschreibung des Aufbaus der in dieser Arbeit entworfenen und aufgesetzten Objektmodellierungsumgebung und ihrer technischen Bestandteile bildete dann die Grundlage sowohl für die Vorstellung der vorhandenen Interaktionsmodi und ihrer potentiellen Anwendungen als auch für die interaktive Objektmodellierung auf der Basis von zu bestimmenden Attributwerten. Besonders hervorzuheben ist hier die Modellierung von stabilen Objektflächen mittels des Inertial-Joysticks, eines eigens entwickelten Interaktionsgerätes, sowie die intuitive, interaktive Modellierung von Manipulationseinschränkungen unter Verwendung eines magnetfeldbasierten Positionssensors. Eine kurze Darstellung zur Systemintegration schloss diesen Teil der Arbeit ab.

Als Anwendungsbeispiel für die interaktive semantische Objektmodellierung wurde die Domäne *Küche* betrachtet. Nach einer kurzen einleitenden Darstellung zu den Grundfragen des Ontologieentwurfs und detaillierten Überlegungen zu den Besonderheiten dieser Domäne wurde dann Ausschnitte aus einer im Rahmen der vorliegenden Arbeit entwickelten Taxonomie von Begrifflichkeiten für Serviceroboter in der Küche vorgestellt. Dies ermöglicht zum einen einen Einblick in die Anwendung der hier entwickelten Objektrepräsentation, zum anderen dient diese Domänenspezifikation aber auch als Grundlage für die Systemevaluation im darauf folgenden Kapitel.

Zur Evaluation und Bewertung der Ergebnisse der Arbeit wurden die semantische Objektrepräsentation und das interaktive Objektmodellierungssystem IOMOS separat untersucht. Für die semantische Objektrepräsentation ergab sich dabei, dass die aus den Zielen der Arbeit heraus entwickelten Anforderungen an eine solche Repräsentation erfüllt werden konnten. Ebenso erfüllte auch IOMOS die im Verlauf der Arbeit herausgearbeiteten Anforderungskriterien. Zusätzlich wurden die beiden oben genannten, besonders interessanten Interaktionsmodi einer näheren Betrachtung unterzogen: die interaktive Modellierung stabiler Flächen mit dem Inertial-Joystick und die interaktive Modellierung von Manipulationseinschränkungen mit dem magnetfeldbasierten Positionssensor. In der Evaluation mit Versuchspersonen

ergaben sich dabei sehr gute Ergebnisse für die Intuitivität und Angemessenheit dieser interaktiven Modellierungsformen sowie sehr gute Modellierungsergebnisse durch die Versuchspersonen. Auch die Nutzungsmöglichkeiten und die Verwendung der im Rahmen der Arbeit erstellten semantischen Objektmodelle wurden hier dargestellt und diskutiert. Abschließend wurde hier ein Anwendungsbeispiel für eine Modellierungssequenz in IOMOS präsentiert, die Modellierung der Objektklasse TASSE. Eine kritische Bewertung der erzielten Ergebnisse rundete die Evaluation ab.

## 8.2 Diskussion

Wie im vorhergehenden Abschnitt bereits erläutert, ergab sich bei der Evaluation der einzelnen Bestandteile dieser Arbeit ein sehr gutes Resultat. Insbesondere konnten die semantische Objektrepräsentation und das interaktive Objektmodellierungssystem IOMOS die gestellten Anforderungen erfüllen. Hierbei bewährten sich auch die untersuchten Interaktionsmodi, die teilweise für diese Arbeit neu konzipiert wurden.

Es konnten aber bei der kritischen Betrachtung der Ergebnisse der Arbeit nicht nur die getroffenen Entwurfs- und Implementierungsentscheidungen bestätigt werden, sondern auch Evidenz für die zu Beginn der Arbeit aufgestellten Thesen gefunden werden. Festhalten lässt sich hierbei:

- Die Flexibilität der semantischen Objektrepräsentation und ihre intuitive Zugänglichkeit für den menschlichen Benutzer machen einen sinnvollen Einsatz in menschenzentrierten, dynamischen Umgebungen wie Haushalten erst möglich. Es wurde hierbei bewusst eine umfassende Modellierung gewählt, die sich nicht nur auf einzelne Objektparameter beispielsweise für die Objekterkennung beschränkt, sondern die ein Objekt möglichst als Ganzes mit seiner semantischen Struktur und seinen funktionalen Aspekten abbildet. Das resultierende Objektmodell kann dabei immer nur in Abhängigkeit vom gegebenen Kontext und den gestellten Anforderungen erstellt und betrachtet werden.
- Aufgrund der angesprochenen Dynamik der Einsatzumgebungen müssen die Objektmodelle nicht nur flexibel die jeweils erforderlichen Objektbestandteile abbilden können, sondern es müssen gegebenenfalls auch neue Informationen integriert oder bereits bekannte Informationen an neues Wissen angepasst werden können. Dies ist in der hier entwickelten semantischen Objektrepräsentation ebenfalls gegeben.
- Eine Modellierung von semantischen Objektinformationen kann nur in Interaktion mit dem Menschen erfolgen, da viele dieser Informationen für ein maschinelles System nur schwer oder überhaupt nicht zugänglich sind. Hierzu gehören beispielsweise benutzerspezifisches Wissen, komplexes Handlungswissen oder kontextabhängiges Wissen.
- Die Interaktion mit dem Benutzer wird wesentlich erleichtert, wenn die Objektrepräsentation an dessen Erfordernisse und Denkstrukturen angepasst wird. Durch die Abbildung von Objekteigenschaften in den Objektmodellen und deren zentrale Stellung wird das Verständnis des Benutzers für das jeweilige Objektmodell und dessen Grundstruktur wesentlich erleichtert.

- Die Integration funktionaler Objektbestandteile erweist sich für den Einsatz der Objektmodelle in realen Kontexten als unabdingbar, um z. B. eine Handlungsplanung durchzuführen. Dabei muss Objektinformation zwingend integriert werden, gleichzeitig ist aber die Verbindung zwischen Objekten und Handlungen in vielen gängigen Objektmodellen nur schwer zu bewerkstelligen. Die Verknüpfung von potentiell ausführbaren Handlungen mit Eigenschaften erleichtert die Anbindung von Handlungsmodellen an Objektmodelle, und die Vor- und Nachbedingungen der Handlungen im Objektmodell können direkt bei der Handlungsplanung eingesetzt und über die Objektmodelle sensorisch abgeprüft werden.
- Durch den Einsatz von innovativen, intuitiven Interaktionsmodi können auch nicht speziell vorgebildete Benutzer Objektinformationen erfolgreich an ein interaktives Objektmodellierungssystem übermitteln. Hierbei können durch geschickte Ausnutzung der sensorisch ermittelten Objektinformationen Routinetätigkeiten weitestgehend vermieden werden, indem beispielsweise die bereits erfasste dreidimensionale Objektgeometrie als Grundlage für die Ermittlung weiterer Objektattributwerte eingesetzt wird. Dem Benutzer verbleibt hierdurch die Eingabe der semantischen Grundstruktur von Objekten, von maschinell nicht oder nur schwer zu ermittelnder Information sowie die vollständige Kontrolle über das System mit der Möglichkeit, Informationen jederzeit zu löschen, zu korrigieren oder zu ergänzen.

Das solcherart ermittelte Objektwissen kann dann nutzbringend in praktischen Anwendungen wie einem Serviceroboter im Haushalt integriert werden. In der vorliegenden Arbeit gelang es daher, die Bedeutung sowohl einer semantischen Objektmodellierung als auch einer gelungenen Mensch-Maschine-Interaktion zum Lernen von Objektinformationen für den Einsatz in realen, entsprechend komplexen und dynamischen, menschenzentrierten Umgebungen deutlich zu machen.

### 8.3 Ausblick

Bei der Arbeit mit IOMOS und den daraus resultierenden Objektmodellen ergaben sich einige wünschenswerte Verbesserungen und Ergänzungen, die im Folgenden kurz dargestellt werden:

- Bisher können die Objektgeometrie oder Objektansichten nur durch weitere Aufnahmen ergänzt werden, solange die Position des Objektes in der Modellierungsumgebung noch nicht verändert wurde. Insbesondere muss, wenn das Objekt nachträglich erneut auf dem Drehteller platziert wurde, um einzelne Aufnahmen zu ergänzen, jeweils die komplette Aufnahmeserie neu gestartet werden, da eine Registrierung der neuen Objektposition und -rotation gegenüber der alten nicht möglich ist. Dies ist keine größere Einschränkung, da das System die spezifizierten Aufnahmen automatisch abarbeitet, kostet aber doch gelegentlich unnötig Zeit. Hier wäre es wünschenswert, mit Methoden zur Registrierung von Punktwolken bzw. von Texturen veränderte Objektpositionen



und -rotationen automatisch zu erfassen und die Winkel der neuen Aufnahmen entsprechend zu korrigieren. Derzeit wird bereits an der Implementierung eines Algorithmus zur Punkteinpassung (engl.: Iterative Closest Point Algorithm, ICP) gearbeitet, der diese Problematik teilweise abdecken wird.

- Bisher werden Objektattribute nur objektspezifisch abgelegt. In realen Anwendungen ergeben sich allerdings situations- bzw. kontextabhängige oder zustandsspezifische Attributwerte. Hierzu gehören unter anderem Manipulationseinschränkungen wie der maximale Kippwinkel eines Behälters, der vom aktuellen Füllstand des Objektes abhängt, oder die bereits angesprochene Auswahl valider stabiler Objektflächen, die stark mit dem Kontext variieren kann. Ein gefülltes Glas z. B. sollte in der Regel nicht kopfüber abgestellt werden, beim Einräumen in die Spülmaschine ist dies aber die bevorzugt auszuwählende stabile Fläche. Hier ergibt sich Raum für Erweiterungen des Systems und der Objektmodelle, durch deren flexibel angelegte Struktur solche ergänzenden Informationen aber zukünftig leicht eingebunden werden können.
- Bei Bedarf können weitere Interaktionsmodi in die Modellierungsumgebung integriert werden, so z. B. die Nutzung von Greifinformationen aus dem Datenhandschuh, für dessen Integration bereits die Grundlagen gelegt wurden. Weiterhin bieten die bereits vorhandenen Interaktionsmodi Potential für andere Modellierungsanwendungen, so sollen zukünftig anhand der mit dem Inertial-Joystick bestätigten Position wie oben schon erläutert auch der Drehteller und der Drehbügel in die gewünschte Stellung für die weiteren Objektaufnahmen gebracht werden.
- Gleiches gilt für die Einbindung ergänzender Sensorsysteme, die aufgrund der erweiterbaren Struktur von IOMOS jederzeit möglich ist. Denkbar ist hier beispielsweise eine digitale, an das Modellierungssystem angebundene Waage zur automatischen Erfassung des Objektgewichts.
- Bereits in Betrieb genommene Interaktionsmodi können ebenfalls teilweise noch verbessert werden, indem sie unter Ausnutzung der bisher gemachten Erfahrungen mit der interaktiven Modellierungsumgebung noch intuitiver und benutzerfreundlicher gestaltet werden. Dies betrifft vor allem den eigens entwickelten Inertial-Joystick, der von Versuchspersonen zwar als intuitiv und für die gegebene Aufgabe sehr geeignet gelobt wurde, aber teilweise nicht mit der nötigen Präzision gedreht werden konnte. Insbesondere das Drücken der Joystick-Knöpfe zum Bestätigen einer Position führt regelmäßig zu Ungenauigkeiten. Hier ist der Aufbau eines verbesserten Gerätes geplant, in das die bisher gemachten Erfahrungen einfließen. Zusätzlich ist an dieser Stelle auch eine Kombination mit der Spracheingabe denkbar, um das Drücken eines Knopfes zur Bestätigung gänzlich zu vermeiden.

Insgesamt ergaben sich im Rahmen dieser Arbeit sehr vielversprechende Ergebnisse, die nicht nur die konzeptuelle Bedeutung einer interaktiven semantischen Objektmodellierung belegen, sondern eine solche auch für den praktischen Einsatz empfehlen. Aufgrund dieser Erkenntnisse bleibt zu hoffen, dass zukünftig nicht nur die aus einer solchen Modellierung resultierenden Objektmodelle auf Servicerobotern in Haushalten und vielfältigen anderen menschenzentrierten Umgebungen zum Einsatz kommen, sondern dass auch der Vorgang

der interaktiven semantischen Modellierung selbst auf längere Sicht auf solchen Robotersystemen integriert werden kann. Sowohl die entsprechenden Objektmodelle als auch der Modellierungsvorgang an sich werden der Benutzer-Akzeptanz und den Einsatzmöglichkeiten eines Serviceroboters sehr zugute kommen.

# Anhang A

## Technische Daten der verwendeten Hardware

### A.1 Farbkameras *Marlin 145C2* von Allied Vision Technologies



Abbildung A.1: Allied Vision Technologies *Marlin 145C2*, aus: Allied Vision Technologies (2007)

Technische Merkmale	<i>Marlin 145C2</i>
Aufnahmeeinheit	Progressive Scan CCD
max. Bildgröße	1392 × 1040 Pixel
Schnittstelle	IEEE 1394
max. Framerate	10 Hz
Abmessungen (L × B × H)	72 mm × 44 mm × 29 mm
Gewicht	< 120 g (ohne Linse)

Tabelle A.1: Technische Daten der Allied Vision Technologies *Marlin 145C2* nach Allied Vision Technologies (2007)

## A.2 Laserscanner Minolta *VI-900*



Abbildung A.2: Minolta *VI-900*, aus: Konica Minolta (2007)

Technische Merkmale	<i>VI-900</i>
Messmethode	Lasertriangulation
Objektive	tele f = 25,5 mm; mittel f = 14,5 mm; weit f = 8 mm
Entfernung zum Objekt	60 cm bis 120 cm
Objektgröße	max. 130 cm × 110 cm
geometrische Genauigkeit	x = 0,17 mm, y = 0,17 mm, z = 0,047 mm (bei 60 cm Entfernung)
Auflösung	640 × 480
Anzahl der Messpunkte	76800 (schnell) bzw. 307200 (fein)
Messdauer	0,3 s (schnell) bzw. 2,5 s (fein)
Schnittstelle	SCSI II
Abmessungen	213 mm × 413 mm × 271 mm
Gewicht	ca. 11 kg

Tabelle A.2: Technische Daten des Minolta *VI-900* nach Konica Minolta (2007)

## A.3 Rotationseinheit Isel *RF-1*



Abbildung A.3: Isel *RF-1*, aus: Isel Automation (2007)

Technische Merkmale	<i>RF-1</i>
Antrieb	Servomotor
Bremse	Magnetbremse
Min. Schrittweite (Positioniergenauigkeit)	2 arcmin
Schnittstelle	seriell
Abmessungen	27,5 cm × 40 cm × 20 cm
Gewicht	14,6 kg

Tabelle A.3: Technische Daten des Isel *RF-1* nach Isel Automation (2007)

## A.4 Rotationsgelenk Amtec *PowerCube PR 090*

Abbildung A.4: Amtec *PowerCube PR 090*, aus: Amtec Robotics (2007)

Technische Merkmale	<i>PowerCube PR 090</i>
Abtriebs-Nennmoment	72 Nm
Max. Abtriebsmoment	206 Nm
Max. Winkelgeschwindigkeit	149 °/s
Auflösung	4 Winkelsec/Inc
Lageerfassung	894 Inc/°
Schnittstelle	CAN-Bus
Abmessungen	9 cm × 9 cm × 18 cm
Gewicht	3,2 kg

Tabelle A.4: Technische Daten des Amtec *PowerCube PR 090* nach Amtec Robotics (2007)

## A.5 Inertialsensor Intersense *InertiaCube2*



Abbildung A.5: Intersense *InertiaCube2*, aus: InterSense (2007)

Technische Merkmale	<i>InertiaCube2</i>
Freiheitsgrade	3
Winkelbereich	360°
max. Winkelgeschwindigkeit	1200°/s
effektive Genauigkeit	1°
effektive Winkelauflösung	0,01°
Wiederholungsrate	180 Hz
Schnittstelle	RS-232
Abmessungen	28,89 mm × 24,38 mm × 33,91 mm
Gewicht	25 g

Tabelle A.5: Technische Daten des Intersense *InertiaCube2* nach InterSense (2007)

## A.6 Plasmadisplay Panasonic *TH-50PF9EK* mit berührungssensitivem Aufsatz *TY-TP50P8-S*

Technische Merkmale	<i>TH-50PF9EK</i>
Bildschirmgröße (B × H)	1106 mm × 622 mm
Anzahl Pixel	1920 x 1080
Abmessungen (B × H × T)	1210 mm × 724 mm × 95 mm
Gewicht	ca. 42 kg

Tabelle A.6: Technische Daten des Panasonic *TH-50PF9EK* nach Panasonic (2007)

Technische Merkmale	<i>TY-TP50P8-S</i>
Typ	Optisches Touch-Panel
Erkennungssystem	Infrarot-Obstruktions-Erkennungssystem
Effektiver Erkennungsbereich (B × H)	1106 mm × 622 mm
Auflösung (B × H)	2209 × 1241 Pixel
Minimale Stiftgröße (B × H)	6 mm × 6 mm
Schnittstelle	USB
Abmessungen (B × H × T)	1257 mm × 773 mm × 69 mm
Gewicht	ca. 5,8 kg (ohne Halterungen)

Tabelle A.7: Technische Daten des Panasonic *TY-TP50P8-S* nach Panasonic (2007)

## A.7 Datenhandschuh Immersion *CyberGlove*

Abbildung A.6: Immersion *CyberGlove*, aus: Immersion (2004)

Technische Merkmale	<i>CyberGlove</i>
Freiheitsgrade	22
Auflösung	0,5°
Wiederholgenauigkeit	1°
Datenrate	150 Hz (ungefiltert)
max. Übertragungsrate	115 kBaud
Schnittstelle	RS-232
Kabellänge	3,0479 m (10 ft)

Tabelle A.8: Technische Daten des Immersion *CyberGlove* nach Immersion (2004)

## A.8 Magnetfeldbasierter Positionssensor Polhemus *Fastrack*



Abbildung A.7: Polhemus *Fastrack*, aus: Polhemus (2007)

Technische Merkmale	<i>Fastrack</i>
Anzahl Freiheitsgrade	6
Wiederholungsrate	120 Hz
Effektive Positionsgenauigkeit	0,03 in
Effektive Orientierungsgenauigkeit	0,15°
Positionsaufösung	0,0002 in pro in Transmitter-Empfänger-Entfernung
Orientierungsaufösung	0,025°
Verzögerung	4 ms
max. Entfernung vom Stationärteil	1,52 m
Schnittstelle	RS-232, USB

Tabelle A.9: Technische Daten des Polhemus *Fastrack* nach Polhemus (2007)



# Abbildungsverzeichnis

2.1	Beispielhafte Objekttaxonomie für Manipulationsaufgaben (Rogalla (2003), S. 78) . . . . .	11
2.2	Beispielhafte Darstellung von Objekten und deren möglichen Attributen (Ehrenmann (2003), S. 59) . . . . .	12
2.3	Repräsentation von Objektfunktionalitäten bei Rivlin et al. (Rivlin et al. (1995), S. 5) . . . . .	14
2.4	Repräsentation eines Objektes und seiner Funktionalitäten bei Bogoni und Bajcsy (Bogoni und Bajcsy (1995), S. 30) . . . . .	18
2.5	Darstellung der Wissensbasis bei Bley et al. (Bley et al. (2006), S. 414) . . . . .	22
2.6	Interaktion zwischen Mensch und Roboter BIRON (aus: Haasch et al. (2004), S. 1) . . . . .	30
2.7	Silhouetten statischer Referenzgesten zur Gestenerkennung bei Ehrenmann (Ehrenmann (2003), S. 142) . . . . .	32
2.8	Markerbasierte Bewegungserfassung mit Kameras. <i>Links</i> Motion Captor-System von Meta Motion (aus: Meta Motion (2006b)). <i>Rechts</i> : das Trackingsystem von VICON in der Anwendung (aus: Stein et al. (2006), S. 14) . . . . .	33
2.9	Magnetfeldbasierte Positionssensoren, <i>links</i> : Flock of Birds von Ascension (aus: Ascension (2006)), <i>rechts</i> : von Polhemus (aus: Polhemus (2006)) . . . . .	33
2.10	Datenhandschuhe, <i>links</i> : CyberGlove II von Immersion (aus: Immersion (2006)), <i>Mitte</i> : Datenhandschuh von Fifth Dimension Technologies (aus: Fifth Dimension Technologies (2006)) und <i>rechts</i> : Gypsy-System von Meta Motion als Beispiel für ein Exoskelett (aus: Meta Motion (2006a)) . . . . .	34
2.11	<i>Links</i> : Taxonomie statischer Griffe aus Cutkosky (1989), S. 273. <i>Rechts</i> : Taxonomie dynamischer Griffe aus Zöllner (2005), S. 80 . . . . .	35
2.12	<i>Links</i> : Serviceroboter ARMAR III mit einer künstlichen Haut, aus Göger et al. (2006), S. 6. <i>Rechts</i> : taktile Sensoren in einem Datenhandschuh aus Zöllner et al. (2001), S. 69 . . . . .	36
2.13	<i>Links</i> : Zeigestab mit Markern aus Giesler (2005), S. 105. <i>Rechts</i> : Ergebnis der interaktiven Objektmodellierung mittels Erweiterter Realität in Giesler (2005), S. 147 . . . . .	37

2.14	Sprachliche und gestische Interaktion zwischen Mensch und Roboter bei Dillmann et al. (aus: Dillmann et al. (2002), S. 1) . . . . .	38
2.15	Schematische Darstellung der multimodalen Benutzerschnittstelle bei Perzanowski et al. (aus: Perzanowski et al. (2001), S. 18) . . . . .	39
3.1	Ableitung von Modellen konkreter Objekte aus der Objektrepräsentation . . . . .	46
3.2	Grundstruktur der semantischen Objektrepräsentation . . . . .	50
3.3	Schematischer Aufbau eines Objektattributes . . . . .	52
3.4	Schematische Darstellung einer Eigenschaft mit ihren Bestandteilen . . . . .	56
3.5	Beispielhafte Darstellung von Vererbung zwischen Objektklassen . . . . .	57
3.6	Komponenten eines Objektmodells im Detail . . . . .	58
3.7	Vererbung zwischen Objektklassen und -instanzen und Ausprägung von Attributwerten konkreter Objekte . . . . .	59
4.1	Ablaufschema der Erstellung eines Objektmodells im interaktiven Objektmodellierungssystem IOMOS . . . . .	65
4.2	Details der Prozesskette zur Erstellung eines Objektmodells . . . . .	66
4.3	Architektur des interaktiven Objektmodellierungssystems IOMOS. Die dunkel hinterlegten Komponenten wurden in dieser Arbeit entwickelt und implementiert. . . . .	67
4.4	Schematische Darstellung der Datenablage in der Objektmodelldatenbank und des externen Zugriffs mittels Corba-Schnittstelle . . . . .	70
4.5	Beispielhafter Ausschnitt aus einem in Form eines XML-Datensatzes abgelegten Objektmodell . . . . .	72
5.1	Gesamtansicht der interaktiven Objektmodellierungsumgebung . . . . .	81
5.2	Abmessungen der Modellierungsumgebung . . . . .	82
5.3	Schematische Darstellung der Erfassungsumgebung mit integrierter Sensorik von vorne, nach: Eisenberg (2006) . . . . .	82
5.4	Schematische Darstellung der Erfassungsumgebung mit integrierter Sensorik von oben, aus: Eisenberg (2006) . . . . .	83
5.5	Schwenken des Stereokamerasystems zur Objektaufnahme aus unterschiedlichen Sichten . . . . .	83
5.6	Unterschiedliche Kameraperspektiven mit jeweils korrespondierenden Objektansichten . . . . .	84
5.7	Schematische Darstellung des Drehtellers zur Rundumerfassung von Objekten . . . . .	85

---

5.8	Unterschiedliche Beleuchtungssituationen in einer Objektmodellierungssequenz (1). <i>Oben</i> : dunkelste Einstellung, <i>Mitte</i> : hellste Einstellung, <i>unten</i> : Lichteinfall von links . . . . .	87
5.9	Unterschiedliche Beleuchtungssituationen in einer Objektmodellierungssequenz (2). <i>Oben</i> : Lichteinfall von links und vorne, <i>Mitte</i> : von oben, <i>unten</i> : von rechts	88
5.10	Bedienung der graphischen Benutzerschnittstelle auf dem berührungssensitiven hochauflösenden Plasmabildschirm . . . . .	90
5.11	Ausschnitt aus dem Hauptfenster der graphischen Benutzerschnittstelle: Modellierung von Objektklassen . . . . .	91
5.12	Joystick mit integriertem Inertialsensor . . . . .	93
5.13	Einsatz des Inertial-Joysticks zur Rotation von dreidimensionalen Objektmodellen . . . . .	93
5.14	Datenhandschuh mit integrierter Kraftsensorik (Zöllner (2005)) und magnetfeldbasiertem Positionssensor . . . . .	94
5.15	Anwendung des magnetfeldbasierten Positionssensors zur intuitiven Objektmodellierung . . . . .	95
5.16	Punktwolke einer Tasse und zugehöriges Dreiecksnetz . . . . .	96
5.17	Graphische Bedienoberfläche zur Einstellung der Aufnahmeparameter von 3D-Geometrie und Objektansichten . . . . .	97
5.18	Kameraaufnahme eines Objektes, daraus extrahierter Objektblob und resultierende Silhouette . . . . .	98
5.19	Potentielle stabile Flächen einer Tasse, aus: Kasper (2007) . . . . .	100
5.20	Unterschiedliche Objektansichten in der graphischen Oberfläche zur interaktiven Modellierung stabiler Flächen . . . . .	103
5.21	Sequenz zur Modellierung der stabilen Flächen einer quaderförmigen Lebensmittelverpackung . . . . .	104
5.22	Hauptachsen einer Tasse, aus: Kasper (2007) . . . . .	105
5.23	Graphische Oberfläche zur interaktiven Bestimmung der Hauptachsen von Objekten . . . . .	106
5.24	Assistent zur interaktiven Modellierung von Manipulationseinschränkungen, Schritt 2: Auswahl der zu bestimmenden Manipulationseinschränkungen in Bezug auf die angezeigten Objektkoordinatenachsen . . . . .	109
5.25	Assistent zur interaktiven Modellierung von Manipulationseinschränkungen, Schritt 3: Vorführung der Bewegungen mit textueller und graphischer Darstellung der ermittelten Ergebnisse in Echtzeit . . . . .	110
6.1	Entwurfsschritte zur Erstellung der Domänenspezifikation . . . . .	115

6.2	Ausschnitt aus der Hierarchie gebräuchlicher Küchenobjekte (1): Besteck . . .	118
6.3	Ausschnitt aus der Hierarchie gebräuchlicher Küchenobjekte (2): oberste Ebene der Hierarchie . . . . .	119
6.4	Ausschnitte aus der Hierarchie gebräuchlicher Küchenobjekte (3): Elektrogeräte . . . . .	119
6.5	Ausschnitt aus der Hierarchie gebräuchlicher Küchenobjekte (4): Funktionale Elemente . . . . .	120
6.6	Ausschnitt aus der Hierarchie gebräuchlicher Küchenobjekte (5): Geschirr . .	121
6.7	Ableitung von wichtigen Objektattributen aus gebräuchlichen Handlungen in der Küche, (aus: Kasper (2007), S. 25) . . . . .	125
6.8	Beispiele für Handlungen in der Domäne <i>Küche</i> . . . . .	127
6.9	Beispiele für Zustände in der Domäne <i>Küche</i> . . . . .	127
6.10	Eigenschaften der Objektklasse TASSE . . . . .	128
6.11	Eigenschaft <b>ist_befüllbar</b> der Objektklasse TASSE . . . . .	129
6.12	<i>füllen</i> , <i>abstellen</i> und <i>stapeln</i> als Beispiele für Handlungen der Objektklasse TASSE . . . . .	130
6.13	<i>voll</i> , <i>halbvoll</i> und <i>leer</i> als Beispiele für Zustände der Objektklasse TASSE	130
7.1	Bezugspunkte der von den Probanden ermittelten stabilen Ebenen, farblich getrennt nach Probanden aufgetragen (aus: Kasper (2007), S. 79) . . . . .	137
7.2	Normalenvektoren der von den Probanden ermittelten stabilen Ebenen, farblich getrennt nach Probanden aufgetragen (aus: Kasper (2007), S. 79) . . . . .	138
7.3	Einbindung der Objektmodelle in die Gesamtarchitektur eines Robotersystems im Sonderforschungsbereich SFB 588 „Humanoide Roboter“ . . . . .	143
7.4	Im Internet zur Verfügung gestellte Objektmodelle mit 3D-Vorschau, Homepage erstellt von A. Kasper . . . . .	145
7.5	Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (1). <i>Oben</i> : Erstellen von Attributschablonen, <i>Mitte</i> : Erstellen von Eigenschaftschablonen, und <i>unten</i> : Benutzerin beim Anlegen der Objektklasse auf dem berührungssensitiven Bildschirm . . . . .	147
7.6	Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (2). <i>Oben</i> : Hinzufügen von Eigenschaften zur Objektklasse, <i>Mitte</i> : Platzieren des Objektes in der Modellierungsumgebung und <i>unten</i> : Auswählen der Parameter zur Erfassung der dreidimensionalen Objektgeometrie und der Objektansichten und -silhouetten . . . . .	148
7.7	Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (3). <i>Oben</i> : Einsatz von Drehbügel, Drehteller und Beleuchtungsvariationen zur Objektmodellierung, <i>Mitte</i> : Resultierende Punktwolke und <i>unten</i> : Beispiel für erfasste Objektansicht und -silhouette . . . . .	149

---

7.8	Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (4). <i>Oben:</i> Interaktive Modellierung der Hauptachsen, <i>Mitte:</i> Modellierung der stabilen Flächen durch eine Benutzerin und <i>unten:</i> Resultierende stabile Flächen	150
7.9	Anwendungsbeispiel: Interaktive Modellierung der Objektklasse TASSE (5). <i>Oben:</i> Auswahl der zu modellierenden Manipulationseinschränkungen, <i>Mitte:</i> Modellierung dieser Attribute durch eine Benutzerin und <i>unten:</i> Anzeige des Objektmodells nach Abschluss der Modellierung . . . . .	151
A.1	Allied Vision Technologies <i>Marlin 145C2</i> , aus: Allied Vision Technologies (2007)	159
A.2	Minolta <i>VI-900</i> , aus: Konica Minolta (2007) . . . . .	160
A.3	Isel <i>RF-1</i> , aus: Isel Automation (2007) . . . . .	160
A.4	Amtec <i>PowerCube PR 090</i> , aus: Amtec Robotics (2007) . . . . .	161
A.5	Intersense <i>InertiaCube2</i> , aus: InterSense (2007) . . . . .	162
A.6	Immersion <i>CyberGlove</i> , aus: Immersion (2004) . . . . .	163
A.7	Polhemus <i>Fastrack</i> , aus: Polhemus (2007) . . . . .	164

# Tabellenverzeichnis

6.1	Wichtige Relationen zwischen Objekten in der Domäne <i>Küche</i> . . . . .	121
6.2	Typische Objekteigenschaften in der Domäne <i>Küche</i> (1) . . . . .	123
6.3	Typische Objekteigenschaften in der Domäne <i>Küche</i> (2) . . . . .	124
6.4	Wichtige Werttypen in der Domäne <i>Küche</i> . . . . .	126
7.1	Bearbeitungszeiten für die Modellierung der stabilen Flächen . . . . .	136
7.2	Bewertung der Modellierung der stabilen Flächen durch die Versuchspersonen	138
7.3	Bearbeitungszeiten für die Modellierung von Manipulationseinschränkungen .	139
7.4	Bewertung der Modellierung von Manipulationseinschränkungen durch die Versuchspersonen . . . . .	140
A.1	Technische Daten der Allied Vision Technologies <i>Marlin 145C2</i> nach Allied Vision Technologies (2007) . . . . .	159
A.2	Technische Daten des Minolta <i>VI-900</i> nach Konica Minolta (2007) . . . . .	160
A.3	Technische Daten des Isel <i>RF-1</i> nach Isel Automation (2007) . . . . .	161
A.4	Technische Daten des Amtec <i>PowerCube PR 090</i> nach Amtec Robotics (2007)	161
A.5	Technische Daten des Intersense <i>InertiaCube2</i> nach InterSense (2007) . . . .	162
A.6	Technische Daten des Panasonic <i>TH-50PF9EK</i> nach Panasonic (2007) . . .	162
A.7	Technische Daten des Panasonic <i>TY-TP50P8-S</i> nach Panasonic (2007) . . .	163
A.8	Technische Daten des Immersion <i>CyberGlove</i> nach Immersion (2004) . . . .	163
A.9	Technische Daten des Polhemus <i>Fastrack</i> nach Polhemus (2007) . . . . .	164

# Algorithmenverzeichnis

5.1	Ablauf der Silhouettenextraktion . . . . .	99
5.2	Ablauf der interaktiven Modellierung stabiler Flächen . . . . .	100
5.3	Berechnung einer stabilen Fläche aus ausgewählten Objektpunkten . . . . .	102
5.4	Ablauf der interaktiven Modellierung von Manipulationseinschränkungen . . . . .	107
5.5	Berechnung der gesuchten maximalen Geschwindigkeiten . . . . .	107
5.6	Berechnung der gesuchten maximalen Beschleunigungen . . . . .	107

## Literaturverzeichnis

- [Allied Vision Technologies 07] Allied Vision Technologies. AVT Marlin 145C2. [http://www.alliedvisiontec.de/files/pdf/produkte/Marlin\\_F145B2\\_C2.pdf](http://www.alliedvisiontec.de/files/pdf/produkte/Marlin_F145B2_C2.pdf), 2007. Version vom 06.02.2007.
- [Amtec Robotics 07] Amtec Robotics. Powercube. [http://www.powercube.de/schwenken\\_pr\\_de.html](http://www.powercube.de/schwenken_pr_de.html), 2007. Version vom 06.02.2007.
- [Arsenio 03] Arthur M. Arsenio. Embodied vision – perceiving objects from actions. Tagungsband: *Proceedings of the 2003 IEEE International Workshop on Robot and Human Interactive Communication*, Millbrae, California, USA, Oktober/November 2003.
- [Arsenio 04] Arthur M. Arsenio. Learning task sequences from scratch: Applications to the control of tools and toys by a humanoid robot. Tagungsband: *Proceedings of the 2004 IEEE International Conference on Control Applications*, Taipei, Taiwan, September 2004.
- [Ascension 06] Ascension. Flock of birds. <http://www.ascension-tech.com/products/flockofbirds.php>, 2006. Version vom 9.11.2006.
- [Asoh 01] Hideki Asoh, Yoichi Motomura, Futoshi Asano, Isao Hara, Satoru Hayamizu, Katsunobu Itou, Takio Kurita, Toshihiro Matsui, Nikos Vlassis, Roland Bunschoten, Ben Kröse. Jijo-2: An office robot that communicates and learns. *IEEE Intelligent Systems*, 16(5):46–55, 2001.
- [Azad 06] Pedram Azad, Tamim Asfour, Rüdiger Dillmann. Combining appearance-based and model-based methods for real-time object recognition and 6D localization. Tagungsband: *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 06)*, Beijing, China, Oktober 2006.
- [Ballard 91] Dana H. Ballard. Animate vision. *Artificial Intelligence*, 48(1):57–86, Februar 1991.
- [Bauckhage 98] Christian Bauckhage, Franz Kummert, Gerhard Sagerer. Modeling and recognition of assembled objects. Tagungsband: *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society*, Seiten 2051–2056, Aachen, September 1998.
- [Becher 03] Regine Becher, Peter Steinhaus, Rüdiger Dillmann. Interactive object modelling for a humanoid service robot. Tagungsband: *Proceedings of the Third IEEE International Conference on Humanoid Robots (Humanoids 03)*, Karlsruhe, Germany, Oktober 2003.
- [Becher 04] Regine Becher, Peter Steinhaus, Rüdiger Dillmann. The Collaborative Research Center 588: „Humanoid Robots – Learning and Cooperating Multimodal Robots“. *International Journal on Humanoid Robotics*, 1(3):429–448, September 2004.
- [Becher 06] Regine Becher, Ingo Boesnach, Peter Steinhaus, Rüdiger Dillmann. From subjects to objects and back – combining human motions and object properties to understand user actions. Tagungsband: *Proceedings of the Human-Centered Robotic Systems (HCRS)*, München, Germany, Oktober 2006.



- [Biedermann 87] Irving Biedermann. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.
- [Bley 06] Florian Bley, Volker Schmirgel, Karl-Friedrich Kraiss. Mobile manipulation based on generic object knowledge. Tagungsband: *Proceedings of the 15th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN06)*, Seiten 411–416, Hatfield, UK, September 2006.
- [Bogoni 94] Luca Bogoni, Ruzena Bajcsy. Active investigation of functionality. Tagungsband: *Proceedings of the International Conference on Vision and Pattern Recognition*, Seattle, June 1994.
- [Bogoni 95] Luca Bogoni, Ruzena Bajcsy. Interactive recognition and representation of functionality. *Computer Vision and Image Understanding: CVIU*, 62(2):194–214, 1995.
- [Brezinka 84] Wolfgang Brezinka. „Modelle“ in den Erziehungstheorien. *Zeitschrift für Pädagogik*, 30(6):835–858, 1984.
- [Chandrasekaran 94] Balakrishnan Chandrasekaran. Functional representation: A brief historical perspective. *Applied Artificial Intelligence*, 8:173–197, 1994.
- [Chandrasekaran 97] Balakrishnan Chandrasekaran, John R. Josephson. Representing function as effect. In Mohammed Modarres, Hrsg., Tagungsband: *Proceedings of the Functional Modeling Workshop*, Paris, Frankreich, 1997.
- [Chandrasekaran 99] Balakrishnan Chandrasekaran, John R. Josephson, V. Richard Benjamins. What are ontologies, and why do we need them? *IEEE Intelligent Systems*, 14(1):20–26, Januar/Februar 1999.
- [Chella 02] Antonio Chella, Massimo Cossentino, Roberto Pirrone, Andrea Ruisi. Modeling ontologies for robotic environments. Tagungsband: *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*, Seiten 77–80, 2002.
- [Coradeschi 99] Silvia Coradeschi, Alessandro Saffiotti. Anchoring symbolic object descriptions to sensor data. Problem statement. *Linköping Electronic Articles in Computer and Information Science*, 4(9), August 1999.
- [Cranefield 01] Stephen Cranefield, Stefan Haustein, Martin Purvis. Uml-based ontology modelling for software agents. Tagungsband: *Proceedings of the Workshop on Ontologies in Agent Systems, 5th International Conference on Autonomous Agents*, 2001.
- [Cutkosky 89] Mark R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *IEEE Transactions on Robotics and Automation*, 5(3):269–279, Juni 1989.
- [Dahm 06] Markus Dahm. *Grundlagen der Mensch-Computer-Interaktion*. Pearson Education, München, 2006.
- [Dillmann 02] Rüdiger Dillmann, Raoul Zöllner, Markus Ehrenmann, Oliver Rogalla. Interactive natural programming of robots: Introductory overview. Tagungsband: *DREH 2002*, Toulouse, Frankreich, Oktober 2002.

- [Dillmann 04] Rüdiger Dillmann, Regine Becher, Peter Steinhaus. Armar II – A learning and cooperative multimodal humanoid robot system. *International Journal on Humanoid Robotics*, 1(1):143–155, 2004.
- [Dix 06] Alan Dix, Janet Finlay, Gregory D. Abowd, Russell Beale. *Human-Computer Interaction*. Pearson Education, Harlow, England, 2006.
- [Drygajlo 03] Andrzej Drygajlo, Plamen J. Prodanov, Guy Ramel, Mathieu Meisser, Roland Siegart. On developing a voice-enabled interface for interactive tour-guide robots. *Advanced Robotics*, 17(7):599–616, 2003.
- [Ehrenmann 03] Markus Ehrenmann. *Handlungsbeobachtung zur Instruierung von Robotersystemen*. Dissertation, Universität Karlsruhe (TH), Herdecke, 2003.
- [Eisenberg 06] Florian Eisenberg. Entwurf und Implementierung eines Modellierungscenars zur semiautomatischen und interaktiven Objektmodellierung. Diplomarbeit, Universität Karlsruhe (TH), 2006.
- [Emde 91] Werner Emde. *Modellbildung, Wissensrevision und Wissensrepräsentation im Maschinellen Lernen*. Springer-Verlag, Berlin, 1991.
- [Fautz 02] Michael Fautz. *Objekt- und Texturrekonstruktion mit einer robotergeführten Kamera*. Dissertation, Universität Karlsruhe (TH), Aachen, 2002.
- [Fifth Dimension Technologies 06] Fifth Dimension Technologies. Data gloves. <http://www.5dt.com/hardware.html#glove>, 2006. Version vom 10.12.2006.
- [Fitzpatrick 03a] Paul Fitzpatrick. First contact: an active vision approach to segmentation. Tagungsband: *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'03)*, Las Vegas, Nevada, USA, Oktober 2003.
- [Fitzpatrick 03b] Paul Fitzpatrick. Object lesson: discovering and learning to recognize objects. Tagungsband: *Proceedings of the Third IEEE International Conference on Humanoid Robots (Humanoids'03)*, Karlsruhe, Germany, Oktober 2003.
- [Friedrich 98a] Holger Friedrich. *Interaktive Programmierung von Manipulationssequenzen*. Dissertation, Universität Karlsruhe (TH), Herdecke, 1998.
- [Friedrich 98b] Holger Friedrich, J. Holle, Rüdiger Dillmann. Interactive generation of flexible robot programs. Tagungsband: *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '98)*, Leuven, Belgien, 1998.
- [Fry 98] John Fry, Hideki Asoh, Toshihiro Matsui. Natural dialogue with the Jijo-2 office robot. Tagungsband: *Proceedings of the 1998 IEEE International Conference on Intelligent Robots and Systems (IROS '98)*, Seiten 1278–1283, Victoria, B.C., Kanada, Oktober 1998.
- [Fügen 06] Christian Fügen, Petra Gieselmann, Hartwig Holzapfel, Florian Kraft. Natural human robot communication. Tagungsband: *Proceedings of the Human-Centered Robotic Systems (HCRS'06)*, München, Oktober 2006.

- [Galitz 97] Wilbert O. Galitz. *Essential guide to user interface design*. Wiley Computer Publishing, New York, 1997.
- [Gangemi 02] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, Luc Schneider. Sweetening ontologies with DOLCE. Tagungsband: *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, Sigüenza, Spanien, Oktober 2002.
- [Geib 06] Christopher Geib, Kira Mourao, Ron Petrick, Norbert Krueger, Florentin Wörgötter. Object action complexes as an interface for planning and robot control. Tagungsband: *Proceedings of the International Conference on Humanoid Robots (Humanoids 2006), Workshop „Toward Cognitive Humanoid Robots“*, Genua, Italien, Dezember 2006.
- [Giesler 04] Björn Giesler, Tobias Salb, Peter Steinhaus, Rüdiger Dillmann. Using augmented reality to interact with an autonomous mobile platform. Tagungsband: *Proceedings of the International Conference on Robotics and Automation (ICRA 04)*, New Orleans, USA, 2004.
- [Giesler 05] Björn Giesler. *Brillenbasierte Erweiterte Realität zur Unterstützung der kommandierenden und kommentierenden Zusammenarbeit mit Servicerobotern*. Dissertation, Universität Karlsruhe (TH), Waabs, 2005.
- [Göger 06] Dirk Göger, Karsten Weiß, Catherina Burghart, Heinz Wörn. Sensitive skin for a humanoid robot. Tagungsband: *Proceedings of the Human-Centered Robotic Systems (HCRS '06)*, München, Oktober 2006.
- [Gruber 93] Thomas R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5(2):199–220, 1993.
- [Haasch 04] Axel Haasch, S. Hohenner, S. Hüwel, M. Kleinhagenbrock, S. Lang, I. Toptsis, Gernot A. Fink, Jannik Fritsch, Britta Wrede, Gerhard Sagerer. BIRON – The Bielefeld Robot Companion. In G. Prassler, G. Lawitzky, P. Fiorini, M. Haegele, Hrsg., Tagungsband: *Proceedings of the International Workshop on Advances in Service Robotics*, Seiten 27–32, Stuttgart, Mai 2004. Fraunhofer IRB Verlag.
- [Harnad 90] Stevan Harnad. The symbol grounding problem. *Physica*, D 42:335–346, 1990.
- [Hashimoto 00] Hideki Hashimoto. Haptic interfaces. <http://dfs.iis.u-tokyo.ac.jp/Research/Haptic/index.html>, 2000. Version vom 25.7.2000.
- [Heinecke 04] Andreas M. Heinecke. *Mensch-Computer-Interaktion*. Carl Hanser Verlag, München, 2004.
- [Henrich 05] Christian Henrich. Semantische Modelle zur Repräsentation von Handlungen. Studienarbeit, Universität Karlsruhe (TH), 2005.
- [Higuchi 04] Mirai Higuchi, Shigeki Aoki, Atsuhiko Kojima, Kunio Fukunaga. Scene recognition based on relationship between human actions and objects. Tagungsband: *Proceedings of the 17th International Conference on Pattern Recognition (ICPR'04)*, 2004.

- [Hommel 01] Bernhard Hommel, Jochen Müsseler, Gisa Aschersleben, Wolfgang Prinz. The Theory of Event Coding (TEC): A framework for perception and action planning. *Behavioral and Brain Sciences*, 24:849–937, 2001.
- [Hommel 04] Bernhard Hommel. Event files: feature binding in and across perception and action. *Trends in Cognitive Science*, 8(11):494–500, November 2004.
- [Iba 05] Soshi Iba, Christiaan J. J. Paredis, Pradeep K. Khosla. Interactive multi-modal robot programming. *International Journal of Robotics Research*, 24(1):83–104, Januar 2005.
- [Immersion 04] Immersion. CyberGlove spezifikation. <http://www.immersion.com>, 2004.
- [Immersion 06] Immersion. CyberGlove II. [http://www.immersion.com/3d/products/cyber\\_glove.php](http://www.immersion.com/3d/products/cyber_glove.php), 2006. Version vom 10.11.2006.
- [InterSense 07] InterSense. InertiaCube2. <http://www.intersense.com>, 2007. Version vom 06.02.2007.
- [Isel Automation 07] Isel Automation. Rotationseinheit RF-1. <http://www.iselautomation.de/products/category.php?lang=de&ID=c57>, 2007. Version vom 06.02.2007.
- [Kasper 07] Alexander Kasper. Untersuchung intuitiver Mensch-Maschine-Schnittstellen zur interaktiven Objektmodellierung. Diplomarbeit, Universität Karlsruhe (TH), 2007.
- [Keyson 97] David Keyson. Touch in user interface navigation. Tagungsband: *Developments in Tactile Displays. IEE Colloquium proceedings No. 96/012*, London, 1997.
- [Kitahashi 05] Tadahiro Kitahashi, Atsuhiko Kojima, Mirai Higuchi, Kunio Fukunaga. A cooperative recognition of human movements and related objects guided by functional models of objects. Tagungsband: *Proceedings of the 2005 International Conference on Cyberworlds (CW'05)*, 2005.
- [Komatsu 92] Lloyd K. Komatsu. Recent views of conceptual structure. *Psychological Bulletin*, 112(3):500–526, 1992.
- [Konica Minolta 07] Konica Minolta. Optische 3D-Messtechnik. <http://www.konicaminolta-3d.com/>, 2007. Version vom 06.02.2007.
- [Krüger 02] Norbert Krüger, Marcus Ackermann, Gerald Sommer. Accumulation of object representations utilizing interaction of robot action and perception. *Knowledge Based Systems*, 13(2), 2002.
- [Lakoff 87] George Lakoff. *Women, fire and dangerous things: What categories reveal about the mind*. University of Chicago Press, Chicago, 1987.
- [Leibe 03] Bastian Leibe, Bernt Schiele. Analyzing appearance and contour based methods for object categorization. Tagungsband: *Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)*, Band 2, Seiten 409–415, Juni 2003.

- [Li 05] Shuyin Li, Axel Haasch, Britta Wrede, Jannik Fritsch, Gerhard Sagerer. Human-style interaction with a robot for cooperative learning of scene objects. Tagungsband: *Proceedings of the International Conference on Multimodal Interfaces*, Seiten 151–158, Trento, Italien, 2005. ACM Press.
- [Löbner 03] Sebastian Löbner. *Semantik. Eine Einführung*. Walter de Gruyter, Berlin, 2003.
- [Lömker 02] Frank Lömker, Gerhard Sagerer. A multimodal system for object learning. In L. Van Gool, Hrsg., Tagungsband: *Proceedings of the 24th DAGM Symposium on Pattern Recognition*, Band 2449 of *Lecture Notes in Computer Science*, Seiten 490–497, Zürich, Schweiz, September 2002. Springer, Berlin.
- [Lopes 00] L. Seabra Lopes, A. Teixeira. Human-robot interaction through spoken language dialogue. Tagungsband: *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, Seiten 528–534, 2000.
- [Mädche 01] Alexander Mädche, Steffen Staab, Rudi Studer. Ontologien. *Wirtschaftsinformatik. WI-Schlagwort*, 4, 2001.
- [Mädche 03] Alexander Mädche, Boris Motik. Repräsentations- und Anfragesprachen für Ontologien – eine Übersicht. *Datenbank-Spektrum*, 6:43–53, 2003.
- [Mann 98] Richard Mann, Allan Jepson. Towards the computational perception of action. Tagungsband: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Seiten 794–799, Santa Barbara, CA, USA, Juni 1998.
- [Mellott 99] Kevin Mellott. Glove-based input interfaces. <http://www.geocities.com/mellott124/glove1>, 1999. Version vom 18.6.1999.
- [Meta Motion 06a] Meta Motion. Gypsy 5. <http://www.metamotion.com/gypsy/gypsy-motion-capture-system.htm>, 2006. Version vom 10.11.2006.
- [Meta Motion 06b] Meta Motion. Motion Captor. <http://www.metamotion.com/captor/motion-captor.htm>, 2006. Version vom 10.11.2006.
- [Miller 03] A.T. Miller, S. Knoop, H.I. Christensen, P.K. Allen. Automatic grasp planning using shape primitives. Tagungsband: *Proceedings of the International Conference on Robotics and Automation (ICRA)*, Seiten 1824–1829, Taipei, 2003.
- [Minar 04] Jan Minar. Visualisierung von Zuständen im Serviceroboter Albert. Diplomarbeit, Universität Karlsruhe (TH), 2004.
- [Modayil 04] Joseph Modayil, Benjamin Kuipers. Bootstrap learning for object discovery. Tagungsband: *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '04)*, Seiten 742–747, Sendai, Japan, September/Oktober 2004.
- [Moll 01] Mark Moll, Michael A. Erdmann. Reconstructing shape from motion using tactile sensors. Tagungsband: *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '01)*, Oktober/November 2001.

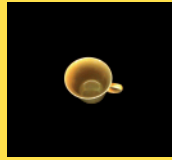
- [Moore 99] Darnell J. Moore, Irfan A. Essa, Monson H. Hayes. Exploiting human actions and object context for recognition tasks. Tagungsband: *Proceedings of the 7th IEEE International Conference on Computer Vision*, Band 1, Seiten 80–86, September 1999.
- [Müller 06a] Roland Müller. Der Modellbegriff: Definitionen, Bedeutungs- und Funktionsvielfalt. <http://www.muellerscience.com/MODELL/Definitionen/Modellbegriff.Definition.htm>, 2006. Version vom 26.4.2006.
- [Müller 06b] Roland Müller. Modellgeschichte ist Kulturgeschichte: Eine Chronik von Modellgebrauch und Modellbegriff. <http://www.muellerscience.com/MODELL/Begriffsgeschichte/ModellgeschichteistKulturgeschichte.htm>, 2006. Version vom 26.4.2006.
- [Müller 77] Roland Müller. Was ist ein Modell? *Schweizer Rundschau*, Seiten 3–10, Dezember 1977.
- [MySQL AB 07] MySQL AB. MySQL-Homepage. <http://www.mysql.com>, 2007. Version vom 09.01.2007.
- [Natale 04] Lorenzo Natale. *Linking Action to Perception in a Humanoid Robot: A Developmental Approach to Grasping*. Dissertation, LIRA-Lab, DIST, University of Genua, 2004.
- [Natale 05a] Lorenzo Natale, Francesco Orabona, Fabio Berton, Giorgio Metta, Giulio Sandini. From sensorimotor development to object perception. Tagungsband: *Proceedings of the International Conference on Humanoid Robotics*, 2005.
- [Natale 05b] Lorenzo Natale, Francesco Orabona, Giorgio Metta, Giulio Sandini. Exploring the world through grasping: a developmental approach. Tagungsband: *Proceedings of the 6th CIRA Symposium*, Espoo, Finnland, Juni 2005.
- [Nickel 03] Kai Nickel. Erkennung von Zeigegesten basierend auf 3D-Tracking von Kopf und Händen. Diplomarbeit, Universität Karlsruhe (TH), 2003.
- [Noy 01] Natalya F. Noy, Deborah L. McGuinness. Ontology development 101: A guide to creating your first ontology. Technischer Bericht SMI technical report SMI-2001-0880, Medical Informatics, School of Medicine, Stanford University, 2001.
- [Noy 97] Natalya F. Noy, Carole D. Hafner. The state of the art in ontology design – A survey and comparative review. *AI Magazine*, 36:53–74, Herbst 1997.
- [Object Management Group (OMG) 07] Object Management Group (OMG). Corba-Spezifikation. <http://www.omg.org>, 2007. Version vom 09.01.2007.
- [Oel 02] Peter Oel. *Optimierung der Bedienzeit von mausgesteuerten grafischen Mensch-Maschine-Schnittstellen*. Dissertation, Universität Karlsruhe (TH), 2002.
- [Okamura 01] Allison M. Okamura, Mark R. Cutkosky. Feature detection for haptic exploration with robotic fingers. *The International Journal of Robotics Research*, 20(12):925–938, Dezember 2001.

- [PACO-PLUS 06] PACO-PLUS. PACO-PLUS – Perception, action and cognition through learning of Object-Action-Complexes. <http://www.paco-plus.org/>, 2006.
- [Panasonic 07] Panasonic. Plasmadisplay und berührungssensitiver Aufsatz. <http://www.panasonic.de>, 2007. Version vom 06.02.2007.
- [Pease 02] Adam Pease, Ian Niles. IEEE Standard Upper Ontology: A progress report. *Knowledge Engineering Review, Special Issue on Ontologies and Agents*, 17:513–526, 2002.
- [Perzanowski 01] Dennis Perzanowski, Alan C. Schultz, William Adams, Elaine Marsh, Magda Bugajska. Building a multimodal human-robot interface. *IEEE Intelligent Systems*, 16(1):16–21, Januar/Februar 2001.
- [Polhemus 06] Polhemus. Motion tracking. <http://www.polhemus.com>, 2006. Version vom 10.12.2006.
- [Polhemus 07] Polhemus. Fastrack. [http://www.polhemus.com/?page=Motion\\_Fastrak](http://www.polhemus.com/?page=Motion_Fastrak), 2007. Version vom 06.02.2007.
- [Pope 94] Arthur R. Pope. Model-based object recognition – A survey of recent research. Technischer Bericht Technical Report 94-04, Department of Computer Science, University of British Columbia, Januar 1994.
- [Preim 99] Bernhard Preim. *Entwicklung interaktiver Systeme: Grundlagen, Fallbeispiele und innovative Anwendungsfelder*. Springer, Berlin, 1999.
- [Riepp 97] Markus Riepp. Wissensbasierte Parametrierung von Operatorsequenzen. Diplomarbeit, Universität Karlsruhe (TH), 1997.
- [Rivlin 95] Ehud Rivlin, Sven J. Dickinson, Azriel Rosenfeld. Recognition by functional parts. *Computer Vision and Image Understanding*, September 1995.
- [Rogalla 03] Oliver Rogalla. *Abbildung von Benutzerdemonstrationen auf variable Roboterkonfigurationen*. Dissertation, Universität Karlsruhe (TH), Herdecke, 2003.
- [Rosch 75] Eleanor Rosch, Carolyn B. Mervis. Family resemblances: Studies in the internal structure of categories. *Cognitive Psychology*, 7:573–605, 1975.
- [Salb 03] Tobias Salb. *Risikoreduktion in der Mund-Kiefer-Gesichts-Chirurgie mittels rechnerbasierter Modellierung und Erweiterter Realität*. Dissertation, Universität Karlsruhe (TH), 2003.
- [Schaude 97] Horst Schaude. *Überprüfung und Überwachung in der Telerobotik*. Dissertation, Universität Karlsruhe (TH), 1997.
- [Scheerer 93] Eckart Scheerer. Mentale Repräsentation: Umriß einer Begriffsgeschichte. In Hans Jörg Sandkühler, Hrsg., *Repräsentation und Modell. Formen der Welterkenntnis*, Band 14 of *Schriftenreihe*, Seiten 9–38. Zentrum Philosophische Grundlagen der Wissenschaft, Bremen, 1993.
- [Shneiderman 00] Ben Shneiderman. The limits of speech recognition. *Communications of the ACM*, 43(9):63–65, September 2000.

- [Siskind 95] Jeffrey M. Siskind. Grounding language in perception. *AI Review*, 8(5-6):371–391, 1995.
- [Smith 81] Edward E. Smith, Douglas L. Medin. *Categories and Concepts*. Harvard University Press, Cambridge, Mass., 1981.
- [Soltau 01] Hagen Soltau, Thomas Schaaf, Florian Metze, Alex Waibel. The ISL evaluation system for Verbmobil-II. Tagungsband: *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, USA, Mai 2001.
- [Stachowiak 73] Herbert Stachowiak. *Allgemeine Modelltheorie*. Springer-Verlag, Wien, 1973.
- [Stark 91] Louise Stark, Kevin Bowyer. Achieving generalized object recognition through reasoning about association of function to structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(10):1097–1104, Oktober 1991.
- [Stein 06] Thorsten Stein, Andreas Fischer, Klaus Bös, Veit Wank, Ingo Boesnach, Jörg Moldenhauer. Guidelines for motion control of humanoid robots: Analysis and modelling of human movements. *International Journal of Computer Science in Sports*, 5(1), August 2006.
- [Steinfeld 04] Aaron Steinfeld. Interface lessons for fully and semi-autonomously mobile robots. Tagungsband: *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA'04)*, Seiten 2752–2757, New Orleans, USA, April 2004.
- [Steinhaus 04] Peter Steinhaus, Regine Becher, Rüdiger Dillmann. Sonderforschungsbereich 588: Humanoide Roboter – Lernende und kooperierende multimodale Roboter. *it – Information Technology*, 46(2):94–100, April 2004.
- [Strube 96] Gerhard Strube, Barbara Becker, Christian Freksa, Udo Hahn, Klaus Opwis, Günther Palm, Hrsg. *Wörterbuch der Kognitionswissenschaft*. Klett-Cotta, Stuttgart, 1996.
- [Takahashi 98] Takuya Takahashi, Satoru Nakanishi, Yoshinori Kuno, Yoshiaki Shirai. Helping computer vision by verbal and nonverbal communication. Tagungsband: *Proceedings of the Fourteenth International Conference on Pattern Recognition*, Band 2, Seiten 1216–1218, Brisbane, Qld., Australien, August 1998.
- [Trolltech 07] Trolltech. Qt-Homepage. <http://www.trolltech.com/products/qt>, 2007. Version vom 06.02.2007.
- [van der Vet 98] Paul E. van der Vet, Nicolaas J. I. Mars. Bottom-up construction of ontologies. *IEEE Transactions on Knowledge and Data Engineering*, 10(4):513–526, Juli/August 1998.
- [Vicon 06] Vicon. Motion capture systems. <http://www.vicon.com/>, 2006. Version vom 10.12.2006.



- [Wachsmuth 05] Sven Wachsmuth, Marc Hanheide, Sebastian Wrede, Christian Bauckhage. From images via symbols to contexts: Using augmented reality for interactive model acquisition. In D. Paulus, D. Droege, Hrsg., Tagungsband: *Proceedings of the KI 2005 Workshop on Mixed-reality as a Challenge to Image Understanding and Artificial Intelligence*, Seiten 41–46, 2005.
- [Wachsmuth 99] Sven Wachsmuth, Gerhard Sagerer. Connecting concepts from vision and speech processing. Tagungsband: *Proceedings of the Conference on Integration of Speech and Image Understanding*, Seiten 1–19, Korfu, Griechenland, 1999.
- [Wessel 98] Ivo Wessel. *GUI-Design: Richtlinien zur Gestaltung ergonomischer Windows-Applikationen*. Hanser, München, 1998.
- [Wilson 03] Andrew Wilson, Steven Shafer. Xwand: UI for intelligent spaces. Tagungsband: *Proceedings of the International Conference on Human Factors in Computing Systems*, Seiten 545–552, 2003.
- [Winston 83] Patrick H. Winston, Thomas O. Binford, Boris Katz, Michael Lowry. Learning physical descriptions from functional definitions, examples and precedents. Technischer Bericht M.I.T. AI Memo 679, Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Januar 1983.
- [Wrede 04] Britta Wrede, Axel Haasch, N. Hofemann, S. Hohenner, S. Hüwel, M. Kleinehagenbrock, S. Lang, Shuyin Li, I. Tóptsis, Gernot A. Fink, Jannik Fritsch, Gerhard Sagerer. Research issues for designing robot companions: BIRON as a case study. In P. Drews, Hrsg., Tagungsband: *Proceedings of the IEEE Conference on Mechatronics and Robotics*, Band 4, Seiten 1491–1496, Aachen, September 2004. Eysoldt-Verlag.
- [Zetie 95] Carl Zetie. *Practical user interface design: making GUIs work*. McGraw-Hill, Maidenhead, 1995.
- [Zöllner 01] Raoul Zöllner, Oliver Rogalla, Rüdiger Dillmann. Integration of tactile sensors in a Programming by Demonstration system. Tagungsband: *Proceedings of the International Conference on Robotics and Automation (ICRA '01)*, Seoul, Korea, Mai 2001.
- [Zöllner 02] Raoul Zöllner, Oliver Rogalla, Rüdiger Dillmann, Marius Zöllner. Understanding user intentions: Programming fine manipulation tasks by demonstration. Tagungsband: *Proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'02)*, Lausanne, Schweiz, 2002.
- [Zöllner 05] Raoul Daniel Zöllner. *Erlernen zweihändiger feinmotorischer Handhabungen*. Dissertation, Universität Karlsruhe (TH), 2005.



Beim Einsatz von Servicerobotern in menschenzentrierten Umgebungen wie beispielsweise Haushalten stellen sich spezifische Probleme für die Objektmodellierung.

In der vorliegenden Arbeit wird als Lösung ein Konzept für eine interaktive semantische Objektmodellierung vorgeschlagen. Die zugrunde liegende flexible und erweiterbare Objektrepräsentation ermöglicht die Modellierung funktionaler und semantischer Objektinformationen durch die Darstellung von Objekteigenschaften, die menschliche Begriffe und Kategorien abbilden und die Verbindung von Objekten mit Handlungen und mit sensoruell erfassbaren Attributen herstellen. Dadurch werden der Einsatz von Robotern in menschenzentrierten Umgebungen und ihre Interaktion mit dem Menschen wesentlich erleichtert.

Das im Rahmen der Arbeit entwickelte interaktive Objektmodellierungssystem IOMOS schließlich erlaubt durch intuitive Interaktionsmodi auch ungeübten Benutzern die Erstellung von semantischen Objektmodellen.