



Xpert.press

Bernd Ludwig

# Planbasierte Mensch-Maschine- Interaktion in multimodalen Assistenzsystemen

 Springer Vieweg

Xpert.press

Die Reihe **Xpert.press** vermittelt Professionals in den Bereichen Softwareentwicklung, Internettechnologie und IT-Management aktuell und kompetent relevantes Fachwissen über Technologien und Produkte zur Entwicklung und Anwendung moderner Informationstechnologien.

---

Bernd Ludwig

# Planbasierte Mensch- Maschine-Interaktion in multimodalen Assistenzsystemen

 Springer Vieweg

Bernd Ludwig  
Lehrstuhl für Informationswissenschaft  
Professur für Informationslinguistik  
Universität Regensburg  
Regensburg, Deutschland

ISSN 1439-5428

ISBN 978-3-662-44818-2

ISBN 978-3-662-44819-9 (eBook)

DOI 10.1007/978-3-662-44819-9

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Springer Vieweg

© Springer-Verlag Berlin Heidelberg 2015

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen.

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier.

Springer-Verlag GmbH Berlin Heidelberg ist Teil der Fachverlagsgruppe Springer Science+Business Media  
([www.springer.com](http://www.springer.com))

---

# Vorwort

Das vorliegende Buch handelt von Assistenzsystemen. Die ursprüngliche Bedeutung von *assistieren* ist *beistehen* oder *helfen*. Ein Assistent ist also jemand, der andere durch eigenverantwortliche Durchführung bestimmter Teilaufgaben in einem komplexen Zusammenhang unterstützt. In diesem Sinn sind alle menschliche Assistenten zu verstehen: ein wissenschaftlicher Assistent eben so wie ein technisch medizinischer Assistent oder ein Assistent der Geschäftsführung. Charakteristisch für diese Beispiele ist dabei, dass sie zu lösende Aufgabe von der Person definiert wird, der Assistenz zu leisten ist. Diese Idee gibt es auch im technischen Bereich und bei Computersoftware. Um also den Begriff *Assistenz*, so wie er in diesem Buch verstanden wird, zu definieren, ist es notwendig, zu überprüfen, welche Eigenschaften von Assistenz, wie sie aus dem Alltag bekannt sind, auch in technischen Assistenzsystemen zu finden sind.

In diesem Sinn thematisiert dieses Buch die Frage, wie *Assistenzsysteme* als *Softwaresysteme* so konzipiert werden können, dass sie den Anforderungen an die zu lösende Aufgabe erfüllen und dabei auch ergonomischen Randbedingungen genügen. Interessant dabei ist ein sehr abstrakter Ansatz, der untersucht, welche Methoden und Verfahren der Informatik, insbesondere der Künstlichen Intelligenz und der Mustererkennung, wie einzusetzen sind, um ein generisches Konzept zu entwickeln, wie Assistenzsysteme systematisch implementiert werden können.

Um dieses Ziel zu erreichen, wird untersucht, welchen Einfluss Ergebnisse zu Assistenzsystemen aus der Psychologie, der Kognitionswissenschaft, des *software engineering* und des *usability engineering* auf die zu verwendenden Verfahren der Informatik haben. Die entwickelte Systematik zur Konstruktion von Assistenzsystemen integriert daher geeignete Methoden aller Disziplinen in einem ganzheitlichen Ansatz.

Die beschriebenen Konzepte werden zurzeit bei der Entwicklung eines Assistenzsystems zur Navigation von Fußgängern umgesetzt. Aktuelle Software und Publikationen dazu finden sich auf der Webseite des Projekts: <http://urwalking.ur.de>. Weitere Software, Beispieldaten und Anwendungen der Konzepte u. a. für Thymio-Roboter finden sich auf der Begleitwebseite zu diesem Buch: <http://www.ur.de/assi>.

An der Entstehung dieses Buchs haben viele Personen Anteil. Neben meiner Familie sei allen voran Ute Schmid, Joachim Hertzberg und Günther Görz für ihre langjährige Unterstützung gedankt. Auch viele Studentinnen und Studenten haben durch ihre Fragen

wertvolle Anregungen und Ideen eingebracht und dazu beigetragen, komplizierte Zusammenhänge zu entwirren. Schließlich geht mein inniger Dank an Nicole für zahllose Tippfehler, die sie gefunden hat, ihre immerwährende Unterstützung und ihr liebevolles Verständnis.

---

# Inhaltsverzeichnis

<b>1</b>	<b>Überblick</b> .....	1
<b>2</b>	<b>Interaktive Assistenzsysteme</b> .....	5
2.1	Typen von Assistenzsystemen .....	5
2.1.1	„Autonome“ Assistenz .....	6
2.1.2	Assistenz mit einem fixierten Ziel .....	7
2.1.3	Assistenz durch Intentionserkennung .....	8
2.2	Assistenz aus der Sicht des Nutzers .....	10
2.3	Anforderungen an Assistenzsysteme .....	15
2.3.1	Interaktivität .....	16
2.3.2	Fähigkeit zur Diagnose .....	24
2.3.3	Fähigkeit zur Korrektur .....	29
2.3.4	Fähigkeit zur Erklärung .....	32
2.3.5	Fähigkeit zur Relaxation .....	42
2.4	Zusammenfassender Überblick .....	44
	Literatur .....	45
<b>3</b>	<b>Interaktion mit Assistenzsystemen</b> .....	47
3.1	Paradigmen für zweckrationale Mensch-Maschine-Interaktion .....	48
3.1.1	Interaktion über graphische Benutzeroberflächen .....	48
3.1.2	Interaktion in natürlicher Sprache .....	50
3.1.3	Die nutzerzentrierte Wende: Erweiterung von Benutzerschnittstellen um Aufgabenanalysen .....	51
3.2	Wissensrepräsentation in der Mensch-Maschine-Interaktion .....	52
3.2.1	Explizites Faktenwissen .....	52
3.2.2	Inhalte der beschreibenden Situation – Beispielfall Layoutprogramm .....	53
3.2.3	Konflikte zwischen beschriebener und beschreibender Situation – Beispielfall Fernsehbedienung .....	56
3.2.4	Unvollständige und hypothetische Information in der beschreibenden Situation .....	58



3.2.5	Explizites Wissen über Handlungen . . . . .	59
3.3	Pragmatische Vorgänge in der Mensch-Maschine-Interaktion . . . . .	59
3.3.1	Vorgänge, die der Nutzer plant und durchführt . . . . .	60
3.3.2	Informationsvorgänge . . . . .	61
3.3.3	Auswahlvorgänge . . . . .	61
3.3.4	Planungsvorgänge . . . . .	61
3.3.5	Ausführungsvorgänge . . . . .	62
3.3.6	Kommunikationsvorgänge . . . . .	63
3.4	Konsequenzen für die Kontrolle von Mensch-Maschine-Interaktion . . . . .	64
3.5	Exemplarische Annotation eines kooperativen Dialogs . . . . .	65
3.5.1	Ein Dialog aus dem TRAINS-Corpus . . . . .	66
3.5.2	Diskussion der Annotation . . . . .	67
3.5.3	Empirische Bestätigung der Thesen über Dialogführung . . . . .	69
3.6	Zusammenfassung . . . . .	69
	Literatur . . . . .	70
<b>4</b>	<b>Wissenserwerb und -repräsentation für interaktive Assistenzsysteme . . . . .</b>	<b>73</b>
4.1	Task-Analyse . . . . .	74
4.2	Notationen zur Repräsentationen für Task-Analysen . . . . .	81
4.3	<i>Concurrent Task Trees</i> . . . . .	84
4.3.1	Was sind <i>Concurrent Task Trees</i> ? . . . . .	84
4.3.2	Vorteile von <i>Concurrent Task Trees</i> für die Formalisierung von Task-Analysen . . . . .	84
4.3.3	Syntaktische Konzepte für <i>Concurrent Task Trees</i> . . . . .	85
4.3.4	Eine Semantik für <i>Concurrent Task Trees</i> ? . . . . .	87
	Literatur . . . . .	89
<b>5</b>	<b>Simulation von Handlungen in Assistenzszenarien . . . . .</b>	<b>91</b>
5.1	Grundlagen des Situationskalküls . . . . .	91
5.1.1	Formalisierung der beschriebenen Situation . . . . .	92
5.1.2	Vorbedingungen von Aktionen . . . . .	94
5.1.3	Effekte von Aktionen . . . . .	94
5.1.4	Das <i>Qualification Problem</i> . . . . .	95
5.1.5	Das <i>Frame Problem</i> . . . . .	97
5.1.6	Erklärungsvollständigkeit . . . . .	99
5.2	Eine Implementierung des Situationskalküls in Prolog . . . . .	103
5.2.1	Definitionen und definitorische Theorien . . . . .	104
5.2.2	Simulation von Aktionen in GOLOG . . . . .	107
5.2.3	Alternativen zu GOLOG . . . . .	112
5.2.4	Komplexe Aktionen in GOLOG . . . . .	112
5.3	Variablen . . . . .	119
5.4	Prozeduren . . . . .	121

---

5.5	Eine GOLOG-Semantik für <i>Concurrent Task Trees</i> . . . . .	124
5.5.1	Darstellung und Aktivierung von Tasks in GOLOG . . . . .	125
5.5.2	Elementare Tasks . . . . .	128
5.5.3	Abstrakte Tasks . . . . .	129
5.6	Zusammenfassung . . . . .	140
	Literatur . . . . .	142
<b>6</b>	<b>Interaktive Planung in Assistenzszenarien mit unsicherer Information</b> . .	<b>143</b>
6.1	Unsicheres Wissen . . . . .	144
6.1.1	Unsicherheit aus Sicht der formalen Logik . . . . .	144
6.1.2	Unsicherheit in der initialen Situation . . . . .	146
6.1.3	Unsicherheit bei der Ausführung von Aktionen . . . . .	150
6.2	Bewertung von Aktionsfolgen . . . . .	153
6.2.1	Wahrscheinlichkeiten von Aktionsfolgen . . . . .	153
6.2.2	Einschätzung von Nutzen und Risiko von Aktionsfolgen . . . . .	155
6.3	Simulation und Beobachtung von Aktionen . . . . .	159
6.4	Assistenz durch Planung . . . . .	162
6.4.1	Instantiierung einer Aufgabe und eines Planungsziels . . . . .	163
6.4.2	Planen von Aktionsfolgen . . . . .	164
6.5	Planung durch Maximierung des zu erwartenden Nutzens . . . . .	166
6.5.1	Grundsätzliche Vorgehensweise . . . . .	167
6.5.2	Markov-Prozesse . . . . .	168
6.5.3	Technische Realisierung . . . . .	169
6.5.4	Verarbeiten unvollständiger Information . . . . .	173
6.5.5	Bewertung des Ansatzes für Assistenzsysteme . . . . .	186
6.5.6	Zusammenhang zwischen <i>Concurrent Task Trees</i> und Markov- Prozessen . . . . .	190
6.6	Planung durch Generieren von Programmen („Klassisches Planen“) . . .	195
6.6.1	Prinzipielle Vorgehensweise beim klassischen Planen . . . . .	195
6.6.2	Realisierung . . . . .	196
6.6.3	Die Zustandsübergangsrelation $\gamma$ . . . . .	199
6.6.4	Suchalgorithmen für Planungsprobleme . . . . .	200
6.6.5	Ansätze zur Effizienzsteigerung . . . . .	202
6.6.6	Die GraphPlan-Repräsentation des Suchraums . . . . .	203
6.6.7	Heuristische Planextraktion . . . . .	208
6.6.8	Gegenüberstellung der Restriktionen des Klassischen Planens und der Anforderungen an Assistenzsysteme . . . . .	215
6.6.9	Bewertung des Ansatzes für Assistenzsysteme . . . . .	218
6.6.10	Zusammenhang zwischen <i>Concurrent Task Trees</i> und Klassischem Planen . . . . .	220
6.6.11	Planen und Planausführung in Assistenzszenarien . . . . .	232

6.7	Planung mit hierarchischen Task-Netzwerken . . . . .	233
6.7.1	Prinzipielle Idee . . . . .	234
6.7.2	Realisierung . . . . .	235
6.7.3	Zusammenhang zwischen <i>Concurrent Task Trees</i> und HTN-Planung . . . . .	238
6.7.4	Konfrontation mit den Anforderungen an Assistenzsysteme . . . . .	241
6.8	Ziele in Assistenzszenarien . . . . .	242
	Literatur . . . . .	252
<b>7</b>	<b>Planbasierter Dialog . . . . .</b>	<b>255</b>
7.1	Fallstudie: Aufnahme einer Sendung . . . . .	255
7.2	Diskussion des Plans . . . . .	258
7.3	Relaxation des Planungsproblems . . . . .	262
7.4	Hierarchisierung des Planungsproblems . . . . .	263
7.4.1	Das Problem mit dem Domänenwachstum . . . . .	263
7.4.2	Relaxation durch Versuch und Irrtum . . . . .	265
7.4.3	Durchführung der Verifikation . . . . .	267
7.5	Planausführung . . . . .	270
7.5.1	ALGO – eine Sprache zur Ausführung von Plänen . . . . .	271
7.5.2	MADL – eine Sprache zur Berechnung von Entscheidungen . . . . .	280
7.6	Simulation von Entscheidungen . . . . .	288
7.6.1	Domänenspezifische MADL-Prozeduren gegen domänenunabhängige Planungsheuristiken – Wettkampf oder Symbiose? . . . . .	289
7.6.2	Entscheidung über alternative Pläne . . . . .	290
7.6.3	Alternativen zum Einsatz von MADL bei der Konstruktion von Plänen . . . . .	294
7.7	Diagnose . . . . .	296
7.7.1	Fallstudie: Verpasst der Nutzer den Bus? . . . . .	297
7.7.2	Vergleich mit REITERS <i>Theory of Diagnosis from First Principles</i> . . . . .	300
7.8	Neuplanung . . . . .	303
7.8.1	Mehrere Ziele . . . . .	303
7.8.2	Relaxation von Zielen . . . . .	305
	Literatur . . . . .	306
<b>8</b>	<b>Task-Analysen in der Mensch-Maschine-Interaktion . . . . .</b>	<b>309</b>
8.1	Realisierung pragmatischer Vorgänge in der Mensch-Maschine-Interaktion . . . . .	309
8.2	Kollaboration und Interaktion als eigenständige Domänen . . . . .	316
8.3	Vergleich mit strukturellen Verfahren zur Dialoganalyse . . . . .	319
8.4	Fazit: Zur Verwandtschaft zwischen Assistenz- und Dialogsystemen . . . . .	324
	Literatur . . . . .	326

---

<b>Definition der Syntax von ALGO</b> .....	329
<b>Definition der Syntax von MADL</b> .....	339

Assistenzsysteme sollen ihre Nutzer bei der Durchführung einer zielgerichteten Tätigkeit durch Übernahme von Teilaufgaben der Tätigkeit unterstützen. Dazu ist in unterschiedlichem Umfang je nach Tätigkeit und Konzeption des Assistenzsystems Interaktion zwischen Mensch und Maschine notwendig. Viele aktuelle Ansätze, Mensch-Maschine-Interaktion zu verstehen, konzentrieren sich auf die Analyse von in Experimenten gesammelten Dialogcorpora, also Protokollen der Interaktionen zwischen Mensch und Maschine bei der kooperativen Durchführung einer Tätigkeit. Nicht protokolliert jedoch werden die gleichzeitig durchgeführten Tätigkeiten, die dazu verwendete Information und das zur erfolgreichen Durchführung erforderliche Wissen.

Auf diese Weise wird ignoriert, dass Interaktion nur ein Mittel zum Zweck ist, Informationen auszutauschen oder Kooperationspartner zu einer Handlung zu motivieren, die der Erreichung eines kooperativen Ziels, nämlich der erfolgreichen Durchführung einer Tätigkeit, dient. Betrachtet man nur die Interaktion, die während der (zweckrationalen) Kooperation geführt wird, bleibt unerklärlich, welche Gründe einen Kooperationspartner zu einem bestimmten Interaktionsschritt veranlassen. Diese liegen aber gerade in seiner Interpretation, wie weit die (kooperative) Durchführung einer Tätigkeit fortgeschritten ist.

Kooperatives Handeln und die dazu notwendige Interaktion sollen in dieser Monographie zusammengeführt werden. Dazu wurde ein Ansatz der Analyse von Mensch-Maschine-Interaktion entwickelt, der den der Interaktion zugrunde liegenden pragmatischen Prozess simuliert. Auf diese Weise können einzelne Interaktionsschritte aus dem Verständnis des Prozesses und der kognitiven Vorgänge heraus erklärt werden, mit Hilfe derer die Kooperationspartner diesen Prozess interpretieren und auf das gemeinsame Ziel ausgerichtet gestalten.

Zweck der Entwicklung des Ansatzes ist die Konzeption eines konfigurierbaren interaktiven Assistenzsystems, das den Nutzer durch Kooperation bei der Ausführung von Vorgängen in einem Anwendungsszenario unterstützt.

In Kap. 2 wird anhand typischer Beispiele recherchiert, welche Typen von Assistenzsystemen nach dem aktuellen Stand der Technik existieren. Komplementär dazu wird

untersucht, welche Formen von Assistenz Nutzer bei der Durchführung ihnen unvertrauter Vorgänge erwarten. Aus beiden Analysen wird ein Katalog von Anforderungen an Assistenzsysteme entwickelt, die den Nutzer interaktiv bei auf ein konkretes Ziel gerichteten Vorgängen unterstützen sollen.

Welche Rolle Interaktion bei der Ausübung von Assistenz spielt, ist Thema von Kap. 3. Es wird offenbar, dass Interaktion zwischen den Kooperationspartnern (in unserem Fall also zwischen Mensch und Maschine) nur ein Typ kognitiver Prozesse ist, die den Vorgang im Anwendungsszenario vorantreiben. In Dialogmodellen, die nur Sprechaktfolgen untersuchen oder von einem schematischen Ablauf der Interaktion für alle gleichartigen Vorgänge eines Anwendungsszenarios ausgehen, fehlt gerade die Analyse der (verborgenen) kognitiven Prozesse, die nicht anhand von Sprechhandlungen beobachtbar sind. Um dieses Defizit zu beheben, werden in Kap. 3 die kognitiven Prozesse typisiert, die zur Erfüllung der Anforderungen an Assistenzsysteme erforderlich sind. Aus dieser Typisierung wird abgeleitet, dass jede Implementierung eines Assistenzsystems heterogene Problemlöseverfahren integrieren muss.

Damit Problemlöseverfahren Ergebnisse ermitteln können, müssen sie mit domänenrelevantem Wissen konfiguriert werden. Bei interaktiver Assistenz besteht dieses Wissen in der Kenntnis über die von Nutzern durchgeführten Vorgänge, bei denen ein Assistenzsystem Unterstützung leisten soll. In einem noch nicht formalisierten Anwendungsszenario ist zunächst unklar, woher dieses Wissen kommen und wie es zur Konfiguration eines Assistenzsystems akquiriert werden kann. Als Lösung für diese Frage wird in Kap. 4 vorgeschlagen, auf das im *usability engineering* etablierte Modell der *Concurrent Task Trees* zurückzugreifen, mit denen Ergebnisse von so genannten Task-Analysen strukturell formalisiert werden können.

Damit *Concurrent Task Trees* von einem konfigurierbaren Assistenzsystem benutzt werden können, müssen sie in einer interpretierbaren Form vorliegen. In Kap. 5 wird gezeigt, wie *Concurrent Task Trees* in Programme des Situationskalküls, genauer gesagt in die GOLOG-Variante des Situationskalküls, übersetzt werden können. Diese formale Theorie des *reasoning about actions* liefert eine semantische Basis für die Interpretation von *Concurrent Task Trees*.

In Kap. 6 wird diskutiert, wie GOLOG-Programme auch bei unsicherem Wissen über die aktuelle Situation und den Ausgang von einzelnen Schritten des zu unterstützenden pragmatischen Prozesses eingesetzt werden können. Für die Anforderungen an Assistenzsysteme hat jedoch auch diese erweiterte Form von GOLOG ein wesentliches Manko: Es können zwar vorgegebene Programme interpretiert werden, aber damit wird nicht die Frage gelöst, wie ein Assistenzsystem ermitteln kann, welche Folge von Schritten einen pragmatischen Prozess bildet, der das zur Lösung einer Aufgabe zugehörige Ziel erreichen kann. Dazu muss das Assistenzsystem planen können. Auf Basis der GOLOG-Simulation von Prozessen wird erörtert, wie *Concurrent Task Trees* in Planungsdomänen übersetzt werden können. Die *Concurrent Task Trees* bilden damit die Nahtstelle zwischen der Modellierung von *user interfaces*, Wissensverarbeitung durch das Assistenzsystem und der Interaktion zwischen Nutzer und Assistenzsystem.

Anschließend werden verschiedene symbolische und entscheidungstheoretische Planungsverfahren verglichen, inwieweit sie für die Anforderungen an Assistenzsysteme geeignet sind. Als Fazit ergibt sich, dass symbolische und entscheidungstheoretische Planungsverfahren miteinander kombiniert werden müssen: symbolische Verfahren erlauben die Konstruktion von Handlungsfolgen, die spezifizierten Vorbedingungen und Effekten genügen, bereiten aber Schwierigkeiten bei der Verarbeitung von unsicherem Wissen. Entscheidungstheoretische Verfahren lösen unsicheres Wissen auf, indem sie Planung, Planausführung und Beobachtung der Umgebung integrieren, entscheiden aber nach stochastischen Kriterien, die verhindern, auf jede konkrete aktuelle Situation individuell eingehen zu können. Als Konsequenz aus dem Fazit wird ein Kontrollalgorithmus entwickelt, der mit Hilfe symbolischer Planungsverfahren eine Sequenz von Aktionen und Entscheidungen der Kooperationspartner plant, die das gemeinsame Ziel erreichen können.

Aktionen in einer geplanten Sequenz werden dabei unterschieden nach vom Nutzer auszuführenden und vom System auszuführenden: Erstere stellen aus der Perspektive des Assistenzsystems nichtdeterministische Aktionen dar und müssen dementsprechend simuliert werden. Letztere werden als „privater“ Vorgang des Assistenzsystems verstanden. Damit sind sie mit denselben Methoden formalisierbar wie *Concurrent Task Trees*. Die wesentliche Konsequenz aus dieser Analogie ist, dass erstens die Wissensbasis des Assistenzsystems modularisiert werden kann, und zweitens Systemaktionen als „lokales“ Planungsproblem interpretiert werden. Letztendlich wird damit das Leisten von Assistenz auf ein hierarchisches Planungs- und Planausführungsproblem zurückgeführt: Die Beobachtung der Umgebung bei der Ausführung eines Plans für das kooperative Ziel besteht darin, dass das Assistenzsystem für alle im Plan enthaltenen Systemaktionen stets überprüft, ob in der aktuellen Situation für sie ein „lokaler“ Plan existiert.

Bei der Planausführung werden im erstellten Plan enthaltene zukünftige Entscheidungen simuliert, indem domänenspezifische heuristische Regeln ausgewertet werden. Sie verallgemeinern die Auswahl nach dem Kriterium der größten Wahrscheinlichkeit zu einer Auswahl nach Multi-Attribut-Kriterien. Die Kriterien werden aus den während des Wissenserwerbs durchgeführten Task-Analysen abgeleitet. In Kap. 7 wird besprochen, wie die Kriterien als Regeln in einer speziellen Programmiersprache für Multi-Attribut-Probleme realisiert werden. Der Kontrollalgorithmus des Assistenzsystems greift auf diese Regeln zu, wenn während der Planausführung rationale Entscheidungen getroffen werden müssen. In diesem Zusammenhang dienen die Regeln als heuristische Optimierungsfunktion, mit deren Hilfe aus den vorhandenen Optionen eine optimale Auswahl getroffen werden kann, um unsicheres Wissen in der aktuellen Situation zu beseitigen.

In Kap. 8 werden der bisher entwickelte Kontrollalgorithmus und die damit verbundene Methodik des Wissenserwerbs dazu genutzt, die in Kap. 2 identifizierten kognitiven Prozesse für in vielen Anwendungsszenarien wiederkehrende Aufgaben bei der Durchführung von Interaktion zu modellieren. Bei der Verarbeitung gerade von natürlichsprachlichen Eingaben ergeben sich Aufgaben, die mit dem eigentlichen gemeinsamen Ziel nur insofern zu tun haben, als die Interaktion dazu dient, unsicheres Wissen aufzulösen. Da die

Sprachverarbeitung aber selbst mit Unsicherheit behaftet ist, entstehen in einem Sprachdialogsystem zusätzliche gemeinsame Ziele – nämlich gerade diese Unsicherheit zu beseitigen. Die dazu nötigen pragmatischen Vorgänge können mit denselben Mitteln formuliert werden wie die Vorgänge im Anwendungsszenario. Fazit dieser Beobachtung ist, dass ein konfigurierbares Assistenzsystem aufgrund der modularen Wissensrepräsentation zu natürlichsprachlicher Interaktion befähigt und damit als Basis für Sprachdialogsysteme eingesetzt werden kann. Damit ist der angekündigte neuartige Ansatz zur Analyse zweckrationaler Mensch-Maschine-Interaktion realisiert.



Gerade wegen der vielen Assoziationen aus dem Alltag ist die Idee der *Assistenz* sehr schwer zu fassen, einzuschränken und präzise zu definieren. In den folgenden Abschnitten soll daher versucht werden, den Begriff der Assistenz anhand einer Analyse der beim menschlichen Problemlösen entscheidenden Handlungsphasen zu erfassen. Grundsätzlich kann in jeder dieser Phasen Assistenz gewährt werden. Aus der Wirkung der Assistenz für den Problemlösevorgang leitet sich dann ihr Typ ab. Die verschiedenen Typen von technischen Assistenzsystemen sollen in dieser Einleitung an Beispielen vorgestellt werden. Aus dem Vergleich der verschiedenen Arten geleisteter Assistenz wird dann für die softwaretechnische Implementierung von Assistenz im Sinn der Taxonomie von WANDKE (siehe [1]) ein Forderungskatalog an ein für bestimmte Aufgaben konfigurierbares multimodales Assistenzsystem abgeleitet. Dessen Konzeption wird später beschrieben.

---

## 2.1 Typen von Assistenzsystemen

Den Begriff „Assistenz“ zur Charakterisierung einer speziellen Klasse von technischen Systemen aus dem umgangssprachlichen Verständnis des Wortes heraus zu definieren, führt nicht zum Erfolg. Bedeutet nämlich Assistenz die Zunahme von externen Fähigkeiten für die Lösung einer Aufgabe, so ist grundsätzlich jedes technische Gerät, insbesondere aber auch jedes Softwaresystem, ein Assistenzsystem. Schon ein Schraubenzieher eröffnet dem Nutzer Möglichkeiten, die er mit bloßen Händen vermutlich nicht hat. Ähnliches gilt für ein Telefaxgerät, mit dem der Nutzer eine Fotokopie an einen beliebigen Ort versenden kann, ohne sich eines Postboten bedienen zu müssen. Mit einer Rakete können Menschen sogar andere Planeten erreichen – ein technisches Werkzeug, das den menschlichen Handlungsspielraum deutlich erweitert. Die Beispiele machen deutlich, dass dieser Versuch, Assistenz zu definieren, keine Charakterisierung einer Systemklasse ermöglicht, weil zwischen Schraubenzieher und Raumfähre jedes technische Artefakt als Assistenzsystem charakterisiert werden kann.

Wandke [1] eröffnet in seiner Arbeit einen besseren Zugang zum Begriff „Assistenz“: Über eine Taxonomie von Assistenzsystemen zeigt er, wie ein Versuch zur Klassifikation von Assistenzsystemen aussehen könnte, der sich am Grad der Autonomie des analysierten Systems orientiert:

- Assistenz kann dadurch erreicht werden, dass einfache Funktionen automatisch, das heißt ohne Auslösung durch den Benutzer, ausgeführt werden. In diese Klasse fallen Systeme wie Autopiloten in Flugzeugen, Antiblockiersysteme in Kraftfahrzeugen oder automatische Abschaltungssysteme für technische Geräte in Haushalt oder Produktionsstätten.
- Die nächstkomplexere Klasse von Assistenz kombiniert verschiedene elementare, nicht notwendigerweise autonome Funktionen eines Geräts. Assistenz besteht in diesem Fall also darin, den Nutzer bei der Durchführung eines vorab definierten, nicht modifizierbaren Anwendungsfalls durch geeignete Bündelung von Funktionen für eine komplexe Aufgabe zu unterstützen. Typische Beispiele hierfür sind Installationsassistenten von Softwareprogrammen, die Kombination von verschiedenen Funktionen wie Telefon, Camera Radio, Navigationssystem und Mailclient in einem Mobiltelefon, automatische Heizungssteuerung in Wohnräumen, oder unterstützende Funktionen für behinderte Personen wie selbstfahrende Rollstühle, die durch einfache Fingerbewegungen gesteuert werden können, oder die Kommunikation von Information über sprachliche oder taktile Modalitäten.
- Die komplexeste Form von Assistenz versucht, die Intention des Nutzers einzelner durchgeführter Schritte zu erkennen, daraus die aktuelle Aufgabe, die gelöst werden soll, abzuleiten und geeignete Schritte vorzuschlagen, mit denen eine effiziente Lösung möglich ist. Typische Assistenten dieser Klasse sind Onlinehilfen für Softwaresysteme, die teilweise mithilfe animierter Charaktere ein anthropomorphes Aussehen annehmen.

In den folgenden Abschnitten soll jede dieser drei Klassen anhand eines Beispiels näher illustriert werden.

### 2.1.1 „Autonome“ Assistenz

Die historisch frühesten, am weitesten verbreiteten und in ihrer Wirkungsweise offensichtlichsten Assistenzsysteme stammen aus dem Bereich der Steuerung von technischen Geräten und Anlagen. Die fortschreitende Digitalisierung und Verbreitung von platzsparender Elektronik hat dazu geführt, dass viele Assistenzsysteme auch für private Nutzer finanzierbar und einsetzbar werden. Vor allem im Automobilbau ist die Idee des Assistenten sehr weit verbreitet. Benmimoun et al. [2] führen wichtige Fahrerassistenzsysteme auf. Fast alle dieser Assistenten zeichnen sich durch autonome Funktionalität aus. Die bekanntesten Beispiele unter ihnen sind: adaptives Kurvenlicht, Reifendruckkontrollsys-

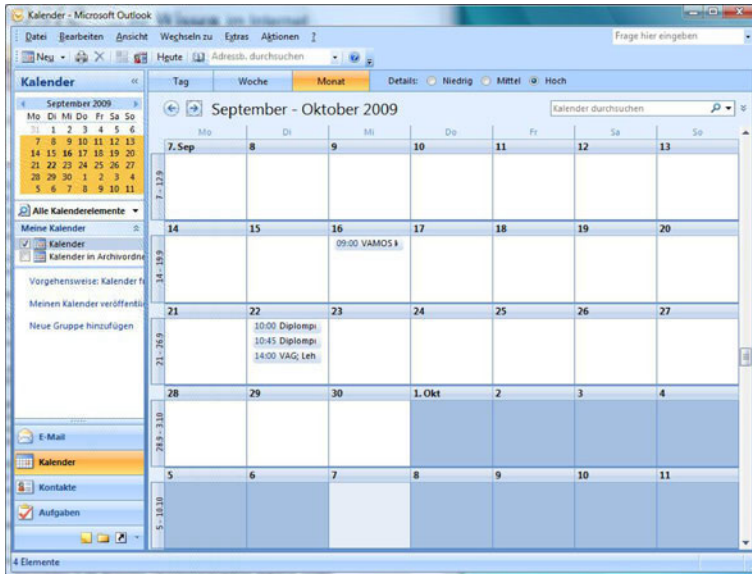
teme, Antiblockiersysteme, elektronische Stabilitätsprogramme, Kollisionswarnsysteme und seit kürzester Zeit auch Nachtsichtsysteme.

Assistenten dieses Typs führen immer genau eine Funktion aus und unterstützen damit den Fahrer bei der Kontrolle seines Fahrzeugs während der Fahrt. Die Assistenten verbessern die Möglichkeiten des Fahrers zur Aufnahme von Informationen und damit auch zur Erkennung der aktuellen Situation und tragen damit zur Erhöhung der Datensicherheit bei, indem sie die sensorischen Fähigkeiten des Fahrers ergänzen. Nach der Klassifikation von WANDKE leisten Fahrerassistenzsysteme also genau in den Handlungsweisen der Aufnahme und der Integration von Information Assistenz. Sie tragen jedoch nicht zur Entscheidung, zur Aktionsausführung und zur Kontrolle der Handlungseffekte bei. Manche der Systeme sind aktive Assistenzsysteme, beispielsweise das Antiblockiersystem, andere wiederum sind passiv wie etwa ein Nachtsichtgerät, das nur Informationen zur Verfügung stellt, also eine Anzeigefunktion wahrnimmt, aber keine weitere Assistenz leistet. Der Benutzer hat in der Regel keine Möglichkeit, das Assistenzsystem in seiner Funktionsweise wesentlich zu beeinflussen.

### 2.1.2 Assistenz mit einem fixierten Ziel

Komplexere Assistenzsysteme stellen mehrere Assistenzfunktionen gleichzeitig zur Verfügung. Typische Beispiele hierfür sind Funktionen in komplexen Softwaresystemen. Dazu zählen Funktionen zum Druck von Dateien, die automatisch die Kommunikation mit dem auf dem System installierten Druckertreiber übernehmen genauso, wie Systeme zur Unterstützung der Produktionsplanung und -steuerung. Auch Software, die komplette Abläufe in der Betriebswirtschaft organisiert, wie etwa die Softwarepakete von SAP, leisten Assistenzfunktion, und zwar bei der Ausführung von Aktionen und bei der Kontrolle von Effekten. Jedes Expertensystem, wie es aus der Künstlichen Intelligenz bekannt ist (siehe [3–5]), ist ein Assistenzsystem, da es bei Handlungen Entscheidungsunterstützung anbietet. Offensichtlich ist die Liste an Beispielen beliebig verlängerbar, ebenso wie die Liste an Handlungen, die Menschen mit der Hilfe von Werkzeugen durchführen.

Typisch für alle Assistenzsysteme mit fixiertem Ziel ist jedoch, dass sie nicht für andere als die vorgesehenen Aufgaben eingesetzt werden können. Der Grad an Konfigurierbarkeit des Systems ist sehr unterschiedlich, je nachdem, wie komplex das Assistenzsystem ist. Ein bekanntes Beispiel für einen Assistenten, der verschiedene Funktionen kombiniert, ist das Microsoft-Programm *Outlook* (siehe Abb. 2.1). Es integriert Verwaltung von E-Mails, Kontakte, zu erledigende Aufgaben und einen Terminkalender. Anders als autonome Assistenzsysteme eignen sich als Softwaresysteme realisierte Assistenten für beliebig komplexe und parametrisierbare Assistenzfunktionen. Ein weiteres Charakteristikum: viele Assistenzsysteme, die bei komplexen Abläufen Unterstützung leisten, kommunizieren mit dem Benutzer, um Informationen über die aktuelle Situation zu erhalten – ein unverzichtbarer Vorgang, wie folgendes Beispiel illustriert: ein Buchhaltungsprogramm kann nur in Aktion treten, wenn Buchungsvorgänge, die außerhalb der Aktivität des Pro-



**Abb. 2.1** Der Bildschirm des komplexen Assistenten *Outlook*

gramms entstehen, in irgendeiner Art und Weise eingegeben werden. Trotz der Mensch-Maschine-Interaktion ist die Assistenz mit einem fixierten Ziel in der Regel passiv, d. h. der unterstützende Ablauf und die damit verbundene Assistenz werden vom Nutzer angestoßen. Assistenzsysteme mit einem fixierten Ziel funktionieren also tatsächlich wie ein (unterschiedlich komplex einsetzbares) „Werkzeug“.

### 2.1.3 Assistenz durch Intentionserkennung

Eine interessante weitere Klasse von Assistenzsystemen bietet aktive Assistenz an und versucht dabei, die Aufgabe, bei der Benutzer Unterstützung benötigen könnten, anhand seiner Handlungen zu erkennen.

Ein weitverbreitetes Beispiel dafür ist der Office-Assistent, der aus dem Programmpaket Microsoft Office bekannt ist. Genau genommen besteht der Office-Assistent aus einer ganzen Familie von Assistenten, die für jeweils eigene Aufgaben spezialisiert sind<sup>1</sup>. Während die algorithmischen Einzelheiten der aktuellen Version des Office-Assistenten zu den Betriebsgeheimnissen von Microsoft zählen, ist ein Vorläufer-Prototyp in der Literatur beschrieben worden: Horvitz et al. [6] berichten, dass der Vorläufer mit dem Namen

<sup>1</sup> Details zu den Assistenten stehen auf der Webseite <http://office.microsoft.com/de-de> zu Microsoft Office (Letzter Aufruf der Seite: 15.06.2015)

*Lumière* während des Betriebs einer Office-Applikation versucht, aus den Aktionen des Benutzers seine aktuellen Ziele zu erschließen. Damit dies möglich wird, müssen die über die Benutzerschnittstelle beobachtbaren Aktionen wie das Bewegen der Maus, das Aufrufen eines Menüpunkts, das Aktivieren eines Fensters und auch Untätigkeit des Nutzers über einen gewissen Zeitraum in Bezug auf Handlungen, die mit Office-Applikation durchgeführt werden können, interpretiert werden. Die pragmatische Interpretation von Aktionen auf der Ebene der graphischen Benutzeroberfläche umfasst Themen wie Suche, Verschieben des Aufmerksamkeitsfokus, Nachdenken über die nächsten Schritte und das Rückgängigmachen erwünschter Effekte. Alle genannten Themen beschreiben Problemlösestrategien der Anwender für komplexe Bedienungsaufgaben. Die Strategien wurden experimentell aus der Analyse von vielen aufgezeichneten Sitzungen von Probanden mit *Lumière* gewonnen. Dabei mussten Probanden die Tabellenkalkulation Excel benutzen und konnten auf einem zweiten Bildschirm die Assistenzvorschläge eines neuartigen Systems betrachten und in ihre Vorgehensweise bei der Lösung der Aufgabe einbeziehen. Tatsächlich jedoch wurden die Vorschläge nicht von einem softwarebasierten Hilfesystem ermittelt, sondern von einem menschlichen Experten, einem so genannten *Wizard-of-Oz* (siehe [7]), der Probanden auf einem getrennten eigenen Monitor beobachten konnte. Aus diesen Studien konnten die Entwickler von *Lumière* Problemlösestrategien ungeübter Nutzer identifizieren, die von einem Experten viel einfacher und schneller gelöst werden können.

Auffällig an den Benutzerstudien war, dass Experten zwar die Fähigkeit haben, Ziele und Bedürfnisse von Nutzern bei der Lösung einer wichtigen Aufgabe anhand der von den Nutzern durchgeführten Aktionen zu identifizieren. Aber es zeigte sich auch, dass die Experten typischerweise bei Beginn einer Aktionssequenz sehr unsicher waren, welches Ziel der Nutzer gerade verfolgte. In dieser Phase konnten selbst Experten nur schlecht Assistenz leisten. Erst nach einigen Bedienschritten verbesserte sich die Situation, und die Vorschläge wurden präziser und zielgerichteter. Die Studien zeigten darüber hinaus, dass schlechte Assistenz den Benutzern erhebliche Schwierigkeiten bereitete. Obwohl sie davon unterrichtet waren, dass sich das Hilfesystem in einem experimentellen Zustand befand, nahmen sie alle Vorschläge sehr ernst und bezogen sie in ihre Problemlösung mit ein. Falsche Vorschläge sind aber offensichtlich sehr abträglich für die effiziente Lösung einer Aufgabe. Diese Ergebnisse machen sehr deutlich, dass ein wesentliches Kriterium für ein leistungsfähiges Assistenzsystem in seiner Fähigkeit besteht, Vorschläge für Handlungen auf ein Ziel hin zu orientieren.

Aus technischer Sicht besteht die Schwierigkeit, präzise Vorhersagen über das vom Benutzer verfolgte komplexe Ziel zu treffen, wenn erst sehr wenige Schritte bekannt sind, darin, dass einzelne Aktionen mehreren typischen Bedienzelen zugeordnet werden können. Dieser Nichtdeterminismus besteht schon bei der Zuordnung beobachtbarer Interaktionen zu einem komplexen Ziel. Wenn beispielsweise ein Nutzer mehrere Menüs kurz hintereinander aufruft, kann er sich eventuell unsicher sein, wie er eine bestimmte Programmfunktion aktivieren kann. Vielleicht will er aber auch bisher noch nicht

genutzte Funktionen entdecken – etwa deswegen, weil ihm nicht klar ist, mit welchen Schritten eine komplexe Aufgabe gelöst werden kann. Bei der Entwicklung von *Lumière* wurde daher versucht, diese Zusammenhänge mit Wahrscheinlichkeiten zu beschreiben.

Die Absichten des Nutzers rechtzeitig zu erkennen, ist, wie Studien mit Benutzern bei Microsoft ja gezeigt haben, ein entscheidender Faktor dabei, ob ein Assistenzsystem in der Lage ist, Nutzer dabei zu unterstützen, mithilfe eines Softwarewerkzeugs ihnen gestellte Aufgaben effizient zu lösen. Absichten des Nutzers zu erkennen, aber nicht überprüfen zu können, ob die von Nutzern gewählten Schritte zielführend sind, wäre absurd. Somit ist die Fähigkeit eines Assistenzsystems, Wege zu finden, wie in einer gegebenen Situation eine Aufgabe gelöst werden kann, eine ebenso wichtige Eigenschaft eines leistungsfähigen Assistenzsystems. Natürlich kann ein Assistenzsystem nicht jede denkbare Aufgabe erkennen, sondern muss sich darauf beschränken, bestimmte Aufgabenklassen, für die das Softwarewerkzeug ein Hilfsmittel sein kann, zu unterstützen. Bei der Konzipierung von Assistenz ist also sowohl der Abstraktion bei der Handlungsphase der Zielbildung wie auch der Ausführung von Aktionen eine Grenze gesetzt. Sie hängt davon ab, welche Art von Assistenz zur Verfügung gestellt, und wie detailliert Umsetzung der Assistenz in konkreten Situationen realisiert werden soll.

---

## 2.2 Assistenz aus der Sicht des Nutzers

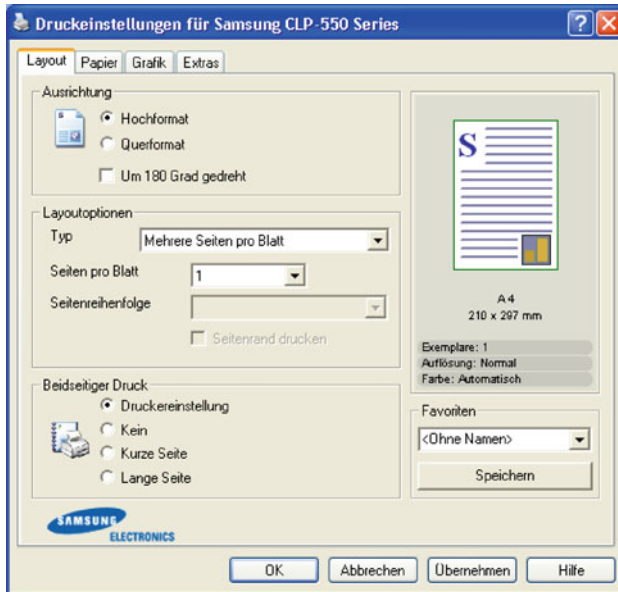
Die Beispiele zeigen deutlich, dass eine Definition von Assistenz über das Spektrum der Funktionen, deren Durchführung ein Gerät oder System für den Nutzer erledigen soll, um damit eine Aufgabe zu lösen, wegen der endlosen Zahl an Möglichkeiten unmöglich ist.

Eine brauchbare Taxonomie darf aber nur eine endliche, ja sogar möglichst kleine Zahl von Unterscheidungen benötigen, um ein überschaubares Begriffssystem zu konstruieren. Aus diesem Grund schlägt Wandke [1] vor, eine Taxonomie von Assistenzsystemen auf einem anderen Ordnungsprinzip als der Funktionsvielfalt aufzubauen. Der entscheidende Punkt liegt laut WANDKE darin, ob ein Benutzer die von einem System angebotenen Funktionen überhaupt einsetzen kann beziehungsweise sie einzusetzen im Stande ist. Dies kann aus verschiedenen Gründen schwierig sein:

- Die vom Nutzer gewünschte Funktion steht überhaupt nicht zur Verfügung.
- Dem Nutzer sind nicht alle zur Verfügung stehenden Funktionen bekannt.
- Der Nutzer ist nicht in der Lage, eine Funktion durchzuführen, weil ein zu hoher sensorischer, kognitiver oder motorischer Aufwand erforderlich ist.

Die Zugänglichkeit von Funktionen ist aber von praktischer Relevanz, weil sie ermöglicht, ein technisches System vollständig, effektiv und effizient einzusetzen.

Assistenz hat also vorrangig die Aufgabe, die Kluft zwischen Systemfunktionalität und Fähigkeiten des menschlichen Nutzers zu überbrücken, d. h. also zu erreichen, dass



**Abb. 2.2** Dialogfenster eines Druckertreibers: Mit Hilfe graphischer Mittel kann bei der Gestaltung von Mensch-Maschine-Schnittstellen dem Nutzer Assistenz bei der Interpretation interner Parameter in Bezug auf eine zu lösende Aufgabe geboten werden

der Nutzer die im Rahmen seines zielgerichteten Handelns notwendigen Funktionen des Systems mit möglichst geringem Aufwand einsetzen kann<sup>2</sup>. Ein Beispiel dafür ist die in Abb. 2.2 zu sehende Schnittstelle, die Assistenz bei der Ansteuerung eines Druckers leistet.

Dass der Nutzer tatsächlich zielgerichtet handelt, ist eine Grundannahme für die Entwicklung interaktiver Assistenzsysteme. Sie lässt sich aus der einem Werkzeug – und genau das ist ja ein technisches Gerät oder Softwaresystem – inhärenten Existenzberechtigung ableiten:

- Ein Schraubenzieher dient dazu, eine Schraube in ein vorgebohrtes Loch hineinzudrehen, um zwei Werkstücke miteinander zu verbinden.
- Ein Flugzeug dient dazu, lange Distanzen in kurzer Zeit zu überwinden.
- Ein Textverarbeitungssystem dient dazu, Text einzugeben, zu formatieren, graphisch zu gestalten und abzuspeichern bzw. zu vervielfältigen.

Aus diesen Beispielen lässt sich verallgemeinern, was in dieser Arbeit unter einer *Aufgabe* und dem *Lösen einer Aufgabe* verstanden wird.

<sup>2</sup> Siehe dazu [8], Kap. 11 und [9].

**Aufgabe (für die ein Assistenzsystem Unterstützung anbieten kann)**

Seit Einführung des GOMS-Modells (*goals, operators, methods, and selection rules*) durch CARD, MORAN und NEWELL in [10] wird die kognitive Fragestellung, wie ein gegebener Zustand eines Systems durch Ausführung von Aktionen in einen gewünschten Zustand überführt werden kann, **Aufgabe** genannt.

**Lösen einer Aufgabe (Problemlösung)**

Das **Lösen einer Aufgabe** besteht in der Ermittlung einer Folge von Aktionen, deren Ausführung die beabsichtigte Zustandsüberführung herstellen kann, und in der Ausführung der ermittelten Folge von Aktionen unter Einbeziehung technischer Geräte bzw. Systeme als Werkzeuge.

Der zentrale Punkt in dieser Definition ist, dass der menschliche Nutzer für das Stellen der Aufgabe verantwortlich ist, und jedes eingesetzte technische Gerät daher nur eine untergeordnete Rolle als Hilfsmittel spielt. Die menschliche Problemlösekompetenz gibt also den Ausschlag dabei, wie die Lösung einer Aufgabe versucht werden soll. Für die Effizienz und Qualität der Lösung ist entscheidend, wie gut der Nutzer die verfügbaren Funktionen der einsetzbaren Hilfsmittel in den Lösungsablauf integrieren kann. Die Definition beschreibt also keine Systeme, die vollautomatisch funktionieren. Wesentlich ist aber, dass Assistenz in der Lage ist, sich an menschlichem Problemlösen und den dazugehörigen Problemlösestrategien zu orientieren, weil nur auf diese Weise sichergestellt werden kann, dass das technische System und ein dazugehöriges Assistenzsystem harmonisch in den Ablauf des Lösungsvorgangs integriert werden können<sup>3</sup>.

Bei der zweckrationalen Lösung von Aufgaben nehmen nach übereinstimmenden Analysen in der Literatur zur kognitiven Ergonomie (siehe [1]) sechs sich ergänzende Aspekte eine zentrale Stellung ein:

- **Motivation, Aktivierung und Zielsetzung**

Ohne Zielsetzung ist keine Aufgabe definiert; dass dieser Aspekt wichtig ist, leuchtet also unmittelbar ein. Motivation und Aktivierung sind jedoch zwei weitere wichtige Voraussetzungen für die Durchführung von – oft komplizierten – Aufgaben, die mit Hilfe von technischen Systemen gelöst werden sollen. Aktivierung spielt beispielsweise eine wesentliche Rolle bei der Überwachung von prozesstechnischen Anlagen. Im Regelfall läuft alles nach Plan, aber bei Störungen ist es für eine Minimierung von Folgeschäden ausschlaggebend, wie rasch das Personal auf die Störung aufmerksam wird, wie präzise der Gefährlichkeitsgrad eingeschätzt werden kann, und wie effizient

<sup>3</sup> Siehe [8], Kap. 11, [11], Kap. 10, 14, und 15 sowie [10].



Gegenmaßnahmen ergriffen werden können. Immer wieder wird bei größeren technischen Störungen deutlich, dass sowohl zu niedrige als auch zu starke Aktivierung zu schwerwiegenden Konsequenzen führen, weil entweder gar keine oder – oft aus Panik – übertriebene oder sogar falsche Gegenmaßnahmen ergriffen werden. Die Kollision eines DHL-Transportflugzeugs mit einer russischen Tupolew Tu-154M der Bashkirian Airlines am 1. Juli 2002 ist ein Beispiel für die Schwierigkeit und Bedeutung der Aktivierung: weil der Fluglotse mit einem Landeanflug auf den Flughafen Friedrichshafen beschäftigt war, übersah er, dass beide Flugzeuge dieselbe Höhe hatten, und eine Kollision bevorstand.

- **Wahrnehmung**

Damit der Nutzer während der Bearbeitung einer Aufgabe zu jeder Zeit richtig reagieren kann, muss er Zugang zur relevanten Information haben. Die Wahrnehmung aller wichtigen Fakten zur richtigen Zeit kann oft schwierig sein, weil entweder der Nutzer mit der Verarbeitung anderer Signale überlastet ist (wie eben auch im Beispiel oben), oder Signale gar nicht oder nur schlecht wahrgenommen werden können (z. B. interner Zustand technischer Geräte, Hindernisse auf der Straße im Dunklen).

- **Integration von Information unter Berücksichtigung der aktuellen Situation**

Die bloße Wahrnehmung elementarer Signale reicht nicht aus, um eine Situation *interpretieren* zu können. Die Signale müssen mit Information aus dem (Langzeit)-Gedächtnis des Nutzers über geltende Tatsachen in Beziehung gebracht werden. Diese Interpretation muss auch erlauben, eine Bewertung der Zielorientierung vorzunehmen. Ein einfaches Beispiel dafür ist die Signalisierung einer Geschwindigkeitsüberschreitung durch ein Navigationssystem. Interpretation von Signalen erfordert also immer Bezugnahme auf Welt- bzw. Domänenwissen (der Messwert des Tachometers hat die Einheit km/h; damit wird eine Geschwindigkeit ausgedrückt; eine Geschwindigkeit von 70 km/h ist innerorts unzulässig, eine Geschwindigkeit von 200 km/h in einer engen Rechtskurve wird so große Fliehkräfte erzeugen, dass das Auto ins Schleudern geraten wird).

- **Fällen von Entscheidungen, Auswahl von Aktionen**

Aus der Interpretation von Information muss zur Lösung einer Aufgabe in zielorientierter Weise eine Reihe von Handlungen abgeleitet werden, mit deren Hilfe die Aufgabe gelöst werden kann – das heißt also, der Nutzer muss Entscheidungen über sein weiteres Vorgehen treffen und diese Entscheidungen auch umsetzen. Bei der Ermittlung erfolgversprechender Handlungen ist Assistenz besonders hilfreich. Assistenz kann darin bestehen, alle sinnvollen Entscheidungsoptionen anzubieten, aus allen Optionen ein(ig)e geeignete herauszufinden oder eine (optimale) Aktion autonom auszuführen. Diese Form von Assistenz sorgt für den denkbar höchsten Automatisierungsgrad.

- **Ausführung von Aktionen**

Werden Aktionen nicht autonom vom Assistenzsystem ausgeführt, so kann auch bei der Ausführungen von Aktionen Assistenz geleistet werden. Beispiele dafür sind Bremsassistenten oder elektronische Vorrichtungen zur Motordrosselung bei Erreichen der erwünschten Höchstgeschwindigkeit in besonders stark motorisierten Autos. Allgemein

geht es bei diesem Assistenztyp meistens um die „Feindosierung“ bei der Steuerung von Aktoren oder um die Vereinfachung der Mensch-Maschine-Interaktion (beispielsweise Spracheingabe oder multimodale Eingabe für Behinderte).

- **Verarbeitung von Effekten der Aktionen und *feedback* darauf**

Die Effekte von Handlungen sind oft durch Wahrnehmung beobachtbar. Zur Kontrolle der Wirkung einer Aktion ist es entscheidend, überprüfen zu können, ob diese Wahrnehmungen mit der erwarteten Wirkung der Handlung übereinstimmen. Nicht immer ist eine direkte Wahrnehmung möglich, über die auf Erfolg oder Misserfolg einer Handlung geschlossen werden kann. Assistenz in dieser Handlungsphase bezieht sich also oft auf die Erweiterung der Möglichkeiten zur Wahrnehmung (Ist ein Datum in einem Gerät abgespeichert worden, Ist beim Einparken der Abstand zur Hausmauer noch ausreichend?). Eine erweiterte Assistenz zur Effektkontrolle bewertet die Auswirkungen von Handlung bezüglich einer zu lösenden Aufgabe.

In Tab. 2.1 wird ein Überblick über die verschiedenen Handlungsphasen, die damit verbundenen Aspekte eines Problemlöseprozesses und dafür vorstellbare Typen von Assistenz gegeben. Die Tabelle fasst die von Wandke [1] erstellte Taxonomie von Assistenzfunktionen zusammen. WANDKE legt den Fokus seiner Untersuchungen auf die Frage, welche Typen von Assistenz es grundsätzlich gibt, ohne Aspekte der Implementierung von Assistenzsystemen auf der Basis von Software zu erörtern. Dies ist angebracht, da Assistenzsysteme, wie die bereits besprochenen Beispiele deutlich machen, nicht nur im Bereich von Softwarelösungen vorstellbar sind. Historisch gesehen ist es sogar genau andersherum: Assistenzsysteme gab es zunächst bei technischen, insbesondere großen technischen Geräten und Apparaturen. Für die Künstliche Intelligenz steht hingegen die Frage im Vordergrund, wie man die vorgeschlagenen Assistenztypen algorithmisch realisiert könnte. Diese Frage stellt das Kernproblem dar, das in diesem Buch detailliert zu besprechen sein wird. In Abschn. 2.4 werden zunächst noch sehr allgemein, ohne auf algorithmische Details einzugehen, Anforderungen an die Algorithmik eines Assistenzsystems aufgeführt, die sich aus WANDKES Typen und einigen weiteren Anforderungen, die im Folgenden entwickelt werden, ableiten lassen.

Bevor dies jedoch geschieht, ist ein weiterer wichtiger Aspekt zu betrachten. Es handelt sich darum, *wann* ein Assistenzsystem Assistenz leisten soll: Nach Nitschke [12] hat jeder der Aspekte aus Tab. 2.1, sobald er in einem Assistenzsystem verfügbar ist, noch zwei Dimensionen, wie er ausgestaltet sein kann:

- **Initiative:** Die Assistenz kann *passiv* (also nur auf explizite Anforderung des Nutzers) oder *aktiv* (also auch autonom) ausgeführt werden.
- **Anpassbarkeit:** Die Assistenzfunktionen in einem System können *statisch* (d. h. nicht veränderbar), *adaptierbar* (d. h. durch den Nutzer parametrisierbar) oder *adaptiv* (d. h. vom System selbst anhand von Informationen aus dem Kontext adaptierbar) sein.

**Tab. 2.1** Taxonomie für Assistenzfunktionen. Jede Phase einer zielorientierten Handlung benötigt eine spezielle Form der Unterstützung, die von einem Assistenzsystem angeboten werden kann

<b>Motiv- und Zielbildung</b>	
Schaffung eines optimalen Aktivierungsniveaus	Aktivierungsassistenz
Verstärkung eines Motivs	Coach-Assistenz
Hemmung eines Motivs	Warn-Assistenz
Anregung eines Zielwechsels	Orientierungsassistenz
<b>Informationsaufnahme</b>	
Bereitstellung von Signalen	Anzeigefunktion
Signalverstärkung	Verstärkungsassistenz
Erzeugung von Redundanz	Wiederholungsassistenz
Transformation von Signalen in andere Modalitäten	Präsentationsassistenz
<b>Integration von Information, Berücksichtigung der aktuellen Situation</b>	
Bereitstellung von Erklärungen	Beschriftungen, Anleitungen, Hilfetexte
Bereitstellung externer Bezugssysteme	Übersetzungsassistenz
Erklärung von Systemausgaben	Erklärungsassistenz
<b>Entscheidung über Auswahl einer Aktion</b>	
Information über alle Optionen	Angebotsassistenz
Information über ausgewählte Optionen	Filterassistenz
Vorschlag einer Option	Beraterassistenz
Vorschlag und Ausführung, wenn der Nutzer zustimmt	Delegationsassistenz
Vorschlag und Ausführung, wenn der Nutzer nicht widerspricht	Übernahmeassistenz
Ausführung mit Information an den Nutzer	Informierende Ausführungsassistenz
Ausführung ohne Information an den Nutzer	Stille Ausführungsassistenz
<b>Aktionsausführung: Wie soll die Aktion durchgeführt werden?</b>	
Verstärken von Aktionen	<i>Power</i> -Assistenz
Verkürzen einer Aktionsfolge	<i>Short cut</i> -Assistenz
Alternative Modalitäten bereitstellen	Eingabeassistenz
<b>Effektkontrolle</b>	
Auswirkungen wahrnehmbar machen	Rückmeldungsassistenz
Grad der Zielerreichung bewerten	Kritikassistenz

## 2.3 Anforderungen an Assistenzsysteme

Die Frage, wann Assistenz zu leisten ist, lässt sich ähnlich schwierig beantworten, wie die Fragen *wozu* und *wie*, die am Beispiel des *Lumière*-Projekts bereits erörtert wurden: Denn auch die Auswahl des richtigen Zeitpunkts ist eine Entscheidung, die von unsicherer Information über den Nutzer bestimmt wird. Ein Assistenzsystem muss dabei zwei Aspekte unter einen Hut bekommen: einerseits die Integration von Wahrnehmungen in den Entscheidungsprozess, obwohl sie oft zu mehrdeutigen Interpretationen Anlass

geben. Andererseits ist das Assistenzsystem gezwungen, eine Assistenzstrategie umzusetzen, die zu einer eindeutig festgelegten nächsten Handlung führt, aber auch Kriterien der Adäquatheit von Assistenz und Kooperation berücksichtigt (siehe [8], Kap. 11). Dieser immer präsente Umstand beeinflusst die algorithmische Umsetzung der in Tabelle 2.1 aufgeführten Assistenzfunktionen. Um in dieser Situation dennoch einer algorithmischen Realisierung von Assistenz näher zu kommen, soll im Folgenden diskutiert werden, welche Inferenz-Fähigkeiten ein Assistenzsystem besitzen muss, um Assistenzfunktionen samt der dafür notwendigen Entscheidungen über Zeitpunkt, Zweck sowie Art und Weise ihrer Durchführung ausführen zu können.

### 2.3.1 Interaktivität

Auch wenn die Interpretation von Beobachtungen nicht eindeutig ist, muss sich ein Assistenzsystem für eine Hypothese entscheiden, kann sich aber nicht sicher sein, welche die richtige ist. Wie kann es diesem Dilemma entgehen? Eine Möglichkeit ist, die aktuelle Situation wiederzuerkennen und sich zu merken, wie oft in dieser Situation bei Ausführen einer bestimmten Handlung welcher Effekt eingetreten ist. Wenn das Assistenzsystem nun die Wahrnehmung so interpretiert, dass es denjenigen Effekt als gegeben ansieht, der bisher unter identischen Umständen am häufigsten aufgetreten ist, trifft es eine rationale Entscheidung, indem es die Erwartung maximiert, die richtige Hypothese ausgewählt zu haben.

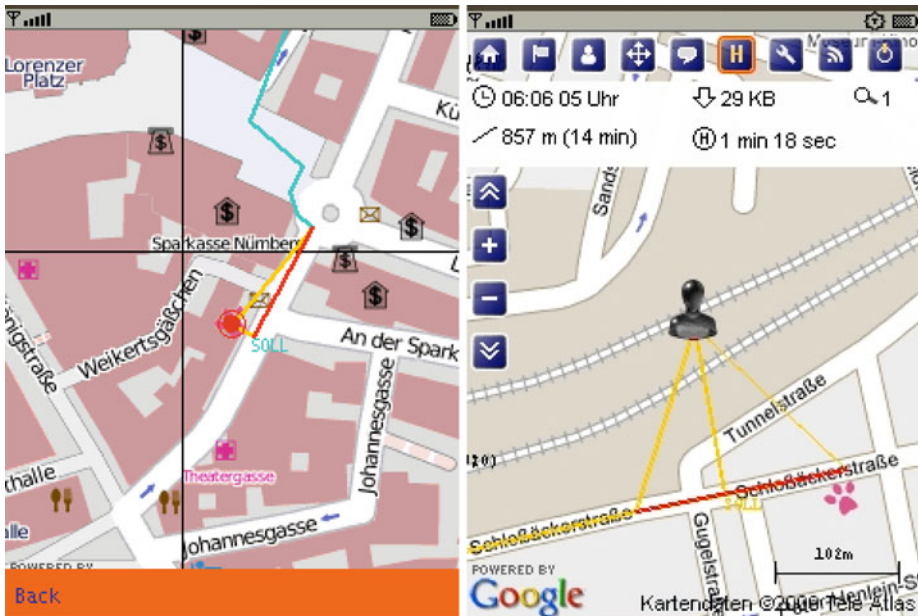
Die beschriebene Strategie soll am Beispiel des Fußgängernavigationssystems ROSE<sup>4</sup> diskutiert werden. ROSE aktualisiert während der Navigation zu einem Ziel ständig ein Benutzermodell, mit dessen Hilfe das aktuelle Interesse des Nutzers abgeschätzt werden soll, dass er weiterhin Assistenz zum Erreichen seines Ziels benötigt. Für den Interessen-Status des Nutzers werden dabei folgende vier möglichen Werte angenommen:

- *c*: Das Ziel soll noch erreicht werden.
- *i*: Der Nutzer möchte zwischenzeitlich ein anderes Ziel erreichen.
- *a*: Der Nutzer hat das Ziel aufgegeben.
- *e*: Der Nutzer wird von einem äußeren Umstand aufgehalten.

Hypothesen über den Status gewinnt ROSE, indem es aus Wahrnehmungen inferiert. Wie beim Office-Assistenten besteht auch in dieser Anwendung die Schwierigkeit, dass die Wahrnehmungen keine eindeutigen Schlüsse auf den Status zulassen. ROSE hat nämlich im Wesentlichen nur die Möglichkeit, die GPS-Koordinaten für den aktuellen Standort des Nutzers festzustellen. Aus dieser Information lassen sich unter Berücksichtigung einer Zeitreihe von GPS-Koordinaten und Karteninformation über den aktuellen Standort folgende Daten ermitteln:

---

<sup>4</sup> Das am Lehrstuhl für Künstliche Intelligenz entwickelte System ist u. a. in [13] beschrieben.



**Abb. 2.3** Präsentationsassistenz des mobilen Auskunftssystems ROSE. Der linke *screen shot* zeigt, wie ROSE GPS-Daten in eine für den Nutzer leichter erfassbare Modalität übersetzt und mit Zusatzinformation über nahe liegende *points of interest* anreichert. Der rechte *screen shot* illustriert eine Warnassistenzfunktion von ROSE: auf dem Bildschirm wird graphisch verdeutlicht, dass der Nutzer sich nicht mehr auf dem direkten Weg zum Ziel befindet

- Ist der Nutzer gerade in Bewegung oder steht er an einem Ort?
- Befindet sich in der Nähe des Nutzers ein für ihn interessanter Ort?

ROSE zeigt diese Information für den Nutzer auf dem Display seines Mobilgeräts an, wie im linken *screenshot* in Abb. 2.3 zu sehen. Über diese reine Präsentationsassistenz hinaus kann ROSE auch noch ermitteln, ob sich der Nutzer auf dem Weg zum Ziel befindet oder nicht (siehe rechten *screenshot* in Abb. 2.3).

Desweiteren kann ROSE den Nutzer durch geeignete Hinweise auf seinem Weg zum Ziel unterstützen. Die Effekte dieser Form von Assistenz kann ROSE auch nur mittelbar erschließen: wenn der Nutzer sich ungefähr auf dem Weg zum Ziel befindet, hat er die Hinweise wohl verstanden; ist die Abweichung groß, dann gibt es ein Missverständnis zwischen ROSE und dem Nutzer. Es sei denn, es ist anzunehmen, dass der Nutzer gar kein Interesse mehr am Ziel hat.

Der eben beschriebene Zusammenhang zeigt, wieso das oben eingeführte Nutzermodell für ROSE so bedeutsam ist: wenn ROSE nämlich ein Missverständnis oder eine Änderung in den Interessen des Nutzers vermutet, ist es nicht sinnvoll, die Effektkontrolle weiter durchzuführen. Vielmehr muss ROSE andere Assistenzfunktionen aktivieren: die

Ziel- und Motivbildung spielt in diesem Zustand wieder eine Rolle. Wenn ROSE nämlich vermutet, dass der Nutzer das Interesse am bisherigen Ziel aufgegeben hat, ist Orientierungsassistentz vorrangig: soll das Ziel gewechselt werden? Wird der Nutzer vermutlich aufgehalten, ist Warnassistentz angebracht: der Nutzer muss daran erinnert werden, dass er zu einem bestimmten Zeitpunkt (bei ROSE meist vor Abfahrt des bei der Zielbildung ausgewählten Nahverkehrsmittels) am Ziel ankommen muss.

Die Diskussion verdeutlicht, dass eine falsche Hypothese über den Interessen-Status des Nutzers dazu führen kann, dass ROSE sämtliche folgenden Handlungen des Nutzers falsch interpretiert, und alle seine Versuche zu assistieren, nur zu Verwirrung führen. Andererseits aber kann ROSE anhand seiner autonom erfassbaren Wahrnehmungen nie so viel Information ansammeln und damit die Unsicherheit über den Interessen-Status so weit reduzieren, dass eine Fehlinterpretation fast sicher ausgeschlossen ist.

Der einzige Ausweg aus diesem Dilemma besteht im Nachfragen beim Nutzer. ROSE muss also in der Lage sein, mit dem Nutzer zu kommunizieren, um sinnvoll Assistenz leisten zu können.

Wie schwierig die Analyse des Interessen-Status nur aufgrund autonom erfasster Wahrnehmungen ist, veranschaulicht das folgende Beispiel für die Implementierung des ROSE-Benutzermodells.

In Abb. 2.4 ist ein Ausschnitt aus dem Benutzermodell von ROSE zu sehen<sup>5</sup>: Um Hypothesen für den Interessen-Status des Nutzers zu gewinnen, bedient sich ROSE eines BAYES-Netzes<sup>6</sup>. In das Netz geht einerseits über Sensoren erfassbare Information ein:

- *Lokalisierung*: Damit wird eine Information bezeichnet, die aus den GPS-Daten ermittelt werden kann: ist der Nutzer in Bewegung oder nicht?
- *POI nahe*: Auch dies ist eine binäre Variable; sie ist wahr, wenn sich in der Nähe der aktuellen Position des Nutzers ein für den Nutzer relevanter *point of interest* befindet, und andernfalls falsch.

Im BAYES-Netz sind andererseits aber auch weitere Größen enthalten, die nicht direkt beobachtbar sind, aber einen Einfluss auf den Interessen-Status des Nutzers haben können:

- *unerwartetes Ereignis*: Während der Nutzer unterwegs zum Ziel ist, ereignet sich ein Vorfall, der den Nutzer darin hindert, weiterzugehen: Z. B. ein unübliches Ereignis auf der Straße; der Nutzer kann stürzen; der Nutzer kann gezwungen sein, einen Umweg zu nehmen. Die Liste möglicher Vorfälle kann beliebig verlängert werden. Allen unerwarteten Ereignissen ist jedoch gemeinsam, dass sie den Nutzer auf dem Weg zum Ziel aufhalten.

<sup>5</sup> Alle BAYES-Netze in diesem Kapitel wurden mit Hilfe des unter <http://aispace.org/bayes/help/> verfügbaren Tools erstellt, das in [14] beschrieben ist (Letzter Aufruf der genannten Webseite am 03.01.2015).

<sup>6</sup> BAYES-Netze werden ausführlich in [15], [16] und [17] dargestellt.