Hendrik Zender

Situated Production and Understanding of Verbal References to Entities in Large-Scale Space

SAARBRÜCKEN DISSERTATIONS
IN COMPUTATIONAL LINGUISTICS AND LANGUAGE TECHNOLOGY

VOLUME 36

Deutsches Forschungszentrum
für Künstliche Intelligenz

German Research Center for
Artificial Intelligence

UNIVERSITÄT
DES
SAARLANDES
Saarland University

Dissertation zur Erlangung des akademischen Grades eines
Doktors der Philosophie der Philosophischen Fakultäten
der Universität des Saarlandes

Saarbrücken, 2011

Tag der letzten Prüfungsleistung: 6. Dezember 2010
Dekan der Philosophischen Fakultät II: Prof. Dr. Erich Steiner
Berichterstatter: Prof. Dr. Hans Uszkoreit, Prof. Dr. Massimo Poesio

ISBN: 978-3-933218-35-3
ISSN: 1434-9353

Abstract

"The robots are coming!"
– "I hope they know where they are going…and what they can and cannot do there!"

The work presented in this thesis addresses the fundamental questions of how machines, such as robots and other autonomous agents, can acquire a mental representation of their environment that allows them to (a) act and navigate in it, and (b) communicate about it with humans in natural language.

We specifically investigate representations of structured environments that cannot be apprehended as a perceptual whole (i.e., *large-scale space*). This comprises, for instance, indoor domestic environments, or building ensembles. By that, the presented work goes beyond situated natural language interaction about an agent's immediate surroundings (i.e., *small-scale space*), such as table-tops or single room spaces.

Situated communication about *entities* – that is, things, places, properties, and events – in large-scale space requires the interlocutors to draw attention to entities that are not currently observable, and to comprehend which remote places and things are being talked about.

We furthermore show how such representations that can be used for spoken interaction with human users also endow autonomous agents with skills for context-aware planning and execution of actions in structured environments that are made by and for humans. To this end, the presented spatial models have been implemented and deployed in integrated systems for intelligent mobile robots.

We then present an approach for natural language generation and understanding that makes use of the acquired spatial models. It allows an agent to successfully *generate and resolve natural language expressions* that refer to entities in large-scale space. The approach is backed by observations from an empirical spoken language production experiment. The thesis concludes with a discussion of ongoing work to transfer the models made for intelligent mobile robots to autonomous virtual agents that act in an online virtual 3D world.

Ausführliche Zusammenfassung

"Die Roboter kommen!"
– "Ich hoffe, sie wissen, wo sie hinwollen…und was sie dort tun können – und was nicht!"

Die vorliegende Arbeit thematisiert die grundlegende Frage, wie Maschinen, wie etwa Roboter oder andere autonome Agenten, ein mentales Bild ihrer Umgebung erwerben können, mithilfe dessen sie (a) in ihrer Umgebung agieren und navigieren, und (b) mit Menschen in natürlicher Sprache über diese Umgebung kommunizieren können.

Insbesondere untersuchen wir Repräsentationen von strukturierten Umgebungen, die nicht unmittelbar als Ganzes wahrgenommen werden können, so genannte *großräumige Umgebungen* (engl. 'large-scale space'). Diese umfassen unter anderem häusliche Umgebungen oder Gebäudekomplexe. Dadurch geht die vorliegende Arbeit einen Schritt weiter als bisherige Modelle für die situierte Interaktion mit Agenten, welche in der Regel auf die unmittelbare Umgebung, d.h. so genannte *kleinräumige Umgebungen* (engl. 'small-scale space'), eines Agenten beschränkt waren. Im Gegensatz zu großräumigen Umgebungen können kleinräumige Umgebungen, zu denen üblicherweise einzelne Zimmer oder auf die Oberfläche eines Tisches begrenzte Szenen zählen, unmittelbar als Ganzes wahrgenommen werden. Die *situierte* Kommunikation über *Dinge* – weit gefasst: Objekte, Orte, Eigenschaften oder Ereignisse – in großräumigen Umgebungen verlangt von Gesprächspartnern einerseits, dass sie die Aufmerksamkeit ihrer Zuhörer auf Dinge lenken, die nicht unmittelbar wahrnehmbar sind, und dass sie andererseits verstehen, über welche an anderer Stelle befindlichen Dinge gerade gesprochen wird.

In dieser Arbeit zeigen wir, wie Raummodelle von strukturierten, von Menschen für Menschen geschaffenen Umgebungen, die zur situierten gesprochenen Interaktion mit Menschen dienen, von autonomen Agenten zur kontextbewussten Planung und Ausführung von Aktionen benutzt werden können. Dazu haben wir die räumlichen Modelle in integrierten Systemen implementiert, die für den Einsatz mit intelligenten mobilen Robotern gedacht sind.

Schließlich präsentieren wir einen Ansatz, der die erworbenen Raummodelle zur situierten Verarbeitung natürlicher Sprache verwendet. Er erlaubt es

einem Agenten, treffende *natürlichsprachliche Ausdrücke zu generieren und zu verstehen*, die auf Dinge in der großräumigen Umgebung referieren. Der Ansatz wird von einem empirischen Sprachproduktionsexperiment unterstützt. Die vorliegende Arbeit behandelt abschließend den im Gange befindlichen Transfer der für intelligente mobile Roboter entwickelten räumlichen Modelle auf virtuelle Agenten in einer internetbasierten virtuellen 3D online-Welt.

In **Kapitel 2** fassen wir den wissenschaftlichen Hintergrund zusammen, auf dem die vorliegende Arbeit aufbaut. Nach einer Kurzeinführung in kognitive Systeme befassen wir uns mit autonomen Systemen, darunter Bereiche der Robotik (insbesondere autonome und intelligente mobile Roboter) und der virtuellen Welten. Danach erörtern wir einige wesentliche Aspekte von verkörperter Kognition (engl. 'embodied cognition'), menschlicher Kategorisierung und Konzeptualisierung, und geben abschließend noch eine Einführung in die ontologische Wissensrepräsentation.

In **Kapitel 3** wird die hervorgehobene Rolle der Gliederung und Kategorisierung großräumiger Umgebungen für ein umfassendes Raumverständnis motiviert. Damit autonome Agenten situierte Dialoge über ihre Umgebung führen können, brauchen sie Raummodelle, die mit den mentalen Raummodellen ihrer menschlichen Gesprächspartner vereinbar sind. Für ein autonomes Verhalten, z.B. Navigation, hingegen benötigen sie Zugriff auf Repräsentationen auf einer viel niedrigeren Ebene. Um diesen beiden Anforderungen gerecht zu werden, führen wir eine Methode zur mehrschichtigen räumlich-konzeptuellen Kartierung ein. Die Beschreibung des Ansatzes ist in eine Erörterung relevanter Forschungsergebnisse aus den Gebieten der menschlichen Raumkognition und der Umgebungsmodellierung für mobile Roboter eingebettet.

In **Kapitel 4** befassen wir uns mit der konzeptuellen Schicht der mehrschichtigen räumlich-konzeptuellen Karte. Wir zeigen, wie Beschreibungslogiken benutzt werden können, um weitere Schlussfolgerungen aus einem symbolischen, mit der menschlichen Umgebungskonzeptualisierung vereinbaren Umgebungsmodell zu ziehen. Des Weiteren zeigen wir, wie prototypische Default-Schlussfolgerungen und Belief-Revision die Fähigkeiten autonomer Agenten bereichern können.

In **Kapitel 5** stellen wir den Explorer vor. Der Explorer ist ein integriertes Roboter-System, das das mehrschichtige räumlich-konzeptuelle Umgebungsmodell implementiert. Die mobile Roboter-Plattform ist mit verschiedenen Sensoren, u.a. zur Kartierung, Objekterkennung und Benutzerinteraktion, ausgestattet. Wir zeigen, wie der Explorer dieses Umgebungsmodell interaktiv in einer so genannten geführten Tour aufbauen kann. Eine wichtige Kernfunktion von Robotern für eine solche geführte Tour ist die Fähigkeit, einem menschlichen Tutor autonom durch die Umgebung folgen zu können. Wir zeigen einen

Ansatz zum situationsbewussten Verfolgen von Personen, der die konzeptuelle Information aus dem mehrschichtigen Umgebungsmodell verwendet. Dies erhöht das wahrgenommene Intelligenzniveau des Roboters.

In **Kapitel 6** präsentieren wir eine Erweiterung des EXPLORER-Systems, die auf PECAS, einer kognitiven Architektur für intelligente Systeme, basiert. PECAS verbindet die Fusion von Informationen aus einer verteilten, heterogenen Architektur mit einer Methode zum fortwährenden Planen als Systemkontrollmechanismus. Wir beschreiben, wie das PECAS-basierte EXPLORER-System das mehrschichtige Umgebungsmodell implementiert. Des Weiteren zeigen wir, wie prototypisches Default-Wissen aus einer auf Beschreibungslogiken basierenden Ontologie abgeleitet werden, und wie dieses, wenn kein Faktenwissen verfügbar ist, zum zielgerichteten Planen für situiertes Agieren in großräumigen Umgebungen eingesetzt werden kann.

In **Kapitel 7** zeigen wir ein Verfahren zur autonomen Erstellung einer räumlich-konzeptuellen Karte. Dies wird durch eine enge Verzahnung von Bottom-up-Kartierung mit logischem Schließen und aktiver Beobachtung der Umgebung erzielt. Das Verfahren erweitert den Ansatz zum mehrschichtigen räumlich-konzeptuellen Kartieren und sieht sowohl einen nicht-monotonen Aufbau des konzeptuellen Umgebungsmodelles als auch eine bidirektionale Verbindung von Wahrnehmung, Kartierung und Inferenz vor. Das Verfahren wurde in dem integrierten Roboter-System DORA implementiert. Es besitzt die Fähigkeit zum beschreibungslogikbasierten Schließen und zur nicht-monotonen Inferenz über eine OWL-Ontologie von räumlichem *Commonsense*-Wissen. Es setzt diese beim aktiven visuellen Durchsuchen und bei einer am Informationsgewinn ausgerichteten Erkundung der Umgebung ein. Das System wurde in mehreren Experimenten getestet, die aufzeigen, wie ein mobiler Roboter sein Umgebungsmodell autonom aufbauen, und wie räumlich-konzeptuelles Wissen sein zielgerichtetes autonomes Verhalten beeinflussen kann.

In **Kapitel 8** behandeln wir eine Methode zur Generierung und Interpretation natürlichsprachlicher Ausdrücke, die auf Dinge in großräumigen Umgebungen referieren. Sie basiert auf der in den vorhergehenden Kapiteln beschriebenen räumlichen Wissensbasis. Bestehende Algorithmen für die Generierung referentieller Ausdrücke versuchen eine Beschreibung zu finden, die den Referenten unter Einbeziehung des aktuellen Kontexts eindeutig identifiziert. Diejenigen Agenten, die wir hier betrachten, agieren jedoch in großräumigen Umgebungen. Das bringt die Herausforderung mit sich, dass ein Zuhörer seinen Kontext entsprechend vergrößern muss, sobald über an anderer Stelle befindliche Dinge gesprochen wird. Hierzu muss der Sprecher genügend Information kommunizieren, um dem Zuhörer zu ermöglichen, den intendierten Referenten korrekt zu identifizieren. Wir präsentieren das Prinzip der topologischen Abs-

traktion (TA) als Lösungsansatz, um geeignete Kontexte sowohl für die Generierung als auch für die Resolvierung referentieller Ausdrücke aufzubauen. Wir zeigen weiterhin, wie unser Ansatz in einem bidirektionalen Dialogsystem für interaktive Roboter verwendet werden kann.

In **Kapitel 9** befassen wir uns mit einer Methode für die Produktion und das Verständnis von Ausdrücken, die auf Dinge in einer großräumigen Umgebung referieren, während eines längeren Diskurses. Die Methode verwendet das TA-Prinzip. Allerdings betrachten wir hier die Identifizierung von Bezugsobjekten sprachlicher Ausdrücke aus einer Diskursperspektive. Zu diesem Zweck schlagen wir zwei Mechanismen vor, die das Lenken der Aufmerksamkeit im Verlaufe eines Diskurses modellieren. Die Mechanismen nennen wir *anchor-progression* und *anchor-resetting*. Wir beschreiben dann die Durchführung und Auswertung eines empirischen Sprachproduktionsexperimentes. Es dient der Evaluation der vorgeschlagenen Mechanismen im Hinblick auf situierte Handlungsanweisungen in kleinräumigen Szenen einerseits und großräumigen Umgebungen andererseits. Abschließend präsentieren wir eine Implementation der Methode.

Acknowledgments

There are many people whom I owe a lot. Writing a thesis like this in an international research project is always a matter of joint work, cooperative thinking and mutual inspiration of all people involved with the project, and a time demanding a great deal from people in one's personal surroundings. Without their support and inspiration, their encouragement and wisdom, this thesis would not have been possible. I wish to thank . . .

. . . my advisor Geert-Jan for his support, his ideas and criticism, his motivation and tireless guidance.

. . . my supervisor and first reviewer Hans Uszkoreit for his constructive remarks and his encouragement during my work on this thesis.

. . . my second reviewer Massimo Poesio for his interest in my work.

. . . my fellow PhD students in and from Saarbrücken Oliver, Maria, Sergio, Pierre, and Alejandro for sharing their ideas and providing positive feedback.

. . . my great colleagues (present and past) from the Talking Robots group: Ivana (thank you for the good discussions, your inspiration, and the great collaborations), Uli (thanks a million for your advice, constructive remarks, and especially your support with the reasoning-related parts of my thesis), Mira (the LaTeX guru), Shanker (thanks for your feedback on the robotics-related parts of my thesis), Bernd, Christopher, Fai, Weijia, and all the others.

. . . my office mates Henrik J and Christian – it's been a pleasure!

. . . the KomParse team at DFKI Berlin Peter, Tina, Feiyu, Xiwen, Torsten.

. . . Patric, Dani, Jeanna, Kristoffer, Andrzej, Alper, and the other PhD students for making my time at the Centre for Autonomous Systems at KTH in Stockholm such an enjoyable, instructive, pleasant, and inspiring stay.

. . . my dear colleagues in the CoSy and CogX projects, especially Henrik C, Jeremy, Nick, Oscar, Marc, Michael B, Michael Z, Alen, and Danijel. You all made the project work, including many, many long, long days, nights, and weeks of integration efforts a good time full of valuable insights.

. . . and last but not least my family and friends. I am deeply indebted to my parents and grandparents, my brother, and especially to Catrin. This thesis is dedicated to you!

I have learned a lot from all of you. *Thank you.*

This work has partly been supported by the European Community under contract number FP6-004250-CoSy in the EU FP6 IST Cognitive Systems Integrated Project "Cognitive Systems for Cognitive Assistants *CoSy*"[1] and under contract number FP7-ICT-215181-CogX in the EU FP7 ICT Cognitive Systems Large-Scale Integrating Project "*CogX* – Cognitive Systems that Self-Understand and Self-Extend" [2] at the DFKI Language Technology[3] lab. Their support is gratefully acknowledged.

[1] `http://www.cognitivesystems.org`
[2] `http://cogx.eu`
[3] `http://www.dfki.de/lt`

Contents

Chapter 1

Introduction

"Spatial cognition is at the heart of our thinking."
(Stephen C. Levinson)

"Language is the dress of thought."
(Samuel Johnson)

"The robots are coming!"
(Alan K. Mackworth)

Many times each day, if not almost constantly, we are thinking about space. We need to do so when we move, when we plan and perform actions, and when we try to accomplish tasks. We always need to know what is where in order to know what we can and cannot do there. This naturally involves the perception of our immediate surroundings. It also comprises access to past experiences and their interpretation that form mental representations of portions of space that are not immediately perceivable. Only this makes it possible to remember and reason about what was where and what to expect there. Spatial cognition is therefore a vital skill and immediately relevant to our survival. The ability to have spatial memories, however, is not unique to humans. Many animals are equipped with sophisticated spatial representations that allow them to find their retreat area or places where they can find food.

Besides mental representations of space, another aspect of cognition dealt with in this thesis is communication – more specifically spatially situated natural language communication. Language is the most common and most prominent means for our everyday communication. Communication is about exchanging information, and as such it is a widespread ability in the animal world as well. Animals communicate, for example, in order to signal danger or courtship, or to coordinate flocking and other behaviors. Animals also communicate spatial information. For instance, an animal can signal the position of a hazard to other members of the group, or it can communicate its own position to potential intruders. The remarkable difference between such forms of communication and

human language is the creative combinatorial potential of the latter. Whereas human language makes use of a generative rule-set and open-ended sets of tokens to create an infinitely large number of different messages, animals can only communicate a finite set of messages with little or no variability at all.[1]

With the emergence of language, i.e., a system to convey arbitrary messages, our ancestors were able to communicate about different spatial topics. This provided the evolutionary advantage of being able to efficiently share the vital aspect of knowledge and experience where dangerous and fruitful places are, and how to best avoid or make use of them, respectively. Even today, a large part of our communication is about space: we talk about things in our environment, we refer to other places – each concrete thing that we talk about has a spatial location. It is not for no reason that foreign language phrasebooks are very elaborate on spatial language: asking for directions, being able to point out which object one is talking about, and even explaining where one comes from are among the basic conversational skills that one needs when talking to strangers. Humans are able to routinely and effortlessly use and understand spatial language.

However, space and spatial cognition affect our language even when we are not directly talking about spatial topics. Spatial language is pervasive and fundamental for all of our communication. We don't have to look at metaphors and collocations to *come to the conclusion* that spatial concepts provide a "structuring tool for other conceptualized domains" (Lakoff and Johnson, 1980). Everyday language makes regular use of spatial relations to express abstract relationships – most prominently through the prepositional system. "Spatial" prepositions are used to express syntactic ("*to* be or not *to* be"), temporal ("the bus arrives *in* five minutes"), as well as causal ("the reasons *behind* his success") relationships, to name just a few.

Many researchers and scientists have investigated the special relation between language and spatial cognition, for example from a cross-linguistic perspective (Levinson, 2003), or under the assumption of an *embodied mind* (Lakoff and Johnson, 1999; Pecher and Zwaan, 2005). The human language capabilities coincide with the human level of conscious intelligence that is tied to higher-level cognitive representations (i.e., concepts) of lower-level mental

[1] For example, certain bees are known to be able to communicate the location of food through a specific form of dance. It allows them to express three kinds of properties of the food location: its relative direction with respect to the sun (expressed through body orientation), the distance to the hive (expressed by the length of certain part of the dance), and the food quality (expressed by the intensity of the display). They are, however, unable to express further circumstances, nor are they able to combine information about two sources of food, let alone specify the position of one food source with respect to another.

categories. The human concept system is available for reasoning and reflection, and thus accessible for manipulation and modification. Humans can verbalize their knowledge and relate external symbols to internal, cognitive concepts to an extent far beyond immediate stimulus-response triggering found in animals. Together, conceptual spatial cognition and natural language communication are fundamental for the kind of intelligence unique to humans.

Artificial intelligence, in contrast, is aimed at recreating intelligence in machines. Language remains the most intuitive, natural and efficient way of exchanging thoughts, commands, and of conveying all kinds of information for humans. The availability and rapid miniaturization of personal computers and their wide distribution has thus challenged researchers with endowing machines with language skills (Carstensen et al., 2010). With the development of autonomous robots, by the same token, the research community has been faced with machines that act in space, that perceive the space around them, that are part of the environment they operate in, and that are endowed with a physical *embodiment*. Teaching such machines basic spatial behaviors was just the beginning of a substantial body of research on robot navigation, mapping, and spatial representation (Choset et al., 2005).

More importantly than for other information technology, natural language capabilities for robots must be grounded in the physical space. If robots are to assist in concrete tasks (rather than reading the news to their users), they must engage in dialogues about things and events in their environment. The ability to refer to objects and to understand which objects are being talked about is at the very core of such spatially situated language use. Understanding, in turn, implies the existence of internal – mental – representations that are linked with the mentioned external entities. As it turns out, autonomous robots for the first time afford researchers with tools that can perceive their environment while being *active* part of the environment. Robots can act in an environment in order to alter it and actively gather new knowledge. And robots offer a degree of anthropomorphism that leads humans to intuitively and naturally employ spatial language when interacting with them. This means that robots offer the possibility to study embodied situated natural language use – but it also means that in order to be successfully deployed as conversational assistants, one of the most important requirements is that robots be able to make use of spatial language.

The work presented in this thesis addresses the fundamental questions of how machines, such as robots and other autonomous agents, can acquire a mental representation of their environment that allows them to (a) act and navigate in it, and (b) communicate about it with humans in natural language. We specifically investigate representations of structured environments that cannot be apprehended as a perceptual whole (i.e., *large-scale space*). This comprises,

for instance, indoor domestic environments, or building ensembles. By that, the presented work goes beyond situated natural language interaction about an agent's immediate surroundings (i.e., *small-scale space*), such as table-tops or single room spaces. *Situated* communication about *entities* – that is, things, places, properties, and events – in large-scale space requires the interlocutors to draw attention to entities that are not currently observable, and to comprehend which remote places and things are being talked about. We furthermore show how such representations that can be used for spoken interaction with human users also endow autonomous agents with skills for context-aware planning and execution of actions in structured environments that are made by and for humans. To this end, the presented spatial models have been implemented and deployed in integrated systems for intelligent mobile robots. We then present an approach for natural language generation and understanding that makes use of the acquired spatial models. It allows an agent to successfully *generate and resolve natural language expressions* that refer to entities in large-scale space. The approach is backed by observations from an empirical spoken language production experiment. The thesis concludes with a discussion of ongoing work to transfer the models made for intelligent mobile robots to autonomous virtual agents that act in an online virtual 3D world.

1.1 Contributions

This work contributes to research in *computational linguistics*, *artificial intelligence* and *robotics* by proposing related and connected approaches to several challenges that arise in the context of human-compatible environment modeling, situated dialogue processing, including natural-language generation and understanding, for autonomous embodied agents. Such agents include autonomous mobile robots and non-player characters for online virtual worlds. The approaches have been successfully deployed in integrated robotic systems. An empirical production experiment was conducted to confirm and refine the proposed models for the production and resolution of referential verbal descriptions in situated discourse about large-scale space.

Specifically we address in this work:

- a multi-layered approach to spatial mapping for mobile robots combining robot-centric spatial representations with higher-level conceptual representations
- a method for ontology-based symbolic representations for the purpose of human-compatible structuring and conceptualization of space

- different reasoning methods for these representations, covering Description Logic-based reasoning, prototypical reasoning, rule-based inference and nonmonotonic reasoning
- observations and experiences from implementing such ontology-based symbolic representations and formalisms in integrated robotic systems
- context- and distractor-set determination for generating and understanding referring expressions in large-scale space using these representations as spatially situated knowledge bases
- a model for attentional anchor-progression in discourse about large-scale space based on an empirical experiment on referring expressions in spatially situated instruction giving

1.2 Publications

The work presented in this thesis is based on the following contributions to research workshops, publications in conference proceedings, scientific journal articles, and book chapters.

Journal articles

- Wyatt, Aydemir, Brenner, Hanheide, Hawes, Jensfelt, Kristan, Kruijff, Lison, Pronobis, Sjöö, Skočaj, Vrečko, Zender, and Zillich (2010). Self-understanding & self-extension: A systems and representational approach. *IEEE Transactions on Autonomous Mental Development*, 2(4):282–303, December 2010.
- Zender, Mozos, Jensfelt, Kruijff, and Burgard (2008). Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, June 2008.
- Kruijff, Zender, Jensfelt, and Christensen (2007b). Situated dialogue and spatial organization: What, where… and why? *International Journal of Advanced Robotic Systems*, 4(1):125– 138, March 2007.

Book chapters

- Pronobis, Jensfelt, Sjöö, Zender, Kruijff, Mozos, and Burgard (2010a). Semantic modelling of space. In Henrik Iskov Christensen, Geert-Jan M. Kruijff, and Jeremy L. Wyatt, editors, *Cognitive Systems*, volume 8 of *Cognitive Systems Monographs*, chapter 5. Springer Verlag, Berlin/Heidelberg, Germany, 2010.

- Kruijff, Lison, Benjamin, Jacobsson, Zender, and Kruijff-Korbayová (2010). Situated dialogue processing for human-robot interaction. In Henrik Iskov Christensen, Geert-Jan M. Kruijff, and Jeremy L. Wyatt, editors, *Cognitive Systems*, volume 8 of *Cognitive Systems Monographs*, chapter 8. Springer Verlag, Berlin/Heidelberg, Germany, 2010.

- Sjöö, Zender, Jensfelt, Kruijff, Pronobis, Hawes, and Brenner (2010). The Explorer system. In Henrik Iskov Christensen, Geert-Jan M. Kruijff, and Jeremy L. Wyatt, editors, *Cognitive Systems*, volume 8 of *Cognitive Systems Monographs*, chapter 10. Springer Verlag, Berlin/Heidelberg, Germany, 2010.

Conference papers

- Zender, Koppermann, Greeve, and Kruijff (2010). Anchor-progression in spatially situated discourse: a production experiment. In *Proceedings of the Sixth International Natural Language Generation Conference (INLG 2010)*, pages 209–213, Trim, Co. Meath, Ireland, July 2010.

- Zender, Kruijff, and Kruijff-Korbayová (2009b). Situated resolution and generation of spatial referring expressions for robotic assistants. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 1604–1609, Pasadena, CA, USA, July 2009.

- Zender, Jensfelt, and Kruijff (2007a). Human-and situation-aware people following. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2007)*, pages 1131–1136, Jeju Island, Korea, August 2007.

- Zender, Jensfelt, Mozos, Kruijff, and Burgard (2007b). An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, pages 1584–1589, Vancouver, British Columbia, Canada, July 2007.

Workshop contributions

- Hanheide, Hawes, Wyatt, Göbelbecker, Brenner, Sjöö, Aydemir, Jensfelt, Zender, and Kruijff (2010). Hanheide, Hawes, Wyatt, Göbelbecker, A framework for goal generation and management. In *Proceedings of the AAAI Workshop on Goal-Directed Autonomy*, Atlanta, GA, USA, July 2010.

- Hawes, Zender, Sjöö, Brenner, Kruijff, and Jensfelt (2009b). Planning and acting with an integrated sense of space. In Alexander Ferrein, Josef Pauli, Nils T. Siebel, and Gerald Steinbauer, editors, *HYCAS 2009: 1st International Workshop on Hybrid Control of Autonomous Systems – Integrating Learning, Deliberation and Reactive Control*, pages 25–32, Pasadena, CA, USA, July 2009.

- Zender, Kruijff, and Kruijff-Korbayová (2009a). A situated context model for resolution and generation of referring expressions. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 126–129, Athens, Greece, March 2009. Association for Computational Linguistics.

- Zender and Kruijff (2007b). Towards generating referring expressions in a mobile robot scenario. In *Language and Robots: Proceedings of the Symposium*, pages 101–106, Aveiro, Portugal, December 2007.

- Mozos, Jensfelt, Zender, Kruijff, and Burgard (2007b). An integrated system for conceptual spatial representations of indoor environments for mobile robots. In *Proceedings of the IROS 2007 Workshop: From Sensors to Human Spatial Concepts (FS2HSC)*, pages 25–32. San Diego, CA, USA, November 2007.

- Mozos, Jensfelt, Zender, Kruijff, and Burgard (2007a). From labels to semantics: an integrated system for conceptual spatial representations of indoor environments for mobile robots. In *Proceedings of the ICRA-07 Workshop on Semantic Information in Robotics (SIR)*, pages 33–40. Rome, Italy, April 2007.

- Zender and Kruijff (2007a). Multi-layered conceptual spatial mapping for autonomous mobile robots. In Holger Schultheis, Thomas Barkowsky, Benjamin Kuipers, and Bernhard Hommel, editors, *Control Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems – Papers from the AAAI Spring Symposium*, Technical Report SS-07-01, pages 62–66, Menlo Park, CA, USA, March 2007. AAAI, AAAI Press.

1.3 Collaborations

Parts of this thesis have resulted from collaboration with other researchers. I am grateful for their ideas, advice, and effort they put into the joint projects.

Specifically, the approach to multi-layered conceptual spatial mapping rests on constant refinements of the low-level maps for robotic navigation developed at the Centre for Autonomous Systems (CAS/CVAP) at the Royal Institute of Technology (KTH) in Stockholm, Sweden, notably by Patric Jensfelt and Kristoffer Sjöö. Furthermore, approaches to hybrid laser- and vision-based object search and localization as well as active visual object search were developed by Dorian Gálvez López (formerly of KTH Stockholm) and Alper Aydemir (KTH Stockholm).

Approaches to sensor-based room categorization that were used as input providers for the ontology- and rule-based room classification were developed by Óscar Martínez Mozos (formerly of the University of Freiburg, now University of Zaragoza) and Andrzej Pronobis (KTH Stockholm).

The dialogue system in which our method for generating and resolving referring expressions in large-scale space was used was developed by my colleagues at DFKI Geert-Jan Kruijff, Ivana Kruijff-Korbayová, and Pierre Lison. Trevor Benjamin (formerly of DFKI) wrote the OpenCCG grammar that is used for parsing and surface realization of natural language. Christopher Koppermann and Fai Greeve (both DFKI) helped conducting and evaluating the empirical experiment on anchor-progression in situated dialogue about large-scale space. The software infrastructure for connecting my models and algorithms with the *Twinity* virtual world is based on the KomParse system developed by my colleagues at DFKI Berlin Peter Adolphs, Tina Klüwer, Feiyu Xu, and Xiwen Cheng, with the help of Torsten Huber and Weijia Shao.

The CAST cognitive architecture design and software were conceived and developed at the University of Birmingham by Nick Hawes, with input from Michael Zillich (now TU Vienna) and Henrik Jacobsson (formerly of DFKI Saarbrücken), and inspired by discussions with Jeremy Wyatt and Aaron Sloman. Approaches to symbolic planning, goal management, and cross-modal content binding are contributed by Michael Brenner (University of Freiburg, continual planning), Marc Hanheide (University of Birmingham, goal-generation and management), and Henrik Jacobsson (cross-modal content binding), respectively.

I will use the first person plural pronouns as *pluralis modestiae*. By consistently adhering to this use throughout my thesis, I want to reflect the fact that this work presents the results of many fruitful discussions with my colleagues and was conducted within European integrated research projects involving many

other people at different sites. The contributions of my collaborators are reproduced, potentially shortened, in as far as they are crucial for the description of the joint work. These contributions are listed separately at the beginning of each chapter. Besides literal quotations, which are marked in the usual way, definitions that are taken from the literature are marked with a citation. Everything else is based on the author's own work.

1.4 Outline

In **Chapter 2**, we present the scientific background that the work in this thesis builds upon. After a short overview of research on cognitive systems, we present an introduction to autonomous agents, including some background in robotics, in particular autonomous and intelligent mobile robots, and virtual worlds. We then discuss relevant aspects of the study of embodied cognition, human categorization and conceptualization. An introduction to ontology-based knowledge representations concludes the chapter.

In **Chapter 3**, we identify structuring of space and categorization of large-scale space as two important aspects of spatial understanding. In order to enable an autonomous agent to engage in a situated dialogue about its environment, it needs to have a human-compatible spatial understanding, whereas autonomous behavior, such as navigation, requires the agent to have access to low-level spatial representations. Addressing these two challenges, we present an approach to *multi-layered conceptual spatial mapping*. The description of our approach is embedded in a discussion of relevant research in human spatial cognition and mobile robot mapping.

In **Chapter 4**, we focus on the *conceptual map layer* of the multi-layered spatial. We show how Description Logics can be used to perform inference on a human-compatible symbolic conceptualization of space. We further propose methods for prototypical default reasoning and belief revision to extend the capabilities of autonomous agents.

In **Chapter 5**, we introduce the EXPLORER robot system. The EXPLORER implements the approach to multi-layered conceptual spatial mapping in an integrated robotic system. The mobile robot base is equipped with different sensors for map building, place and object recognition, and user interaction. We illustrate how the multi-layered map can be acquired interactively in a so-called guided tour scenario. We furthermore present a method for human- and situation-aware people following that makes use of the higher-level information of the multi-layered conceptual spatial map, thus increasing the perceived level of intelligence of the robot.

In **Chapter 6**, we present an extension of the EXPLORER system. The presented implementation makes use of PECAS, a cognitive architecture for intelligent systems, which combines fusion of information from a distributed, heterogeneous architecture, with an approach to continual planning as architectural control mechanism. We show how the PECAS-based EXPLORER system implements the multi-layered conceptual spatial model. Moreover, we show how – in the absence of factual knowledge – prototypical default knowledge derived from a Description Logic-based ontology can be used for goal-directed planning for situated action in large-scale space.

In **Chapter 7**, we present an approach in which a conceptual map is acquired or extended autonomously, through a closely-coupled integration of bottom-up mapping, reasoning, and active observation of the environment. The approach extends the conceptual spatial mapping approach, and allows for a nonmonotonic formation of the conceptual map, as well as two-way connections between perception, mapping and inference. The approach has been implemented in the integrated mobile robot system DORA. It uses rule- and DL-based reasoning and nonmonotonic inference over an OWL ontology of commonsense spatial knowledge, together with active visual search and information gain-driven exploration. It has been tested in several experiments that illustrate how a mobile robotic agent can autonomously build its multi-layered conceptual spatial representation, and how the conceptual spatial knowledge can influence its autonomous goal-driven behavior.

In **Chapter 8**, we present an approach to the task of *generating and resolving referring expressions to entities in large-scale space*. It is based on the spatial knowledge base presented in the previous chapters. Existing algorithms for the generation of referring expressions try to find a description that uniquely identifies the referent with respect to other entities that are in the current context. The kinds of autonomous agents we are considering, however, act in large-scale space. One challenge when referring to elsewhere is thus to include enough information so that the interlocutors can extend their context appropriately. To this end, we present the principle of *topological abstraction* (TA) as a method for context construction that can be used for both generating and resolving referring expressions – two previously disjoint aspects. We show how our approach can be embedded in a bi-directional framework for natural language processing for conversational robots.

In **Chapter 9**, we present an approach to producing and understanding referring expressions to entities in large-scale space during a discourse. The approach builds upon the principle of topological abstraction (TA) presented in Chapter 8. Here, we address the general problem of establishing reference from a discourse-oriented perspective. To this end, we propose *anchor-progression*

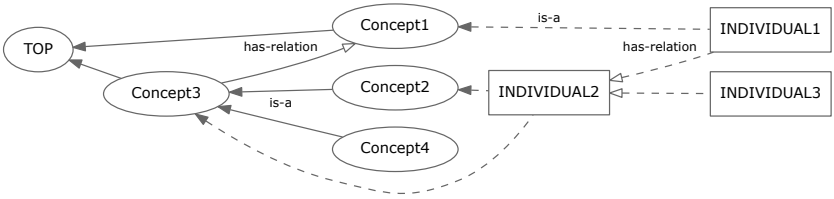
and *anchor-resetting* mechanisms to track the origin of the TA algorithms throughout the discourse that model the way attention-directing information unfolds during the course of a discourse. We present an empirical production experiment that evaluates the utility of the proposed methods with respect to situated instruction-giving in small-scale space on the one hand, and large-scale space on the other. We conclude with a discussion of an implementation of the approach and give examples of its performance with respect to the domain of the production experiment.

In **Chapter 10**, we recapitulate the work presented in this thesis. We describe an ongoing effort to transfer the proposed robotics-oriented models to autonomous virtual agents that act in an online virtual 3D world. We conclude the thesis with a discussion of open issues and opportunities for extensions to the presented work in future research.

1.5 Notational Conventions and Symbols

Ontology, logics and rule-based reasoning

Logics	
Symbol	**Description**
General Description Logics	
$\mathcal{O}$	ontology knowledge base
$\mathcal{T}$	TBox
$\mathcal{A}$	ABox
$\mathcal{R}$	RBox
$\mathcal{D}$	DBox
Δ	domain of discourse
$\mathcal{I}$	interpretation function
$\models$	entailment
$\mathcal{O} \models \ldots$	knowledge base $\mathcal{O}$ entails ...
Abstract DL syntax	
A, B	atomic concepts
C, D	concept definitions
R, S	roles
a, b, c	individuals
x, y, z	free individual variables
n, m	non-negative integers
$A(a)$, $C(a)$	concept assertions
$R(a, b)$, $S(b, c)$	role assertions
Examples in abstract DL syntax	
Concept, ConceptName	concept names
related, hasRelation	role names
INDIVIDUAL, IND1	individual names
Concept(IND1), ConceptName(IND2)	concept assertions
related(IND1, IND2), hasRelation(IND2, IND1)	role assertions
Default Logic	
α, β, γ	first-order logic formulae
δ	default rule
x, y, z	free individual variables
E_i	knowledge base extensions



Solid arrows express relationships between Concepts: concept definitions ("has-relation", hollow arrow head), or subclass relations ("is-a", dark arrow head). Dashed arrows express relationships involving INDIVIDUALs: being an instance of a Concept ("is-a", dark arrow head), or being related to another INDIVIDUAL ("has-relation", hollow arrow head).

Natural language syntax and semantics

Combinatory Categorial Grammar (CCG)	
Symbol	**Description**
X, Y	categories
$/$, $\backslash$	rightward-combining and leftward-combining functors ("slashes")
Hybrid Logic Dependencuy Semantics (HLDS)	
Symbol	**Description**
$@$	satisfaction operator
p, q, r	variables over any hybrid logic formula
i, j, k	variables over states
d_i, h_i	variables over nominals (for dependent and head, respectively)
An, Bn, $\ldots$	nominals (with n being a natural number)

Statistics

Statistics	
Symbol	**Description**
σ	standard deviation
t	t-value for t-test
df	degrees of freedom (usually $n-1$)
p	probability of error

Chapter 2

Background

Summary
In this chapter, we present the scientific background that the work in this thesis builds upon. After a short overview of research on cognitive systems, we present an introduction to autonomous agents, including some background in robotics, in particular autonomous and intelligent mobile robots, and virtual worlds. We give a short definition of the notion of situated dialogue. We then discuss relevant aspects of the study of embodied cognition, human categorization and conceptualization. An introduction to ontology-based knowledge representations concludes the chapter.

The work presented in this thesis is at the intersection of *computational linguistics*, *artificial intelligence*, and *robotics* – an inter- and multi-disciplinary area concerned with the design and implementation of *cognitive systems*. The long-term goal of cognitive systems research is best characterized by the objective of the European Commission's sixth framework programme (FP 6):

> *"to construct physically instantiated … systems that can perceive, understand … and interact with their environments, and evolve in order to achieve human-like performance in activities requiring context- (situation and task) specific knowledge"* (from (Christensen et al., 2010)).

From this objective a number of requirements can be derived. First of all, physical instantiation requires *embodiment* – i.e., the agent is a physical part of the world, it has sensors and effectors to perceive and manipulate its environment. In addition to *perception*, the agent must develop an *understanding* of its environment that allows for context-aware *interaction* with the environment.

This work addresses some of the challenges that an embodied cognitive system, i.e., an *autonomous agent* has to overcome in order to acquire a representation of its large-scale spatial environment that allows it to act and interact

in it. The specific kind of interaction that we focus on in this work is *spatially situated dialogue* with a human. In the following, we present relevant foundational and background work from the involved and related scientific fields.

2.1 Autonomous Agents

Franklin and Graesser (1997) define an *autonomous agent* as "a system situated within and a part of an environment that senses that environment and acts on it, over time, in pursuit of its own agenda and so as to effect what it senses in the future." Autonomy, in turn, is characterized as a property that entails exercising control over one's own actions, and gaining information about the environment. With respect to the taxonomy proposed by Franklin and Graesser (1997), we focus on agents which possess, in addition to autonomy, the following properties.

Goal-orientation Depending on its system state, the agent might have the goal to follow the instructions of a user, or it might come up with its own goals.

Communicativity The agent is able to communicate with other agents. Here, we focus on interactive communication with humans. Diverging from Franklin and Graesser's terminology, we will call this class of agents *conversational agents*.

Mobility A mobile agent is not static, but can actively change its location.

Embodiment As opposed to mere software agents that, for instance, collect information on the world wide web, and communicate with their users like other computer programs, the agents we are interested in have a physical embodiment. They have a "shared presence" with their human users, breaking the so-called "fifth wall" that usually separates the spaces in which the agent and its user operate (Byron and Fosler-Lussier, 2006).

Under these assumptions, we are particularly focusing on two special kinds of autonomous agents, namely *autonomous mobile robots* (cf. Section 2.2), and *virtual embodied agents*, which correspond to non-player avatars in virtual worlds (cf. Section 2.3).

Later, we will use the term *(autonomous) agent* to refer to any of these two kinds of agents. Sometimes, we will also subsume humans (or human-controlled avatars) in the environment under this term. Instead of saying autonomous mobile robot, we will later simply use the term *robot*, when there is no danger of confusion with industrial robots, or when we also want to subsume stationary intelligent robots. Finally, we will call virtual embodied agents simply *virtual agent*, or, in order to stress the difference to human-controlled avatars, *non-player character* (NPC).

2.2 Robotics

The term *robot* for a human-like automaton first appeared in the 1920 play *Rossum's Universal Robots* by the Czech writer Karel Čapek. The term was coined after the Czech word *robota* '(corvée or slave) labor'. However, the concept of artificial people and intelligent automata respectively is much older, ranging back to both Greek and Norse mythology. Leonardo da Vinci's design of a mechanical knight from the late 15th century, as well as the mechanical automata of Jacques de Vaucanson, a French engineer in the 18th century, exhibit features similar to the modern sense of robot.

There are several kinds of modern robots that share certain features, but are different in a lot of ways. All robots have in common that they are mechanical automata capable of performing actions and movements. They differ in the degree of autonomy they have in executing their tasks, ranging from a fixed sequence of actions and motions in typical industrial robots to a high degree of reactiveness and adaptability to a changing world in autonomous robots (Siegwart and Nourbakhsh, 2004).

Industrial robots (cf. Figure 2.1) find their application in automated assembly processes. They mostly perform an invariant sequence of actions, outperforming humans in strength and precision. Industrial robots, however, usually cannot sense changes in their environment and cannot change their behavior de-

Figure 2.1: Factory Automation with industrial robots for palletizing food products like bread and toast at a bakery in Germany.

liberately. Typical industrial applications are robot arms with several degrees of freedom for painting, welding, or assembling component parts.

While industrial robots are typically immobile and perform their task at a fixed position in the assembly work flow, robotic automated guided vehicles (AGV) follow preprogrammed routes in – often instrumented – environments such as large warehouses. The next degree of autonomy, especially concerning mobility in unknown environments, is found in applications that involve tasks that are either too dangerous to be performed by humans, or pertain to environments that are not easily accessible by humans, such as bomb disposal (explosive ordnance disposal robots, EOD) or underwater exploration scenarios (autonomous underwater vehicle, AUV).

2.2.1 Autonomous mobile robots

As mentioned earlier, our focus is on autonomous mobile robots. Autonomy implies that the robot must be equipped with *sensors* for perceiving its environment, whereas mobility implies that the robot must be equipped with *actuators* that provide it with locomotion capabilities.

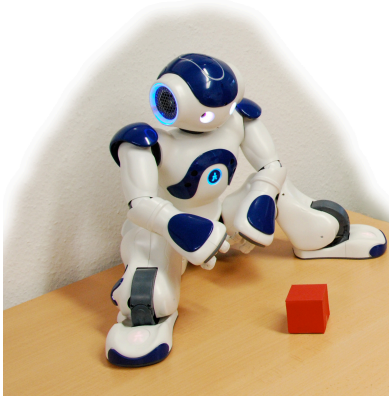
In the context of perception, *exteroceptive* sensors are used to perceive the robot's environment, and *proprioceptive* sensors measure a robot's internal states. Laser range finders, sonar arrays, and infrared distance sensors are the most common exteroceptive sensors used in mobile robots, while various kinds of encoders provide proprioceptive measurements of wheel and/or motor speed and position.

When it comes to *locomotion* – as opposed to *manipulation* – there are several robots that make use of biomimetic mechanisms, such as various kinds of legged walking systems: biped anthropomorphic robots (such as the small humanoid robot Nao,[1] see Figure 2.2a), quadruped reptile or mammal-like robots, even eight-legged walking robots, or swimming robots that mimic the locomotion abilities of a sea snake. Such biomimetic locomotion apparatus are highly specialized for certain terrains.

Wheeled robots or robots equipped with slip/skid steering with caterpillar tracks sacrifice the benefits of biomimetic configurations. Instead, they have the advantage of easier control and more efficient locomotion in structured environments, such as indoor environments. They do not have the problem with balance and stability that most legged robots have. Figure 2.2b shows a MobileRobots P3-DX[2] robotic platform for research and teaching. It is equipped with a two-wheel differential drive and a third caster wheel for stability. The

[1] `http://www.aldebaran-robotics.com/en/node/1160` [last accessed on 2010-05-10]

[2] `http://www.mobilerobots.com/ResearchRobots/ResearchRobots/PioneerP3DX.aspx` [last accessed on 2010-05-10]



(a) Nao by Aldebaran Robotics: a medium-sized (ca. 58 cm) humanoid robot.

(b) P3-DX by MobileRobots Inc.: a wheeled mobile robot.

Figure 2.2: Two mobile robots with different morphologies.

wheel speed and position is tracked by the robot's odometry. The robotic systems presented in Part II are based on robot platforms that are derived from the P3-DX. The main external sensor used in these systems is a laser range finder (not shown in Figure 2.2b).

Robot self-localization and environment mapping

Siegwart and Nourbakhsh (2004) identify *self-localization*, *navigation*, and *path planning* as crucial functional prerequisites of mobile robots. These pre-suppose the existence of an environment map within which to localize, navigate, and plan paths. Consequently, *mapping* is another cornerstone of mobile robotics.

Primitive *localization* methods rely only on *dead reckoning*. Dead reckoning considers an initial position, the distance, speed, time, and directions traveled to estimate the current position. In other words, dead reckoning tries to estimate the robot's current position solely on the basis of odometry readings. Due to external factors, including friction of the mobile robot's wheels on the ground surface, dead reckoning yields rather unreliable position estimates. Together with general sensor noise, these factors lead to a considerable amount of accumulated error, such that reliable self-localization is impossible without additional error correction. Moreover, localization assumes that a map of the operating environment exists. Simple *mapping* of an environment, in contrast, is achieved through exteroception. However, the readings that such sensors

(e.g., camera images, sonar, or laser) give are also prone to errors and noise. In case the environment to be covered is larger than the sensory horizon, or parts of it are occluded by obstacles, the exteroceptive readings must be acquired at different positions in the environment and integrated with each other. This, in turn, poses the question of how to know the exact positions at which the different sensor readings were acquired.

There are several methods that integrate exteroceptive and proprioceptive sensors to build more reliable hypotheses about the environment and the robot's position within it. One such technique is *simultaneous localization and mapping* (SLAM). SLAM is utilized to overcome the limitations of dead reckoning, while also attempting to correct exteroceptive errors. It usually combines odometry readings from the rotation of the robot's wheels and input from at least one exteroceptive sensor. From these sensor readings, the SLAM method extracts features and uses these feature observations for self-localization by tracking and aligning the observations from subsequent sensor readings over time through statistical and probabilistic methods. SLAM is very useful as it provides a position estimate while constructing a map of unknown environment. There are different SLAM algorithms, as mentioned in (Thrun et al., 2005). Their common goal is to construct an optimal metric map of the environment while the robot is exploring that environment. In order to account for sensor noise, other limitations in perception, and the combined error of odometry and external measurement most SLAM methods explicitly represent the uncertainty about the state they are in. The maps created by the SLAM technique range from local patches that are only loosely combined to a global representation, to global metric maps that can represent whole buildings with several floors, e.g., in the approach presented by Frese and Schröder (2006). We show later how SLAM is utilized in our approach for acquiring a spatial representation of the robot's environment.

2.2.2 Service robots

With the ongoing development in the field of artificial intelligence, the previously fictional idea of intelligent, or at least highly autonomous, robots has become an increasingly active subject of research and development. Social robots, domestic robots, and *service robots* are considered future and emerging technologies by government agencies and commercial companies all over the world. The International Federation of Robotics (IFR) Statistical Department (2009) identifies the following major classes of service robots:

- Field robotics (agriculture and forestry)
- Professional cleaning (e.g., floor cleaning and pipe cleaning)

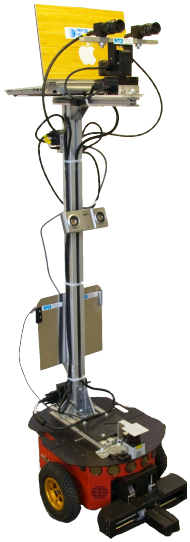
- Inspection and maintenance systems (e.g., for factories or sewers)
- Construction and demolition (including nuclear dismantling)
- Logistic systems (e.g., mail or cargo handling, and factory logistics)
- Medical robotics (most notably, robot assisted surgery and therapy)
- Defense, rescue and security applications (e.g., demining robots, unmanned aerial vehicles (UAVs) and other surveillance robots)
- Underwater systems (i.e., autonomous underwater vehicles (AUVs))
- Public relation robots (e.g., guide robots or library robots)
- Robots for domestic tasks (the largest portion being consumer-level floor cleaning robots)
- Entertainment robots (including toy/hobby robots as well as robots used in education and training)
- Handicap assistance (most notably, robotized wheelchairs)

In total, these amount to a number of approximately 1.6 million sold units in 2007, and approximately 1.7 million sold units in 2008 International Federation of Robotics (IFR) Statistical Department (2009). Out of these, hotel and restaurant robots (estimated installations between 2009 and 2012: 150 units), guide robots (units sold in 2007: 4, units sold in 2008: 5, estimated installations 2009–2012: 350 units), robot butlers and companions (units sold in 2007: 518, units sold in 2008: 600, estimated installations 2009-2012: 40,000 units), robotic wheelchairs (estimated installations between 2009 and 2012: 1,000 units), and other assistive robots (estimated installations between 2009 and 2010: 25,300 units) are the ones that are most likely to be used by non-expert users, and have to operate in non-instrumented environments. The Nesbot™ robot (see Figure 2.3a) by BlueBotics is an example of a domestic service robot. It can operate in known, fully mapped environments. Users can interact with it using a handheld control panel, or a web-based ordering application.

These numbers also illustrate that people are willing to introduce robots into their everyday lives and to permit them to enter their social environments. Such robotic assistants, however, need to have more intelligence than current autonomous vacuum cleaners or robotic pets, which are mere sensor-enabled household tools or toys, respectively. In order to be of real assistance to humans, a robot must be able to act within, interact with, and understand a human-populated environment. This is where cognitive systems come into play. An

(a) The Nesbot robot by BlueBotics: a mobile service robot with limited user interaction capabilities.

(b) BIRON, the Bielefeld Robot Companion: a human-robot interaction research platform.



(c) DORA, the cognitive robotics research platform of the project CogX.

Figure 2.3: Three chest-height sized mobile robots designed for user interaction.

autonomous cognitive assistant has to be aware of the implications that arise when performing actions in domestic environments and must react accordingly.

2.2.3 Intelligent robotic systems

Currently, most robots for domestic use on the market have been developed with security and safety considerations in mind, which is the most important requirement of any consumer-level machine. Intelligence, a high level of autonomy, and sophisticated natural interaction capabilities are still topics of research. Consequently, most existing integrated intelligent robotic systems so far are research prototypes. Besides the integrated systems that are described in more detail in Part II (e.g., DORA shown in Figure 2.3c), a number of other integrated intelligent robotic systems exist that are able to interact with humans in their environment. Figure 2.3 shows different wheeled mobile indoor robots that are designed for interaction with human users.

Rhino (Burgard et al., 1999) and *RoboX* (Siegwart et al., 2003) are robots that are designed to act as tour-guides in museums. Both robots rely on an accurate metric representation of the environment and use limited dialogue to communicate with people. *ShopBot* (Gross et al., 2008) is an interactive in-store assistant robot. Examples of robots with more elaborate dialogue capabilities are *RoboVie* (Ishiguro et al., 2001), *BIRON* (shown in Figure 2.3b)(Haasch et al., 2004; Spexard et al., 2006; Peltason et al., 2009), *Godot* (Bos et al., 2003), *WITAS* (Lemon et al., 2001) and *Mel* (Sidner et al., 2004). Recently, the *Autonomous City Explorer* (Bauer et al., 2009) was deployed in the city center of Munich, where it autonomously navigated across the pedestrian area. It did not have a pre-defined map of the environment and relied solely on gestured direction information from passers-by.

2.3 Virtual Worlds and Agents

The term *virtual world* refers to "an electronic environment that visually mimics complex physical spaces, where people can interact with each other and with virtual objects, and where people are represented by virtual characters" (Bainbridge, 2007).

This comprises games as well as online services that serve the goal of social exchange between the participants (referred to as *players* for the sake of simplicity). The electronic environments can either be fictional, taking place in settings with differing degrees of realism, or modeled on the real world. What they have in common is that their mode of presentation is inherently human-oriented. "[Virtual worlds] approximate aspects of reality – enough for the purposes of immersion (...)" (Bartle, 2003). Spaces in the virtual world are to be understood by humans. They thus must make use of patterns that are familiar and meaningful to the players. The extent to which they do so differs with the intended degree of realism.

Just like autonomous mobile robots, *autonomous virtual agents* (see Section 2.1) therefore operate in human-oriented environments. They too need to make sense of environments that were designed to be understood by humans. Parts of that environment are usually designed by the programmers and providers of the virtual world, which makes it possible to give an agent a precise representation of its environment and the things occurring therein. This is similar to instrumented environments in reality. However, a typical feature of social online virtual worlds is to allow players to customize and shape parts of their environment. This usually comprises the possibility to change the appearance of their characters, but also, more importantly, to own places that serve as virtual homes, which can then be designed, decorated, and modified just like a

Figure 2.4: A conversational agent bartender NPC in the virtual world *Twinity*.

real one. In this case, the autonomous agent cannot receive a fully labeled map, including names for places and objects, from the server.

In Section 10.2 we show ongoing work in applying the methods of this thesis, which were originally designed for autonomous robots, to a non-player character in a virtual world. We have interfaced our algorithms with the NPC control for a virtual online world named *Twinity*, a product of the Berlin start-up company *Metaversum*.[3] Figure 2.4 shows a non-player character in a the *Twinity* virtual world engaged in a dialogue with a human-controlled avatar.

2.4 Situated Dialogue

So far, we have discussed different techniques that endow agents with autonomy to sense and move about their environment. Another aspect of the kinds of autonomous agents we are interested in is their ability to interact verbally with humans about their environment, their current tasks, or plans. In such an interaction process they must be able to communicate about entities in the (physical or virtual) world they are *situated* in.

Formally, dialogue can be seen as "a joint process of communication," which "involves sharing of information (data, symbols, context) between two or more parties" (Lansdale and Ormerod, 1994). Meaning is established if the

[3] `http://www.twinity.com/`, `http://www.metaversum.com/` [last accessed 2010-05-05]

symbols used by each party refer to common concepts and entities. In situated communication the data and symbols exchanged are about entities in the interlocutors' environment. The way the reference between the symbols and the entities in the word is established is determined how the interaction is situated in the environment. We say that the communication is *grounded* in the spatial context. In general, this information sharing can be non-linguistic. Here, however, we focus on natural-language based communication in situated dialogue.

2.5 Categorization and Conceptualization

An important issue in cognitive science, psychology, and linguistics is the question how the mind processes sensorimotor stimuli in order to form abstract representations that are available for higher-level reasoning as well as language production and understanding. A related question is how words, being arbitrary *symbols*, get their meaning and how this meaning is *grounded* in reality, i.e., how words can refer to things and circumstances in the world.

On the lowest level of sensorimotor abstraction, the mind performs *categorization*. Categorization is a basic skill of structuring sensory input by abstraction and simplification. It is an essential capability of every neural system in humans and animals alike, or as Lakoff and Johnson (1999) put it, "every living being categorizes," and every "living system must categorize". By categorization, it is possible to reduce the complexity of the input by relating it to previous input patterns, i.e., past experiences. With more and more experience, more and more categories are formed, and existing ones are refined. Most of category-forming and categorization is a sub-conscious process, while only a small part of it can be subject to conscious, deliberate cognitive action (Lakoff and Johnson, 1999).

Concepts are higher-level cognitive representations of our mental categories. The concept system is accessible for reasoning and inference and thus part of our conscious thinking. Concepts are often formed around *prototypes* – either ideal or average representatives of their concept, or ones that possess only elementary properties. Prototypes allow to draw draw inferences about category members in the absence of any special contextual information (Lakoff and Johnson, 1999).

2.5.1 Basic-level categories and concepts

We are concerned with the question of how one can refer linguistically to a spatial structure – e.g., a room, a place, or an object in a specific location – in a given situation. By naming a referent, people categorize it. Brown (1958) identifies that people in one community prefer the choice of one particular name

for classes of things over the many other possible names. "The most common name is at the level of usual utility" (Brown, 1958). This theory is regarded as the first approach towards a notion of *basic-level categories* further developed by Rosch (1978). The basic-level category of a referent is assumed to provide enough information to establish equivalence with other class members while distinguishing it from non-members. It has also been shown that the concept of an object evokes expectations about how to interact with it (Borghi, 2005).

For our method, we claim that a similar correspondence between room concepts and actions holds. The basic-level category of spatial entities in an environment is determined by functional properties that the members of such an equivalence class afford. The functional affordances of a room which are not directly related to its spatial extension, are in most cases provided by objects that are located therein. For instance, the concept of 'kitchen' applies to rooms that are suited for preparing a meal. The preparation of food, in turn, is afforded by certain objects, such as, e.g., an oven, a stove, a microwave oven, and secondary objects, such as a refrigerator, a countertop, or a sink. Thus, the objects that are located in a room are a basis for determining the appropriate general name to refer to the room. The category of a room can also be determined by several properties pertaining to the room's appearance such as shape, size, or color. For example, corridors are places that allow people to walk from one room to another. This function is not provided by any specific object, but rather by the spatial layout of the corridor, which spatially links as many rooms as possible (and necessary).

We furthermore assume that the basic-level categories that people use to refer to spatial areas are located at one level lower than the more general category 'room'. Of course, rooms can have proper names and it is common usage in office environments to label rooms systematically, e.g., by assigning unique, ordered numbers, but still it is uncommon in everyday talk that people use these proper names to refer to a spatial entity. People instead refer to rooms with their general names, which correspond to basic-level categories such as 'kitchen,' 'library,' or 'lobby.'

2.5.2 Basic spatial relations

The physical properties of containers and surfaces belong to the "first and most frequent spatial concepts taught" to children (Freundschuh and Sharma, 1996). Since these spatial concepts are among the first to be experienced through our own embodiment, they give rise to the basic cognitive schemata for spatial and metaphorical thinking. The so-called *container schema* represents one of the most pervasive and intuitive spatial relations, namely *containment* (Lakoff and Johnson, 1999). Another schema that is acquired early on is the notion of

surface-support, i.e., the *surface schema*. In natural language they are expressed by the topological locatives "in" and "on," which are among the most frequently used prepositions (Coventry and Garrod, 2004).

2.5.3 Ontology-based knowledge representation

Our approach models conceptual knowledge in an ontological taxonomy. It is composed of a commonsense ontology of an indoor environment that describes necessary and sufficient conditions that spatial entities must fulfill in order to qualify for belonging to a certain concept. Our definitions of the concepts in the terminological taxonomy are inspired by the way humans *categorize* space.

An *ontology* as a knowledge representation in computer science is a formal, "explicit specification of a conceptualization" of an area of interest (Gruber, 1993). Ontologies describe classes of objects, their properties, and relations that can hold between them. Ontologies are used to formally define a shared terminology, and to provide a semantic interpretation. They can be used as knowledge base for automated reasoning. *Description Logics (DL)* are a family of logical formalisms for ontology-based reasoning. Ontologies are suitable for representing the knowledge about a given domain in a way that is understandable by humans and processable by computers.

Recent research on formal and applied aspects of ontology-based knowledge representations have resulted in a distinction between two kinds of ontologies. For one, there exist ontologies that are supposed to represent very general concepts and relations that exist independently of a specific domain – so-called *upper ontologies* (Mascardi et al., 2007), such as, e.g., the Generalized Upper Model (GUM) (Bateman et al., 1995) or DOLCE (Masolo et al., 2003). *Domain ontologies*, on the other hand, represent the kinds of objects, along with their properties and relations, that exist in a particular domain. Examples of domain ontologies that are used to formally define the terminology of a specific field are the different food and agriculture ontologies listed by the Food and Agriculture Organization of the United Nations (FAO) (2010), and the Music Ontology proposed by Raimond et al. (2010). Figure 2.5 shows an example ontology (see Section 1.5 for a legend).

The Semantic Web makes wide use of ontologies. The aim of the Semantic Web is to provide machine-processable representations of web content. This processing involves the combination of information, knowledge discovery through inference, and an automatic detection of knowledge gaps and inconsistencies (Antoniou and van Harmelen, 2008). Ontologies provide machine-readable descriptions of their content as well as a semantic interpretation. The *Web Ontology Language OWL* has been introduced specifically to address the requirements of the Semantic Web, and to be mostly compatible to the well-

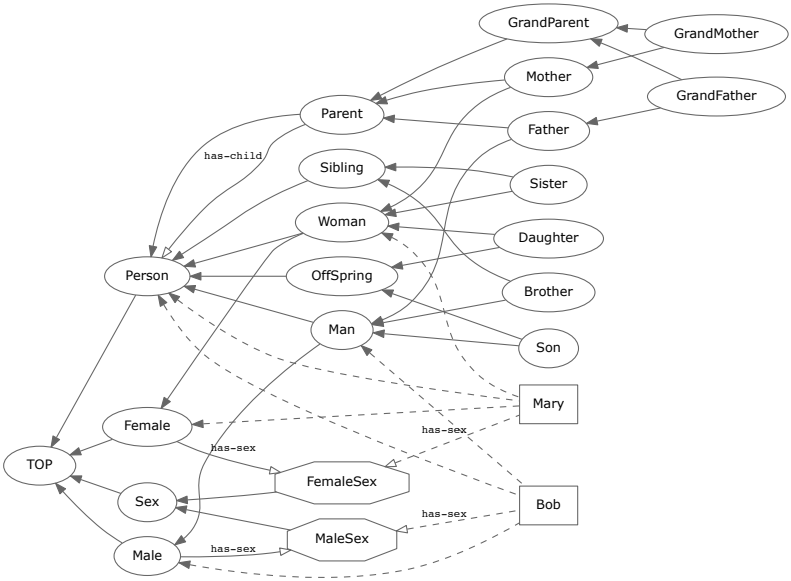


Figure 2.5: Example of an ontology of family relationships.

established standards RDF and RDFS (see Section 4.2.4). It offers a formally defined syntax, as well as formal semantics. Its expressive power strikes a balance to a tractable reasoning support.

2.6 Summary and Outlook

In this chapter, we have presented the scientific background that the work in this thesis builds upon. After a short overview of research on cognitive systems, we have presented an introduction to autonomous agents, including some background in robotics, in particular autonomous and intelligent mobile robots, and virtual worlds. We have then discussed relevant aspects of the study of embodied cognition, human categorization and conceptualization as well as the use of ontologies as a knowledge representation. In the next chapter, we will present an approach to multi-layered conceptual spatial mapping for autonomous mobile robots. The approach combines low-level sensor-based maps and several abstraction layers that are inspired by human categorization and conceptualization. In Chapter 4, we will show will show how we make use of an OWL-DL ontology for the conceptual layer of the multi-layered spatial representation.

Part I

Representing Knowledge for Spatially Situated Action and Interaction

Chapter 3

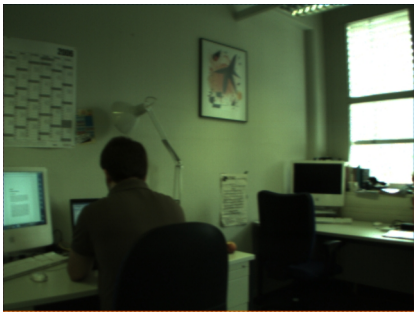
Multi-Layered Conceptual Spatial Mapping

Summary
In this chapter, we identify structuring of space and categorization of large-scale space as two important aspects of spatial understanding. In order to enable an autonomous agent to engage in a situated dialogue about its environment, it needs to have a human-compatible spatial understanding, whereas autonomous behavior, such as navigation, requires the agent to have access to low-level spatial representations. Addressing these two challenges, we present an approach to multi-layered conceptual spatial mapping. We embed our work in a discussion of relevant research in human spatial cognition and mobile robot mapping.

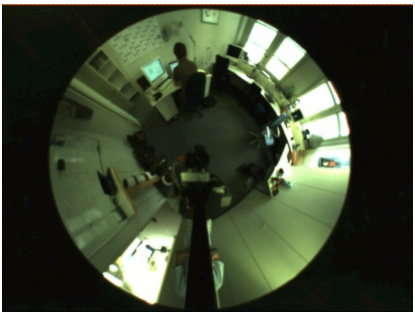
3.1 Motivation and Background

We are driven by the research question of *spatial understanding* and its connection to acting and interacting in indoor environments. We want to endow autonomous embodied agents with the capability to conduct spatially *situated dialogues*. For this the agent must be able to understand space in terms of concepts that can be expressed in, and resolved from natural language.

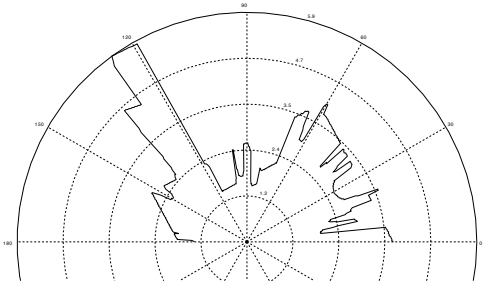
We start from the assumption that the environment is not instrumented in order to facilitate the mapping problem. The kinds of environments that we are interested in are indoor spaces that are designed by humans for humans – and that are intuitively and easily *understood* by humans. This includes ordinary and everyday indoor office environments or apartments that are populated by humans working and living there. This also includes virtual spaces that are designed in such a way that humans who control an avatar using a 3D client software perceive of them as if they were realistic models of natural physical spaces. We call this class of environments that are made and designed by humans for being used and populated by humans *human-oriented environments*.



(a) Perspective image taken from a digital camera mounted on the top platform of the robot (height: 140cm, field of view: 68.9°).



(b) Omnidirectional image taken from a digital camera facing up towards a hyperbolic mirror (height: 116cm, field of view: 360°).



(c) Frontier of the corresponding laser range scan taken at a vertical height of 30cm in parallel to the floor plane (field of view: 180°).

(d) The mobile robot used for acquiring the data. The cameras and the laser scanner can be seen on the top and bottom platforms, respectively.

Figure 3.1: Office environment "seen" by different robot sensors.

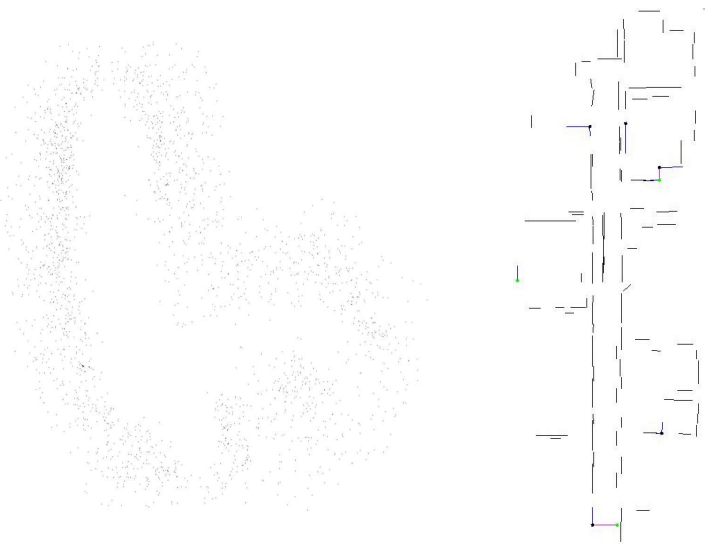
Figure 3.2 demonstrates examples of different human-oriented environments in which autonomous agents have to operate. Figure 3.1 shows how a robot's sensors (cameras and laser range finders) perceive such an environment.

There exist many different approaches for equipping autonomous embodied agents, most notably mobile robots, with spatial models. The problem is that these models are usually specifically tailored for the tasks the agent is supposed to fulfill. This means that the features of the spatial representation are

(a) Autonomous mobile robots – left: EXPLORER (see Chapter 5 and Chapter 6), right: DORA (see Chapter 7) – operating in an office building.

(b) A virtual character in a household environment within the Twinity world (see Section 10.2).

Figure 3.2: Examples of human-oriented environments.



(a) Underwater grid map of a marina in San Pere Pescador, Spain (Keshavdas, 2009).

(b) Line-feature map of an indoor environment.

Figure 3.3: Examples of robotic spatial representations for SLAM.

typically only *meaningful* with respect to the algorithms that work on these representations. These include, for instance, occupancy grid maps (see Figure 3.3a for an example), which address the challenge of representing which parts of an environment are likely to be free and unobstructed, and which ones contain potential obstacles (Thrun et al., 2005), or line maps that represent static features of the environment for the purpose of *simultaneous localization and mapping (SLAM)*, illustrated in Figure 3.3b (see also Section 2.2.1).

In contrast to this, what we need are human-like features. In order to be able to talk in and about space, the agent needs to abstract from its internal, machine-compatible representations of space to a level that is at least comparable to the way humans perceive of space.

Spatial understanding comprises two aspects. For one, it concerns *structuring* of spatial organization. That is, which are the units a human-oriented environment is composed of? Secondly, it concerns *categorization* of space. That is, which are the concepts that describe these spatial units, and how are they determined? We call spatial knowledge representations that address these issues *human-compatible representations* of space.

To this end the work presented in this thesis builds upon and extends the author's previous research on *multi-layered conceptual spatial mapping* (Zender, 2006; Zender and Kruijff, 2007a) in the tradition of approaches like the *(Hybrid) Spatial Semantic Hierarchy* (Kuipers, 2000; Kuipers et al., 2004; Beeson et al., 2007), the *Route Graph* model (Werner et al., 2000; Krieg-Brückner et al., 2005), *hybrid maps* (Buschka and Saffiotti, 2004), and *multi-hierarchical semantic maps* for mobile robots (Galindo et al., 2005, 2007). The approach is inspired by human cognition. On the lower layers it contains sensor-based representations. These are abstracted into basic categories (free space vs. occupied space, areas vs. humans vs. objects, rooms vs. corridors, etc.). The basic spatial relation is spatial containment, corresponding to the container schema, which is among the most prominent, most important, and most fundamental schemata in human cognition (cf. Section 2.5.2 and (Lakoff and Johnson, 1999)).

3.1.1 Structuring space

Research in cognitive psychology addresses the inherently *qualitative* nature of human spatial knowledge. It tries to answer the question how the human mind represents spatial information in a so-called *cognitive map*. Following the results of empirical studies, it is nowadays generally assumed that humans adopt a *partially hierarchical* representation of spatial organization (Stevens and Coupe, 1978; McNamara, 1986). The basic units of such a qualitative spatial representation are *topological* regions (Cohn and Hazarika, 2001), which correspond to more or less clearly bounded spatial areas. The borders may be defined *physically*, *perceptually*, or may be purely *subjective* to the human. It has been shown that even in natural environments without any clear physical or perceptual boundaries, humans decompose space into topological hierarchies by clustering salient landmarks (Hirtle and Jonides, 1985). In our approach, topological areas are the primitive units of the conceptual map that is used for human-robot interaction and dialogue, and the basic spatial relation is topological inclusion.

Recent advances in cognitive neuroscience have found evidence for brain structures that supply the topological representations of the so-called "place-cells" with a metric one encoded in the so-called "grid cells" (Jeffery and Burgess, 2006). This does not contradict the assumption that the global-scale representation of *large-scale space* in the cognitive map is a topological one. It rather provides insight into how local scenes, i.e., *small-scale space*, might be represented in the human mind and speaks in favor of a multi-layered, hybrid representation of space in the cognitive map.

Large-scale space and small-scale space

There is an important distinction to make when investigating any kind of spatially situated behavior, be it acting, planning, observing, learning, or communicating, namely if it pertains to space that constitutes the agent's immediate surroundings, or if it pertains to larger spatial structures. The dichotomy between *small-scale space* and *large-scale space* for human spatial cognition (Herman and Siegel, 1978; Hazen et al., 1978) is central to the work presented in this thesis.

Kuipers (1977) defines large-scale space as "a space which cannot be perceived at once; its global structure must be derived from local observations over time," whereas small-scale space consist of the here-and-now. For example, a drawing is a large-scale space "when viewed through a small movable hole, while a city can be small-scale when viewed from an airplane" (Kuipers, 1977). In more common everyday situations, an office environment, one's house, a city, or a university campus are large-scale spaces. A table-top or a particular corner of one's office are examples of small-scale space.

This distinction is crucial to the work presented in this thesis. We get back to it later when discussing appropriate knowledge representations (in this chapter) and reasoning mechanisms (in Chapter 4); when discussing semantic-driven exploration, navigation, and planning (in Part II); and, ultimately, when discussing strategies for verbally referring to entities in large-scale space (in Part III).

Segmenting and partitioning space

As mentioned earlier, it is important that autonomous agents which are supposed to interact with humans in a human-oriented environment have a notion of spatial units that are also meaningful for humans. Topological regions are such units that are meaningful to humans. We call the units of indoor spaces *areas*. We distinguish between two basic kinds of areas. *Rooms* are spatial areas whose primary purpose is defined by the kinds of actions they afford (see Section 2.5.1). The other major class of indoor areas are *passages* whose primary purpose is to link rooms and provide access to other spatial areas. In Section 5.4.3 we show how this basic distinction provides a basis for situation-aware motion behavior of the robot.

The challenge for intelligent agents is to autonomously build spatial representations that are composed of such areas. The previously mentioned distinction between physical, perceptual and subjective boundaries of topological areas corresponds to a *spatial segmentation* along geometric features versus functional features. In indoor environments, walls are the physical boundaries of areas. They determine the geometric layout of the space they surround. Functional features, as mentioned in Section 2.5.1, can be determined by specific

objects – but also by the spatial layout and the composition of the objects and their surroundings.[1] Similarly, the gateways that link areas can be defined geometrically or on a functional-perceptional basis.

However, as we showed in the previous sections, the sensors of a robot are not particularly geared towards perceiving architectural structures. Neither do computer vision methods exist that allow to visually recognize arbitrary objects – let alone their functional affordances. Currently, the main purpose of robotic exteroceptive sensors is to discriminate free space from physical obstacles, and to provide a means for localizing the robot with respect to local landmarks. It is therefore necessary to make use of other cues to *segment* an environment into topological units.

A special kind of free space are geometrically bounded *gateways*. In a spatial representation that is based upon free space and its inter-connectivity, gateways play an important role in structuring and segmenting free space. In a map that only implicitly represents the boundaries of spatial areas, gateways divide space into regions that belong to one spatial area from regions that belong to other spatial areas. "Cognitively this allows the world to be broken up into smaller pieces" (Chown, 1999). Gateways constitute an important factor for spatial cognition and navigation of autonomous agents in large-scale space (Chown, 2000). Chown et al. (1995) explains the special role of gateways for autonomous robots like this:

> *"In buildings, these [gateways] are typically doorways…Therefore, a gateway occurs where there is at least a partial visual separation between two neighboring areas and the gateway itself is a visual opening to a previously obscured area. At such a [location], one has the option of entering the new area or staying in the previous area."*

Likewise, our approach is based on the assumption of the importance of gateways (especially doorways) for human-compatible spatial representations of human-oriented environments. Later we show how our approach makes use of information about doorways in order to maintain a representation that is composed of rooms and other spatial areas (e.g., corridors).

Hierarchical subdivision of space

One prominent spatial relation we experience physically and abstractly every day is spatial *containment* (see also Section 2.5.2). Egenhofer and Rodríguez

[1] Strictly speaking, the presence of a coffee machine alone does not turn a room into a kitchen – it could as well be a storeroom. The space in the room must afford the preparation of coffee, just as the coffee machine must be reachable and usable.

(1999) consider the space within a room as a small-scale space in which people experience cognitive image schemata, e.g., the *container-surface schema*. However, people routinely employ the same schemata to larger structures, for example when saying "the bench is in the garden" (Lakoff and Johnson, 1999). Similar to objects that are *inside* a room, streets are *in* a city, and several districts form a country. The space around us can thus be decomposed into smaller units, or can combine with other spatial units to larger regions. The container schema can – with a few constraints – also be applied to large-scale space – at least when considering objects of comparable size and similar observation scale (Rodríguez and Egenhofer, 1997).

Containment of objects or spatial units is a productive schema for spatial language (Coventry and Garrod, 2004), and one of the structuring principles in the cognitive map (Stevens and Coupe, 1978; McNamara, 1986). Likewise, hierarchical subdivisions of space are a basic topological relation for *geographical information systems* (GIS) (Marx, 1986; Trainor, 2003).

Topological hierarchies can be expressed as spatial-relation algebras, which, unlike usual computational geometry-based calculations, "rely on symbolic computations over small sets of relations. This method is very versatile since no detailed information about the geometry of he objects, such as coordinates of boundary points or shape parameters, is necessary to make inferences" (Egenhofer and Rodríguez, 1999). This makes them a prime candidate for a basic human-compatible relation to structure and subdivide space.

Conceptually, containment does not form a strict hierarchy. One spatial region can be contained in several different spatial regions, which, in turn, might not be in a containment relation. Consider, for example, an intersection of two corridors. While the intersection itself forms a spatial region, it can also be assumed to be a part of each individual corridor. The representation of spatial abstraction hierarchies is thus rather a *partially ordered set* (poset) (Kainz et al., 1993).

Definition 1 (Partially ordered sets (posets) (Kainz et al., 1993))**.**

Let P be a set. A *partial order* on P is a binary relation $\leq$ on P such that, for every $x, y, z \in P$:

1. $x \leq x$ (reflexive)
2. if $x \leq y$ and $y \leq x$, then $x = y$ (antisymmetric)
3. if $x \leq y$ and $y \leq y$, then $x \leq z$ (transitive)

> A set P with a reflexive, antisymmetric and transitive relation (*order relation*) $\leq$ is called a *partially ordered set* (or *poset*). For every partially ordered set P we can find a new poset, the *dual* of P, by defining that $x \geq y$ is in the dual if $y \leq x \in P$. Any statement about a partially ordered set can be turned into a statement of its dual be replacing $\leq$ with $\geq$, and vice versa. $\geq$ is called the *inverse* of $\leq$. ■

We show later (in Part III) how a hierarchical subdivision of space provides the basic structure for the production and understanding of spatially situated language.

3.1.2 Categorizing space

Aside from the functionality of the cognitive map, another relevant question from cognitive science is how people categorize spatial structures, as illustrated in Section 2.5.1. Categories determine how people can interact with, and linguistically refer to entities in the world. *Basic-level categories* represent the most appropriate name for a thing or an abstract concept. The basic-level category of a referent is assumed to provide enough information to establish equivalence with other members of the class, while distinguishing it from non-members (Brown, 1958; Rosch, 1978). We draw from these notions when categorizing the spatial areas in the robot's *conceptual map*. We are specifically concerned with determining appropriate properties that allow a robot to both successfully refer to spatial entities in a situated dialogue between the robot and its user, and meaningfully act in its environment.

Our work rests on the assumption that the basic-level categories of spatial entities in an environment are determined by the actions they afford. Many types of rooms are designed in a way that their structure and spatial layout afford specific actions, such as corridors, or staircases. Other types of rooms afford more complex actions. These are in most cases provided by objects that are located there. For instance, the concept 'living room' applies to rooms that are suited for resting. Having a rest, in turn, can be afforded by certain objects, such as couches or TV sets. We thus conclude that besides basic geometric properties, such as shape and layout, the objects that are located in a room are a reliable basis for appropriately categorizing that room.

3.2 Representing Space at Different Levels of Abstraction

If an autonomous agent is required to perform navigation tasks, it must have access to low-level spatial representations that are suitable for fine-grained hard-

ware control. These are typically *quantitative* spatial representations, such as *metric coordinate systems*. Metric maps rely on accurately measurable distances and dimensions. The sensors modern robots are typically equipped with, such as time-of-flight cameras or laser range finders, provide quite exact measurements of free and occupied space in the robot's surrounding. Such sensor readings are hence often stored in metric maps of different kinds. Metric maps are also an obvious choice for online avatars because they can have easy access to the virtual world, which typically consists of 3D models.

Humans, on the other hand, use the topological structuring of space to form a more *qualitative* sense of space. This is reflected in natural language, which is full of vague, qualitative spatial expressions. In order to be able to communicate successfully and naturally with humans, autonomous conversational agents must be able to establish such a quantitative spatial understanding on the basis of the low-level maps they can build from their sensory input.

To this end, we present *multi-layered conceptual spatial mapping*. The approach addresses the problems of human-compatible structuring and categorization of space. It comprises spatial representations at different levels of abstraction, ranging from low-level metric maps to symbolic conceptual representations. In Chapter 4 we present reasoning methods that can be performed using such spatial conceptual knowledge. In Part II we show that the implemented approach is suitable for situated action and interaction with humans in human-oriented environments.

The multi-layered conceptual spatial mapping principle has been implemented in two instantiations. The one in Figure 3.4 is the basis for the integrated robotic systems in Chapters 5 and 6. More recently, Pronobis et al. (2010b) presented a refined approach to multi-layered mapping, in which, most notably, the representations of the lower map layers were re-defined. The integrated robotic system in Chapter 7 makes use of this instantiation (illustrated in Figure 3.5).

In the following sections we outline the different spatial representations underlying the individual abstraction layers. The details of the implementation are presented in the chapters of Part II.

3.2.1 Related work

Recently, a number of methods originating in robotics research have been presented that construct multi-layered environment models. These layers range from metric sensor-based maps to abstract conceptual maps that take into account information about objects acquired through computer vision methods. Vasudevan et al. (2007) suggest a hierarchical probabilistic representation of space based on objects. The work by Galindo et al. (2005, 2007) presents an approach containing two parallel hierarchies, spatial and conceptual, connected through

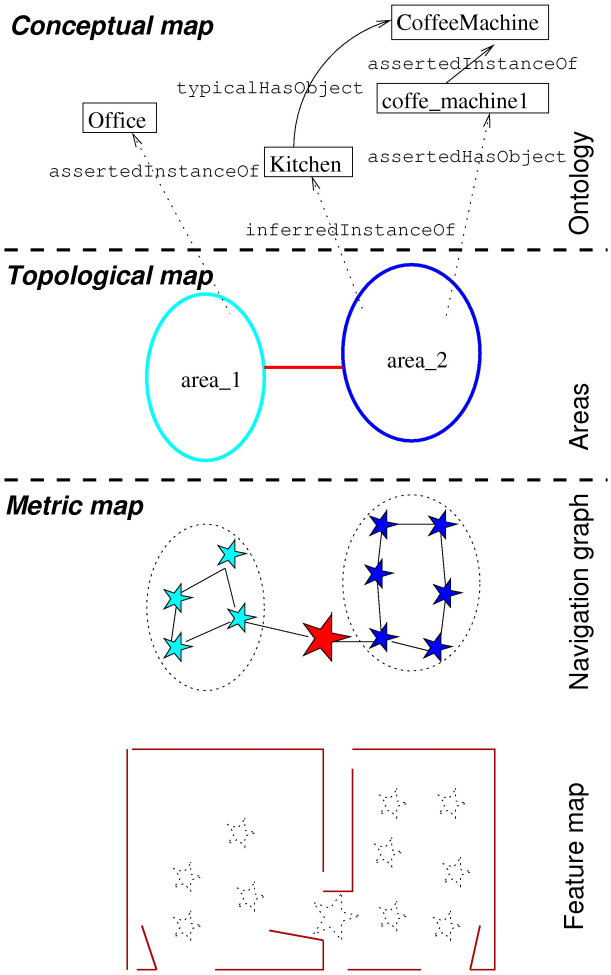


Figure 3.4: Illustration of a multi-layered conceptual spatial map.

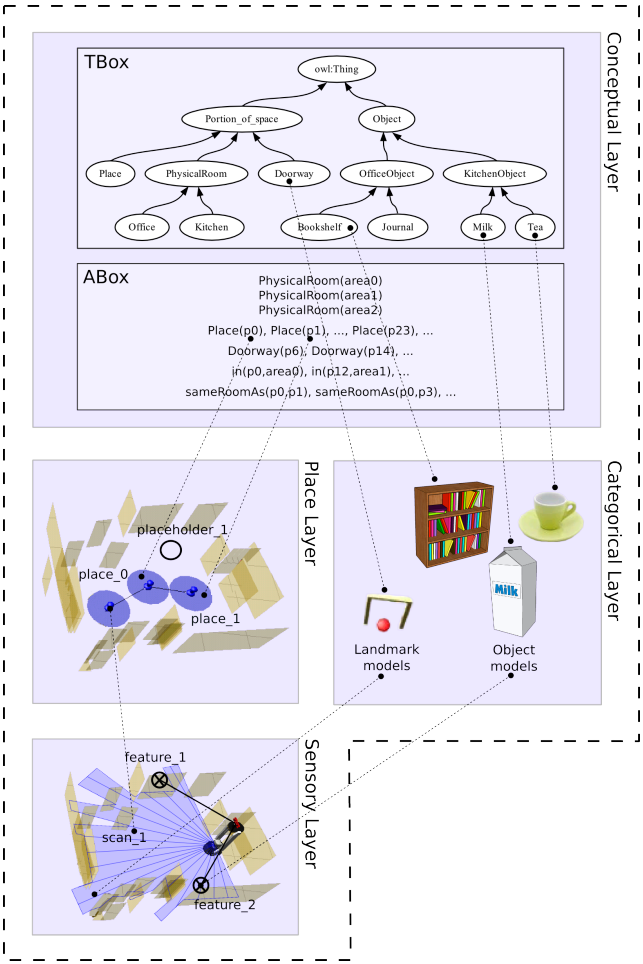


Figure 3.5: COARSE (Cognitive lAyered Representation of Spatial knowledgE) (Pronobis et al., 2010b).

anchoring. Inference about places is based on objects found in them. This approach is based on the Multi-AH-graph model by Fernández and González (2001). The work by Diosi et al. (2005) creates a metric map through a guided tour. The map is then segmented into discrete rooms according to the labels given by the instructor. Furthermore, the *Hybrid Spatial Semantic Hierarchy* (HSSH), introduced by Beeson et al. (2007), allows a mobile robot to describe the world using different representations, each with its own ontology.

3.2.2 The different map layers

In the following, we briefly describe the properties of the individual layers. The conceptual map layer is central to the work presented in this thesis. The other layers, i.e., the metric, navigation, and topological layers will be referred to as the "lower layers" of the spatial model. They are outside the scope of this thesis. While they are important for robot navigation and self-localization, their sole relevance to the work of this thesis is that they provide input to the conceptual layer based on perception of the real world. The robotic systems presented in Part II illustrate implementations of the multi-layered mapping approach using different techniques in the lower layers.

Unless the agent is equipped with a form of external localization – such as robots acting in instrumented environments (which, in turn, are faced with their own challenges (Estrin et al., 2002)), or avatars that operate in the 3D coordinate system of the virtual world – it must be equipped with sensors that allow it to perceive its surroundings. In the simplest case, such sensors are only used to prevent the robot from hitting an obstacle[2] or to enable the robot to move to a fixed target position.[3] This, however, does not amount to much spatial understanding other than a robot-centric frame of reference that captures the here-and-now small-scale space.

An understanding of large-scale space requires that the agent at least be able to represent – i.e., remember and retrieve – landmarks that are outside the currently observable part of space. Some approaches to mapping of large-scale space generate metric maps, ranging from interconnected patches of local maps (Beeson et al., 2010) to larger, global metric maps of the whole operating environment (Frese and Schröder, 2006). In contrast, there are other approaches to mapping of large-scale space that do without local metric maps, but rather represent the positions of landmarks with respect to each other in terms of *control laws* that take the robot from one landmark to another (Kuipers, 2000).

[2] For instance, the e-puck educational robot is equipped with eight infrared (IR) proximity sensors, which measure the presence of nearby obstacles (Mondada et al., 2009).

[3] The iRobot© Roomba© autonomous vacuum cleaner has the capability to find its way to a docking station by sensing the IR signals that the station emits.

Such maps, referred to as either *metric maps* or called the *sensory map layer* serve the principal purpose of allowing the robot to safely navigate its environment while staying localized within its representation of large-scale space. This self-localization can be performed in an absolute frame of reference or in a relative frame of reference with respect to a local landmark. As a result, such maps are essentially representations of *free and reachable* space, rather than faithful models of the architectural structure around that free space.

In order to allow for efficient path planning it is common practice to abstract away from sensor-based metric maps. The first abstraction step is *discretization* of the continuous metric space. Examples of such a discretization are *free-space markers* (Newman et al., 2002) which are used to form a *navigation graph map layer* in the implementations in Chapters 5 and 6. The intermediate map layer used in the robotic system in Chapter 7 is formed out of *places*.

This level of discretization provides a basic notion of the topological structure of an environment. However, the discrete units are not guaranteed to be meaningful to humans. It is thus necessary to aggregate the units of the intermediate layer into *human-compatible spatial units*, such as rooms.

This then provides a *topological partitioning* that can be used for *human-compatible structuring* and *categorization* of space. In this view, the exact shape and boundaries of an area are irrelevant. Basic notions that are represented in such a map are *adjacency* and *connectivity*.

Together, the intermediate discretization layer and the topological layer provide a *symbolic abstraction* over continuous, sensor-based metric data. The symbols correspond to the units of the respective maps (e.g., places, navigation nodes, areas, objects, and landmarks) and the relations that hold between them (e.g., adjacency, inclusion, visibility). These symbols are the basis for the *conceptual map layer*. In the conceptual map, different kinds of symbolic reasoning are used to provide a human-compatible structuring and categorization of space that can be used for situated human-machine interaction.

With each abstraction step, the available spatial information gets coarser, while the conceptual knowledge increases. Apart from immediate adjacency of topological areas, the model is unable to derive a global structure other than containment of one portion of space in another. Specifically this means that the model cannot predict that two known areas are adjacent to each other unless their connectivity has been explicitly recognized. This corresponds, on a smaller scale, to the human performance in novel environments. Imagine the surprise when, e.g., while walking through a large furniture store, one realizes that the bathrooms are behind the bedroom closets. A similar behavior becomes apparent in Chapter 7 when the robot enters a partially explored room through a different door (thus at first believing that yet another new room has been dis-

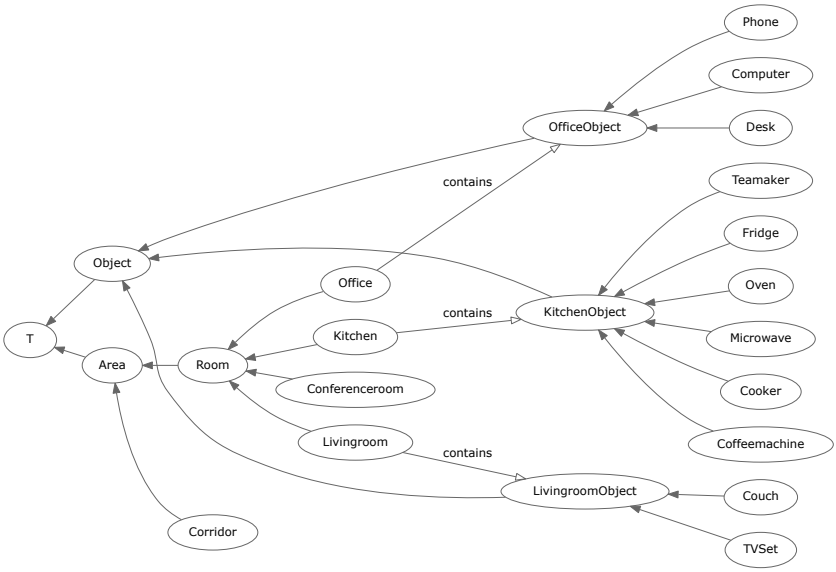


Figure 3.6: Illustration of a part of the commonsense ontology of an indoor office environment. Edges with dark arrow heads denote the taxonomical subclass relation. Edges labeled with contains express that the given subclass of Room is *defined* as containing at least one instance of the pointed-to Object subclass. T stands for the universal top-level concept. See Section 4.2).

covered), and only afterwards arrives at a previously visited place that it knows belongs to an already known room.

In the conceptual map, information stemming from vision and dialogue is related to the spatial units generated in the lower map layers. This allows, for instance, to represent the fact that a specific object was encountered in a specific room together with the information that the human user called that room "the kitchen." Internally, the conceptual map represents information about spatial areas and objects in the environment in an ontological (see Section 4.2) reasoning module. It consists of a commonsense ontology of an indoor environment, which describes *taxonomies* (i.e., *subclass* relations) of room types, and couples room types to typical objects found therein through *contains* relations. Figure 3.6 shows such a commonsense ontology. In the next chapter we give formal definitions of the ontology and its underlying representations and reasoning formalisms.

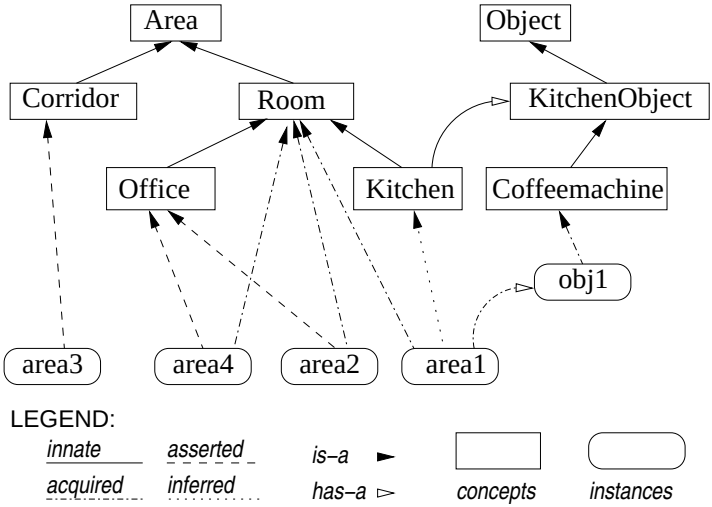


Figure 3.7: Combining different types of knowledge in the conceptual map.

These conceptual taxonomies have been handcrafted and cannot be changed online. However, instances of the concepts are added to the ontology during run-time. Using a reasoner, new knowledge can be inferred. For example, if the robot knows that it is in an area where there is a coffee machine and an oven, it can infer that it can categorize this area as a kitchen. Like this, linguistic references to areas can be generated and resolved (see Part III).

In Chapter 4 we present the different kinds of inferences – including non-monotonic and ontology-based reasoning methods – in the conceptual map in more detail.

3.3 Information Processing

Depending on the origin of a piece of information, we distinguish between *acquired*, *asserted*, *innate*, and *inferred* knowledge. These notions are important for the characterization of the information flow during map acquisition.

- *Acquired knowledge* is derived from the robot's own sensors, including the spatial information encoded in the lower map layers and objects recognized by a computer vision software. The information that an avatar receives from the virtual world engine is another example of acquired knowledge.

- *Asserted knowledge* is provided by another agent, for example a human tutor. It is typically given through verbal input (for example, the tutor might say "you are in the laboratory.").

- *Innate knowledge* is any kind of information that is incorporated into the system in a way that does not allow for on-line manipulation of the knowledge. In our approach, the conceptual ontology (cf. Section 4.2) is an example of innate knowledge.

- Any piece of information that can be derived on the basis of the combination or evaluation of other information provides *inferred knowledge*, such as knowledge inferred by the Description Logic-based reasoning mechanisms in the conceptual map.

Figure 3.7 illustrates how different pieces of information are combined and processed in the conceptual map layer. In the next chapter, we explain the different reasoning mechanisms employed in the conceptual map in more detail.

The individual layers of our multi-layered mapping approach have been implemented within different instantiations of a distributed cognitive architecture. The information processing in these integrated systems is described in more detail in Part II.

3.4 Summary and Outlook

In this chapter, we have presented an approach to multi-layered conceptual spatial mapping for autonomous agents. It addresses the challenges of *structuring space* as well as *categorizing space*, which are prerequisites of *spatial understanding*. Since the kinds of agents we are dealing with have to operate in non-instrumented *human-oriented environments* it is crucial that they be endowed with a *human-compatible* spatial representation in order to engage in meaningful situated dialogues about spatial topics with its human user. Moreover, the presented approach allows for integration with lower-level robotic maps that provide the robot with safe and reliable navigation and control mechanisms, and which take the recent advances in robot sensing, mapping, and motion control into account. In the following chapter, we will explain the representations and formalisms underlying the conceptual map in more detail. In Part II, we will present three integrated robotic systems that make use of the multi-layered mapping approach. In Part III, we will show how the conceptual map can be used as a basis for generation and understanding of natural language in spatially situated human-agent dialogues.

Chapter 4

Reasoning with Changing and Incomplete Conceptual Spatial Knowledge

Summary
In this chapter, we focus on the conceptual map layer of the multi-layered spatial model proposed in Chapter 3. We show how Description Logics can be used to perform inference on a human-compatible symbolic conceptualization of space. We further propose methods for prototypical default reasoning and belief revision to extend the capabilities of autonomous agents. In the subsequent chapters, we will illustrate how these principles can be applied to real, integrated robotic systems.

4.1 Motivation and Background

The kinds of autonomous agents under study in this work operate in dynamic, large-scale environments. These environments are subject to change and cannot be apprehended as a perceptual whole. At the same time, the agents have the possibility to alter the world around them, and to perform actions that allow them to extend their own knowledge. For this to be successful, their knowledge representation must be able to deal with changing and incomplete information.

In Section 4.2, we show how Description Logic-based ontological reasoning can help *overcome the problem of partial information at the sensory-symbol interface*. Nonmonotonic reasoning methods, addressed in Section 4.3, are suitable for drawing more, potentially too strong, conclusions from the knowledge base of an autonomous agent that acts and interacts in a dynamic world. We show how, in particular, default reasoning (Section 4.3.1) can be used for reasoning with *incomplete information*, while belief revision (Section 4.3.2) is suitable for reasoning with *changing information*.

We do not focus on purely epistemological aspects of ontology modeling. While the representations themselves are important, our main focus lies on practical systems; systems that allow an autonomous agent to instantiate these representations, reason with them, and extend them. The emphasis is here on spatial representations that support human-compatible action and interaction. This comprises the capability to verbalize the agent's knowledge, and to relate its internal symbols to entities in the world on the one hand, and to the words that are exchanged with its interlocutors on the other. This also comprises enabling the agent to act and navigate in a human-oriented environment in such a way that its understanding becomes *transparent* to the humans in its surrounding.

To this end, we employ terminological inference, rule-based reasoning, different kinds of nonmonotonic reasoning methods, and and simple spatial reasoning based on topological containment. The challenges of using spatial calculi for qualitative spatial reasoning, and their application to ontology-based spatial representations are beyond the scope of this work.[1] Here we focus on the application of different kinds of ontology-based reasoning methods to the problem of conceptualizing and structuring space for autonomous agents.

4.2 Description Logic-Based Reasoning

We have chosen a Description Logic (DL) based domain ontology (cf. Section 2.5.3) as representation for the conceptual map layer (cf. Section 3.2). It provides a human-compatible symbolic knowledge base that can be used as a basis for interaction with humans. Due to the availability of different OWL reasoning software, its wide acceptance as a standard for ontology engineering, and the resulting re-usability of resources, we adopt OWL-DL as the ontology language for the present work, see Section 4.2.4. DLs comprise a whole family of knowledge representations and associated reasoning formalisms that are based on fragments of first-order logic (Baader et al., 2003).

DL-based knowledge representations distinguish three kinds of knowledge. Firstly, a *taxonomy* of *concepts* represents the so-called terminological knowledge of the domain. This part of the knowledge base is referred to as *TBox* $\mathcal{T}$. Secondly, the *ABox* $\mathcal{A}$ (for assertional knowledge) holds the knowledge about individuals in the domain. Finally, DL ontologies contain a set of *roles* that can hold between individuals, and which are defined over concepts. The hierarchy of roles and their properties are represented in the *RBox* $\mathcal{R}$. While the TBox expresses general, abstract knowledge of the domain, the ABox contains a description of a specific state of affairs of the world. The role definitions and

[1] Katz and Grau (2005) and Grütter and Bauer-Messmer (2007) present extensions to OWL-DL that use the RCC-8 calculus for qualitative spatial reasoning (Cohn and Hazarika, 2001).

role restrictions that are used in concept definitions belong to the TBox. The knowledge about concrete relations between individuals is part of the ABox.

Definition 2 (Ontological knowledge base)**.**
An ontological knowledge base $\mathcal{O}$ consists of a TBox $\mathcal{T}$, an ABox $\mathcal{A}$, and an RBox $\mathcal{R}$: $\mathcal{O} = \mathcal{T} \cup \mathcal{A} \cup \mathcal{R}$. ■

The basic constituents of the TBox are concepts. Another common name for concept is *class*. This gives rise to a more extensional perspective, in which a concept can be represented as the set of its member individuals. Individuals are the basic entities represented in the ABox. We say that an individual a is an *instance* of a concept A if a instantiates A or any of its subconcepts. Likewise, we can also say that a belongs to a class A if it is a member of A or any of its subclasses. An alternative, equivalent formulation that will be used later is to say that a has *type* A.

An important notion we get back to later are *concept definitions*. Using concept constructors and role restrictions, complex *concept descriptions* can be built from atomic concepts. *Named concepts* can then be axiomatically *defined* through concept descriptions. Concepts can also be defined through an extensional enumeration of the individuals belonging to it. In the following, we explain the concept constructors and role restrictions that are relevant for this work. The interested reader can find a more detailed account of the matter in the book chapters by Nardi and Brachman (2003) and Baader and Nutt (2003).

4.2.1 Open-world and non-unique name assumption

Description Logics make two important assumptions regarding the knowledge about the domain they represent.

The *open-world assumption* (OWA) says that the knowledge base can be incomplete with respect to the domain of interest. In particular, the OWA implies that a statement that is not known to be true is not assumed to be false. This effectively means that DLs operate with a three-valued truth system: true, false, and unknown. The OWA is an important prerequisite for a monotonic behavior of DL-based knowledge representations.

In line with the OWA, many DLs make a second assumption, the *non-unique name assumption*. According to this assumption, individuals that have different names are not automatically assumed to be different from each other. They might also be equal.[2] The fact that two individuals are indeed identical – or have different identities – must be explicitly expressed in order to draw more

[2] Frege (1892) uses the famous example of the identity of 'the Morning Star' and 'the Evening Star' with the planet Venus. It is possible to learn and talk about 'the Morning Star' and 'the Evening Star' individually, before recognizing that they are indeed one and the same planet.

conclusions. The non-unique name assumption is crucial for applications where knowledge bases can be fused. OWL-DL implements this assumption in order to account for the decentralized and distributed way in which knowledge is represented in the Semantic Web. The non-unique name assumption is also useful for autonomous agents that have limited perceptual capabilities. For instance, the recognition that an observed object is identical to a previous observation is not trivial to establish. It must thus be possible to accumulate knowledge about both observations independently and, later, once their identity is established, to fuse the knowledge.

4.2.2 DL syntax and semantics

The formal semantics of DL concepts is given by a set-theoretic *interpretation* $\mathcal{I}$ with respect to a domain $\Delta^{\mathcal{I}}$. Concepts are interpreted as sets of individuals, and roles are interpreted as sets of pairs of individuals. According to the OWA, the domain of the interpretation can be infinite. The interpretation of concepts and roles exhibits the relatedness of DL and predicate logic. Since the interpretation assigns to every atomic concept a unary relation, and to every role a binary relation, concepts and roles can be viewed as unary and binary predicates, respectively. Formal definitions of the syntax and semantics of DL concepts and roles are given below. Definition 3 and Table 4.1 on the next page show the syntax and semantics of DL axioms. Definitions 4 and 5 provide the abstract syntax of DL concept and role constructors, respectively, whereas Table 4.2 provides the corresponding semantic interpretations. These definitions cover the subset of DL syntax and semantics relevant for the present work. Baader (2003) provides an exhaustive discussion of DL terminology, along with formal syntax and semantics for DLs of different levels of expressivity.

The taxonomical relations between concepts and between roles are expressed by *terminological axioms*. An *interpretation* $\mathcal{I}$ consists of a non-empty set $\Delta^{\mathcal{I}}$ (the *domain* of the interpretation) and an interpretation function which assigns to every atomic concept A a set $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$ and to every atomic role R a binary relation $R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.

Definition 3 (Terminological axioms (Baader and Nutt, 2003))**.**
Terminological axioms express how concepts and roles are related to each other. There exist two kinds of axioms, *inclusions* and *equalities*. Together they constitute the taxonomy of concepts and roles. Inclusions take the form $C \sqsubseteq D$ or $R \sqsubseteq S$, where C, D are concepts, and R, S are roles. Equality axioms have the form $C \equiv D$ or $R \equiv S$. Their interpretations are given in Table 4.1. ■

Table 4.1: Syntax and semantics of inclusion and equality axioms.

Syntax	**Semantics**	**Name**
$C \sqsubseteq D$	$C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$	concept inclusion
$C \equiv D$	$C^{\mathcal{I}} = D^{\mathcal{I}}$	concept equality
$R \sqsubseteq S$	$R^{\mathcal{I}} \subseteq S^{\mathcal{I}}$	role inclusion
$R \equiv S$	$R^{\mathcal{I}} = S^{\mathcal{I}}$	role equality

Definition 4 (DL concept constructors (Baader, 2003))**.**
The abstract syntax of Description Logic concept descriptions C, D is specified by the following recursive rewrite rules[3]:

$$C, D \rightarrow \quad A \mid \top \mid \bot \mid \neg A \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C \mid \exists R.b \mid \\ \geq nR \mid \leq nR \mid = nR \mid \geq nR.C \mid \leq nR.C \mid = nR.C$$

Table 4.2 lists the inductive definitions for the interpretation. ■

Definition 5 (DL role constructors (Baader, 2003))**.**
Roles, being interpreted as binary relations, can be subject to the usual operations on binary relations (depending on the expressivity of the DL language). In the DL language we use, *OWL-DL*, the following role descriptions can be constructed from a role R:
R^+ (transitive closure), R^- (inverse)
Symmetric roles are specified through an equality axiom between a role and its inverse: $R \equiv R^-$ (symmetry)
Table 4.2 lists the inductive definitions for the interpretation. ■

The previously mentioned *concept definitions* provide a means for formally specifying atomic concepts in terms of complex concept descriptions.

Definition 6 (Concept definitions (Baader and Nutt, 2003))**.**
A concept definition is an equality axiom whose left-hand side is an atomic concept: $A \equiv C$

Concept definitions can thus introduce names for complex but otherwise *anonymous* concepts. ■

[3]For compactness alternative expansions are separated by |. For explanations of the different symbols used, please refer to Table 4.2 on the following page or Section 1.5.

Table 4.2: Syntax and semantics of some DL concept and role constructors.

Concept constructors

Syntax	**Semantics**	**Name**
A	$A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$	atomic concept
$\top$	$\Delta^{\mathcal{I}}$	universal concept
$\bot$	$\emptyset$	bottom concept
$\neg A$	$\Delta^{\mathcal{I}} \setminus A^{\mathcal{I}}$	atomic negation
$C \sqcap D$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$	intersection
$C \sqcup D$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$	union
$\forall R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$	value restriction
$\exists R.C$	$\{a \in \Delta^{\mathcal{I}} \mid \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$	existential quantification
$\exists R.b$	$\{a \in \Delta^{\mathcal{I}} \wedge b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}$	existential quantification with restricted value
$\geq nR.\top$	$\{a \in \Delta^{\mathcal{I}} \mid\mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\} \mid \geq n\}$	cardinality restriction
$\leq nR.\top$	$\{a \in \Delta^{\mathcal{I}} \mid\mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\} \mid \leq n\}$	
$= nR.\top$	$\{a \in \Delta^{\mathcal{I}} \mid\mid \{b \in \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\} \mid = n\}$	

Role constructors

Syntax	**Semantics**	**Name**
R	$R^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$	atomic role
R^+	$\bigcup_{n \geq 1} (R^{\mathcal{I}})^n$	transitive closure
R^-	$\{(b,a) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}} \mid (a,b) \in R^{\mathcal{I}}\}$	inverse

Let us consider some examples from an indoor mapping domain as discussed in the previous chapter. Figure 4.1 shows the TBox $\mathcal{T}_{indoor}$ of the commonsense ontology of an indoor environment. Example 1 gives an example of a concept definition in $\mathcal{T}_{indoor}$. It defines kitchens as those rooms that contain at least one kitchen object.

(1) $\mathsf{Kitchen} \equiv \mathsf{Room} \sqcap \exists \mathsf{contains}.\mathsf{KitchenObject} \in \mathcal{T}_{indoor}$

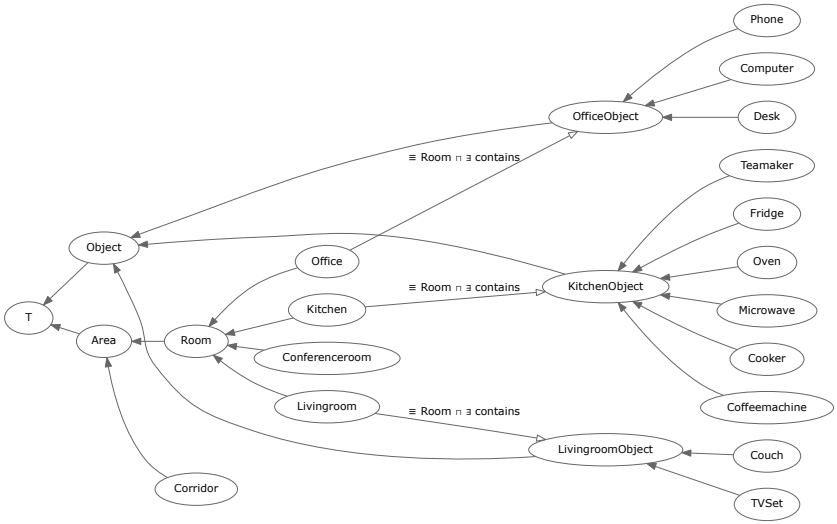


Figure 4.1: Part of the commonsense ontology of an indoor office environment (see also Figure 3.6) in $\mathcal{T}_{indoor}$. Edge labels express concept definitions like in Example 1.

We have already made use of a role: contains. In order to capture the kinds of spatial relations we are interested in (i.e., topological containment), we need to extend the RBox $\mathcal{R}_{indoor}$ further. Example 5 specifies another role, in, as inverse role of contains. Making use of transitivity and inversion, we represent topological inclusion of spatial entities along the definition of a poset, see Definition 1 on page 38. Reflexivity is of minor importance for our purposes. It is left out due to implementational considerations. Anti-symmetry per se is not supported by the Description Logic language we are using (i.e., OWL-DL, see Section 4.2.4).[4]

(2) $\mathsf{in} \equiv \mathsf{contains}^{-} \in \mathcal{R}_{indoor}$

(3) $\mathsf{in} \equiv \mathsf{in}^{+} \in \mathcal{R}_{indoor}$

(4) $\mathsf{contains} \equiv \mathsf{in}^{-} \in \mathcal{R}_{indoor}$

(5) $\mathsf{contains} \equiv \mathsf{contains}^{+} \in \mathcal{R}_{indoor}$

[4]The OWL 1.1 draft contains a construct for anti-symmetry, which rather expresses asymmetry. The kind of anti-symmetry required by Definition 1 on page 38 could be expressed through inference rules (see Section 4.2.6) instead.

4.2.3 DL inferences

DLs are based on a decidable subset of first-order predicate logic. The set-theoretical semantic interpretation of DLs is equivalent to the semantics of predicate logic. Similar to a set of predicate logic axioms, the axioms in a DL knowledge base represent explicit as well as implicit knowledge about the domain. Using logical inference, this implicit knowledge can be made explicit.

Definition 7 (Entailment in DL)**.**
We say that a set of axioms $\mathcal{O}_1$ *entails* another set of axioms $\mathcal{O}_2$, $\mathcal{O}_1 \models \mathcal{O}_2$, iff all interpretations satisfying $\mathcal{O}_1$ also satisfy $\mathcal{O}_2$. ■

Different reasoning engines (henceforth simply called *reasoners*) for the different variants of DL exist. The task of DL reasoners is to perform certain kinds of inferences in both the TBox and the ABox that follow from the semantics of the respective DL formalism used. These inferences are typically based on two kinds of conditions: necessary and sufficient conditions. If the *sufficient* conditions are met, the truth of a conclusion is warranted. Conversely, the falsity of *necessary* conditions imply the falsity of their conclusions. This means that necessary conditions alone do not suffice to guarantee the truth of their conclusions, while sufficient conditions might be by no means necessary for their conclusions. Often, a number of necessary conditions together form a sufficient condition. In such a case, the different conditions are *individually necessary*, but *jointly sufficient*. The task of a reasoner can be paraphrased as inferring all the knowledge that is only implicitly represented in the knowledge base.

The most basic TBox inference is *subsumption checking* between concepts. This inference turns a set of concept definitions into a hierarchical taxonomy in which concepts are related with a subclass/superclass relation. Given the example above, a DL reasoner can infer that Kitchen is a subclass of Room.

(6) $\mathcal{T}_{indoor} \models \mathsf{Kitchen} \sqsubseteq \mathsf{Room}$

In the ABox a DL reasoner establishes concept membership of individuals, the so-called *instance checking* mechanism (Nardi and Brachman, 2003). Individuals are introduced by naming them and asserting properties about them. Such properties include their concept membership and roles that they take part in. Concept membership is expressed using so-called *concept assertions*. $C(a)$ denotes that a belongs to the interpretation of C. A *role assertion* like $R(a, b)$ denotes that the tuple (a, b) belongs to the interpretation of R, cf. Definition 4.

Definition 8 (Satisfiability of assertions (Baader and Nutt, 2003))**.**
$\mathcal{I}$ *satisfies* $C(a)$ if $a^{\mathcal{I}} \in C^{\mathcal{I}}$.
$\mathcal{I}$ satisfies $R(a, b)$ if $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in R^{\mathcal{I}}$. ■

Continuing our example, we assert the following facts about our domain:

(7) The example ABox $\mathcal{A}_{ex}$:
$\mathsf{Room}(\mathsf{AREA1})$
$\mathsf{Oven}(\mathsf{OBJ1})$
$\mathsf{contains}(\mathsf{AREA1}, \mathsf{OBJ1})$
$\mathsf{Office}(\mathsf{AREA2})$

The reasoner can then infer that OBJ1 is also an instance of KitchenObject and hence AREA1 is an instance of Kitchen[5]. Additionally, the reasoner can infer that AREA2 also instantiates Room:

(8) $\mathcal{T}_{indoor} \cup \mathcal{A}_{ex} \models \mathsf{KitchenObject}(\mathsf{OBJ1}),$
$\mathsf{Kitchen}(\mathsf{AREA1}),$
$\mathsf{Room}(\mathsf{AREA2})$

In other words, the subclass relation between Office and Room establishes a sufficient condition for an instance of Office to also instantiate Room. Conversely, being a Room instance consists a necessary condition for being an instance of Office or Kitchen, but it is not sufficient. The concept definition in Example (1) expresses the jointly necessary and sufficient conditions under which individuals belong to the named concept Kitchen. Generally speaking, the subclass relation $A \sqsubseteq B$ expresses a necessary condition $\neg B(x) \rightarrow \neg A(x)$. Put differently, if $A(x)$ is true then $B(x)$ must not be false, or else the knowledge base is inconsistent. Its inverse constitutes a sufficient condition $A(x) \rightarrow B(x)$. Necessary and sufficient conditions $A(x) \leftrightarrow B(x)$ thus constitute mutual subsumption $A \sqsubseteq B \wedge B \sqsubseteq A$, which is the same as full equivalence $A \equiv B$ (cf. Example (1)).

The iterative process in which the different DL reasoning services infer new facts from the TBox, ABox, and RBox axioms is called *expansion*. In pure Description Logics, this is a monotonic process, i.e., the *full expansion* of a knowledge base results from repetitive applications of the DL rules, irrespective of their order. Unless the knowledge base is *inconsistent*, there is exactly one full expansion for a given knowledge base.

4.2.4 OWL and RDF

For the work presented in this thesis, we represent our ontologies in the *Web Ontology Language OWL*, more specifically in its sub-language OWL-DL (Smith et al., 2004). OWL ontologies make use of RDF and RDF Schema (RDFS)

[5] Of course, an OWL-DL reasoner would establish the full type hierarchy for both individuals along the transitive subsumption axis (cf. Figure 4.1). This is left implicit here for ease of reading.

constructs (Brickley and Guha, 2004), as well as XML Schema (XSD) data types. RDF can express binary relations, so-called subject-predicate-object *triples*. RDF itself is composed of three basic elements: resources, properties, and classes. OWL makes use of RDF and RDFS constructions and extends their vocabulary. Consequently, roles are called *properties* in OWL. XSD defines a set of primitive data types, such as string, boolean, and several numerical data types, and provides constructors (list, union, and restriction) for defining complex data types. In OWL, properties that do not range over individuals (so-called object properties) but over data types (so-called datatype properties) are defined in accordance with XSD.

OWL/RDF can either be written in XML syntax or in a more compact notation of RDF triples, e.g., N-Triples (Beckett and Barstow, 2001). Triples encode the RDF graph structure, whereby the subject and object resources are the nodes and the predicate resources are the labeled edges of the RDF graph (RDF Working Group, 2004). In RDF triple notation, XSD data values are typed using a `^^xsd:DATATYPE` suffix, where `DATATYPE` is replaced with the actual data type (e.g., `boolean` or `string`). Further down, Example 12 and Example 13 on page 62 show RDF triples with datatype properties.[6]

Definition 9 (RDF triples (Powers, 2003))**.**

> Each RDF triple is made up of subject, predicate and object.
> Each RDF triple is a complete and unique fact.
> An (RDF) triple is a 3-tuple, which is made up of a subject, predicate and object (…).
> Each RDF triple can be joined with other RDF triples, but it still retains its own unique meaning, regardless of the complexity of the models in which it is included. ■

In the following we make use of a simplified N-Triples notation (Beckett and Barstow, 2001), where namespaces URIs are omitted in favor of short namespace prefixes, and namespace prefixes, in turn, might be omitted for examples that are local to the work presented herein.

We write RDF triples like this:

`subject predicate object .`	(namespaces omitted)
`Kitchen rdfs:subClassOf Room .`	(with RDF-Schema namespace prefix)
`area1 rdf:type Room .`	(with RDF namespace prefix)
`contains owl:inverseOf in .`	(with OWL namespace prefix)
`_:b1 rdfs:subClassOf owl:Thing .`	(with a *blank node*)

[6] In the rest of the thesis, however, we will omit XSD suffixes for ease of reading. Data values are assumed to be implicitly typed with their usual, simple data types.

These graphs can be queried using the SPARQL Protocol and RDF Query Language (Prud'hommeaux and Seaborne, 2008). Without going into the details, a basic SPARQL query consists of a query clause and a graph pattern clause. In the following examples the query clause consists of a `SELECT` clause that specifies variables that will be presented in the query results, and a `WHERE` clause that specifies the graph pattern that should be matched against the data graph. The graph pattern can consist of any number of triples. Example 9 shows a query that matches all RDF triples, whereas Example 10 queries the knowledge base for all ABox individuals. More details about the SPARQL syntax can be found in the official W3C recommendation document by Prud'hommeaux and Seaborne (2008).

(9) `SELECT ?x ?y ?z WHERE {?x ?y ?z .}`

(10) `SELECT ?x WHERE {?x rdf:type owl:Thing .}`

It is important to note that OWL-DL reasoners perform open-world reasoning, whereas SPARQL performs closed-world querying. We show later (see Section 4.3.2) how inferences drawn from the absence of counter-evidence result in a non-monotonic behavior that necessitates *belief revision* mechanisms.

In the previous sections, we introduced the Description Logic axioms as well as the concept and role constructors that are part of OWL-DL. Table 4.3 on the following page shows the correspondence between OWL axioms and DL axioms, and Table 4.4 on the next page contains the OWL equivalents of the DL concept and role constructors. As can be seen in the namespace prefixes, OWL makes use of RDF constructions wherever available.

Restrictions are written as `owl:Restriction` elements with a declaration of `owl:onProperty` that specifies which role is subject to a restriction (left out of the table). The different kinds of restrictions are existential quantifications (`owl:someValuesFrom` and `owl:hasValue`), value restrictions (`owl:allValuesFrom`), and cardinality restrictions. For such cardinality restrictions `owl:minCardinality` can be used to specify a lower bound and `owl:maxCardinality` an upper bound. They can be combined to specify cardinality intervals. If an exact cardinality is to be expressed, `owl:cardinality` can be used instead of specifying an interval with identical lower and upper bounds. Qualified cardinality restrictions (`owl:valuesFrom`) are a non-standard extension to OWL-DL. They are available as suggestions from the Editor's Drafts of the Best Practice and Deployment Documents (Rector and Schreiber, 2005).

Domains and ranges of roles are also straightforwardly represented (`rdfs:domain` and `rdfs:range`, respectively). The individuals in the ABox are asserted to instantiate a concept using `rdf:type` statements.

Table 4.3: OWL-DL axioms.

OWL Axiom	**DL Syntax**	**Example**
`rdfs:subClassOf`	$C \sqsubseteq D$	Oven $\sqsubseteq$ Object
`owl:equivalentClass`	$C \equiv D$	Kitchen $\equiv$ Room $\sqcap \exists$contains.KitchenObject
`owl:disjointWith`	$C \sqsubseteq \neg D$	Room $\sqsubseteq \neg$Object
`owl:sameAs`	$\{a\} \equiv \{b\}$	KITCHEN1 $\equiv$ ROOM128
`owl:differentFrom`	$\{a\} \sqsubseteq \{\neg b\}$	KITCHEN1 $\sqsubseteq \neg$KITCHEN2
`rdfs:subPropertyOf`	$R \sqsubseteq S$	in $\sqsubseteq$ topoIncludes
`rdfs:equivalentProperty`	$R \equiv S$	in $\equiv$ locatedIn
`owl:inverseOf`	$R \sqsubseteq S^-$	contains $\sqsubseteq$ in$^-$
`owl:transitiveProperty`	$R^+ \sqsubseteq R$	in$^+ \sqsubseteq$ in
`owl:symmetricProperty`	$R \equiv R^-$	equals $\equiv$ equals$^-$

Table 4.4: OWL-DL constructors (`owl:` namespace omitted).

OWL Constructor	**DL Syntax**	**Name**
`Thing`, `Nothing`	$\top, \bot$	universal and bottom concept
`complementOf`	$\neg A$	negation
`intersectionOf`	$C \sqcap D$	intersection
`unionOf`	$C \sqcup D$	union
`oneOf`	$\{a_1, \ldots, a_n\}$	enumeration
`allValuesFrom`	$\forall R.C$	value restriction
`someValuesFrom`	$\exists R.C$	existential quantification
`hasValue`	$\exists R.b$	exist. quant. with restr. value
`maxCardinality`	$\geq nR$	cardinality restriction
`minCardinality`	$\leq nR$	
`cardinality`	$= nR$	
`maxCardinality` + `valuesFrom`	$\geq nR.C$	qualified cardinality restriction
`minCardinality` + `valuesFrom`	$\leq nR.C$	
`cardinality` + `valuesFrom`	$= nR.C$	

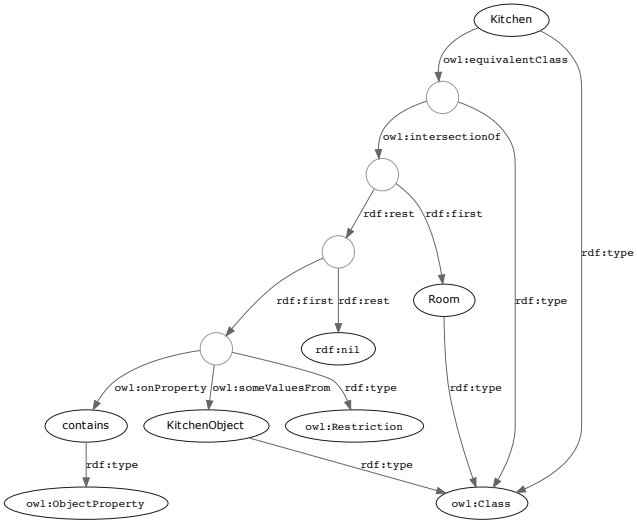


Figure 4.2: RDF graph for a part of $\mathcal{T}_{indoor}$. Empty circles denote blank nodes.

Complex concept descriptions correspond to anonymous classes that can be named through equality axioms. In XML syntax, such class definitions are stored as nested elements, whereas in RDF anonymous classes correspond to blank nodes in the graph. Figure 4.2 shows the RDF graph that corresponds to Example 1 on page 54.

Using OWL-DL ontologies has a number of advantages. First of all, OWL-DL is widely used and accepted due to its status as a W3C standard. This means that many available ontologies – including upper ontologies and domain ontologies alike – are represented in OWL. Furthermore, there exists a large number of reasoners which allow for automatic reasoning over OWL-DL ontologies. These systems offer the aforementioned inference mechanisms (subsumption and instance checking) and in some cases (e.g., forward chaining approaches, see Section 4.2.6) compute fully materialized knowledge bases. This means that ABox, TBox, and RBox contain all facts that are entailed through the OWL-DL semantics by the asserted facts in the ontology definitions. Another advantage of using OWL/RDF is that, on top of such a fully materialized triple store, other inference rules that go beyond the power of OWL-DL can be applied. One could, e.g., specify a domain-specific set of rules that extend the domain-

independent OWL-DL inference rules in order to draw more, domain-specific conclusions. Depending on the actual set-up, facts derived from applying such rules could then again give rise to further OWL-DL based inferences and so forth.

4.2.5 Marking basic-level concepts

One important aspect of human-compatible categorization is the notion of *basic-level categories*, as introduced in Section 2.5.1. If we want to express that a certain class in the ontology corresponds to such a basic-level category, it is important to ensure that this only holds for the respective class and not for its subclasses (unless explicitly stated otherwise). We hence need a way of marking a class as "basic-level" outside the automatic subsumption checking of OWL-DL reasoners. OWL-DL offers so-called *annotation properties*, which are mutually disjoint with the usual terminological properties (i.e., `owl:ObjectProperty` and `owl:DatatypeProperty`), and which must not be used in property axioms (Bechhofer et al., 2004).

The classes in the TBox that are considered basic-level categories can thus be marked by adding RDF triples like the following to the knowledge base. In Example 11, we define an annotation property isBasicLevel which takes Boolean values. Example 12 and Example 13 illustrate how a named class can be marked as a basic-level category.

(11) `isBasicLevel rdf:type owl:AnnotationProperty .`
`isBasicLevel rdfs:range xsd:boolean .`

(12) `Office isBasicLevel "true"^^xsd:boolean .`

(13) `Couch isBasicLevel "true"^^xsd:boolean .`

When querying for basic-level categories we can then perform closed-world queries for classes that have the annotation property isBasicLevel with a boolean value of true (illustrated in Example 14). This has the advantage that the process of creating, maintaining, and editing the ontology can be kept simple: only those classes that are deliberately chosen as basic-level categories must be annotated. The classes for which this annotation property is undefined are assumed to be non-basic-level.

(14) `SELECT ?x WHERE {?x isBasicLevel "true"^^xsd:boolean .}`

4.2.6 Rule-based reasoning

The conceptual map layer represents symbolic knowledge derived from abstractions of lower-level sensor input. DL-based inference combines knowl-

edge about the individuals in the domain, including the types that are derived from the sensor input, and their relationships. There are, however, other regularities that help structure the knowledge that are beyond pure terminological reasoning. Some of these regularities can be expressed in terms of conditional *inference rules*.

Such an inference rule consists of a set of *premises* and a set of *consequences*. It is possible to represent the semantics of lathe parts of OWL-DL in terms of TBox, RBox and ABox inference rules. Using additional rules, it is possible to extend the inferences that are supported by the conceptual map. A reasoner that performs rule-based inference on a knowledge base is called *rule engine*.[7] Rules can be expressed in an abstract syntax that resembles first-order logic conditionals:

Definition 10 (Inference rules)**.**

Let P and Q denote well-formed formulae in the syntax of the rule engine. A rule is a conditional of the form: $P \Rightarrow Q$

P and Q are the *premises* and *consequences* of the rule, respectively. Premises are also called *body terms*, conclusions are called *head terms*. The set of premises can also be referred to as the *antecedent* of a rule, whereas the set of conclusions can be called its *consequent*. Both can – depending on the rule engine – contain procedural primitives for, e.g., evaluation of conditionals, execution of an output operation, or arithmetic operations. ■

There exist different approaches to rule-based reasoning. One distinction of the approaches is between *forward chaining* and *backward chaining*, and it characterizes the direction and order in which the rules are evaluated (Russell and Norvig, 2003). Forward chaining evaluates a rule if its premises are fulfilled, i.e., if all its premises are known to be true with respect to the current explicit knowledge. Backward chaining , in contrast, starts from an assumption (i.e., a set of hypotheses, also called goals) and tries to find a list of rules that will establish the truth of the goals. It does so by inspecting the consequents of the available rules. If a rule is found whose consequent fulfills one or multiple goals and whose antecedent is known to be true, the consequent is added to currently known facts and removed from the goal set. If the consequent of a rule matches a current goal, but its antecedent is not known to be true, backward chaining adds the antecedent to the goal set. This procedure is repeated until either no

[7] Other names for rule engine that stress different properties are *production (rule) system* (where the nature of rules *producing* new knowledge is highlighted) and *pattern-directed inference system* (which stresses the pattern matching necessary to select applicable rules).

more goals are present, in which case the truth of the hypotheses is established, or no additional rules can be selected in order to derive the truth of a hypothesis from the known facts. Another way of saying that a rule antecedent or consequent (in case of forward or backward chaining, respectively) matches is to say that a rule *fires*. Forward chaining is characterized by a *data-driven* approach: the presence of known facts gives rise to concluding new facts, which can then be the basis for additional inferences, and so on. The approach of backward chaining, on the other hand, is *goal-driven*: rules are only evaluated if they support the proof of a given hypothesis. As a consequence, backward chaining will terminate for a larger set of knowledge and rule bases, because it is less likely to run into infinite loops of firing rules. Backward chaining also requires less memory space, but is slower, while forward chaining is faster and consumes more memory space.

Different rule engine algorithms address the issues of optimizing the *pattern matching* strategy for runtime and memory efficiency. A well-known and widely-used pattern matching algorithm for rule engines is the RETE algorithm by Forgy (1982). Furthermore, practical implementations of rule systems must also address the issues of *conflict resolution* and *truth maintenance*. Conflict resolution becomes necessary, for instance, in order to determine the order in which to execute the currently applicable rules – which can have a crucial influence on the efficiency of the rule engine. Truth maintenance becomes necessary if the knowledge base can be modified, or if there are rules whose consequent might invalidate the antecedent of a rule that was already applied. Truth maintenance is a nonmonotonic process that is related to the notion of *belief revision*, which we explain in more detail together with other challenges that arise from reasoning under changing knowledge in Section 4.3.

Using a general rule engine in addition to DL-based reasoning has the advantage that some more domain-specific knowledge can be taken into account for drawing inferences. For example, the data-driven forward chaining approach fits well with the bottom-up map acquisition process in Section 5.3.5. The system can then perform automatic inferences (in addition to DL-based instance and subsumption checking) using the symbols and facts that are asserted from the lower levels. We make use of inference rules for establishing prototypical defaults (see Sections 4.3.1 and 6.2.3) and for maintaining symbols for spatial areas (see Section 7.2.3).

4.3 Nonmonotonic Reasoning

Robotic systems and avatars alike are faced with two challenges for their knowledge representation. For one, the knowledge base cannot always be assumed

to be complete. In fact, it will inevitably be incomplete at any point. Secondly, realistic environments are dynamic. States and positions of things in the world change over time – both actively and passively. Moreover, the agent's perception of the world might be incomplete or error-prone and thus an agent's representation of the world might be initially false and only over time become more accurate. A spatial knowledge base for autonomous agents should thus be able to address these two challenges: *reasoning with incomplete information*, and *reasoning with changing information*. There are two major kinds of nonmonotonic logics that address these two challenges, respectively (Antoniou, 1997).

Default Logic, introduced by Reiter (1980), is an approach to derive more, but not necessarily true, conclusions from a knowledge base. Special inference rules, called *defaults*, that represent *commonsense knowledge* are applied to a set of certain facts. This allows the inference of more knowledge than represented by the known, certain facts. This is a principle that is used commonly in human cognition. "Reasoning with prototypes is, indeed, so common that it is inconceivable that we could function for long without it" (Lakoff and Johnson, 1999). We show how an autonomous agent can make use of such default assumptions when operating with incomplete information in Section 4.3.1.

Belief Revision, on the other hand, provides mechanisms for reasoning with changing information (Gärdenfors, 1988, 1992; Nebel, 1989). This is the case, e.g., in a world that is not static, or if the agent acquires new information that invalidated older, potentially erroneous information. In Section 4.3.2 we show how an autonomous agent can employ nonmonotonic reasoning to "take back conclusions that turned out to be wrong and for deriving new, alternative conclusions instead" (Antoniou, 1997).

It is important to note that reasoning with changing information can become necessary due to two very different reasons. For one, in a dynamic world things change. This means that new facts can become true while old facts might cease to be true. The second kind of changing information is induced by the agent itself. The agent might make erroneous observations that are only later corrected. In our approach, we are focusing on representing the current state-of-affairs and on keeping this representation as accurate as possible over time. The distinction whether an old fact became invalid because the world has changed or because the agent recovered from a mistake is therefore of less importance. Such a differentiation can only be adequately represented in a knowledge base that explicitly represents the states of facts over time. Krieger et al. (2008), for instance, present a diachronic representation of facts about entities and how they correspond over time. Their approach also makes use of OWL-DL ontologies, which they augment with temporal information. Such extensions must

either make use of reification (see also page 72), or extend the representation from triples to n-ary tuples. Reasoning with temporal information also requires further inference rules in addition to standard OWL-DL (Krieger, 2010a,b).

Autonomous agents operating in large-scale space, however, are faced with the problem of partial observability of the world – and therefore might encounter the famous *frame problem* (McCarthy and Hayes, 1969) when trying to adequately model the state of their environment over time. For this work, we thus focus on methods that allow autonomous agents to maintain a faithful representation of their spatial environment with respect to the current state-of-affairs only.

4.3.1 Default reasoning

Default Logic (Reiter, 1980) is a family of nonmonotonic logics. In a nutshell, Default Logic allows to draw *risky* (i.e., potentially false or contradicting) conclusions from a set of certain, but possibly incomplete facts using a special kind of "rules of thumb" called *defaults* (Antoniou, 1997). Inference from defaults differs from usual entailment in that defaults permit the derivation of their consequences based on the absence of counter-evidence for their truth. In combination, the set of certain facts W and the set of defaults D constitute a *default theory* $T = (W, D)$.

The standard syntax of a default δ is:

$$\delta = \frac{\alpha : \beta}{\gamma}$$

α, β, γ are first-order logic formulae. α is the *prerequisite* of the default rule, β is called the *justification*, and γ is its *consequent*. Informally speaking, a default δ can be interpreted like this: if α is true, and if it is consistent to assume β, then conclude γ. This definition, however, does not yet capture the crucial aspect that the application of defaults may alter the knowledge base and thus influence further default applications. Following Antoniou (1997), here is a more precise, formal definition of the semantics of Default Logic.

Definition 11 (Formal semantics of Default Logic (Antoniou, 1997))**.**

> If α is currently known, and if β is consistent with the current knowledge base, then conclude γ. The current knowledge base E is obtained from the facts and the consequents of some defaults that have been applied previously.

$\delta = \frac{\alpha : \beta}{\gamma}$ is *applicable* to a deductively-closed set of formulae E iff $\alpha \in E$ and $\neg\beta \notin E$. The state of knowledge base E at time t is expressed by a subscript E_t. Only one default δ can be applied to E at a time. After applying δ to E_t, the new current knowledge base E_{t+1} results from adding γ to E_t. Defaults that were previously applicable (i.e., to E_t) might not be applicable anymore to E_{t+1}. New defaults, however, might become applicable to E_{t+1}. ■

Satisfiability within a default theory means that a formula must either follow from the facts (i.e., the certain information) or from the consequents of the defaults (i.e., other possible conjectures) that have been evaluated so far. *Extensions* E of a default theory T are sets of possible beliefs about the domain that are consistent with T. The order in which defaults are applied matters. It is, for example, possible that a consequent of one default negates the justification of another default. An extension is always a maximal realization of a possible world, i.e., a state in which no more defaults can be applied. Multiple extensions are possible for the same set of facts and defaults.[8]

Sometimes there are defaults that intuitively and intendedly correspond to more general principles, which could be given up more easily in case a more specific default applies. A common improvement of standard Default Logic is thus to allow for priorities over defaults. This way some order of evaluation of the defaults can yield extensions that correspond more to the intuition behind a default theory. We refer the interested reader to the textbook by Antoniou (1997) and the seminal article by Reiter (1980) for further and more detailed discussions about the different kinds of Default Logics.

A special form of default reasoning is *prototypical reasoning*, which expresses typical properties of instances of a concept (see also Section 2.5). This notion is closely related to the intuition behind the ontological knowledge representation we chose for our conceptual spatial knowledge base. There, we use ontology-based reasoning in order to endow autonomous agents with commonsense knowledge about the spatial organization of indoor environments. Defaults are a way of expressing a different kind of commonsense knowledge. It would be possible to define a set of default rules that make use of the facts and predicates in the ontology and extend the DL-based model with further conclusions. However, the possibility to combine ontological resources, and the ease

[8] Note that there may be default theories that don't have an extension because they contain contradicting defaults. There exist different approaches that attempt to construct consistent default theories from contradictory defaults, such as the computation of weak extensions (Lévy, 1993).

with which existing ontologies can be edited and extended would make the task of maintaining a separate set of default rules consistent with the representations chosen in the ontology a tedious one. Instead we show how generalized introspective mechanisms can be applied to derive defaults from existing OWL-DL ontologies in a principled way.

As stated previously, default knowledge can help infer new facts from incomplete knowledge. There are many cases, in which an autonomous agent should be able to act on incomplete knowledge. This pertains especially to situations in which such an agent is capable of extending its own knowledge. The agent could then attempt to verify or falsify default assumptions through its own perceptive abilities. Another use case are automatic planning algorithms. These usually rely on a number of preconditions and intermediate conditions in order to find a complete plan. Defaults can help overcome situations in which the agent would otherwise be unable to come up with a plan.

Let us start with the following example that expresses the commonsense knowledge that ovens are usually found in kitchens:

$$\delta_{oven} = \frac{Oven(x) \wedge Kitchen(y) : in(x,y)}{in(x,y)} \tag{15}$$

The above default contains free variables. It is a so-called *open default* that represents a set of defaults, where all variables are assigned values. Practically only those substitutions are considered for which the prerequisite is satisfiable, i.e., in our case only oven instances would be used to substitute the free variable x in the first place. The same holds for the other free variable y. Note that this explicitly rules out hypothesizing about unknown *individuals*. Nevertheless such a *closed default* would allow an autonomous agent to hypothesize about the whereabouts of certain objects in case their existence can be assumed. The autonomous agent mentioned before could use this default knowledge to come up with an informed guess where to look first for an oven. This can be helpful both for the purely epistemic goal of achieving a better and more complete knowledge of the world, and for executing a task, such as to fetch a cake from the oven. We show later in more detail how these defaults can be used for goal-directed knowledge gathering, and planning of complex actions with deferred validation of default assumptions.

Deriving defaults from an OWL-DL ontology

Ontologies describe *the way things are.* They describe the concepts within a specific domain, and how they are related. Based on the notion of concept definition, a DL reasoner can infer which (named) classes an individual instantiates.

This is a crisp inference based on facts that are known to be true. This especially means that the `rdf:type` relation between an individual and a class can be inferred from class definitions, but other relations cannot. Only properties that generalize an asserted property can be inferred to hold.

Kolovski et al. (2006) present an implemented approach to extending an OWL-DL reasoner with the capability to apply default reasoning. Their intention is to extend the expressiveness of OWL-DL so that it is possible to represent and reason with terminological default theories. Their evaluation, however, shows that such an approach quickly degrades with the number of possible extensions. In comparison, we are interested in automatically *deriving* default knowledge – more specifically prototypical knowledge – that is implicitly already represented in OWL-DL knowledge bases. Our approach does not require a full-fledged default reasoner. We rather show how introspective mechanisms can be used to exploit standard OWL-DL ontologies for inferring prototypical facts. Moreover, we propose a way of representing such prototypical facts in the same knowledge base as other OWL-DL facts – while ensuring that the default knowledge does not interfere with the agent's innate, acquired, asserted, or inferred crisp knowledge.

Within our indoor ontology example, the presence of an oven in a room legitimates the conclusion that it is a kitchen. If, on the other hand, the agent were told that there exists a coffee machine somewhere, nothing else could be inferred. Using DL-based reasoning alone, it is impossible to apply the commonsense knowledge that coffee machines are *usually* found in kitchens in order to *surmise* that this coffee machine can be found in the kitchen. It simply does not logically follow from the model. And, in fact, such an assumption might turn out to be wrong. Nevertheless it is exactly this kind of commonsense knowledge that a human would apply in a similar situation: unless she knows that there is no coffee machine in the kitchen, this is where she would start looking for one.

Most autonomous agents that operate in human-compatible environments will be faced with situations in which they need to act upon incomplete knowledge. In the example, our agent might be told to fetch a coffee from the coffee machine. In the absence of factual knowledge about its location, a reasonably flexible agent could initiate an exhaustive search of the environment. A somewhat more intelligent agent, however, would know where to start looking.

As it turns out, such default associations are implicitly present in the OWL-DL knowledge base.[9] If we again turn our attention to Example (15) and then

[9] This approach is, however, dependent on the way the respective ontology is modeled. It is only applicable if the ontology makes use of concept definitions through complex concept descriptions.

look at Example (1), it is obvious that it is such a concept definition that contains knowledge about kitchen appliances and kitchens. The relationship between coffee machines and kitchen appliances is then established by the subsumption hierarchy.

In order to generate a default from a concept definition, we propose to use introspective meta-reasoning over necessary conditions. Example (1) can be decomposed into the following two necessary conditions:

(16) $\mathcal{T}_{indoor} \models \mathsf{Kitchen} \sqsubseteq \mathsf{Room}$

(17) $\mathcal{T}_{indoor} \models \mathsf{Kitchen} \sqsubseteq \exists \mathsf{contains}.\mathsf{KitchenObject}$

A kitchen can be assumed to (necessarily) be a room. It can also be assumed that it contains some kitchen object. According to the open-world assumption underlying DLs, the actual objects fulfilling this assumption might still be unknown. Suppose the knowledge base contains an instance of some kitchen object and nothing else is known about it. It then makes sense as a case of *prototypical default reasoning* to assume that the given kitchen object is contained in a known kitchen. Of course it is impossible to know for sure – without actually trying to perceptually verify its truth. That is why it is not desirable to add the consequents as crisp facts to the OWL-DL knowledge base. The default theory must be kept separate so that the agent can act upon default knowledge without regarding it as real facts.

Based on the part of the concept definition in Example 1 on page 54 that describes an anonymous class (i.e., Example 17), we can express our intuitive, prototypical knowledge about usual locations of kitchen appliances like this:

(18) $$\delta_{\mathsf{contains}} = \frac{\mathsf{Kitchen}(\mathsf{x}) \wedge \mathsf{KitchenObject}(\mathsf{y}) : \mathsf{contains}(\mathsf{x},\mathsf{y})}{\mathsf{contains}(\mathsf{x},\mathsf{y})}$$

This default rule is an open default. We are not interested in constructing such default rules *per se*. We are rather interested in drawing default conclusions about known individuals – i.e., substitutions with existing individuals that satisfy the prerequisites and that don't violate the justification. Given our previously introduced example TBox $\mathcal{T}_{indoor}$ with ABox $\mathcal{A}_{ex}$ (see Examples (1)–(8)), and exploiting the RBox definition of contains and in as inverse roles, the open default in Example 18 can be instantiated like this:

(19) $$\delta_{\mathsf{contains}_1} = \frac{\mathsf{Kitchen}(\mathsf{AREA1}) \wedge \mathsf{KitchenObject}(\mathsf{OBJ1}) : \mathsf{contains}(\mathsf{AREA1},\mathsf{OBJ1})}{\mathsf{contains}(\mathsf{AREA1},\mathsf{OBJ1})}$$

(20) $$\delta_{\mathsf{contains}_{1^-}} = \frac{\mathsf{Kitchen}(\mathsf{AREA1}) \wedge \mathsf{CoffeeMachine}(\mathsf{OBJ1}) : \mathsf{in}(\mathsf{OBJ1},\mathsf{AREA1})}{\mathsf{in}(\mathsf{OBJ1},\mathsf{AREA1})}$$

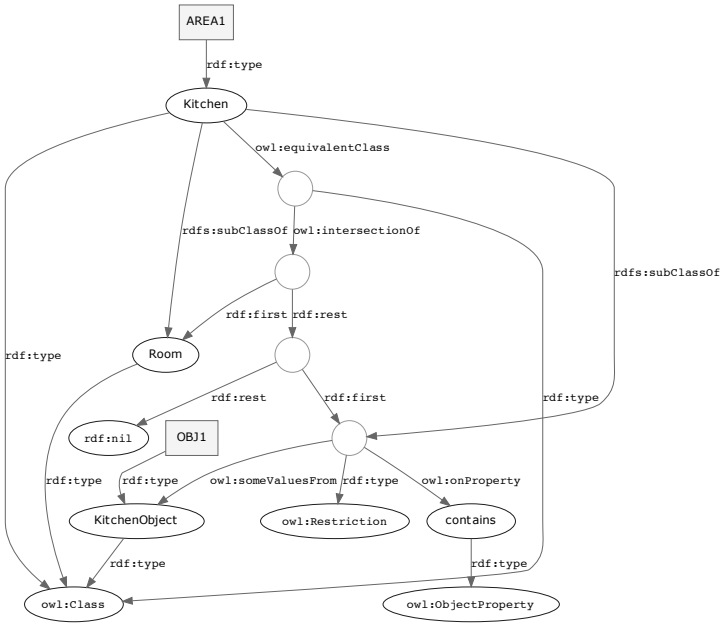


Figure 4.3: RDF graph for a part of $\mathcal{T}_{indoor}$, $\mathcal{R}_{indoor}$ and $\mathcal{A}_{ex}$.

The individuals that satisfy the prerequisites (here: AREA1 and OBJ1) can be easily retrieved from the knowledge base using SPARQL queries (cf. Section 4.2.4). In Figure 4.2 we have already shown an RDF graph for the concept definition of Kitchen. Figure 4.3 shows that RDF graph augmented with further `rdfs:subClassOf` relations (established through *subsumption checking*) and some ABox individuals introduced in Example 7 on page 57.

Concept definitions like the one above and including the other concept constructors (cf. Table 4.2 on page 54) share a common *graph pattern* that can be expressed as a SPARQL query restriction. Example 21 on the following page shows a SPARQL query that returns variable substitutions (i.e., `?x` and `?y`) for open defaults that are based on concept definitions. Additionally the query returns the role (`?property`) that is supposed to hold prototypically between `?x` and `?y`. This provides a productive pattern for deriving closed defaults based on concept definitions in a principled way. Note that we claim that it is al-

ways consistent to assume the validity of prototypical statements. Therefore, satisfiability of β (cf. Definition 11) is never explicitly checked. We rather ensure elsewhere in the system that factual knowledge is favored over prototypical knowledge (see below).

```
(21)  SELECT DISTINCT ?x ?property ?y
      WHERE { ?x rdf:type ?definedClass .
              ?y rdf:type ?roleFillerClass .
              ?definedClass rdfs:subClassOf ?restriction .
              ?restriction rdf:type owl:Restriction .
              ?restriction owl:onProperty ?property .
             ?restriction ?quantification ?roleFillerClass .
              ?roleFillerClass rdfs:subClassOf owl:Thing .
              FILTER (?definedClass != owl:Nothing).
              FILTER (?quantification != rdfs:subClassOf).
              FILTER (!isBlank(?definedClass)).
              FILTER (!isBlank(?roleFillerClass)). }
```

Representing defaults in OWL/RDF through reification

If we want to represent prototypical knowledge within the same knowledge base as the asserted, acquired, and inferred knowledge deduced through OWL-DL reasoning, these two kinds of knowledge must not be mixed up. In particular we are interested in keeping separate certain relations that hold between individuals and prototypical relations that could be assumed in case no other evidence is present. One possibility for this is to introduce new relations in a separate namespace – or in a separate branch of the RBox hierarchy – that are not used in the axioms and class definitions of the OWL-DL part of the ontology. This, however, neglects the fact that there still exists a conceptual relationship between a prototypical relation and its DL counterpart. Moreover, as we see later in Chapter 6, it is desirable that prototypical relations can *stand in* in case their DL counterparts are not deducible – with the constraint that their prototypicality be apparent. Like this, an autonomous agent can perform different, possibly more cautious, actions when the decisions are based on prototypical default knowledge rather than crisp facts. Likewise, a conversational agent can produce different utterances when reporting on prototypical knowledge (e.g., "the coffee machine is probably in the kitchen" – as opposed to saying "the coffee machine is in the kitchen").

It would thus be useful to qualify relations derived from default applications as prototypical, while maintaining the semantics of their non-prototypical, DL counterparts. Unfortunately, OWL/RDF can only express binary relations.

It is not possible to directly express qualifications of concrete relation triples. However, it is possible to circumvent this limitation through *reification*. For reification, a new individual that denotes a specific relation between two other individuals is introduced. This individual is called *statement*. In order to represent the given relation, the statement expresses the related individuals as its subject and object, and the relation as its predicate.

RDF offers reification using the `rdf:Statement` class. Instances of `rdf:Statement` can then be related to its subject and object individuals using `rdf:subject` and `rdf:object`, respectively. `rdf:predicate` denotes the type of relation that the statement is about. `rdf:subject`, `rdf:object`, and `rdf:predicate` are instances of `rdf:Property`. Example (4.3.1) summarizes the structure of a reified statement in RDF triple notation.[10]

(22) Example of reification using RDF statements:

```
statement     rdf:type       rdf:Statement .
statement     rdf:subject    subj .
statement     rdf:predicate  pred .
statement     rdf:object     obj .
```

This construction allows us to refer to relations that hold prototypically. It is straightforward to express relations between OWL individuals as RDF statements. OWL can express two kinds of relations: `owl:ObjectProperty` and `owl:DatatypeProperty`. These are subclasses of the RDF class `rdf:Property`, and OWL individuals are instances of `rdfs:Resource`. In order to qualify a statement as prototypical, we introduce a new class `DefaultStatement` as a subclass of `rdf:Statement` that expresses prototypical statements based on the default rules presented earlier. Example (4.3.1) shows the RDF triple notation of our previous example (see Example (18)).

(23) Asserted knowledge:

```
area1  rdf:type       Kitchen .
obj1   rdf:type       Oven .
```

(24) Inferred knowledge:

```
obj1   rdf:type       KitchenObject .
```

(25) Default knowledge:

```
_:s     rdf:type       DefaultStatement .
_:s     rdf:subject    area1 .
_:s     rdf:predicate  contains .
_:s     rdf:object     obj1 .
```

[10] `subj` and `obj` are instances of `rdfs:Resource`, and `pred` is an instance of `rdf:Property`.

Like this, prototypical knowledge can be stored in the same knowledge base, but it is kept separate from the triples that are subject to the OWL-DL reasoning. A special set of rules must be written to generate and access this kind of knowledge. Conceptually, we refer to this set of knowledge as *DBox*. We can hence extend the definition of our knowledge base from Definition 2 on page 51 as follows.

Definition 12 (Ontological knowledge base (extended))**.**
An ontological knowledge base $\mathcal{O}$ consists of a TBox $\mathcal{T}$, an ABox $\mathcal{A}$, an RBox $\mathcal{R}$, and a DBox $\mathcal{D}$: $\mathcal{O} = \mathcal{T} \cup \mathcal{A} \cup \mathcal{R} \cup \mathcal{D}$. ■

Using rules to automatically instantiate open defaults

We have presented a technique for instantiating prototypical defaults using RDF graph queries. We have also shown how prototypical knowledge can be stored in an OWL/RDF knowledge base while keeping it outside the part of the knowledge base that is subject to direct manipulation by an OWL-DL reasoner. Using a combined OWL-DL and general purpose rule engine as presented in Section 4.2.6 allows us to automatically generate reified defaults while the agent is discovering its environment.

The rules essentially consist of a set of premises that correspond to the graph pattern clause of the SPARQL query (21), and a set of consequences that add a reified statement like the one in Example 25 – with the difference that the object positions of the statement triples are instantiated with the result of the graph pattern query. In Chapter 6, we present an integrated autonomous robot that makes use of prototypical knowledge derived from its spatial knowledge base. We also show how such a rule can be written for the specific rule engine used in the implementation.

4.3.2 Belief revision

Belief revision deals with keeping a knowledge base consistent when new information is acquired. In the simplest case, the new information leads to adding certain facts to the knowledge base without creating any inconsistencies with respect to the current knowledge base. This rather straightforward, monotonic process is called *expansion* (Antoniou, 1997). Expansion happens for example in the ABox when new individuals are added to an OWL-DL ontology, or when new relations between individuals are asserted. In such a case the OWL-DL reasoner can monotonically infer new facts by computing the closure of the new knowledge base. This is the standard case in our scenarios when the agent explores its environment and discovers new objects and their locations. The reasoner then performs the usual instance checking for establishing the type hi-

erarchies of objects, and, possibly, for inferring new types of individuals using OWL-DL concept definitions, as described earlier (cf. Section 4.2).

Another case for expansion of a knowledge base is when new conceptual information is added to the TBox. This can be the case when two ontology resources are combined in an on-line manner. The OWL-DL reasoner must then compute the transitive closure of the combined type taxonomies. In case of a non-empty ABox this also entails the recomputation of the type hierarchy of individuals, which might lead to inference of new types either through simple concept subsumption or more complex concept definitions. However, we are not considering extensions of the TBox during run-time. We thus neglect the belief revision mechanisms necessary to recover from potentially inconsistent conceptual knowledge.

A more difficult to handle process is *contraction* (Antoniou, 1997), which deals with the retraction of facts from a knowledge base, and the resulting problem of determining which other, now invalid, facts must be removed such that the knowledge base is consistent again. Obviously the knowledge base should not only be consistent, but also complete with respect to the certain facts and known rules. Contraction can then be interpreted as the process of removing the minimal set of facts such that these two requirements hold again. This, in turn, can be a hard problem in case there are several possibilities to restore a consistent knowledge base. Contraction happens for example when an individual is removed from the knowledge base (in which case every triple involving the individual must be removed), or when the presence of new information invalidates a left-hand side (i.e., the antecedent) of an inference rule. In the latter case, the derived facts from the consequent side of the rule must be retracted from the knowledge base, along with all the facts that were derived solely from these consequents. It is therefore necessary that the reasoner internally distinguish between asserted knowledge (i.e., either innate, or acquired from processing sensor input, or user-asserted) and facts that were inferred (i.e., either through DL-based reasoning or inference rules).

Since, in our approach, the TBox is only altered off-line and thus innate with respect to the system run-time, there will not be any necessity to recover from changing or even contradictory terminological knowledge. Asserted knowledge is knowledge about individuals that is created on the basis of user input. There might be misunderstandings, or the user might later give inconsistent information. This is a case in which clarification strategies are necessary that are beyond the scope of this thesis.[11] For acquired knowledge, it is possible

[11] Kruijff et al. (2008) present an approach to situated clarification that could potentially be used to resolve inconsistencies or ambiguities that arise from asserted knowledge.

that a lower-level processing module recognizes that the state of an entity in the world has changed, or that information stemming from a lower-level processing module invalidates previous assumptions in upper-level modules. In both cases, the module that handles the input from lower-levels and mediates the ontology-based representation of facts derived from lower-levels must be aware to which individuals this change pertains, and which appropriate steps it needs to perform in order to allow the reasoner to correctly perform knowledge base contraction with respect to facts inferred from the revised facts.

By precluding terminological revision and by actively maintaining asserted and acquired knowledge, the belief revision process is greatly simplified. The reasoner must ensure that inferred facts are 'given up' more easily, or, even more strictly speaking, only contradictory inferences are withdrawn. In the latter case, there must however be a mechanism to retrieve inconsistent, acquired or asserted, knowledge in order to initiate other appropriate steps for recovering a consistent knowledge base, such as interactive clarification.

4.4 Summary and Outlook

In this chapter, we have presented the representations and formalisms underlying the conceptual map layer of the multi-layered spatial model from the previous chapter. We have shown how Description Logics can be used to perform inference on a human-compatible symbolic conceptualization of space. We have further proposed methods for prototypical default reasoning and belief revision to extend the capabilities of autonomous agents. In the next chapters, we will illustrate how these principles can be applied to real, integrated robotic systems. The EXPLORER system that will be presented in Chapter 5 performs DL-based reasoning in its conceptual map. The extensions to the EXPLORER in Chapter 6 consist of exploiting terminological knowledge for deriving default assumptions that are useful for goal-directed planning for situated action in large-scale space. The robot DORA, which will be introduced in Chapter 7, relies on mechanisms for belief revision as well as nonmonotonic symbol creation and maintenance. Finally, in Part III, we will use the conceptual spatial representation as a knowledge base for situated natural language comprehension and production.

Part II

Implementation and Experiences on Integrated Robotic Systems

Chapter 5

Spatial Understanding and Situated Interaction with the EXPLORER

Summary
In this chapter, we introduce the EXPLORER *robot system. The* EXPLORER *implements the approach to multi-layered conceptual spatial mapping from Chapter 3 in an integrated robotic system. The mobile robot base is equipped with different sensors for map building, place and object recognition, and user interaction. We illustrate how the multi-layered map can be acquired interactively in a so-called guided tour scenario. We furthermore present a method for human- and situation-aware people following that makes use of the higher-level information of the multi-layered conceptual spatial map, thus increasing the perceived level of intelligence of the robot.*

This chapter originates from a joint work with Óscar Martínez Mozos (laser-based place classification), Patric Jensfelt (low-level mapping, SLAM, navigation, laser-based people tracking and robot control), and Geert-Jan Kruijff (situated dialogue), cf. (Kruijff et al., 2007b; Zender et al., 2008, 2007a,b; Mozos et al., 2007a,b). It also makes use of the approach to hybrid laser- and vision-based object search and localization developed by Gálvez López (2007).

5.1 Motivation and Background

As discussed earlier, robots that can perform more demanding household tasks and interact with their users must be endowed with a human-compatible representation of their environment. An important aspect of human-compatible perception of the world is the robot's understanding of the spatial and functional properties of human-oriented environments, while still being able to safely act in it. In the previous chapters we have presented an approach to multi-layered conceptual spatial mapping that addresses these requirements.

In this chapter we focus on how a robot can autonomously build an internal representation of the environment by combining innate (possibly human-compatible) concepts of spatial organization with different low-level sensory systems. To this end, we present an implementation of the multi-layered conceptual mapping approach in a mobile robot system called EXPLORER. The multi-layered conceptual spatial representation presented in Chapter 3 contains maps at different levels of abstraction in order to meet both aforementioned requirements – robust robot control and human-compatible conceptualization. In Section 5.4 we show how conceptual spatial information can be used in lower-level processes, such as robot navigation and control, so that the robot can anticipate likely actions performed by a human and adjust its motion accordingly – in an approach to *human- and situation-aware people following*.

5.1.1 The EXPLORER

The EXPLORER is based on a MobileRobots Robotics Pioneer 3 PeopleBot[1] research robot platform (see also Section 2.2.1), which is equipped with various sensors for *odometry* (i.e., measuring the distance traveled) and *exteroception* (i.e., perceiving external stimuli). It has been implemented on two different PeopleBots – the DFKI-based "Robone" (cf. Figure 5.1) and "Minnie" at KTH Stockholm. Figure 5.11 shows both robots side-by-side.

The laser range finder, a SICK LMS200, is the main exteroceptive sensor in our system. It is mounted at a vertical height of 30 cm and faces the forward direction in parallel to the floor plane. It covers a semi-circle at the robot's front with time-of-flight measurements of 360 laser pulses with an angular resolution of 0.5° at 36 Hz. The laser scanner detects solid objects in the direct line of sight. Glass objects, however, cannot be detected. Figure 3.1c shows the frontier of such a laser range scan.

The two drive wheels allow for convenient motion control of the robot, while the third wheel – a caster wheel at the back of the robot – preserves stability and balance. The wheels are equipped with rotation encoders, which provide odometry readings. The wireless ethernet component ensures the teleoperability of the robotic system when running the software architecture off-board.

Both robots feature a pan-tilt unit (PTU) bearing a camera. On "Minnie" this PTU is mounted upside down below the top platform, on "Robone" the PTU carries a stereo-vision camera and is mounted on top of the top platform of the robot.

[1] `http://www.mobilerobots.com/ResearchRobots/ResearchRobots/PeopleBot.aspx` [last accessed on 2010-04-23]

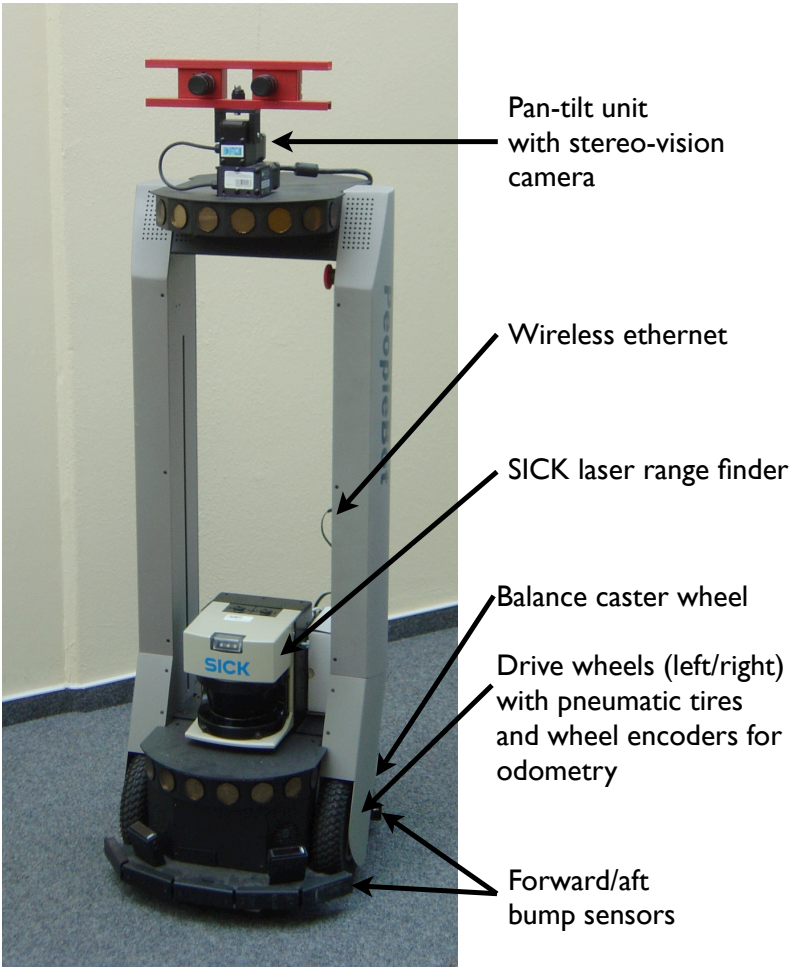


Figure 5.1: Features and accessories of the PeopleBot "Robone."

5.1.2 Related Work

In (Kruijff et al., 2007b) we present the cognitive architecture of the implementation of the EXPLORER system as used in this chapter, and give details on its dialogue capabilities. We furthermore discuss how these components are used for interactive map acquisition. In addition to this, the present work focuses on an implementation of our method for representing the environment on several levels of abstraction as introduced in the previous chapters. Where necessary, some details about the computer vision algorithms used for object detection, about the processing of sensory input from a laser scanner, and about the principles of knowledge processing in the conceptual map layer are given. We present another extension to the robotic system in (Zender et al., 2007a). The extension consists of the approach to human- and situation-aware people following that makes use of the semantic information represented in the conceptual spatial map.

People detection, tracking and following

There are several techniques that address *detection*, *tracking*, and *following of persons* in a robot's environment. They differ from the present work not only in the sensors used, but also in the degree of mobility of the robot. Kleinehagenbrock et al. (2002) present a person tracking approach that fuses information from a laser-range based leg detection mechanism and a vision-based face recognition module to keep track of a person. Fritsch et al. (2004) extend this work by adding a stereo-microphone setup that locates persons through the speech sounds they produce. One reason for combining multiple sensors for tracking a person is the lack of occlusion handling of their laser-range based people tracker. They present an experimental setup in which a static robot has to keep track of a person partially occluded by office furniture while manipulating a typical office object. Although they achieve a fair degree of robustness in the experiments, there is no evaluation of the performance of the approach when used on a moving robot. Moreover, their approach does not have the predictive capabilities to anticipate actions of a tracked person. Topp and Christensen (2005) present an evaluation of a laser-based people tracking method that allows for multiple people in the environment and temporary occlusion of tracked persons, similar to the algorithms of Schulz et al. (2003). The experiments show that an approach that is only relying on laser data is a good choice for mobile robots that will be operating under different lighting conditions and will have to interact with previously unknown people. However the experiments also reveal the disadvantages of a purely laser-based method: in a typical office environment laser readings of many structures at a height of 30 cm resemble laser readings typical for legs at that height. Arras et al. (2007) present

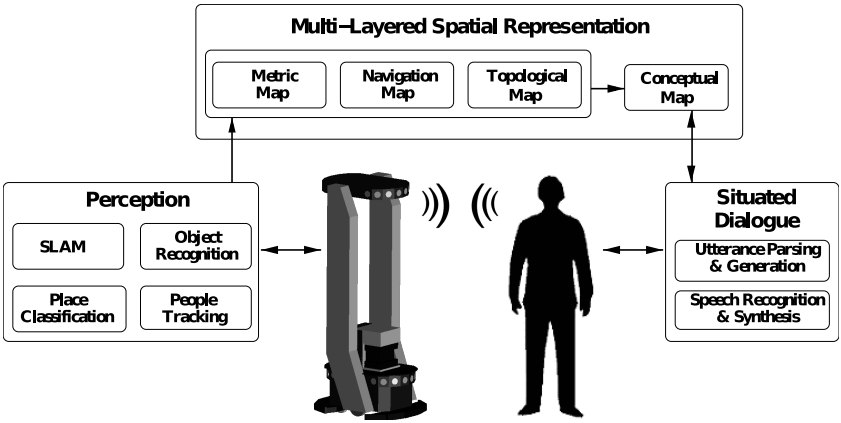


Figure 5.2: Overview of the components of the EXPLORER robotic system.

a machine learning approach to acquire features for person detection from laser scans that could overcome some of these drawbacks.

Here we present an approach to people following which builds forth on the research cited above. We opted for a laser range finder as the main sensor for our method, as it imposes the least requirements on the clothing of people, their body posture with respect to the robot, and the lighting conditions of the surroundings. Based on sensing input, the robot maintains an awareness of the current situational context. This forms the core, and the novelty, of our approach: a combination of both *human awareness* and *situation awareness* to yield a comprehensible, socially acceptable following behavior, which includes keeping an acceptable personal distance (based on Hall's notion of *proxemics* (Hall, 1966; Pacchierotti et al., 2005)), establishing eye contact, providing verbal feedback, and applying situation-aware interpersonal behaviors.

5.2 System Overview

This multi-layered spatial representation from Chapter 3 is the centerpiece of the EXPLORER integrated robotic system. It is created using information coming from different input modalities, as shown in Figure 5.2. The individual modalities range from low level robot control and perception modules to a communication subsystem for spoken dialogue with the user. There are three main subsystems involved in constructing, maintaining, and using the spatial representation: the *perception subsystem* (presented in more detail in Section 5.2.1) for evaluation of sensory input, the *communication subsystem* (see Section 5.2.2)

for situated spoken dialogue, and the subsystem for *multi-layered conceptual spatial mapping* that bridges the gap between sensor-based maps and a *human-compatible* spatial representation. The main techniques used in the perception and communication subsystems and the structure of the multi-layered spatial representation that sits at the core of our system are explained in more detail in the following sections.

5.2.1 Perception

The perception subsystem gathers information from the laser range scanner and from a camera. Different techniques are used for evaluation of the sensory input. The laser data is processed and used to create the low level layers of the spatial representation. At the same time the input from the laser scanner is used by a component for detecting and following people. Finally, the images acquired by the camera are analyzed by a computer vision component for object recognition.

Simultaneous localization and mapping

To reach a high level of autonomy the robot needs the ability to build a map of its environment that can be used to safely navigate and stay reliably localized. To this end we use the *simultaneous localization and mapping* (SLAM) technique of Folkesson et al. (2005). In our system the SLAM module extracts geometric primitives from laser range scans and applies an Extended Kalman Filter (EKF) framework for the integration of feature measurements. The geometric features used in our approach are lines, which typically correspond to walls and other straight structures that that appear as a line segment at the height of the laser scanner. Since walls are in most cases static, these invariant features of the environment are used to keep the robot localized. The line features are stored in a global metric map with an absolute frame of reference. Figure 3.3b shows an example of such a line map created using this method.

Place classification

Apart from line features, i.e. walls, other features can be derived from the laser range data. These features are useful to *semantically interpret* the position at which they were detected. The integrated robotic system presented here uses a laser-based method that classifies observations into belonging to either a doorway, a corridor, or a room.

Doorways indicate the transition between different spatial regions (Chown, 1999). They are detected and added whenever the robot passes through an opening of a certain width. This width is selected such that it corresponds to the width of standard doorways in the environment (e.g., around 80 cm). Informa-

tion about the door opening (width and orientation) is stored in the map (see Section 5.3.2) along with the detected position of the doorway.

Corridors and rooms are classified according to the laser observation that the robot takes at that location. The main idea of this approach is to extract simple geometrical features from the laser scans and their polygonal approximation. The overall approach, which lies outside the scope of this thesis, is presented in more detail in (Mozos et al., 2005; Mozos, 2010).

People tracking

In order to follow its user, the robot must be able to detect and track the positions of people in its vicinity. Here, we focus on the interaction with a single person – the user – which simplifies the tracking problem. To handle the challenges that arise with occlusions and people moving close to each other, a more advanced tracking algorithm such as the one by Schulz et al. (2003) is needed. For the present work, we apply a method for people tracking that is akin to (Lindström and Eklundh, 2001; Wang and Thorpe, 2002). The exact details of the method are outside the scope of this thesis. They are described in more detail in (Zender et al., 2007a).

Object recognition

In a nutshell, the visual object detection system uses SIFT features (Lowe, 2004) to recognize previously learned objects, such as a television set, a couch, or a coffee machine. Objects play an integral role in the conceptual map, as the information of recognized objects is used for inferring subconcepts (e.g., Kitchen or Livingroom) for rooms in the environment. Object recognition has been and still is a very active area of research. Since the implementation of the computer vision modules is not part of this thesis, we refer the reader to (Zender et al., 2008; Gálvez López, 2007) for more details of the approach.

5.2.2 Situated dialogue

In this section, we discuss the functionality which enables a robot to carry out a *situated dialogue* in natural language with a human user. A core characteristic of our approach is that the robot builds up a meaning representation for each utterance. The robot interprets it against the dialogue context, relating it to previously mentioned objects and events, and to previous utterances in terms of "speech acts" (dialogue moves). Because dialogues in human-robot interaction are inherently situated, the robot also tries to ground the utterance content in the situated context – including past and current visuo-spatial contexts (reification of visuo-spatial references), and future contexts (notably, planned events and

states). Below we only highlight several aspects; for more detail, we refer the reader to (Kruijff et al., 2007a,b).

Speech recognition yields a string-based representation for spoken input, which is subsequently parsed using the *OpenCCG* Combinatory Categorial Grammar (CCG) parser (see Section 8.1.2 and (Baldridge and Kruijff, 2003)). The parser analyzes the utterance syntactically and derives a semantic representation in *Hybrid Logics Dependency Semantics* (HLDS) (see Section 8.1.3 and (Baldridge and Kruijff, 2002)). The semantic representation is a *logical form* (LF) in which propositions are assigned ontologically sorts, and related along typed relations (e.g. "Location", "Actor").

The LFs yielded by the parser are interpreted further, both within the dialogue system and against information about the *situated spatial context*. Objects and events in the logical form are related against the preceding context (coreference resolution), as is the dialogue move of the utterance. The resulting dialogue model is similar to that proposed by Asher and Lascarides (2003) and Bos et al. (2003). The robot also builds up a temporal-aspectual interpretation for events, relating it to preceding events in terms of how they temporally and causally follow on each other (see also (Kruijff and Brenner, 2007)). In combination with the dialogue model, this is closely related to the approach of Sidner et al. (2004).

Example 26 shows the meaning representation of typical assertion of the human user. Example 27 presents the robot's answer to the question "where are you?". It illustrates that the robot conveys its inferred conceptualization of the current area as a referring expression (see Section 8.1.1). The examples show the semantic analysis of these utterances in HLDS, which are described in detail in Section 8.1.3.

(26) HLDS logical form of the utterance "we are in the office."

$$
\begin{aligned}
@_{be1:ascription}(&\textbf{be} \wedge \\
&\langle Mood \rangle \textbf{ind} \wedge \\
&\langle Tense \rangle \textbf{pres} \wedge \\
&\langle Copula - Restr \rangle (w1 : person \wedge \textbf{we}) \wedge \\
&\langle Copula - Scope \rangle (in1 : m - location \wedge \textbf{in} \wedge \\
&\quad \langle Anchor \rangle (office1 : e - place \wedge \textbf{office} \wedge \\
&\quad\quad \langle Delimitation \rangle \textbf{unique} \wedge \\
&\quad\quad \langle Num \rangle \textbf{sg} \wedge \\
&\quad\quad \langle Quantification \rangle \textbf{specific})) \wedge \\
&\langle Subject \rangle (w1 : person))
\end{aligned}
$$

(27) HLDS logical form of the utterance "I am in the living room."

$$
\begin{aligned}
@_{be9:state}(&\textbf{be} \wedge \\
&\langle Mood\rangle\textbf{ind} \wedge \\
&\langle Tense\rangle\textbf{pres} \wedge \\
&\langle Copula-Restr\rangle(r1 : person \wedge \textbf{I}) \wedge \\
&\langle Copula-Scope\rangle(in1 : m-location \wedge \textbf{in} \wedge \\
&\quad \langle Anchor\rangle(livingroom1 : e-place \wedge \textbf{livingroom} \wedge \\
&\quad\quad \langle Delimitation\rangle\textbf{unique} \wedge \\
&\quad\quad \langle Num\rangle\textbf{sg} \wedge \\
&\quad\quad \langle Quantification\rangle\textbf{specific})) \wedge \\
&\langle Subject\rangle(r1 : person))
\end{aligned}
$$

5.3 Multi-Layered Spatial Representation

The EXPLORER system presented in this chapter is endowed with a multi-layered conceptual spatial map, ranging from a low-level metric map for robot localization and navigation (SLAM), to a conceptual layer that provides a human-compatible decomposition and categorization of space. Figure 5.3 depicts the main layers of the representation.

The lower layers of our model are derived from sensor input (Section 5.2.1). Different methods are used to gradually construct more abstract representations. On the highest level of abstraction, we regard topological regions and spatially located objects as the primitive entities of a spatial conceptualization that is compatible with how humans perceive space. In order for a robot to meaningfully act in, and talk about an environment, it must be able to assign human-compatible categories to spatial entities. Below, the individual layers of our spatial representation are addressed more closely.

5.3.1 Metric map

At the lowest layer of our spatial model, we have a *metric map*. In this map, lines are the basic primitive to represent the boundaries of open space. The metric line map supports self-localization of the robot. It is maintained and used by the SLAM component, as described in Section 5.2.1. As can be seen in Figure 3.3b and Figure 5.4, the line based metric map gives a rather sparse description of the environment. Moreover, since the global co-ordinate system of the metric map is purely internal to the robot and since humans are not able to easily (i.e., without additional tools) evaluate its underlying quantitative spatial represen-

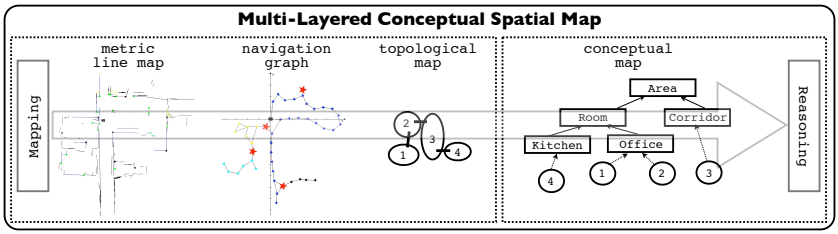


Figure 5.3: Our multi-layered map, ranging from sensor-based maps to a conceptual abstraction.

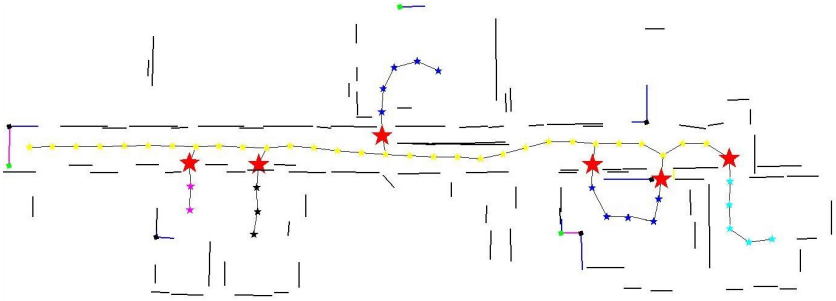


Figure 5.4: The navigation graph overlayed on the metric line-feature map. The navigation nodes are represented by stars. Different colors represent different areas separated by doors, which are marked by bigger red stars.

tation, the metric map alone does not provide a suitable level of abstraction for human-robot dialogues.

5.3.2 Navigation graph

The next layer of our representation is composed of a *navigation graph*, which establishes a model of free space and its connectivity, i.e., reachability. It is based on the notion of a "roadmap of virtual free-space markers" as described in (Latombe, 1991; Newman et al., 2002). As the robot navigates through the environment, a marker called *navigation node* is added to the map at the robot's current position whenever it has traveled a certain distance from the closest existing node. Nodes are connected following the order in which they were generated. This order is given by the trajectory that the robot follows during the map acquisition process. Navigation nodes are anchored in the coordinate system

of the metric map, as illustrated in Figure 5.4. Using standard graph search algorithms[2] the navigation graph can be used for path planning and autonomous navigation in the already visited part of the environment.

In the navigation graph the robot's spatial representation is augmented with sensor-based semantic environment information. The approach presented in Section 5.2.1 for semantic classification assigns a label (i.e., either corridor or room) to each pose of the robot during a trajectory. In order to make the classifications more robust, we store the classification of the last N poses in a short term memory. Using a majority vote approach over these memorized classifications, we then assign a class to each navigation node.

Objects detected by the computer vision component are also stored on this level of the map. Whenever an image is matched to a training image of an object, the pose of the robot is used to determine the position of the corresponding detected object. The positions of objects are are associated with the navigation node that is closest to their estimated metric position.

5.3.3 Topological map

The *topological map* provides a level of abstraction that approximates a human-like qualitative segmentation of an indoor space into distinct regions, as discussed in 3.1.1. In this view, the exact shape and boundaries of an area, as represented in the lower map layers, are abstracted to a coarse categorical distinction between rooms and corridors.

The topological map divides the set of nodes in the navigation graph into areas. An area consists of a set of interconnected nodes which are separated by a node classified as a doorway. In Figure 5.4, the topological segmentation is represented by the coloring of the nodes. In order to determine the category of an area, we take a majority vote approach of the classification results of all nodes in the given area. The topological areas, along with detected objects (cf. Section 5.2.1), are passed on to the conceptual map, where they are represented as instances of their respective classes.

5.3.4 Conceptual map

On the highest level of abstraction resides the *conceptual map*. For one, it describes *taxonomies* of room classes and typical objects found in an office environment. Second, information extracted from sensor data and given through situated dialogue about the actual environment is stored as symbols that instantiate these classes. It represents conceptual knowledge in an OWL-DL ontology

[2] such as, e.g., A^* (see (Russell and Norvig, 2003))

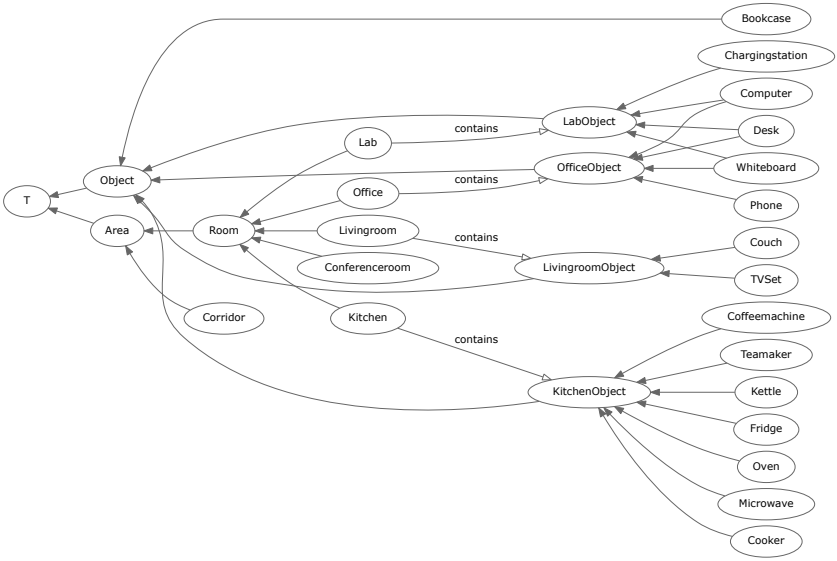


Figure 5.5: Illustration of the TBox $\mathcal{T}_{explorer}$ of the indoor commonsense ontology used in the EXPLORER system. See also Figure 3.6.

of an indoor office environment (see Figure 5.5), and makes use of the reasoning principles introduced in Chapter 4.

In line with the way humans categorize space (see Section 3.1.2), our ontology defines room types on the basis of the objects they contain. The conceptual ontology in the TBox constitutes *innate* knowledge, which has to be predefined and cannot be changed during run-time. However, while the robot operates in its environment, the sensors constantly *acquire* new information, which is then represented as instance knowledge in the ABox. Through situated dialogue the robot can obtain *asserted* knowledge from its user. A description logic reasoner can then fuse this knowledge in order to *infer* new knowledge about the world that is neither given verbally nor actively perceived.

5.3.5 Spatial knowledge processing

Below, we describe the information processing principles (see also Section 3.3) for these individual types of knowledge in more detail. Figure 3.7 shows an example of how spatial knowledge from different sources in our robotic architecture converges in the conceptual map.

Interactive map acquisition

The map acquisition process exemplifies how information and knowledge is exchanged between the different parts of the robotic architecture. The multi-layered representation is created using an enhanced method for concurrent semi-supervised map acquisition, i.e., the combination of a user-driven supervised map acquisition process with autonomous exploration discovery by the robot. This process is based on the notion of *human-augmented mapping*, as introduced by Topp and Christensen (2005). We additionally use a linguistic framework that actively supports the map acquisition process and is used for situated dialogue about the environment (Kruijff et al., 2007b). The map can be acquired during a so-called *guided tour scenario* in which the human tutor shows the robot around and continuously teaches the robot new places and objects. During such a guided tour, the user can command the robot to follow him or to explore an area autonomously. This mixed control strategy – referred to as *sliding autonomy* (Heger et al., 2005), or *adjustable autonomy* (Crandall and Goodrich, 2001) – combines the robot's autonomous capabilities where appropriate with different levels of user control where needed. However, the human user preserves full control over the robot. She can always stop the robot or give new commands.

Our system does not require a complete initial guided tour. It is as well possible to incrementally teach the robot new places and objects at any time the user wishes. With every new piece of information, the robot's internal representations become more complete. Still, the robot can always perform actions in, and conduct meaningful dialogue about, the aspects of its environment it already knows about. This amounts to an *anytime behavior*, in which there is no distinction between the learning phase and the operation phase typical for other machine learning approaches. Rather, the robot acts according to the principle of *discovery* for map acquisition and navigation (Maio and Rizzi, 1992). It constantly combines acquired and user-asserted knowledge with its conceptual taxonomy and draws further conclusions according to the principles explained in Part I.

Let us consider an example guided tour of the 7th floor of the CAS building at KTH in Stockholm. A full description of the run can be found in (Kruijff et al., 2007b).[3] Initially, the robot starts with the TBox $\mathcal{T}_{explorer}$ shown in Figure 5.5. The ABox $\mathcal{A}_{CAS7}$ and the RBox $\mathcal{R}_{CAS7}$ are empty. The user then activates the robot and teaches it the position of the charging station. After that, the robot is told to follow the user, who enters another room, the so-called living room.

[3] A video that shows the full run can be found on-line [last accessed on 2010-04-23] under: `http://video.google.com/videoplay?docid=4538999908591170429`

Figure 5.6: The tutor activating the robot for a guided tour.

There, the robot first asks for clarification of an erroneously detected doorway, which then leads to an update of the spatial representation. After that, the user asks the robot several questions in order to check the robot's current knowledge of the environment. For instance, when first asked where the robot thinks it is, it answers "I am in a room" because it only has the laser-based place classification information. Then the user asks the robot to have a look around. This initializes visual object recognition. After a while, the robot detects a couch and a TV set, which lead to the inference that the containing area is a living room. After asking again for the robot's belief about its whereabouts, which it correctly describes as living room, the user sends the robot back to the charging station. The video also shows the robot's human- and situation aware person following behavior, which are described in more detail in Section 5.4, when the user is approaching a known doorway.

Innate conceptual knowledge

We have handcrafted an ontology (see Figure 5.5) that models conceptual commonsense knowledge about an indoor office environment. On the top level of the conceptual taxonomy, there are the two general concepts Area and Object. The concept Area can be further partitioned into the concepts Room and Corridor. The basic-level categories, i.e., the subclasses of Room, are *defined* (see Section 4.2) by the Object instances that are found there, as represented by the contains relation.

Acquired knowledge

While the robot moves around constructing the metric and topological maps, our system derives higher-level knowledge from the information in these layers. The bottom-up acquisition of the spatial representation is done in a fix sequence of processing steps. The metric map constructed in the SLAM module works on input from the laser range finder and from the wheel odometry. Within the multi-layered map, the SLAM module enables the robot to acquire knowledge about solid structures, as well as free and reachable space in its environment. Through use of a simple door detection mechanism, the free and reachable space is partitioned into topological areas. As soon as the robot acquires the knowledge about a new area in the environment, this information is pushed from the topological map to the conceptual map. Each topological area is represented in the conceptual map as an instance of the class Area. Furthermore, as soon as reliable information about the laser-based geometric-semantic classification of an area is available (cf. Section 5.3.3), this is reflected in the conceptual map by assigning the area's instance a more specific category (Room or Corridor).

(28) The robot starts in the corridor (see Figure 5.6). The laser-based place classifier recognizes the current area as a corridor:
$\mathcal{A}_{CAS7} = \mathcal{A}_{CAS7} \cup \{\mathsf{Corridor}(\mathsf{AREA0})\}$

Information about recognized objects stemming from the vision subsystem (cf. Section 5.2.1) is also represented in the conceptual map. Whenever a new object in the environment is recognized, a new instance of the object's type, e.g., TVSet, is added to the ABox. Moreover, the object's instance and the instance of the area where the object is located are related via the contains relation.

(29) Later in the run, the robot is asked to have a look around and finds a television (see Figure 5.7):
$\mathcal{A}_{CAS7} = \mathcal{A}_{CAS7} \cup \{\mathsf{TVset}(\mathsf{OBJ1}), \mathsf{contains}(\mathsf{AREA1}, \mathsf{OBJ1})\}$

Asserted knowledge

During a guided tour with the robot, the user typically names areas and certain objects that she believes to be relevant for the robot. Typical assertions in a guided tour include "we are in the kitchen," or "this is the charging station" (see Example 30). Any such assertion is analyzed by the subsystem for situated dialogue processing (see Section 5.2.2). In case an assertion about the spatial environment is made, the dialogue subsystem pushes these assertions on to the conceptual map, where the ontology is updated with the new information

Figure 5.7: "Aha. I see a television."

– either by specifying the type of the current area or by creating a new object instance of the asserted type and linking it to the area instance with the contains relation, as illustrated in Examples (31) and (32).

(30) HLDS logical form of the utterance "this is the charging station."

$$
\begin{aligned}
@_{be9:state}(&\textbf{be} \wedge \\
&\langle Mood\rangle\textbf{ind} \wedge \\
&\langle Tense\rangle\textbf{pres} \wedge \\
&\langle Copula-Restr\rangle(t1 : thing \wedge \textbf{this} \wedge \\
&\quad \langle VisualContext\rangle(v1 : visualobject \wedge \\
&\quad\quad \langle Proximity\rangle\textbf{proximal})) \wedge \\
&\langle Copula-Scope\rangle(c1 : thing \wedge \textbf{chargingstation} \wedge \\
&\quad \langle Delimitation\rangle\textbf{unique} \wedge \\
&\quad \langle Num\rangle\textbf{sg} \wedge \\
&\quad \langle Quantification\rangle\textbf{specific}))
\end{aligned}
$$

(31) determine current location: AREA0

(32) assert the fact that there is a charging station:
$\mathcal{A}_{CAS7} = \mathcal{A}_{CAS7} \cup \{\mathsf{Chargingstation(OBJ0)}, \mathsf{contains(AREA0, OBJ0)}\}$

Inferred knowledge

Based on the knowledge representation in the ontology, our system uses DL-based reasoning, which allows us to move beyond a pure labeling of areas. Combining and evaluating acquired and asserted knowledge within the context of the innate conceptual ontology, the reasoner can infer more specific classes for known areas. E.g., combining the acquired information that a given area is classified as Room and contains a television, with the innate conceptual knowledge given in our commonsense ontology, the reasoner infers that this area can be categorized as being an instance of Livingroom (see Example 33). Conversely, if an area is classified as Corridor and the user shows the robot a charging station in that area, no further inference can be drawn. The most specific category the area instantiates will still be Corridor (see Example 34).

(33) $\mathsf{Room}(\mathsf{AREA1}), \mathsf{TVSet}(\mathsf{OBJ1}), \mathsf{contains}(\mathsf{AREA1}, \mathsf{OBJ1}) \in \mathcal{A}_{\mathsf{CAS7}}$
$\mathcal{T}_{explorer} \cup \mathcal{A}_{CAS7} \models \mathsf{Livingroom}(\mathsf{AREA1})$

(34) $\mathsf{Corridor}(\mathsf{AREA0}), \mathsf{Chargingstation}(\mathsf{OBJ0}), \mathsf{contains}(\mathsf{AREA0}, \mathsf{OBJ0})$
$\in \mathcal{A}_{\mathsf{CAS7}}$

In principle, our method allows for multiple possible classifications of any area[4] because the main purpose of the reasoning mechanisms in our system is to facilitate human-robot interaction. The way people refer to the same room can differ from situation to situation and from speaker to speaker, as observed by Topp et al. (2006b). What one speaker prefers to call the kitchen might be referred to as the recreation room by another person. Allowing for multiple classifications of are instances ensures that is is possible to resolve all such possible references.

5.3.6 Discussion: conceptualizing areas

What does it mean to recognize an area? What *defines* an area? In SLAM-based approaches, the notion of area roughly corresponds to an "enclosed space." Observed linear structures are interpreted as walls, delineating an area. This yields a purely geometrical interpretation of the notion of area, based on its perceivable physical boundaries. Doorways are regarded as transitions between distinct topological areas. Although this is already a suitable level of abstraction, it is not yet sufficient for discriminating between areas. Another observation from human spatial cognition is that humans tend to categorize space not only geometrically, but also functionally. This functionality is often a result of the different objects inside an area, like home appliances or furniture, that afford these functions. In order to achieve a functional-geometric interpretation, a

[4] This is achieved by avoiding disjointness axioms (see Section 4.2.4) between classes in the TBox.

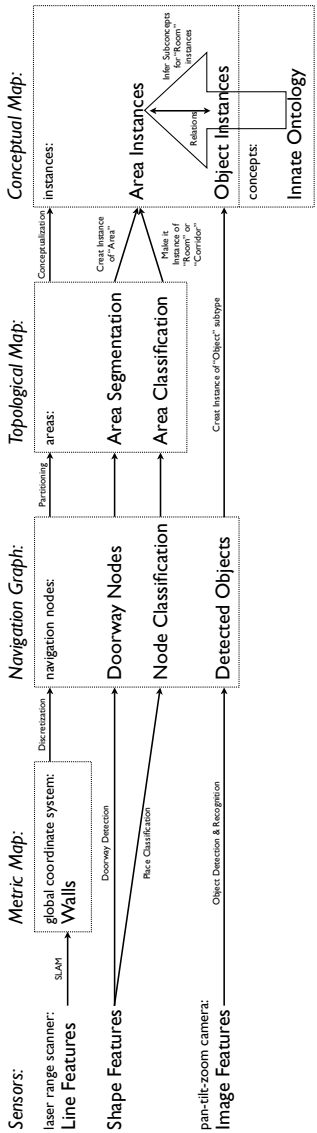


Figure 5.8: Process diagram showing convergence on consistent interpretation across levels of spatial abstraction.

robot thus has to integrate its knowledge about distinct topological areas with its knowledge about the presence of certain objects.

Figure 5.8 illustrates the way in which the modules in our system (cf. Section 5.2.1) contribute to the individual layers of our conceptual spatial map (cf. Section 5.3), and how additional pieces of knowledge are combined to achieve a more complete conceptualization of space. In our example guided tour, this is illustrated when the user repeatedly asks the robot about where it thinks it is. At first, the only thing the robot knows is that the current area is classified as Room, and answers "I am in a room." However, after the robot has recognized the couch and the TV set in the current room, its ontological reasoning capabilities can infer the appropriate subconcept Livingroom (see Example 33). Hence the robot can produce an answer that contains more information: "I am in the living room."

Our approach thus not only creates a qualitative representation of space that is similar to the way humans conceptualize it. It also serves as a basis for successful dialogues by allowing the robot to successfully refer to spatial entities using natural language expressions. In Part III we present the details of the natural language processing methods we use. Finally, experiments highlighted the need for nonmonotonic reasoning (as discussed in Section 4.3), that is, knowledge must not be written in stone. As erroneous acquired or asserted knowledge will otherwise lead to irrecoverable errors in inferred knowledge. In the next chapters, we show how our system can make use of different nonmonotonic reasoning methods, namely prototypical reasoning and belief revision.

5.4 Interactive People Following

A key functionality of an interactive mobile robot is to recognize its user and follow her around the environment – commonly referred to as *people following* behavior. It is useful in a range of situations, for example, when the user intends to perform a collaborative action with the robot elsewhere in the environment, and first needs to take the robot to this place. Another case where people following is necessary is the the interactive map acquisition process presented in the previous sections.

Here we describe how the already present information in the map can be used to make such a people following behavior more intelligent – by incorporating a notion of *human-* and *situation awareness*.

5.4.1 People tracking

The people tracking module keeps a list of the current dynamic objects. A dynamic object is represented as a tuple $o_i = \langle id, x, y, \theta, v \rangle$ where (x, y) is its

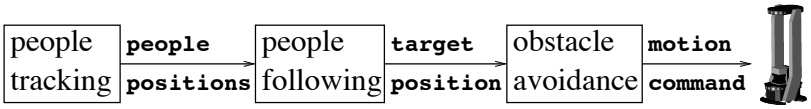


Figure 5.9: Information flow for robot control in people following mode.

position in the metric map, θ the direction of motion, v its speed and id a unique identifier to keep track of objects over time. This information is processed by the people following module, calculating a target robot position $p_t = (x_t, y_t)$ which is at a distance D_p from the person followed. The value of D_p is determined according to the situation as described below. The calculated target location is then passed on to the obstacle avoidance modules to calculate the appropriate motion commands. The basic motion control algorithm used for obstacle avoidance is the Nearness Diagram (Minguez and Montano, 2004) which is able to handle very cluttered scenes. Figure 5.9 illustrates this.

5.4.2 Social awareness

The people following behavior presented here preserves socially acceptable distances from its human user, and gives *readable social cues* (gaze, speech) indicating how the robot tries to maintain engagement during following.

The user can initiate the people following behavior by asking the robot to follow him (e.g., "Come with me!" or "Follow me!"). Following is initialized by selecting and then tracking the closest dynamic object, which is assumed to be the user. The behavior is *interactive* in that the robot actively gives the person feedback about its internal state. Verbal grounding feedback (e.g., "Yes", "Okay!") signals that the robot has understood the command and is ready to follow the user. During the execution of the people following behavior, the pan-tilt unit (PTU) is moved to simulate a gaze that follows the user. This signals that the robot is aware of its user's position and provides additional feedback about which person the robot assumes as its guiding person. The pan and tilt angles are adjusted such that the camera that is mounted on the PTU (cf. Figure 5.11) points towards the head of the tracked person. We assume the head of the person to be at $\sim$1.7m above ground at the x-y-position of the tracked person.

In accord with Pacchierotti et al. (2005), the motion control algorithms of our approach employ a control strategy that reflects the notion of *proxemics* (Hall, 1966). We only initiate a motion to follow the user if the person is more than 1.2m away from the robot – that is, when the user leaves the *personal distance*. Inside the personal distance, which we assume to be appropriate for interaction with a domestic service robot, the robot will turn its "head" to pro-

vide gaze feedback showing its user awareness. As long as she stays within the personal distance boundary, the robot will turn in place if the change in angle to the user is larger than an angle α (we use $\alpha = 10°$) in order to keep the user in its field of view. As soon as the user is further away than 1.2m, we take this as an indication that the robot should continue following her. For approaching the user, we determine a target point at distance $D_p = 50$cm, thus preserving a personal distance without violating the *intimate distance boundary*. The user can stop the robot at any time ("Halt!", "Stop!").

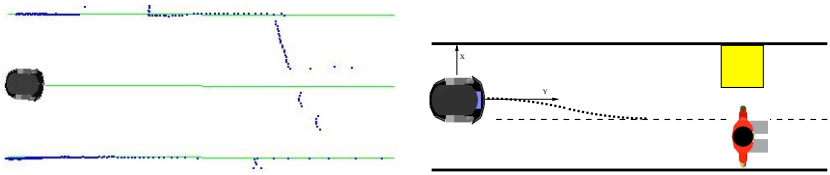
5.4.3 Situation awareness

Situation awareness can be paraphrased as "knowing [the important aspects of] what is going on around you", where 'importance' is "defined in terms of the goals and decision tasks for [the current] job" (Endsley and Garland, 2000). Endsley defines three levels of situation awareness: *perception*, *comprehension*, and *projection*. In the following paragraphs we explain how our robotic system uses perception and comprehension of the current situation to anticipate projected future states. The two example situations are embedded into the context of following a human user in a known indoor environment.

Smart handling of doors

When the user approaches a door, the robot can cause problems if it continues in normal following mode. It is a frequently observed behavior – e.g., reported by Mahani and Topp (2010) – of the kind of robot that we are using in the kind of guided tour scenario we are using it in. If the user intends to close an open door or open a closed door the robot might end up in a situation where it blocks the user from, for example, swinging open a closed door leaf. A smart robot should be aware of this danger and take appropriate action.

Since the only sensor used is a laser scanner, it is nearly impossible to detect whether the user intends to open, close, or pass through a door when approaching it. However, a safe assumption is to make room so that the user can perform any such action with the door. The navigation graph contains the position of doors in the environment. Our solution is hence to increase the desired distance between the robot and the user ($D_p = 2$m) when the user is in or close to a door. If the user moves through the door in one motion – i.e., not manipulating the door – the increased distance will not be visible and the robot follows through. If on the other hand the user stops in the door, the robot will also stop and even back off to keep a long distance from the user, thus making room for the user's actions. As soon as the robot detects that the user passed through the door, continuing his or her way, the robot will decrease the desired distance to the user again ($D_p = 0.5$m) and resume its people following behavior.



(a) Detecting the main walls and center line from laser data. The small dots depict the laser range readings.

(b) Calculating a smooth trajectory along the corridor. The yellow square represents a box close to the corridor wall.

Figure 5.10: Corridor follow mode.

Following in a corridor

Moving in a corridor is different from general motion in open space or in a more cluttered environment like a room. If the robot is able to take advantage of this situation, a smoother, faster and more intuitive motion can be achieved. The main assumption underlying our approach (cf. Figure 5.10) is that the robot can make much better predictions about the motion of the person being followed in the corridor than in a general environment: motion in a corridor is known to be along the corridor. The motion control problem is thus reduced to determining the speed along the corridor and the position across the corridor. For the obstacle avoidance method this means that a standard approach that is governed by the robot's local surround is not suitable. This would sometimes result in large corrections to the direction of motion when some new structure or person enters into the immediate surrounding of the robot. In a corridor, however, obstacles on the robot's path can be detected from a fair distance. In our approach, the motion planning method can look ahead in the corridor and make corrections to the path autonomously without relying on detecting that the user adjusts his/her course. The lateral position in the corridor is controlled so that the robot follows a safe *lane* along the corridor. For detecting upcoming obstacles, naturally, the user is not considered.

Another observation that can be made is that corridors are transportation roads for people where the speed of travel tends to be a bit higher and where people are used to moving a bit closer to each other when passing each other. The upper bound of the robot's speed along the corridor, v_{rob}, is controlled according to

$$v_{rob} = v_p + k(D - D_p) \tag{5.1}$$

where D and D_p are the current and desired distance respectively between person and robot, v_p is the current speed of the user and k is the controller gain (here

$k = 0.5$). Experiments show (cf. Section 5.4.4) that increasing the robot's maximal speed when moving in a corridor yields a better performance. Following a person in a corridor reduces the motion control problem to adjusting the speed along the corridor and position across the corridor. Predicting how the user will move is also simpler and the robot can initiate an obstacle avoidance maneuver much earlier.

Determining when to switch from normal following mode to corridor following mode can be based on the node classification from the navigation graph. We also require that the parameters defining the corridor, i.e., direction and width, can be found. This is done based on angle histograms similar to (Hinkel and Knieriemen, 1988). Figure 5.10a shows an example where the direction and the main walls of the corridor have been found.

5.4.4 Implementation and evaluation

The approach to interactive people following was implemented and tested on the two robots "Robone" and "Minnie" presented in Section 5.1.1. In both cases the mounted cameras are not used in the experiments. The pan-tilt unit however serves to provide *gaze feedback* by moving the camera to "look at" the user. The robots had a map of their environment that had been acquired beforehand. Below we discuss some results obtained from the experimental runs.

The top velocity for the robots as recommended by the manufacturer is 0.5m/s. Tests have clearly shown that it is not advised to violate this upper bound in normally cluttered space, e.g., inside an office room. In line with Section 5.4.3, however, we claim that an increased top speed of 0.8m/s is reasonable when the robot is moving in a corridor employing the proposed control algorithm.

The experiments were run by people familiar with the system as the main purpose was to validate the usefulness of the proposed algorithms. All the experiments start with the user asking the robot to follow him. The robot acknowledges its understanding ("Okay.") and initiates the people tracking and following mechanisms. The user then guides the robot around the environment, moving inside and between rooms and corridors, to create those situations we are interested in, i.e., passing through doors and moving along corridors.

The experiments (see Figure 5.12) consist of two separate episodes, demonstrating the smart handling of doors and the corridor follow mode. The zero point of the time axis is set to the point when the robot is in the center of the door in order to facilitate the comparison of the two episodes of the individual experiments. In the first episode the robot follows its user through an office, keeping a longer distance while the user is close to the known door. Figure 5.11(right) shows two screen-shots that illustrate how the robot keeps a longer distance to

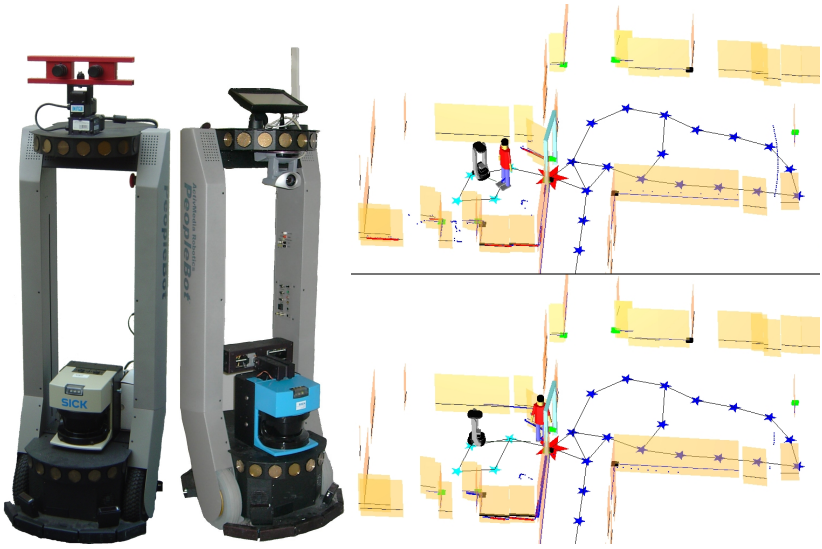
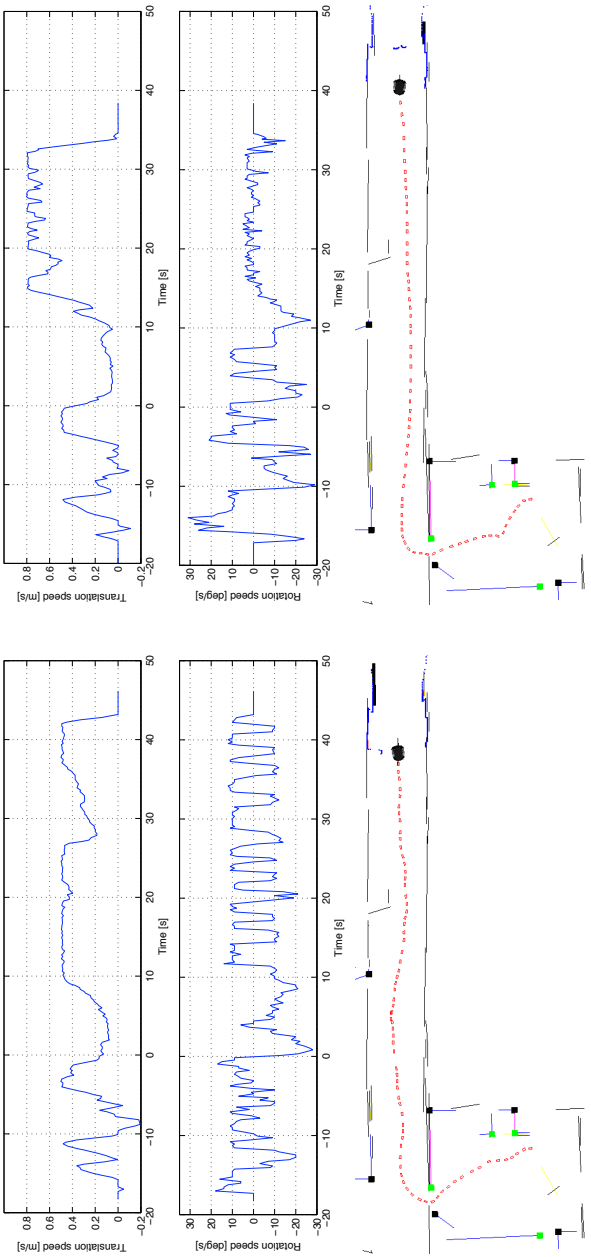


Figure 5.11: Left: the two robots "Robone" and "Minnie" used in the experiments. Right: screen-shots taken during the experiments showing the robot and the user in a room that has a doorway which leads into a hall. Note how the robot increases its distance while the user is close to the door (bottom).

its user in order to allow the user to open the door. In the second episode, the robot has to follow its user down a corridor. Our tests clearly show that the proposed motion planning algorithm for following in a corridor outperforms the standard people following mode which the robot has to rely on when moving in unknown or cluttered areas.

In both experiments, the user and the robot started in a room (lower left corner in the map). The user first guided the robot through the door into a hall and then down a corridor extending out from the hall. This first part (episode) of the experiments was used to demonstrate the robot's awareness of the door (Time$<$0s). The second episode (Time$>$0s) was used to compare the robot's performance with active corridor awareness against its performance in the non-aware follow mode.

In the first episode of both runs, the robot increased its distance or kept a longer distance to the user while the user was close to the door, which can be seen in the translation speed profiles (e.g., at -8s in Figure 5.12a, and at -17s and -9s in Figure 5.12b). In this phase, the robot also turned a lot in both



(a) Speed profiles and trajectory for normal follow mode.

(b) Speed profiles and trajectory for corridor follow mode.

Figure 5.12: Two experimental runs with inactive (a) and active (b) corridor follow mode.

experiments to keep a posture facing the user, which can be seen in the rotation speed profiles.

The behaviors of the robot differed in the second episode. As an additional obstacle, a box had been placed at the side of the corridor (see Figure 5.10).

In the normal follow mode the robot's translation speed was limited to 0.5m/s, which was reached rather quickly and maintained until the robot came close to the box, which it only late considered an obstacle (at approx. 28s). The robot corrected its heading very often and in a rather shaky manner, as can be seen by the amplitude of the rotation speed curve. The end position was reached only after 42s.

The corridor mode resulted in a shorter trajectory, which took the obstacle much earlier into account. The slow motion between 0s and 10s can be explained by the robot originally facing the wrong direction of the corridor and having to turn around almost in place. From 10s on the robot detected the corridor and started aligning itself in it. After that it accelerated and reached the increased translation speed of 0.8m/s. It only slowed down while passing next to the box. The smooth trajectory planning lead to only small adjustments to the robots rotation speed. The end position was reached after 33s.

5.5 Summary and Outlook

In this chapter, we have introduced the EXPLORER robot system. The EXPLORER implements the approach to multi-layered conceptual spatial mapping from Chapter 3 in an integrated robotic system. The mobile robot base is equipped with different sensors for map building, place and object recognition, and user interaction. We have illustrated how the multi-layered map can be acquired interactively in a so-called guided tour scenario. We have furthermore presented a method for human- and situation-aware people following that makes use of the higher-level information of the multi-layered conceptual spatial map, thus increasing the perceived level of intelligence of the robot. In the next chapter, we will present an extension of the EXPLORER system, in which we make use of prototypical default knowledge for goal-directed planning for situated action in large-scale space.

Chapter 6

Planning and Acting with Spatial Default Knowledge in the EXPLORER

Summary
In this chapter, we present an extension of the EXPLORER *system introduced in the previous chapter. The presented implementation makes use of PECAS, a cognitive architecture for intelligent systems, which combines fusion of information from a distributed, heterogeneous architecture, with an approach to continual planning as architectural control mechanism. We show how the PECAS-based* EXPLORER *system implements the multi-layered conceptual spatial model from Chapter 3. Moreover, we show how – in the absence of factual knowledge – prototypical default knowledge derived from a Description Logic-based ontology using the method presented in Chapter 4 can be used for goal-directed planning for situated action in large-scale space.*

This chapter originates from a joint work with Nick Hawes (CAST architecture and goal-generation), Kristoffer Sjöö (low-level mapping and navigation), Michael Brenner (continual planning), Geert-Jan Kruijff (situated dialogue), and Patric Jensfelt (navigation and robot control), cf. (Hawes et al., 2009b). It also makes use of the concept of cross-modal binding developed by Henrik Jacobsson et al. (2008).

6.1 Motivation and Background

If we wish to build a mobile robotic system that is able to act in a real environment and interact with human users we must overcome several challenges. From a system perspective, one of the major challenges lies in producing a single intelligent system from a combination of heterogeneous specialized modules, e.g., natural language processing, reasoning and planning, conceptual spatial mapping, hardware control, computer vision, etc. Ideally this must be done in a general-purpose, extensible and flexible way, with the absolute minimum of

hardwired behaviors. Additionally, taking account of the "human in the loop" poses the challenge of relating robot-centric representations to human-centric conceptualizations, such as the understanding of large-scale space.

Here, we introduce *PECAS*, an architecture for intelligent systems, and its application in the interactive mobile robot EXPLORER. PECAS is a new architectural combination of information fusion and *continual planning*, which addresses the need to integrate multiple competences into a single robotic system. PECAS plans, integrates and monitors the asynchronous flow of information between multiple concurrent systems. Information fusion provides a suitable intermediary to robustly couple the various reactive and deliberative forms of processing used concurrently in the EXPLORER . The EXPLORER instantiates PECAS around a hybrid spatial model combining SLAM, visual search, and conceptual default inference. We describe the elements of this model, and demonstrate on an implemented scenario how PECAS provides means for flexible control. Section 6.3 presents a complete system run from our implementation, demonstrating how the flow of information and control passes between low and high levels in our system.

We use this scenario to illustrate the usefulness of the spatial model presented in Chapter 3 and how it can be instantiated in a distributed cognitive architecture like PECAS. A special emphasis is put on the utility of *prototypical default reasoning* over OWL-DL ontologies for goal-directed, situated planning and acting in large-scale space. The EXPLORER system presented in this chapter makes use of an approach to continual planning, which offers the possibility to *assert* facts, which can be used to construct an initial plan that leads to the desired goal state, but which also need to be verified as part of the plan execution. Our method for deriving prototypical default knowledge from conceptual terminological knowledge (as described in Section 4.3.1) can provide the planner with such tentative information. Just like the possibility that default knowledge can be invalidated a property of defeasible and nonmonotonic reasoning, continual planning explicitly allows for the falsification of asserted facts. In that case, re-planning is triggered and a different plan that does not make use of the now unavailable assertion is generated.

6.1.1 The PECAS architecture

Recent work in the "CoSy" research project on cognitive systems for intelligent robotics has led to the development of the PLAYMATE/EXPLORER CoSy Architecture Sub-Schema (**PECAS**).[1] PECAS is an information-processing architecture

[1] PECAS is the result of many collaborations of many researchers in the mentioned projects, and, by itself, it is outside the scope of this thesis. We will hence only summarize its main features and properties.

suitable for situated intelligent behavior (Hawes et al., 2009a). The architecture is designed to meet the requirements of scenarios featuring situated dialogue coupled with table-top manipulation (the **P**layMate focus (Hawes et al., 2007)) or situated action and interaction in large-scale space (the **E**xplorer focus (Zender et al., 2008), see also the previous chapter). It is based on the CoSy Architecture Schema (**CAS**), which structures systems into *subarchitectures* (SAs) that cluster *processing components* around *shared working memories*, as proposed by Hawes et al. (2007). In PECAS, SAs group components by function (e.g., communication, computer vision, or navigation). All these SAs are active in parallel, typically combining reactive and deliberative forms of processing, and all operating on SA-specific representations (as is necessary for robust and efficient task-specific processing).

These disparate representations are unified, or *bound*, by a *subarchitecture for binding* (binding SA), which performs abstraction and cross-modal information fusion on the information from the other SAs using the approach of Jacobsson et al. (2008). PECAS makes it possible to use the multiple capabilities provided by a system's SAs to perform many different user-specified tasks. In order to give the robots a generic and extensible way to deal with such tasks, we treat the computation and coordination of overall (intentional) system behavior as a *planning* problem. The use of planning gives the robot a high degree of autonomy: complex goal-driven behaviors need not be hard-coded, but can be flexibly planned and executed by the robot at run-time. The robot can autonomously adapt its plans to changing situations using *continual planning* (Brenner and Nebel, 2009) and is therefore well suited to dynamic environments. Relying on automated planning means that tasks for the robot need to be posed as goals for a planner, and behavior to achieve these goals must be encoded as actions that the planner can process. The following sections expand upon these ideas.

6.1.2 Cross-modal binding

Cross-modal binding is an essential process in information-processing architectures that allow multiple task-specialized (i.e., *modal*) representations to exist in parallel.[2] Although many behaviors can be supported within individual modalities, two cases require representations to be shared across the system via binding. First, the system requires a single, unified view of its knowledge in order to plan a behavior that involves more than one modality (e.g., following

[2] The approach to cross-modal binding itself is outside the scope of this thesis, but it underlies much of the design and implementation of the Explorer system. We will thus summarize its main features in so far as they are relevant for the descriptions in this chapter. More details can be found in (Hawes et al., 2009b) and (Jacobsson et al., 2008).

a command to do something relative to the object or area). Second, binding is required when a subsystem needs information from another one to help it solve a problem.

Each PECAS SA that wishes to contribute information to the shared knowledge of the system must implement a *binding monitor*. This is a specialized processing component which is able to translate from an arbitrary modal representation (e.g., one used for spatial modeling or language processing) into a fixed *amodal* (i.e., behavior neutral) representation. Binding monitors deliver their abstracted representations into the binding SA as binding *proxies* and *features*. Features describe the actual abstract content (e.g., color, concept, or location) in our amodal language, whilst proxies group multiple features into a single description for a piece of content (such as an object, room, or person), or for relationships between two or more pieces of content. The binding SA collects proxies and then attempts to fuse them into binding *unions*. Unions are structures which group multiples proxies into a single, cross-system representation of the same entity. Groupings are determined by feature matching. Figure 6.1 illustrates this principle: the subarchitecture for low-level mapping and navigation (nav SA) and the subarchitecture for conceptual mapping and reasoning (coma SA), provide their information to the binding SA.

Throughout this process links are maintained between all levels of this hierarchy: from modal content, to features and proxies, and then on to unions. These links facilitate access to information content regardless of location. Binding thus supports the two identified cases for cross-modal binding: the collection of unions provide a single unified view of system knowledge, and cross-subsystem information exchange is facilitated by linking similarly referring proxies into a single union.

6.1.3 Planning for action and processing

For PECAS we assume that we can treat the computation and coordination of overall system behavior as a planning problem. This places the following requirements on PECAS:

1. it must be able to generate a state description to plan with;

2. system-global tasks for the robot need to be posed as goals for a (symbolic) planner;

3. and behavior to achieve these goals must be encoded as actions which can be processed by the planner.

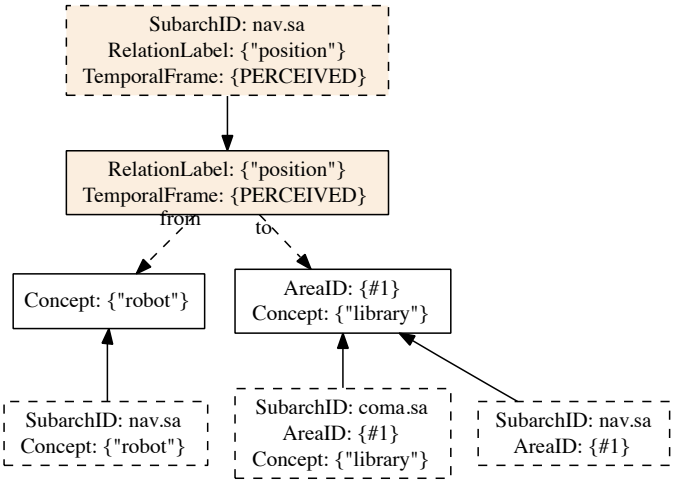


Figure 6.1: Binding localization and conceptual information: "the robot is in the library." Proxies have dashed borders, unions solid borders. Relation proxies and relation unions are colored.

In our implementation we use the MAPSIM continual planner and its Multi-Agent Planning Language (MAPL) as described by Brenner and Nebel (2009).[3] In MAPL, we can model beliefs and mutual beliefs of agents as well as operators affecting these, i.e., perceptual and communicative actions. The continual planner actively switches between planning, execution, and monitoring in order to gather missing goal-relevant information as early as possible.

To provide a planning state, the planning SA automatically translates from the unions in the binding SA into MAPL. The planner thus automatically receives a unified view of the system's current knowledge. The links from amodal unions via subarchitecture-specific proxies back to modal content keep the planning state, and therefore the plans, grounded in representations close to sensors and effectors. In PECAS, planning goals arise as modal *intentional content* which is then abstracted via binding monitors and placed in the planning SA's working memory. From here we use the same translation method as is used on the planning state to produce MAPL goals for the planner.

[3] Continual planning itself is beyond the scope of this thesis. However, we illustrate the benefit of using prototypical reasoning in absence of crisp facts together with a continual planning approach that postpones sub-goal resolution.

The use of continual planning in PECAS is essential, as it is intended for deployment in complex, dynamic situations. In these situations continual planning allows a system to cope with both expected and unexpected change in its environment. In the latter case, execution monitoring detects action failures and triggers re-planning. In the former, novel constructs called *assertions* allow the planning agent to postpone decision making until more information is available.

While the traditional use of planning is achieving goals through physical actions, such direct interpretations of behavior are the exception rather than the rule in cognitive robotics (cf. Shanahan (2002)). Here, where information is incomplete, uncertain, and distributed throughout subsystems, much of the actions to be performed by the system are to do with processing or moving *information*. Whilst some information processing may be performed continually (e.g., SLAM (cf. Section 2.2.1)), much of it is too costly to be performed routinely and should instead be performed only when relevant to the task at hand, i.e., it should be planned based on context.

As each SA in the decentralized PECAS approach is effectively a self-contained processing unit, our design leads naturally to an integration strategy: each SA is treated as a separate agent in a multi-agent planning problem. A crucial feature of this strategy is that each subarchitecture's knowledge is separate within the planning state, and can only be reasoned about using epistemic operators (i.e., operators concerned with knowledge). Likewise, goals are often epistemic in nature, e.g., when a human or a SA wants to query the navigation SA for the location of an object.

To realize internal and external information exchange each subarchitecture can use two epistemic actions, TELL-VALUE and ASK-VALUE, coupled with two facts about SAs, PRODUCE and CONSUME. The actions provide and request information respectively. The facts describe where certain types of information can come from and should go. For example, if a human teacher tells our robot that "this is the kitchen," this gives rise to the motivation that all SAs which consume room knowledge – i.e., the subarchitecture for conceptual mapping (coma SA, see Section 6.2.3) – should know the type of the room in question. This may lead to a plan in which the SA for situated dialogue (comsys SA) uses a TELL-VALUE action to give the coma SA this information.

Using this design, planning of information-processing becomes a matter of planning for epistemic goals in a multi-agent system. This gives the robot more autonomy in deciding on the task-specific information flow through its subsystems. But there is another assumption underlying this design: whilst the binding SA is used to share information throughout the architecture, not all information in the system can or should be shared this way. The principle of *data parsimony* ensures that the system is not overwhelmed with (currently)

irrelevant information. Thus, in order to restrict the knowledge the planner gives "attention" to without losing important information, it needs to be able to *extend* its planning state on-the-fly, i.e., during the continual planning process. In PECAS state extension is supported through the ASK-VALUE and TELL-VALUE actions, and results in a process we call *task-driven state generation*.

6.2 The EXPLORER Instantiation

The binding and planning SAs described above are system and scenario independent. We now discuss the EXPLORER -specific SAs to describe concrete functionality and how this relates to system control. All SAs have been implemented in CAST[4] and tested on a PeopleBot research robot (see also Section 5.1.1). Figure 6.2 shows all subarchitctures used in the EXPLORER PECAS instantiation.

For a mobile robotic system that is supposed to act and interact in large-scale space, an appropriate spatial model is key. The EXPLORER maintains a multi-layered conceptual spatial map of its environment as described in Part I. It serves as a long-term spatial memory of large-scale space. Its individual layers represent large-scale space at different levels of abstraction (cf. Chapter 3), including low-level metric maps for robot motion control, a navigation graph and a topological abstraction used for high-level path planning, and a conceptual representation suitable for symbolic reasoning (cf. Chapter 4) and situated dialogue with a human. In the EXPLORER , different SAs represent the individual map layers. Details on the interactive process of human-augmented map acquisition are given in the previous chapter.

6.2.1 nav SA

The SA for navigation and low-level spatial mapping hosts the three lowest levels of the spatial model (metric map, navigation map, and topological layer). For low-level, metric mapping and localization the nav SA contains a module for laser-based SLAM. The nodes and edges of the *navigation map* represent the connectivity of visited places, anchored in the metric map through x-y-coordinates. Topological areas, corresponding roughly to rooms in human terms, are sets of navigation nodes. This level of abstraction in turn feeds into the conceptual map layer that is part of the coma SA.

The nav SA contains a module for laser-based people detection and tracking as described in the previous chapter. The nav SA binding monitor maintains the robot's current spatial position and all detected people, as proxies and relations on the binding SA. The smallest spatial units thus represented are areas. This

[4]CAST is an open-source toolkit implementing the CAS schema, see [last accessed 2010-04-26]: `http://www.cs.bham.ac.uk/research/projects/cosy/cast`

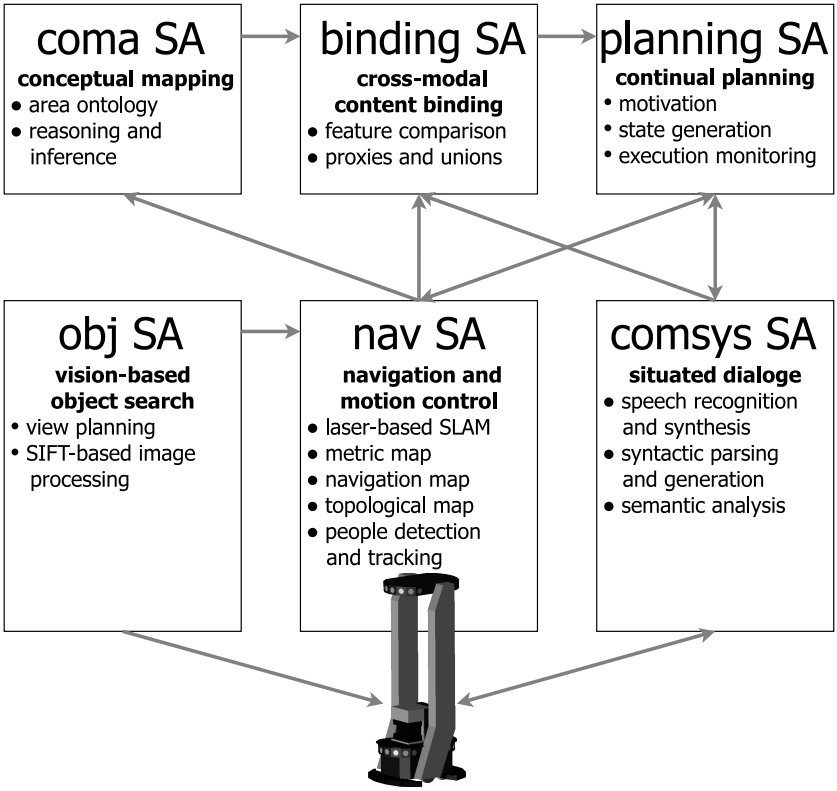


Figure 6.2: The EXPLORER architecture.

provides the planner with a sufficiently stable and continuous description of the robot's state. The planning SA can pose MOVE commands to the nav SA. The target location is defined based on the current task which might be to follow a person, move to a specific point in space, etc. Move commands are executed by a navigation control module, which performs path planning on the level of the navigation graph, but automatically handles low-level obstacle avoidance and local motion control.

6.2.2 obj SA

The subarchitecture for vision-based object search contains the components for finding objects using computer vision. It consists of a module for view planning and one for visual search. The view planning component creates a plan for

Figure 6.3: The "Borland book."

which navigation nodes to visit, in what order and in what directions to look. Details of the process can be found in (Gálvez López et al., 2008). Object instances that are to be detected must be trained off-line in a supervised manner. Figure 6.3 shows a training image used in the implementation: it is assigned the label "borland_book" and the concept "Book." The object recognition makes use of the SIFT feature matching method by Lowe (2004). Objects that are found during run-time of the system are published on the obj SA working memory. The nav SA detects this and in turn extends the spatial model with the new objects. This then propagates the information to the coma SA and, if and when necessary, to the binding SA.

6.2.3 coma SA

The subarchitecture for conceptual mapping and reasoning maintains an abstract symbolic representation of space suitable for situated action and interaction. It represents spatial areas (from nav SA), objects in the environment (from obj SA), and abstract properties of persons (e.g., ownership relations) in a combined TBox, ABox and RBox reasoning framework based on an OWL-DL reasoner. It allows to infer more specific concepts for the area instances as described in Chapter 4. For our implementation, we use the Jena reasoning framework[5] with its built-in OWL reasoning and rule inference facilities. Internally, Jena stores

[5] `http://jena.sourceforge.net/` [last accessed on 2010-04-26]

the facts of the ontology as RDF triples (see Section 4.2.4). The knowledge base can be queried through SPARQL queries (see also Section 4.2.4).

The coma SA makes its information available to the binding SA on demand; i.e., whenever planning SA sends the respective ASK-VAL command, coma SA will add its knowledge about spatial entities, especially their most specific concepts, to the binding SA memory. From there the information then enters the current planning state. In our system the explicit definitions of area concepts through occurrences of certain objects are also used to raise expectations about typical occurrences of certain objects. If the planning SA needs to know the location of an object that has not been encountered before, it can query the coma SA, which will then provide a *prototypical* location of the object in question. This is done using the approach described in Section 4.3.1. An example of this is discussed in Section 6.3.

The default rules in the Jena rule format are listed below. They instantiate closed defaults based on concept definitions that involve role restriction concept constructors (i.e., value restrictions, existential quantifications, and cardinality restrictions, cf. Table 4.2 in Section 4.2.2) from the TBox (see Example 35). Exploiting RBox knowledge, closed defaults for the inverse roles of the ones involved in the concept constructors are also generated (see Example 36).

(35)
```
[closedDefaultsRule:
  (?definedClass rdfs:subClassOf ?def),
  (?def rdf:type owl:Restriction),
  (?def owl:onProperty ?prop),
  (?def ?restr ?restrictedClass),
  (?restrictedClass rdfs:subClassOf owl:Thing),
  notEqual(?definedClass, owl:Nothing),
  notEqual(?restr, rdfs:subClassOf),
  (?ind1 rdf:type ?definedClass),
  (?ind2 rdf:type ?restrictedClass),
  noValue(?y rdf:type default:DefaultStatement),
  noValue(?y rdf:subject ?ind1),
  noValue(?y rdf:predicate ?prop),
  noValue(?y rdf:object ?ind2),
  makeTemp(?x)
  ->
  (?x rdf:type default:DefaultStatement),
  (?x rdf:subject ?ind1),
  (?x rdf:predicate ?prop),
  (?x rdf:object ?ind2)]
```

(36)
```
[inverseDefaultsRule:
  (?def rdf:type default:DefaultStatement),
  (?def rdf:subject ?ind1),
  (?def rdf:predicate ?prop),
  (?def rdf:object ?ind2),
  (?prop owl:inverseOf ?inverseProp),
  noValue(?y rdf:type default:DefaultStatement),
  noValue(?y rdf:subject ?ind2),
  noValue(?y rdf:predicate ?inverseProp),
  noValue(?y rdf:object ?ind1),
  makeTemp(?x)
  ->
  (?x rdf:type default:DefaultStatement),
  (?x rdf:subject ?ind2),
  (?x rdf:predicate ?inverseProp),
  (?x rdf:object ?ind1)]
```

These rules match an RDF graph pattern that is common to OWL-DL concept definitions involving role restrictions. For implementational reasons, the `noValue(...)` clauses must ensure that only one instance of any closed default is generated. The `makeTemp(?x)` functor generates a blank node and binds it to the given variable (here: `?x`). This blank node corresponds to the reified prototypical statement introduced by the rule. As soon as individuals that instantiate the concepts involved in any given concept definition are added to the ABox, the two rules above are triggered and add prototypical statements about these individuals to the DBox.

6.2.4 comsys SA

The subarchitecture for situated dialogue processing has a number of components concerned with understanding and generation of natural language utterances.[6] *Speech recognition* converts audio to possible text strings, which are subsequently parsed. *Parsing* produces a packed representation of logical forms (LFs) (see Section 8.1.3) that correspond to possible semantic interpretations of an utterance. Finally, the semantics are interpreted against a model of the dialogue context. Content is connected to discourse referents, being objects and events talked about over the course of an interaction. In the dialogue context model, both the content of the utterance and its intent are modeled.

[6] More details on the dialogue system used in this work can be found in (Kruijff et al., 2010). Here, we only briefly describe its main properties.

All this information is communicated to the planning and binding SAs through proxies representing the indexical and intentional content of the utterances. In rough terms the *indexical content* (information about entities in the world) is used by the binding SA to link with information from other modalities. Meanwhile the *intentional content* (information about the purpose of the utterance) is used by the planning SA to raise goals for activity elsewhere in the system.

6.3 Example: Finding a Book

This section presents a scenario in which a human asks the EXPLORER to perform a task. It shows how PECAS controls system behavior and information-processing, and illustrates how prototypical knowledge derived from an OWL ontology can be used in goal-directed continual planning. The example is taken directly from our implemented system, showing system visualizations (with minor post-processing).

We assume that the EXPLORER has already acquired a map of its environment, including a number of rooms and a corridor:

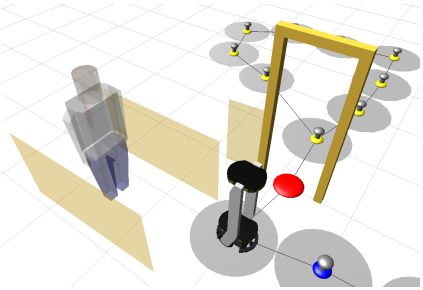
(37) $\mathsf{Corridor}(\mathsf{AREA0})$

(38) $\mathsf{Library}(\mathsf{AREA1}) \in \mathcal{A}_{PECAS}$

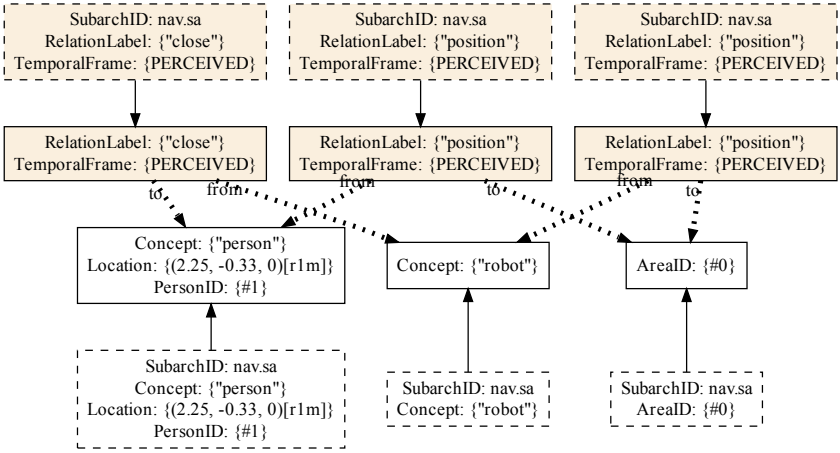
The EXPLORER starts in the spatial context and binding state visualized in Figure 6.4: the robot and person are in the same area, and the person is close to the robot. The ROBOT PROXY is provided by the nav SA which it abstracts from its representation of the robot pose. The PERSON PROXY is provided by the nav SA because a person is being tracked. In addition to these, the nav SA makes available a proxy for the AREA in which one of these proxies occurs, linking them with a POSITION relation proxy. Finally, the CLOSE relation proxy connects the robot proxy to the proxy of the person because the person is geometrically close to the robot. Note that no objects are present, nor are other areas except the current area.

Next, the human approaches the robot and says "find me the Borland book." The comsys SA interprets this utterance, presenting the elements of its interpretation to the rest of the system as proxies. Figure 6.5 shows the results. The EXPLORER itself (the recipient of the order) is represented by a proxy with Concept ADDRESSEE, which binds to the robot proxy already present. The word "me" refers to the speaker, and generates a "person" proxy identified by the Name feature I. The expression referring to the book is given by a "borland_book" proxy, not yet bound to any other proxies at this point.

The comsys SA can determine the intention of this utterance, and separates the intentional elements of the interpretation from the aforementioned descrip-



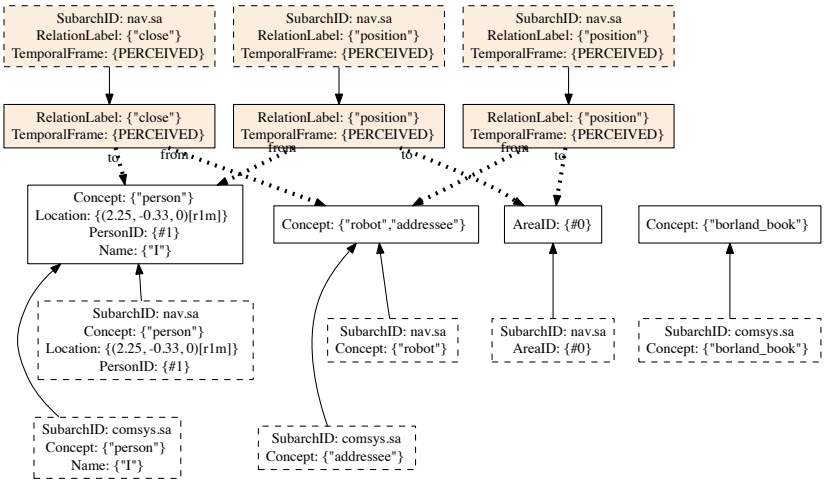
(a) Screenshot of the visualization tool.



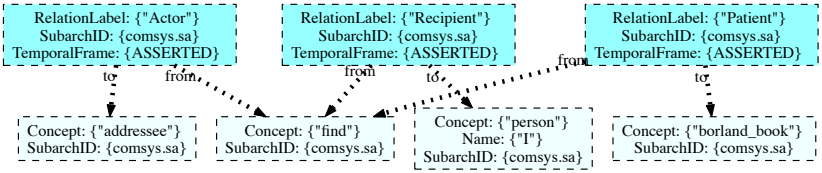
(b) Contents of binding working memory.

Figure 6.4: Initial situation: the user approaches the robot.

tive proxies. This intentional content is written to planning SA as a proxy structure with links back to the binder. The structure of this *motive* can be seen in Figure 6.5b. Planning SA, detecting a new motive, begins the process of creating a plan to fulfill it. First, it converts the information on the binder (shown in Figure 6.5a) to the MAPL representation in Example 39 on the next page. In this process, unions become objects and predicates in the planning state. For instance, as the person union is related by a position relation union to an area union, this is expressed as `(perceived-pos gensym4 : area_0)` in the planning language, where `gensym4` is an auto-generated planning symbol referring to the person, and `area_0` refers to the area. The planner similarly converts the



(a) State of binding SA.



(b) Representation of intentional content (as a *motive*).

Figure 6.5: After the user has uttered the command "Find me the Borland Book."

motive from Figure 6.5b into a MAPL goal `(K gensym4 (perceived-pos gensym6))`. This can be read as the EXPLORER having the goal of the person knowing the position of the book. We use this interpretation of the command "Find me...," as the robot does not have the ability to grasp objects.

(39) Planning state after processing the intentional content from Figure 6.5b.

```
Objects:
(area_id_0 - area-id)
(gensym0 - robot)
(gensym1 - area-name)
(gensym4 - person)
(gensym6 - movable)
```

```
Facts:
(area-id gensym1 : area_id_0)
(area-name area_id_0 : gensym1)
(perceived-pos gensym0 : area_id_0)
(perceived-pos gensym4 : area_id_0)
(close gensym4 gensym0 : true)
```

Given this state and goal, the MAPSIM planner produces the following plan:

(40)
```
L1: (negotiate_plan gensym0 coma_sa)
L2: (tell_val_asserted-pos coma_sa gensym0 gensym6)
L3: (find_a gensym0 gensym6 gensym0)
L4: (tell_val_perceived-pos gensym0 gensym4 gensym6)
```

This plan states that the EXPLORER must find the location of the book (L3), then report this location to the person (L4). Before it does this, it must negotiate with the coma SA (as each subarchitecture is treated as a separate agent) to provide a location where it might be able to find the book (L1, L2). The reasoning behind this plan is that EXPLORER must provide the person with a perceived location for the book (as is specified in the goal), and, having not seen it recently, the only way to obtain a perceived location is via its object search functionality. To perform an object search the system must have both an object to search for (the book in this case) and an area to search.

TYPICAL positions of objects (as opposed to their PERCEIVED positions) are derived from the ontology in coma SA. According to the data parsimony principle (cf. Section 6.1.3), not all knowledge encoded in the coma SA knowledge base should be made available via binding all the time. This would add many extra and redundant facts to the planning state, which is not desirable. Rather, prototypical knowledge is offered by coma SA using a PRODUCE fact (see Section 6.1.3). This allows the planner to query coma SA for prototypical positions when it requires them. One advantage of this on-demand state generation is that the comsys SA could also be used to provide the same knowledge, e.g., through asking a human (and would be if the book was not found initially).

In the above plan, the planner makes use of this by getting the coma SA to TELL-VAL the typical position of the Borland book to the binding SA. Using a set of rules, the coma SA knowledge base always contains prototypical knowledge from instantiating open defaults (see Section 4.3.1 and Section 6.2.3). When asked by the planner for this information (L2), coma SA queries its knowledge base for relevant reified prototypical statements.

Only instantiated prototypical knowledge, i.e., closed defaults, is stored in the knowledge base. In the present case, the robot's ABox contains the fact that

there exists an area (AREA1) which belongs to the class Library. The knowledge that there exists an instance BORLAND_BOOK of type Book is also already part of the knowledge base. Any other concrete information apart from its existence, like its position, is unknown. This knowledge, being ABox knowledge, is not *innate* like the conceptual taxonomy in the TBox. The computer vision component must be trained on specific object instances that it will later be able to detect and recognize. We thus assume that the computer vision component, when loading its available models (and their labels and categories) during initialization, informs the conceptual map about any instance knowledge it has (see Section 6.2.2). In this sense, it can be characterized as *asserted* knowledge: during its (supervised) training phase, the computer vision component was told by a human tutor the corresponding labels for each training image.

Together, the instance knowledge about AREA1 and BORLAND_BOOK, and the terminological knowledge about libraries[7] trigger the instantiation of the default rule and yields the reified statement in Example 44. Moreover, knowing that contains and in are inverse roles yields the statement in Example 45.

(41) $\mathsf{Book}(\mathsf{BORLAND\_BOOK}),\ \mathsf{Library}(\mathsf{AREA1}) \in \mathcal{A}_{PECAS}$

(42) $\mathsf{Library} \equiv \mathsf{Room} \sqcap \geq 100\ \mathsf{contains}.\mathsf{Book} \in \mathcal{T}_{PECAS}$

(43) $\mathcal{T}_{PECAS} \cup \mathcal{A}_{PECAS} \cup \mathcal{D}_{PECAS} \models \delta_{\mathsf{book}_1} =$

$$\frac{\mathsf{Library}(\mathsf{AREA1}) \wedge \mathsf{Book}(\mathsf{BORLAND\_BOOK}) : \mathsf{contains}(\mathsf{AREA1}, \mathsf{BORLAND\_BOOK})}{\mathsf{contains}(\mathsf{AREA1}, \mathsf{BORLAND\_BOOK})}$$

(44)
```
_:book1   rdf:type      DefaultStatement .
_:book1    rdf:subject    area1 .
_:book1    rdf:predicate  contains .
_:book1    rdf:object     borland_book .
```

(45)
```
_:book1i  rdf:type      DefaultStatement .
_:book1i   rdf:subject    borland_book .
_:book1i   rdf:predicate  in .
_:book1i   rdf:object     area1 .
```

Upon receiving the TELL-VAL request, the coma binding monitor then, in absence of factual knowledge, retrieves the prototypical statement from the closed default in Example 45 and translates it to a proxy structure, which it then publishes on the binding SA working memory: coma SA adds one proxy for its BORLAND_BOOK individual, one proxy for its AREA1 individual, and a relation

[7] Note that OWL-DL does not allow for expressing *vague* knowledge like "libraries are rooms that contain *many* books." For the purpose of the present implementation, we chose a minimal cardinality of 100 – and assume that the robot was informed verbally by its user that the room with AreaID #1 is the library.

proxy with TemporalFrame feature TYPICAL, marking the relation as prototypical. Together with the comsys SA proxy already present, and the proxy for Area #1 automatically produced by nav SA (upon noticing that coma SA published a proxy with an AreaID feature), the binding SA generates the structure below.

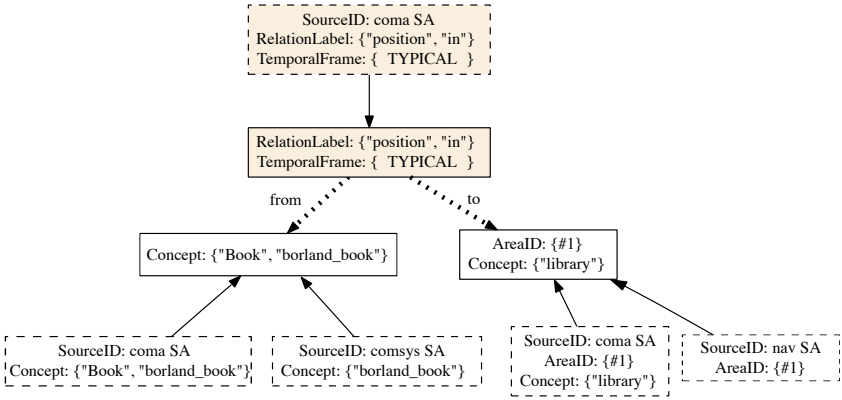


Figure 6.6: Hypothetical position of the Borland book.

Given this hypothesis for the book's location, MAPSIM uses a replanning step to expand the initial plan to include steps to move the robot to the library, search there for the book, then move back and report to the user. The updated plan and planning state are now as follows:

(46) Planning state:

```
Objects:
(area_id_0 - area-id)  (area_id_1 - area-id)
(gensym0 - robot)      (gensym1 - area-name)
(gensym4 - person)     (gensym6 - borland_book)
(gensym6 - movable)    (gensym7 - area-name)

Facts:
(area-id gensym1 : area_id_0)
(area-id gensym6 : area_id_1)
(area-name area_id_0 : gensym1)
(area-name area_id_1 : gensym7)
(asserted-pos gensym6 : gensym7)
(perceived-pos gensym0 : area_id_0)
(remembered-pos gensym4 : gensym1)
```

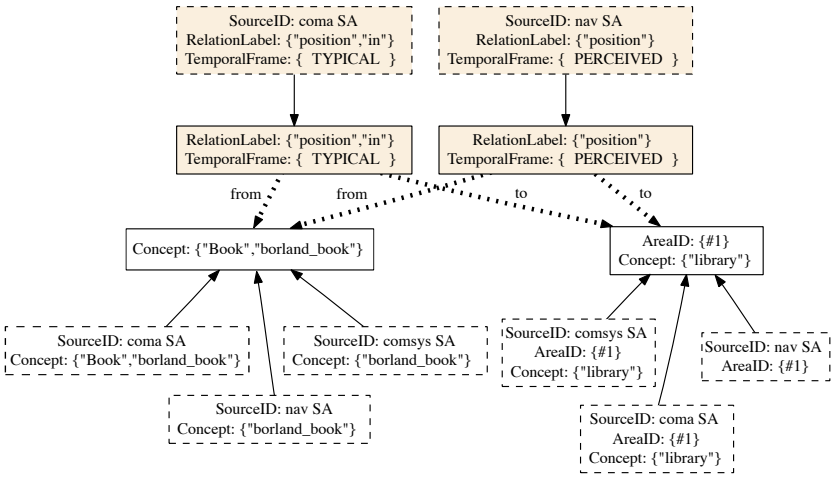


Figure 6.7: Perceived location of the book

(47) Plan:

```
L1: (move gensym0 area_id_1 area_id_0)
L2: (object-search-in-room gensym0 gensym6 area_id_1)
L3: (approach-person gensym0 gensym4 area_id_0)
L4: (tell_val_perceived-pos gensym0 gensym4 gensym6)
```

In the above, `gensym7` is the binding union of the library. Using the AreaID feature from the this union, the planner issues a command to the nav SA which moves the robot to the library (fulfilling step `L1`). As with all other steps in the plan (including the information-processing ones), the results of this action are checked by MAPSIM to determine whether it has completed successfully or whether replanning is required. This check is performed by inspecting the planning state and comparing it to the expected state. This means that all actions must have effects that are visible on the binding SA (for subsequent translation). Once the check has passed for `L1` (confirming the robot has arrived in the library), the planner issues an object search command to the object SA. The EXPLORER searches the room as described previously. Once the object is found, the nav SA adds it to the navigation graph. Since it is part of the current spatial context, it is also exported to the binder in the form of an object proxy, which is connected to the room's proxy by a new position relation proxy. This position relation proxy has a PERCEIVED temporal frame.

The new proxies generated by object search bind to the existing complex (depicted in Figure 6.6), resulting in the structure in Figure 6.7. This binding provides the original comsys SA book proxy with a perceived position (in addition to its prototypical one). With this knowledge in the planning state (i.e., the effect of `L2` is verified, which satisfies one precondition of `L4`), the planner is able to trigger the remaining steps in the plan: moving to the user and reporting the perceived position. A move command is sent to the nav SA referencing the Location feature from the person proxy. Once close to this location, a TELL-VAL is sent to the comsys SA to communicate the book's location to the person. The content generation component in the comsys SA uses the contents of the binding SA working memory (see Figure 6.7) to generate the utterance "the Borland book is in the library", thus completing the plan and satisfying the original goal (that the person knows the position of the book).

6.4 Summary and Outlook

In this chapter, we have presented an extension of the EXPLORER system introduced in Chapter 5. The presented implementation makes use of PECAS, a cognitive architecture for intelligent systems, which combines fusion of information from a distributed, heterogeneous architecture, with an approach to continual planning as architectural control mechanism. We have shown how the PECAS-based EXPLORER system implements the multi-layered conceptual spatial model from Chapter 3. Moreover, we have shown how – in the absence of factual knowledge – prototypical default knowledge derived from a Description Logic-based ontology using the method presented in Chapter 4 can be used for goal-directed planning for situated action in large-scale space. In the next chapter, we will present DORA, a robotic system that is also based on the CAS cognitive architecture. Using a different instantiation of the multi-layered mapping approach, DORA can autonomously acquire a spatial representation of its environment.

Chapter 7

Autonomous Semantic-driven Indoor Exploration with DORA

Summary
In this chapter, we present an approach in which a conceptual map is acquired or extended autonomously, through a closely-coupled integration of bottom-up mapping, reasoning, and active observation of the environment. The approach extends the conceptual spatial mapping approach presented in the previous chapters. It allows for a nonmonotonic formation of the conceptual map, as well as two-way connections between perception, mapping and inference. The approach has been implemented in the integrated mobile robot system DORA. *It uses rule- and DL-based reasoning and nonmonotonic inference over an OWL ontology of commonsense spatial knowledge, together with active visual search and information gain-driven exploration. It has been tested in several experiments that illustrate how a mobile robotic agent can autonomously build its multi-layered conceptual spatial representation, and how the conceptual spatial knowledge can influence its autonomous goal-driven behavior.*

This chapter originates from a joint work with Kristoffer Sjöö (place-based mapping, placeholder creation, and navigation), Alper Aydemir (active visual object search), Patric Jensfelt (low-level navigation and robot control), Marc Hanheide (goal generation and management), and Nick Hawes (cognitive architecture, CAST).

7.1 Motivation and Background

Several approaches to *human-augmented mapping* (see also Section 5.3.5) have recently been proposed. A human guides a robot around an indoor environment, and the robot uses the information obtained through interaction with the human to semantically annotate its map. BIRON (Peltason et al., 2009), ISAC (Kawa-

mura et al., 2008), and the EXPLORER (described in Chapter 5 and Chapter 6) are just a few examples of such mobile robots. But what happens after the "home tour?" After a tour, the robot typically only has a partial representation of the environment. Experience shows that human users do not necessarily visit every place or talk about every object (Topp et al., 2006a). Even when they do, they still might be blocking the robot's view by standing close to the laser scanner or the camera. Ontological reasoning can be used to deal with this partiality, to an extent. It can infer defaults, e.g., what objects can be prototypically found by default in a given location, as described in Chapter 6. But that does not yet provide a fully instantiated map.

This chapter presents an extended approach to semantic mapping in which the robot can autonomously build an instantiated map. The approach presents a closely-coupled integration of several forms of cognitive functionality in a single system. The approach combines the bottom-up construction of a *conceptual map*, typical for a home tour, with *autonomous exploration* and top-down mechanisms for guiding *active visual search*. Visual search, and lower levels of sensor data abstraction such as the building of topological structure, can make the mapping construction process *nonmonotonic*. This is a natural consequence of the uncertainty and partiality of observations the robot is dealing with. Structural and conceptual abstractions may need to be reconsidered in the light of new evidence. The approach we present is capable of such nonmonotonic reasoning for conceptual map construction and revision. Existing approaches for human-augmented mapping do not provide this functionality.

In this chapter, we introduce another integrated robotic system: "DORA the Explorer." It is based on a MobileRobots P3-DX[1] robot platform (see also Figure 2.2b), and is equipped with a custom-built upper structure that holds a pan-tilt unit with a stereo-vision camera. Figure 7.1 as well as Figures 2.3c and 3.2a show the DORA robot. Apart from the usual proprioceptive odometry encoders, its main exteroception sensor is a Hokuyo URG-04LX[2] laser range scanner.

In the following, we first provide an example to illustrate the problems, and connect this to relevant background on semantic mapping. We note shortcomings, and address these in our approach. The full implementation in a mobile robot system is then presented, with a discussion of experimental results obtained in simulation. We focus here on the mapping approach. The use of internal goal generation and management, as well as planning processes for controlling exploration is only briefly highlighted.

[1] `http://www.mobilerobots.com/ResearchRobots/ResearchRobots/PioneerP3DX.aspx` [last accessed on 2010-05-10]

[2] `http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx.html` [last accessed on 2010-05-10]

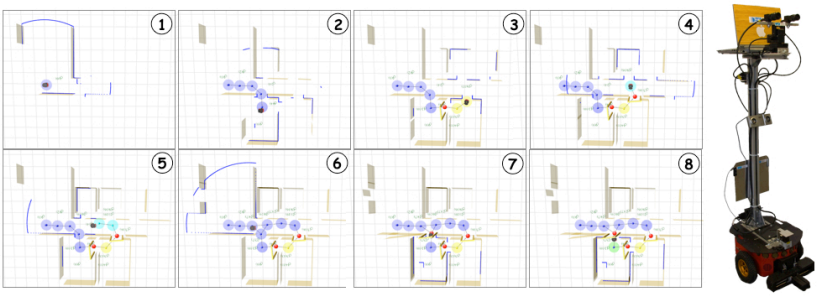


Figure 7.1: Screenshots from an exploration sequence. Red nodes are doorways, colored circles are free space nodes. Nodes having the same color are interpreted as belonging to the same room. Color changes of a node indicates a revision of a room hypothesis, e.g., fusion of nodes into a single room (5 $\rightarrow$ 6) or separation into a new room after observing a doorway (7 $\rightarrow$ 8).

7.1.1 Motivating example

Figure 7.1 illustrates the inherent nonmonotonic nature of the autonomous semantic mapping process we model. (1) shows the initial state. Blue points indicate laser range readings, gray rectangles are walls, and colored circles are (linked) nodes on a navigation graph. If nodes have the same color, they are interpreted as belonging to the same room. (2) shows a sequence of nodes formed after moving around. All nodes belong to a single room, a "corridor," because the robot failed to detect the door it was passing through. In (3) the robot has passed through, and successfully detected, a doorway (red node). This triggers the creation of a new room. In (4) the robot has exited this room through another doorway, re-entering the corridor. At this point, the robot is unaware that it has returned to the same corridor as before. Only in (6) nodes become fully connected. Now, the hypothesis for a new room raised in (4) is fused with the already existing corridor hypothesis, creating a single room. In (7), the robot detects the doorway that it had not spotted earlier, in (2). This leads to a separation of already observed nodes, creating a new room (8).

This is just a short example of an exploration of a confined, previously unknown environment. Nevertheless, any robot that acts is a dynamic environment and operates under the principle of *discovery* for map acquisition and navigation (i.e., there are no distinct learning and operation phases, see also Section 5.3.5) faces similar challenges. As discussed earlier in Chapter 4, in realistic environments the world changes, which means that the agent has to

revise its model of the world. Similarly, the agent cannot rely on its perceptual capabilities to be perfect. Erroneous perceptions and subsequently derived knowledge must be retractable. Finally, any agent operating in large-scale space is faced with the fact that its operating environment is only partially observable. All in all, this means that every mobile agent operating in a sufficiently realistic large-scale environment must be able to deal with *incomplete* and *changing information*. At every point in time, its representation of the world should be as faithful as possible.

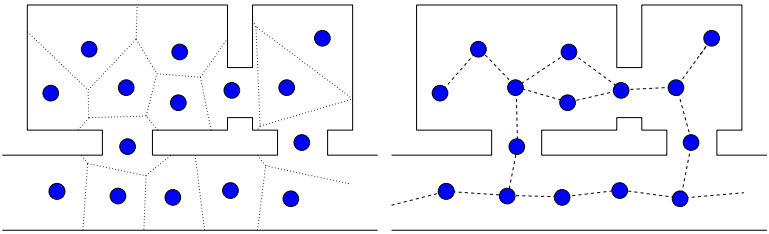
The conceptual mapping approach presented here manages the potentially nonmonotonic formation and maintenance of room representations. It uses topological information to establish the spatial extent of a room. Ontological inference is used to reason about the concept of a room, and what objects it might contain. This in turn guides active visual search. The observations help extend the conceptual map with more instance information.

7.2 Design

The design we present can be considered an improvement over the semantic mapping approach implemented in the EXPLORER system, cf. Chapter 5. That approach still assumed a strongly supervised setting, in which a conceptual map layer was built in a strictly monotonic way. Below, we present a new algorithm for managing the formation of a conceptual map layer in a way that allows for nonmonotonicity. The algorithm uses the notion of topological *Places* and *Placeholders*. These are, in their turn, abstractions from metric mapping data. We first discuss the topological structure, then the conceptual mapping algorithm. The incremental way in which the model is constructed implies that over time, a habituation effect will be observable: with decreasing uncertainty at the lower layers, the conceptual representations will change less often due to error-recovery, and will only be changed to reflect changes in the environment.

Whereas Places in our approach are spatial units that are meaningful for the robot only, rooms are adequate for interaction with a human. Rooms are a human category, and rooms can be conceptualized in a way that is meaningful to humans. Knowledge about rooms and their concepts is thus important for robots that need to perform tasks in common human-oriented environments.

For the purposes of this work, we only consider conceptual room structures to apply to disjoint sets of Place nodes in a topological graph. As a consequence, a flat conceptual map is built, without a partonomic hierarchy based on topological inclusion. This assumption can, however, be easily lifted, along the lines of the approach presented in Part I. In the current implementation, DORA can only reach, explore, and represent areas on a single floor. An approach like the



(a) Places defined by nodes. (b) Graph of Place nodes.

Figure 7.2: Places.

one by Karg et al. (2010) that explicitly represents floors as units in a SLAM-based map could be integrated straightforwardly with our conceptual mapping approach.

7.2.1 Places

The robot uses laser range data to autonomously build a 2D metric map. This map is divided into discrete regions called *Places*. A Place provides a basic form of spatial abstraction, cf. (Pronobis et al., 2009). Here, we define each Place in terms of a map point called a *node*. Nodes indicate "free space," and are created at regular intervals along the robots' trajectory (see also Section 3.2.2). A node defines a Place as the Voronoi cell (Aurenhammer, 1991) surrounding it, as illustrated in Figure 7.2a.

Nodes are connected into a *navigation graph* as the robot transits from one Place to another. Figure 7.2b illustrates such a graph. Graph-edges indicate adjacency of Places, and the possibility of moving between them. This connectivity is used in planning and conceptual reasoning.

7.2.2 Placeholders

Space that has not yet been explored by the robot has no Place nodes in it. Nevertheless, high-level processes like reasoning and planning do need symbols representing areas that could potentially be explored. We facilitate this by giving unexplored space its own representation in *Placeholders*. A Placeholder symbolizes an unexplored direction that the robot might move in – which may or may not yield new Places. Placeholders are stored internally in the form of a position in the map termed a node *hypothesis*, generated in space that is reachable from the current Place, but which is devoid of other nearby Place nodes. This process is illustrated in Figure 7.3.

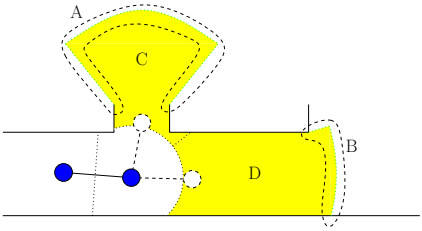


Figure 7.3: Placeholder creation. Dashed circles are hypotheses, each representing one Placeholder. A and B are frontier length estimates, C and D are coverage estimates for the respective Placeholders.

Placeholders and Places have the same high-level representation, as do the links connecting them. Placeholders are ascribed the additional attribute of being *unexplored*, as well as two quantitative measures of estimated information gain should the robot explore them. These are used by the goal-management system, described in Section 7.3. The quantitative measures used are the coverage estimate and the frontier length estimate, cf. Figure 7.3. The former is obtained by measuring the free space visible from the current node and not near to any existing node, and assigning it to the closest hypothesis. This heuristically estimates the number of new Places that would result from exploring that direction. The frontier length estimate is analogously extracted from the length of the border to unknown space. By prioritizing these two measures differently, the goal management mechanism can produce different exploratory behaviors.

7.2.3 Conceptual mapping

Conceptual mapping uses the Place-based topological organization to perform two reasoning tasks.

One, it maintains a representation that groups Places into rooms. This is a difference to the implementation in the EXPLORER system, where the conceptual map layer did not have access to small spatial units (like nodes or Places) and instead relied on the navigation graph layer to provide a segmentation into larger units (namely, areas). As explained in Section 3.1.1, gateways play a special role when structuring and segmenting space within spatial models that are based on representing free space. In our approach, Places represent free space, and the system contains a dedicated module for detecting doorways (see Section 5.2.1 for a summary). The method for structuring and segmenting Places into discrete rooms is based on doorways, which are a typical gateways in indoor environments.

Two, using observations of salient objects, it can infer possible concepts for a room, and objects that are likely to be present by default. Performing these tasks yields a conceptual map of the environment, with room organization, instances, and default expectations. This works similarly to the methods presented in Chapter 5 and Chapter 6, respectively. Much like in the previous chapters, the TBox contains *innate* terminological knowledge, whereas the ABox contains knowledge that is either *acquired* autonomously or *asserted* by a tutor during run-time. On the basis of this knowledge, a reasoner can then discover new, *inferred* knowledge.

However, in this work we specifically take into account that the ongoing construction of the conceptual map is a potentially nonmonotonic process. The overall room organization may be revised on the basis of new observations. The representation underlying the conceptual map is an OWL-DL ontology (as introduced in Section 4.2, consisting of a taxonomy of concepts (the *TBox*) and the knowledge about individuals in the domain (the *ABox*), as well as the knowledge about the properties that can exist in the domain (the *RBox*), and prototypical statements about individuals (in the so-called *DBox*).

The TBox taxonomy from Figure 5.5 and Figure 3.6 must be extended in order to reflect the fact that in the DORA system the spatial units that serve as input to the conceptual map layer are Places. Therefore, the following additional concepts are part of the TBox $\mathcal{T}_{Dora}$.

(48) $\mathcal{T}_{Dora} \supseteq \mathcal{T}_{explorer}$

(49) $\mathsf{PortionOfSpace} \sqsubseteq \top \in \mathcal{T}_{Dora}$

(50) $\mathsf{Area} \sqsubseteq \mathsf{PortionOfSpace} \in \mathcal{T}_{\mathsf{Dora}}$

(51) $\mathsf{Place} \sqsubseteq \mathsf{PortionOfSpace} \in \mathcal{T}_{Dora}$

(52) $\mathsf{Door} \sqsubseteq \mathsf{Place} \in \mathcal{T}_{Dora}$

Place instances are generated in a bottom-up fashion each time a new Place node is created, or an existing Placeholder turns into an explored Place. In the architecture used, this is signaled by the place layer to the conceptual map layer. The latter can then add the respective information to its knowledge base (cf. Example 54 on the next page). Whenever the system detects a doorway at a given Place, this is represented in the ABox in a similar way (cf. Example 55 on the following page). Likewise, if a Place ceases to exist (e.g., if it gets merged with another Place), its Place instance is removed from the ABox. Edges between Place nodes in the navigation graph are the basis for asserting their adjacency (cf. Example 56 on the next page) with the symmetric adjacent role (see Example 53 on the following page). In case a Place instance is deleted from the ABox, the reasoner must perform ABox *contraction* as described in Section 4.3.2 in

order to remove all facts involving the deleted individual (such as adjacency relation assertions). Moreover, doorway detection can go wrong, and thus the belief about rooms in the environment might also change nonmonotonically.

(53) $\mathsf{adjacent} \equiv \mathsf{adjacent}^{-} \in \mathcal{R}_{Dora}$

(54) $\mathcal{A}_{Dora} = \mathcal{A}_{Dora} \cup \{\mathsf{Place}(\mathsf{PLACE1}), \mathsf{Place}(\mathsf{PLACE2})\}$

(55) $\mathcal{A}_{Dora} = \mathcal{A}_{Dora} \cup \{\mathsf{Door}(\mathsf{PLACE2})\}$

(56) $\mathcal{A}_{Dora} = \mathcal{A}_{Dora} \cup \{\mathsf{adjacent}(\mathsf{PLACE1}, \mathsf{PLACE2})\}$

Besides the usual inferences performed by the OWL-DL reasoner (see Section 4.2.3), namely *subsumption checking* for concepts in the TBox (i.e., establishing subclass/superclass relations between concepts) and *instance checking* for ABox members (i.e., inferring which concepts an individual instantiates), an additional *rule engine* (see Section 4.2.6) is used to form and maintain Room instances based on adjacency of Places. Based on adjacency, a rule maintains another role, the symmetric and reflexive sameRoomAs, that expresses which Places belong to the same room. The rule engine, monitors the knowledge base and adds a sameRoomAs fact, whenever two Place instances fulfill the antecedent (left-hand side) of the respective rule. These rules (listed in Figure 7.4) are interpreted nonmonotonically: whenever a previously true antecedent turns false, its consequent (right-hand side) statements are retracted from the ABox.

Continuing our example, let us assume that the robot discovers some more Places, and represents that some of them are adjacent (see Examples 57 and 58). According to the rules 1, 2 and 3 (from Figure 7.4), the knowledge base $\mathcal{O}_{Dora}$ then contains the facts in Example 60.

(57) $\mathcal{A}_{Dora} = \mathcal{A}_{Dora} \cup \{\mathsf{Place}(\mathsf{PLACE3}),$
$\mathsf{Place}(\mathsf{PLACE4}), \mathsf{Place}(\mathsf{PLACE5})\}$

(58) $\mathcal{A}_{Dora} = \mathcal{A}_{Dora} \cup \{\mathsf{adjacent}(\mathsf{PLACE3}, \mathsf{PLACE4}),$
$\mathsf{adjacent}(\mathsf{PLACE4}, \mathsf{PLACE5})\}$

(59) $\mathcal{A}_{Dora} \cup \mathcal{R}_{Dora} \models \{\mathsf{adjacent}(\mathsf{PLACE4}, \mathsf{PLACE3}),$
$\mathsf{adjacent}(\mathsf{PLACE5}, \mathsf{PLACE4})\}$

(60) $\mathcal{O}_{Dora} \models \{$
$\mathsf{sameRoomAs}(\mathsf{PLACE3}, \mathsf{PLACE3}), \mathsf{sameRoomAs}(\mathsf{PLACE4}, \mathsf{PLACE4}),$
$\mathsf{sameRoomAs}(\mathsf{PLACE5}, \mathsf{PLACE5}), \mathsf{sameRoomAs}(\mathsf{PLACE3}, \mathsf{PLACE4}),$
$\mathsf{sameRoomAs}(\mathsf{PLACE4}, \mathsf{PLACE3}), \mathsf{sameRoomAs}(\mathsf{PLACE3}, \mathsf{PLACE5}),$
$\mathsf{sameRoomAs}(\mathsf{PLACE5}, \mathsf{PLACE3}), \mathsf{sameRoomAs}(\mathsf{PLACE4}, \mathsf{PLACE5}),$
$\mathsf{sameRoomAs}(\mathsf{PLACE5}, \mathsf{PLACE4})\}$

Rule 1 for Place instance x:
$\mathsf{Place}(x)$ & $\neg\mathsf{Door}(x)$
$\Rightarrow \mathsf{sameRoomAs}(x, x)$

Rule 2 for Place instances x, y:
$\mathsf{adjacent}(x, y)$ & $\neg\mathsf{Door}(x)$ & $\neg\mathsf{Door}(y)$
$\Rightarrow \mathsf{sameRoomAs}(x, y)$

Rule 3 for Place instances x, y, z:
$\mathsf{adjacent}(x, z)$ & $\mathsf{sameRoomAs}(y, z)$
& $\neg\mathsf{Door}(x)$ & $\neg\mathsf{Door}(y)$ & $\neg\mathsf{Door}(z)$
$\Rightarrow \mathsf{sameRoomAs}(x, y)$

Rule 4 for Place instances x, y, and Room instance z:
$\mathsf{sameRoomAs}(x, y)$ & $\mathsf{contains}(z, x)$
$\Rightarrow \mathsf{contains}(z, y)$

Rule 5 for Place instance x:
$\neg\mathsf{Door}(x)$ & $\neg\mathsf{contains}(y, x)$
$\Rightarrow$ **generateNewInstance(z)** &
$\mathsf{Room}(z)$ & $\mathsf{contains}(z, x)$ & $\mathsf{hasSeedPlace}(z, x)$

Rule 6 for Room instances x, y, and Place instance z:
$\mathsf{hasSeedPlace}(x, z)$ & $\mathsf{contains}(y, z)$ & $x \neq y$
$\Rightarrow$ **deleteInstance(y)**

Figure 7.4: Rules for room segmentation. Internally, the rules perform closed-world reasoning: negation is interpreted as absence of the positive facts. The rules ensure that only Places that are transitively interconnected (i.e., adjacent) without passing a doorway Place are asserted to belong to the same room. The reflexive sameRoomAs role thus provides an extensional, bottom-up definition of which segments of space consist a room.

If DORA then detects a door at PLACE4 that it previously did not spot, the now invalid consequents of the rules are removed from the knowledge base:

(61) $\mathcal{A}_{Dora} = \mathcal{A}_{Dora} \cup \{\mathsf{Door}(\mathsf{PLACE4})\}$

(62) $\mathcal{O}_{Dora} \models \{$
$\mathsf{sameRoomAs}(\mathsf{PLACE3}, \mathsf{PLACE3}), \mathsf{sameRoomAs}(\mathsf{PLACE5}, \mathsf{PLACE5})\}$

Based on the extensional sameRoomAs role, another rule is responsible for asserting the containment of Places in rooms (through the contains/in roles). Rule 5 handles the creation of new Room instances. This is where an external function (**generateNewInstance**) needs to be executed in order to introduce a new symbol to the ABox. In case two existing rooms are merged, rule 6 applies, which takes care of deleting one room symbol and all the relations it has through an external function (**deleteInstance**). For this, we make use of the notion of *seed Place*, which is usually the first Place found in a room. All places in the same room (according to rule 3 and 5) are then asserted to belong to the same room instance as the seed place. The creation of a new room symbol cannot be expressed in the first-order logic-like rule syntax. That's why room creation and maintenance relies on external functions. Since the results of external functions are not transparent to the rule engine, rules 5 and 6 cannot be undone by simple belief revision. Rule 6 hence provides an explicit trigger for the deletion of Room instances previously created by rule 5. Together, rule 5 and rule 6 alter the knowledge base in a nonmonotonic way. Further deletions based on the contraction initiated by rule 6 are then again handled automatically by the reasoning and rule engine.

(63) after Example 60 on page 132:
$\mathcal{O}_{Dora} \models \{ \mathsf{contains}(\mathsf{ROOM1}, \mathsf{PLACE3}), \mathsf{contains}(\mathsf{ROOM1}, \mathsf{PLACE4}), \mathsf{contains}(\mathsf{ROOM1}, \mathsf{PLACE5}) \}$

(64) after Example 62 on the previous page:
$\mathcal{O}_{Dora} \models \{ \mathsf{contains}(\mathsf{ROOM1}, \mathsf{PLACE3}), \mathsf{contains}(\mathsf{ROOM2}, \mathsf{PLACE5}) \}$

Rooms are usually *extended* as the robot keeps exploring its environment. *Splitting* of rooms occurs when a doorway is correctly detected only later. *Merging* of rooms occurs when the robot enters the same room from a different side, which leads to the creation of a new room instance, and then closes the connection to the already existing places in that room. The newer one of the two merged room instances is then deleted.

7.3 Implementation

Below we discuss how the above design has been implemented in a cognitive system running on a mobile robot platform. The implementation combines the spatial mapping functionality with active visual search, and goal generation and management mechanisms to autonomously drive exploration. Goal generation is based on planning. It uses information gain and the current state of the map to decide whether to plan for further spatial exploration (achieved through ex-

ploring Placeholders), or for obtaining more categorical information (achieved through active visual search).

7.3.1 Architecture design

The integrated system is built using the cognitive robotics software framework CAST.[3] CAST is an event-driven architecture, built from one or more subarchitectures (Hawes and Wyatt, 2010), as described in more detail in the previous chapter. In a nutshell, each subarchitecture (SA) provides a certain functionality. It consists of independently executing software processing components, and a common working memory through which the components exchange information. SAs can likewise exchange information through read/write-operations on each other's working memories. We use the "Player/Stage" middleware [4] to integrate sensorimotor I/O and control into the CAST system.

Our system incorporates five subarchitectures. The spatial SA constructs the representations of spatial knowledge. The active visual search (AVS) SA finds objects using computer vision and view planning. The binding SA serves to fuse information from different modalities, into singular amodal representations (Jacobsson et al., 2008) (see also Section 6.1.2 for more details on the approach). The goal-management and planning SAs use the data from binding to generate goals and plans for achieving them (see Section 6.1.3). Line in the EXPLORER system presented in the previous chapter, the planning SA performs high-level symbolic planning using the amodal information from the binding SA as planning state. The goals are given by the goal-management SA, which uses introspective mechanisms to determine possible actions that extend the agent's knowledge. The planning SA issues action commands to the AVS and spatial SAs.

7.3.2 goal-management SA

The goal-management SA is an architectural concept for *goal selection*.[5] In the context of exploration as discussed here, it decides on a behavioral level which exploration goal to pursue next. Basically, we consider two types of goals: exploration to extend the spatial coverage of the map, or exploration to increase the amount of conceptual instance information in the conceptual map.

Symbolic planning itself has been widely researched. Yet, comparatively little attention has been paid to where the goals for planning processes come

[3] `http://baltcast.sourceforge.net/` [last accessed on 2010-04-15]

[4] `http://playerstage.sourceforge.net/` [last accessed on 2010-04-15]

[5] Goal generation, selection and management is beyond the scope of this thesis. We summarize the relevant principles that are used in the present implementation. Further details can be found in (Hanheide et al., 2010)

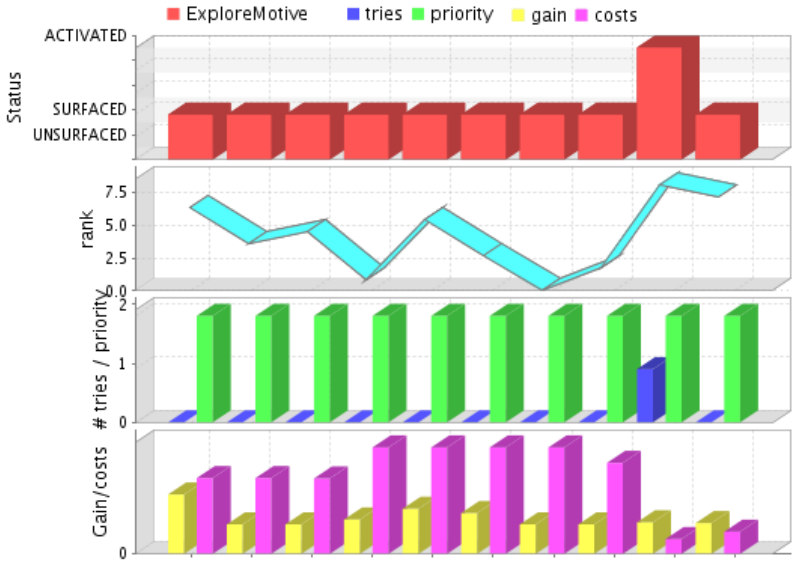


Figure 7.5: Visualization of a goal-management SA state with multiple competing goals (here: "motives" for exploring different Placeholders).

from. We propose an architecture for goal generation and management based on work by Wright et al. (1996). In brief, this architecture is composed of reactive *goal generators*, *filters*, and *management mechanisms*. The goal generators create new goals from modal content in spatial SA, and amodal content on binding SA. The filters do a first pass selection of goals to be considered for activation. Management mechanisms determine which of these goals should be *activated*, i.e., planned for. The system can generate multiple new goals asynchronously, e.g., when a new area of space is sensed, or when a command is given. At the same time it also determines which collection of goals the system should currently try to achieve, e.g., which space to explore, or whether exploration or categorization goals should be pursued. Figure 7.5 shows an example of multiple competing system goals.

7.3.3 spatial SA

The SA most central to exploration is the spatial SA. Its components work together to extract abstract representations from raw sensory data, and to translate high-level actions back to low-level motor commands. Figure 7.6 illustrates the data flow in the Spatial SA. It is organized in a layered manner along the princi-

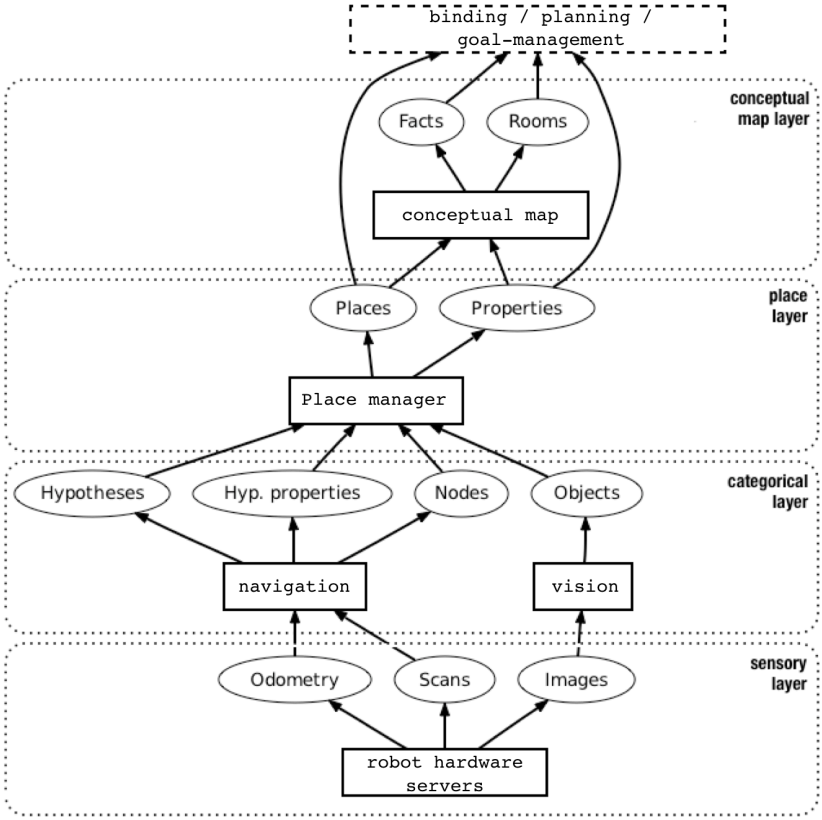


Figure 7.6: Data flow in the spatial SA.

ples presented in Chapter 3. It makes use of the concepts proposed by Pronobis et al. (2009).

The *sensory layer* provides continuous low-level readings from sensors. Readings are clustered and classified quantitatively in the *categorical layer*. The results are used in the *place layer* to form discrete Places and Placeholders, and their associated properties. The components of the *conceptual map layer* perform qualitative reasoning over these abstractions. Firstly, the conceptual map layer segments interconnected Places into rooms and maintains Room instance representations, as described earlier. Second, the reasoner tries to infer more special categories for rooms, e.g., Office or Kitchen. It makes use of in-

ference mechanisms described in Section 4.2. The combined rule and OWL reasoning is done using the Jena reasoner.[6]

The output of both place and conceptual layers are presented to the system at large, through the amodal representations of the binding SA. The goal-management and planning SAs use this information to decide on how to continue autonomous exploration. The goal-management SA selects a goal to be pursued, e.g., for a certain Placeholder to be explored, and the planning SA constructs a plan that will fulfill it. The actions that make up this plan are then fed back into the spatial SA, and turned into concrete continuous-space motor commands by the respective layers (not shown).

7.3.4 AVS SA

The active visual search SA is responsible for finding objects in rooms. AVS is triggered when the goal-management SA selects a goal for categorizing a certain room. The process maintains several information flows between the AVS SA and the spatial SA. One, observed objects are provided to the conceptual map in the spatial SA, to infer more specific concepts than Room. Two, given a goal to locate a particular object, the AVS SA uses information from the place- and conceptual map layers to determine in which rooms the object is likely to be found (e.g., coffee machines in kitchens), making use of prototypical knowledge (see also Section 4.3 and Chapter 6). Like this, conceptual knowledge can feed back to sensory modalities and provide a kind of *attentional priming*.

Our implemented algorithm is a derivation of the one by González-Banos and Latombe (2001).[7] Once the robot is in a room that is to be searched, the AVS SA identifies parts of the room where objects can more likely be found. The idea behind such indirect search is that the time cost of finding possibly object-rich parts of a room is almost always smaller than a full scale random search over the whole area (Tsotsos, 1992). Free space is assumed to be devoid of objects, and, conversely, obstacles and landmarks on the low-level map are likely to include objects. The search plan hence starts from positions which provide the most coverage of seen obstacles, and generates view points in an art-gallery problem fashion (Shermer, 1992; O'Rourke, 1987).

7.4 Experiment

The integrated robot system described here has run for many hours at different sites in different countries, being one of the demonstrator scenarios of the re-

[6] `http://jena.sourceforge.net/` [last accessed on 2010-04-15]

[7] Active visual search is beyond the scope of this thesis. We hence only provide a short introduction to the problem and sketch the approach chosen for the DORA system.

search project "CogX." However, in order to eliminate noise from environmental variation and to gather as much comparable data as possible, we evaluated our approach using the "Player/Stage" simulator. The goal was to assess the accuracy and appropriateness of our nonmonotonically built spatial representation. The system consisted of precisely the same implementation as used on the robot albeit with simulated sensor and motor interfaces. The test environment was a floor-plan map of one of our office environments, shown in Figures 7.7 and 7.8. The map consisted of eight rooms: a corridor, a terminal room, a lab, two offices, two restrooms, and a printer room. This constitutes the ground truth for our tests of the accuracy of the room maintenance. The robot was ordered to perform an autonomous exploration, controlled by a symbolic planner. The top-level goal-management system would select appropriate locations for exploration based on the notion of Placeholders. To assess the coverage that this exploration yields, we determined a gold standard of 60 Place nodes to be generated in order to fully, densely and optimally cover the simulated environment. We achieved this by manually steering the robot to yield an optimal coverage, staying close to walls and move in narrow, parallel lanes.

We performed three runs with the robot in different starting positions, each time with an empty map. Each run was cut-off after 30 minutes. The robot was then manually controlled to take the shortest route back to its starting position.

For the evaluation, the system state was logged after fix intervals. At each such step, the generated map was compared to the ground truth for the room representation and to the gold standard for Place node coverage. The first Room instance to cover part of a ground-truth room is counted as *true positive (TP)*. If that Room instance extends into a second room, it is counted as TP only once, and once as a *false positive (FP)*. Each additional Room instance inside a ground-truth room is also counted as FP. *False negatives (FN)* are ground-truth rooms for which no instance exists. Using these measures, precision P, recall R and the balanced f-score F for the room maintenance are as follows. Moreover, we compute a normalized value for coverage.

$$P = \frac{|\,TP\,|}{|\,TP\,| + |\,FP\,|} \tag{7.1}$$

$$R = \frac{|\,TP\,|}{|\,TP\,| + |\,FN\,|} \tag{7.2}$$

$$F = 2 \times \frac{P \times R}{P + R} \tag{7.3}$$

$$coverage = \frac{|\,nodes\,|}{60} \tag{7.4}$$

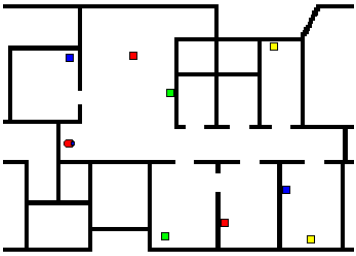
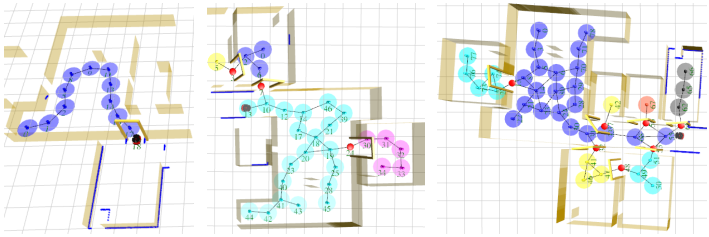


Figure 7.7: Stage simulation model used in the experiments



(a) During run 1 (b) During run 2 (c) During run 3

Figure 7.8: Screenshots acquired at different stages of the experiments.

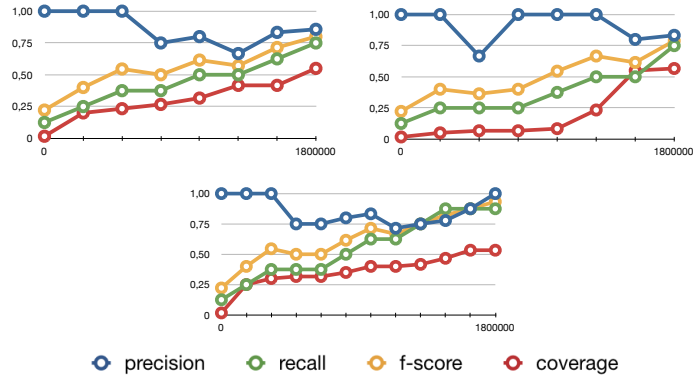


Figure 7.9: Plots for precision, recall, balanced f-score and coverage of each of the three experimental runs. The Y-axis shows the normalized values for precision, recall, balanced f-score, and coverage (0–1). The X-axis is time, in milliseconds.

Figure 7.9 shows the progression of these measures during the three experimental runs. As can be seen, the accuracy (balanced f-score) of the representation is increasing towards a high end result (0.80, 0.79 and 0.93, respectively). The increases and decreases in precision during the individual runs are due to the introduction and retraction of false room instances. Recall can be interpreted as coverage in terms of room instances. After 30 minutes, the exploration algorithm yielded a relatively high recall value (0.75, 0.75 and 0.88, resp.), i.e., most of the rooms had been visited. A recurring problem here was that the two smallest rooms were often only entered by a few decimeters. This was enough to consider the corresponding Placeholder to be explored, but not enough to create an additional Place node beyond the doorway – which would have been the prerequisite for the creation of a Room instance. The node coverage that the algorithm achieved after 30 minutes (33, 34, 32 out of 60, respectively) can be attributed partly to the 30-minutes cut-off of the experiment, and partly to the exploration strategy which goes for high information gain Placeholder first. These tend to be in the middle of a room rather than close to its walls, which means that larger areas are covered with less Place nodes than maximally possible.

7.5 Summary and Outlook

We have presented an approach that integrates several levels of cognitive functionality for a mobile robot system. The robot is able to (a) explore an indoor environment, (b) autonomously construct a multi-layered map of that environment, and (c) deliberate on the basis of the state of the map whether to explore new space, or categorize known rooms.

The integrated robotic system DORA we have introduced here is an extension of the EXPLORER system presented in the previous chapters. We have presented a new algorithm that is capable of dealing with the partiality and uncertainty inherent to mapping. It can handle the nonmonotonicity in forming and maintaining rooms. It uses an instance of the multi-layered conceptual spatial mapping approach from Chapter 3, and it makes use of OWL-DL and rule-based reasoning (as presented in Chapter 4) for room maintenance. This provides the basis for a possible integration with other functionality, such as situated dialogue processing in human-robot interaction, which will be presented in the following chapters.

Part III

Establishing Reference to Spatial Entities

Chapter 8

Situated Resolution and Generation of Referring Expressions

Summary
In this chapter, we present an approach to the task of generating and resolving referring expressions to entities in large-scale space. It is based on the spatial knowledge base presented in Part I. Existing algorithms for the generation of referring expressions try to find a description that uniquely identifies the referent with respect to other entities that are in the current context. The kinds of autonomous agents we are considering, however, act in large-scale space. One challenge when referring to elsewhere is thus to include enough information so that the interlocutors can extend their context appropriately. We address this challenge with a method for context construction that can be used for both generating and resolving referring expressions – two previously disjoint aspects. We show how our approach can be embedded in a bi-directional framework for natural language processing for conversational robots.

This chapter originates from joint work with Geert-Jan M. Kruijff (overall communication system, and the CCG grammar) and Ivana Kruijff-Korbayová (utterance planning and utterance production), cf. (Zender and Kruijff, 2007b; Zender et al., 2009a,b). More people have participated in the design and implementation of the communication system for situated natural language processing, cf. (Kruijff et al., 2010).

8.1 Motivation and Background

The robots described in the previous chapters so far needed only limited dialogue capabilities. Once such robots are supposed to assist people in more demanding everyday tasks, they will need to be endowed with further natural langauge capabilities. For example, imagine a robot that can deliver objects,

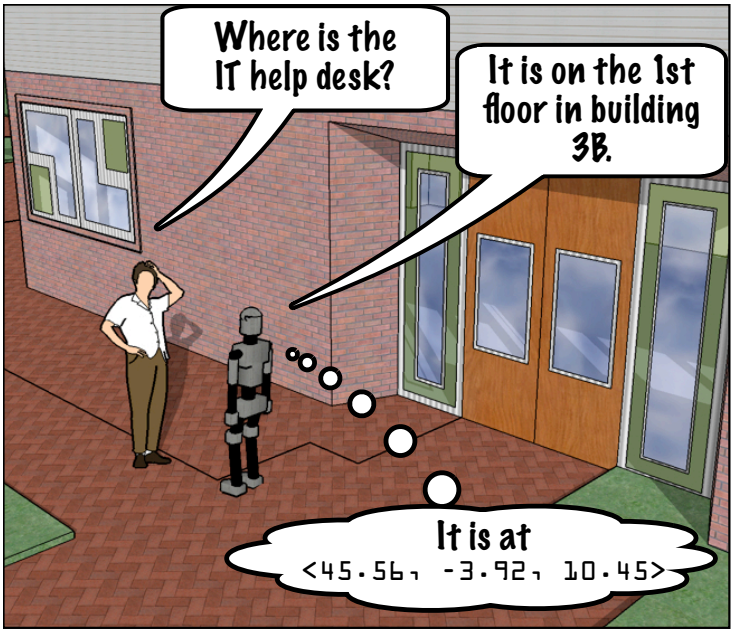


Figure 8.1: Situated dialogue with a campus service robot.

and give directions to visitors on a university campus. Such a robot must be able to verbalize its knowledge in a way that is understandable by humans, as illustrated in Figure 8.1, in an interaction setting that is *situated* in a *human-oriented environment*.

In Part I, we showed how a machine can be endowed with a *human-compatible knowledge representation*. But there is more to a successful verbal interaction than making use of human-compatible concepts. In a physically situated dialogue setting, the location of the interlocutors as well as the things being talked about, has a crucial influence on how mutual reference – that is, a common understanding of the interlocutors which things in the world are being talked about – to entities in the environment is established. The focus of this work lies on reference to entities that are outside the currently observable scene. In other words, we pick up the dichotomy between *small-scale space* and *large-scale space* as defined in Section 3.1.1.

Despite large-scale space being not fully observable, people can nevertheless have a reasonably complete mental representation of, e.g., their domestic

or work environments in their *cognitive maps*. Details might be missing, and people might be uncertain about particular things and states of affairs that are known to change frequently. Still, people regularly engage in a conversation about such an environment, making successful references to spatially located entities.

There are conceivably many ways in which a physically situated agent might refer to things in the world, but many such expressions are unsuitable in many situations. Consider the following set of examples:

1. "position $P = \langle 45.56, -3.92, 10.45 \rangle$"
2. "the area"
3. "Peter's office at the end of the corridor on the third floor of the Acme Corp. building 7 in the Acme Corp. complex, 47 Evergreen Terrace, Calisota, Earth, (...)"

Clearly, these REs are valid descriptions of the respective entities in the agent's world representation. Still they fail to achieve their *communicative goal* (Grice, 1975), which is to specify the right amount of information so that the hearer can easily uniquely identify what is meant.

The following expressions *might* serve as more appropriate variants of the previous examples (*in certain situations!*):

1. "the IT help desk"
2. "the large hall on the first floor"
3. "Peter's office"

The question then remains how a *natural language processing* (NLP) system can generate such expressions which are suitable in a given situation. In this chapter, we identify some of the challenges that an NLP system for situated dialogue about *large-scale space* needs to address, and discuss a model that addresses these.

Specifically, we present a situated model for generating and resolving referring expressions, with a special focus on how a conversational mobile robot can produce and interpret such expressions against an appropriate part of its acquired knowledge base. A benefit of our approach is that most components, including the situated model and the linguistic resources, are bi-directional, i.e., they use the same representations for comprehension and production of utterances. This means that the proposed system is able to understand and correctly resolve all the referring expressions that it is able to generate.

8.1.1 Referring expressions

Referring expressions (REs) are *definite descriptions* that serve the communicative goal of enabling the hearer to "pick out whom or what [the speaker] is talking about" (Donnellan, 1966). Strawson (1950) identifies the following class of referring expressions:

> *"singular demonstrative pronouns ('this' and 'that'); proper names (e.g., 'Venice', 'Napoleon', 'John'); singular personal and impersonal pronouns ('he', 'she', 'I', 'you', 'it'); and phrases beginning with the definite article followed by a noun, qualified or unqualified, in the singular (e.g., 'the table', 'the old man', 'the king of France')."*

More recently, the notion of referring expressions has been broadened and narrowed in different aspects. First of all, as explained by Reiter and Dale (1992), *felicitous* referring expressions are characterized by their communicative purpose. A definite description is a referring expression "if and only if (...) it is *intended* to identify the object it describes to the hearer," the so-called *intended referent*. Appelt (1985) divides this intention further: either the "speaker intends to refer to a mutually known object," or the "speaker has an implicit assumption that the hearer identify a referent." This *referential* use of definite descriptions contrast with their *attributive* use, in which "the speaker has an implicit intention that the hearer not identify a referent" (Appelt, 1985). This class of definite descriptions falls outside the scope of this work. We only consider referring expressions, i.e., referential definite descriptions.

Secondly – extending the above list by Strawson (1950) – plural expressions, constituting referring expressions that refer to sets of objects, have recently become an active subject of research (Stone, 2000; van Deemter, 2000).

Moreover, the distinction between *anaphoric* and *exophoric* references has recently received attention (Krahmer and Theune, 2002). Whereas exophora express reference to entities outside the discourse context – thus introducing *referents* into the discourse context – anaphora refer back to already introduced referents. Vieira and Poesio (2000) distinguish between "direct anaphora," which use the same head noun to pick up a previous referent, "bridging descriptions" using a different head noun for picking up an already introduced entity, and *"discourse-new"* references. These are defined as "first-mention definite descriptions that denote objects not related by shared associative knowledge to entities already introduced in the discourse" (see also (Prince, 1992)). Exophora are discourse-new references. The distinction between direct anaphora and bridging descriptions is more important from a stylistic viewpoint. The

use of bridging descriptions, for instance, can be advisable, depending on the intended register or text genre, in order to avoid repetitions. When resolving references, however, it is important that a reference resolution mechanism successfully identify such a *coreference*.

In *natural language generation* (NLG) the task of *generating referring expressions* (GRE) is finding an appropriate verbal expression that successfully identifies an *intended referent* to the hearer on first mention. This implies that the description must be chosen in a way that prevents it from referring to another entity in the current *context*. This context includes the discourse context, the immediate surroundings of the interlocutors (the visual context), but also world knowledge and other implicit referents that the speaker or hearer might have in mind. Other entities in the context that are not intended referents could be mistakenly assumed as referents by the hearer. These are called *potential distractors*. A successful referring expression must thus distinguish the intended referent from its potential distractors.

Conversely, in natural language comprehension, *resolving referring expressions* (RRE) is concerned with identifying which entity is referred to by the speaker. In everyday language, felicitous uses of definite descriptions are not restricted to already *evoked* entities or previously introduced discourse referents. Poesio and Vieira (1998), Vieira and Poesio (2000), and Poesio et al. (2004) found in several corpus studies, i.e., studies on large collections of written documents (e.g., newspaper articles) that discourse-new definite descriptions make for a large portion of all definite descriptions. We claim that the same is true for situated dialogues about entities in large-scale space. A module for resolving referring expressions in such dialogues must hence be able to establish which entity in the world is being talked about by finding an appropriate referent in its knowledge base, thus going beyond intra-linguistic coreference resolution (Byron and Allen, 2002).

Usually, GRE has been viewed as an isolated problem, focusing on efficient algorithms for determining which information from the domain must be incorporated in a noun phrase such that it allows the hearer to optimally understand which referent is meant. Other challenges addressed in the GRE field involved psycholinguistic plausibility, algorithmic elegance, and representational efficiency. The domains of such approaches usually consist of small, static domains or simple visual scenes.

In their seminal work Dale and Reiter (1995) present the *incremental algorithm* (IA) for generating referring expressions. The IA constitutes an approach to the GRE problem, which they rephrase in terms of the *Gricean Maxims* (Grice, 1975). Inherently, any referring expression should fulfill the Maxim of Quality in that it should not contain any false statements. The algorithm also

ensures that only properties of the referent that have some discriminatory power are realized (Maxim of Relevance). Moreover, they try to fulfill the Maxims of Manner and Quantity in that the produced expressions are short and do not contain redundant information. The incremental algorithm provides a solution to the GRE problem with a reasonable run-time complexity. This is achieved by not trying to find an optimal referring expression, which Dale and Reiter justify by findings in psycholinguistics.

In more recent work, van Deemter (2002), and Krahmer and Theune (2002) propose extensions to the IA that address some of its shortcomings, such as negated and disjoined properties (van Deemter, 2002) and an account of salience for generating contextually appropriate shorter, anaphoric referring expressions (Krahmer and Theune, 2002). Other, alternative GRE algorithms exist (Horacek, 1997; Bateman, 1999; Krahmer et al., 2003). What all these GRE algorithms have in common is that they rely on a given *domain of discourse* that constitutes the current *context*, also called *focus of attention*. The task of the GRE algorithm is then to single out the intended referent against its *potential distractors*. As long as the domains of discourse are small visual scenes or other closed-context scenarios, the intended referents are always in the current focus of attention.

We address the challenge of producing and understanding references to entities that are *outside* the current focus of attention, e.g., because they have not been mentioned yet and are beyond the currently observable scene. Following Appelt (1985) (see above), we make the assumption that *felicitous* references to entities outside the current focus of attention are possible because a) they are mutually known, or b) the hearer accepts and accommodates the presupposition[1] that a uniquely identifiable referent exists. In the first case, the referent is *discourse-new*, but nevertheless part of the interlocutors' *shared knowledge* (Prince (1981) calls this Givenness_k). In the latter case, the hearer will only later be able to resolve the reference to a physical entity. Consequently, the resolution process is deferred until the identity of the referent can be perceptually confirmed. In both cases, however, the hearer's attention must be directed towards an entity that is not immediately perceivable.

Paraboni et al. (2007) are among the few to address the issue of generating references to entities outside the immediate environment. They present an algorithm for *context determination* in hierarchically ordered domains, mainly targeted at producing textual references to entities in written documents (e.g., figures and tables in book chapters). As a result they do not touch upon the chal-

[1] A presupposition can be accommodated if it is not in conflict with the hearer's background knowledge.

lenges of physically and perceptually situated dialogue. Section 8.2 contains a more detailed discussion of the different approaches to context determination in spatial domains. Another shortcoming is that their approach is not easily combinable with any other existing GRE algorithm. We address this by proposing a separate algorithm for determining an appropriate context. This context can then be used to constrain the input to a GRE algorithm.

For completeness' sake, we present two of the widely used general-purpose GRE algorithms in the following sections.

The *incremental algorithm* (IA) of Dale and Reiter (1995)

The main routine of the IA, makeReferringExpression (reproduced in Algorithm 1), relies on three input parameters: the *intended referent* r, the *contrast set* C (defined as the *context set* without the intended referent), and a list of *preferred attributes* P.

Moreover, the IA needs a knowledge base that describes the *properties* of the domain entities through *attributes* and *values*. A special attribute is an entity's *type*. In order to determine appropriate discriminating properties, the algorithm requires a set of interface functions to the knowledge base to get additional information, namely the *taxonomical specialization* of a given attribute, the *basic-level category* of an entity's attribute, which draws from the notion of basic-level categories (see also Section 2.5.1), and a model of the *user's knowledge*.

The conceptual spatial map described in Part I represents the knowledge in its domain as statements about the properties of the individuals in the domain, and their relationships, as well as a taxonomy of concepts. As described in Section 4.2.5, the TBox also contains information about which concepts count as basic-level categories. It is thus straightforward to interface the conceptual spatial map with the IA.

After initialization, the IA iterates through the attribute list in the given order of preference. Within that loop, a number of subroutines, which are not reproduced here, are called. The findBestValue routine determines an appropriate value for the given attribute that has the highest discriminatory power – given that the hearer *knows* about it (checked against a user model by the userKnows routine). It is initially called with the basicLevelValue of the referent's attribute under consideration. Below we give an informal example of the algorithm that also discusses the ideas behind findBestValue and basicLevelValue. Once the best value, which holds for the intended referent and is false for at least one member of the contrast set, has been established for the respective attribute, the algorithm adds the attribute-value pair to the description generated so far. It then also shrinks the contrast set accordingly in order to reflect which

Algorithm 1 The basic incremental algorithm for GRE by Dale and Reiter (1995)
makeReferringExpression(r, C, P)

Input: intended referent r, contrast set C, preferred-attributes-list P
Output: description $DESC$ of a referring expression for r or *failure* if there exists none

Initialize: $DESC := \emptyset$
for each $A_i \in P$ **do**
  $V :=$ findBestValue$(r, A_i,$ basicLevelValue$(r, A_i))$
  if rulesOut$(\langle A_i, V\rangle) \neq nil$ **then**
    $DESC := DESC \cup \{\langle A_i, V\rangle\}$
    $C := C \setminus$ rulesOut$(\langle A_i, V\rangle)$
  end if
  if $C = \{\}$ **then**
    if $\langle type, X\rangle \in DESC$ for some X **then**
      return $DESC$
    else
      return $DESC \cup \{\langle type,$ basicLevelValue$(r, type)\rangle\}$
    end if
  end if
end for
return failure

potential distractors have hence been ruled out. The loop stops once the contrast set is empty, i.e., once the intended referent is the only entity in the context for which the conjunction of the constituents of the generated description is true. Finally, the algorithm makes sure the $type$ of the intended referent is included in the description, irrespective of its potential discriminatory power. The reasoning behind this is that usually an entity's type is realized as a noun – the most important constituent of (most) referring expressions. If the algorithm has successfully eliminated all original members from the contrast set, it terminates and returns the expression generated so far. If the contrast set is non-empty after iterating over all properties, the algorithm fails. For more details on the algorithm, we refer the reader to the work by Dale and Reiter (1995).

Example Imagine that the incremental GRE algorithm has to describe one specific ball among a number of other small objects, including other balls. Let's further assume that the preferred attributes are type and color. The algorithm then first checks which of the intended referent's types (here: beach ball, ball, object, and the top level DL concept $\top$) is most useful for identifying it to the hearer. The basic-level category is "ball", which already rules out any object that is not

a ball. However, descending the type taxonomy to "beach ball" also succeeds in setting the intended referent apart from the other balls that are not beach balls, for instance, basketballs and volleyballs. The algorithm then chooses "beach ball" as the so-called best value for the type attribute, includes it into the description, and shrinks the contrast set, which then only consists of all the beach balls in the domain.The next attribute it checks is color. Let's assume that the knowledge base represents an objects color as RGB values.[2] Without going into the details of research on color perception, let us further assume that the knowledge base knows the ranges in the RGB color space that correspond to the eleven basic color terms in English[3]. It then picks "green" as the basic level color of the intended referent, which rules out all but one other beach ball from the contrast set. Descending the color taxonomy then yields "dark olive green" as a color term that contrasts the intended referent to the other, "lawn green" colored beach ball. The contrast set is now empty, and the IA returns the semantic description of "the dark olive green beach ball" as a referring expression.

The salience-based version of the IA by Krahmer and Theune (2002)

Krahmer and Theune (2002) present a revision of the original IA, which especially aims at being sensitive to the discourse context. To this end, they make use of the notion of *discourse salience*. Krahmer and Theune (2002) discuss different approaches to determining the salience of a discourse referent – the basic assumption being that a recently mentioned entity is more *salient* than entities that have not been mentioned. They then reformulate the strict requirement of the original IA that the generated description discriminate the intended referent against all other entities in the context. The salience-based modified algorithm (reproduced in Algorithm 2 on the following page) only requires that the intended referent be the most salient entity described the generated expression. This allows the algorithm to generate short definite descriptions that act as *anaphoric* references to previously mentioned referents.[4] The overall behavior of the algorithm is similar to the original IA. It is important to note that, besides a model of discourse context, the algorithm makes the same assumption about the properties of the knowledge base and the existence of a given external con-

[2] This is a simplifying assumption. Systems operating in the real world have to recognize the color of the objects in their surroundings autonomously. van de Weijer et al. (2009) present an approach for learning the main color of objects in images retrieved from the web. Vrečko et al. (2009) present an approach for interactive learning of object colors in an integrated robotic system similar to the ones in Part II.

[3] Berlin and Kay (1969) identify black, white, red, green, yellow, blue, brown, purple, pink, orange, grey as the eleven basic color terms of the English language.

[4] Kelleher and van Genabith (2004) and Kelleher (2005) take visual salience into account for generating and resolving referring expressions. Their approach is also based on the original IA.

Algorithm 2 The salience-based version of the IA by Krahmer and Theune (2002)
makeReferringExpression(r, P, s)

Input: intended referent r, preferred-attributes-list P, salience-state s
Output: syntactic $tree$ of a referring expression for r or *failure* if there exists none

Initialize list of properties: $L := \emptyset$
Initialize syntactic tree: $tree := nil$
Initialize indication if current property is contrastive: $contrast :=$ **false**
for each $A_i \in P$ **do**
  $V :=$ findBestValue$(r, A_i,$ basicLevelValue$(r, A_i), s)$
  $contrast :=$ contrastive(r, A_i, V)
  $tree' :=$ updateTree$(tree, V, contrast)$
  if $(|\mathsf{val}(L \cup \{\langle A_i, V\rangle\})| < |\mathsf{val}(L)| \vee A_i = type) \wedge tree' \neq nil$ **then**
    $L := L \cup \{\langle A_i, V\rangle\}$
    $tree := tree'$
  end if
  if mostSalient$(r, L, s) =$ **true then**
    $tree :=$ addDefDet$(tree)$
    return $tree$
  end if
end for
return failure

text. For more details, interested readers are referred to the original work by Krahmer and Theune (2002).

An example for the utility of such an approach is the production task presented in the next chapter (cf. Section 9.3), where participants first produce a long expression, which is then picked up by a short anaphora, as in the following example:

(65) "take the ball from the table in the kitchen, then go to the study and put *the ball* into the box."

Even shorter anaphora are achieved by *pronominalization* like:

(66) "then you take the ball in the kitchen and you put *it* into the box on the table."

8.1.2 OpenCCG

We use *OpenCCG* (White, 2010), an open source implementation of Multi-Modal CCG as presented in (Baldridge and Kruijff, 2003). It is based on the

Combinatory Categorial Grammar (CCG) formalism (Steedman, 2000), which in turn is an extension of the traditional Categorial Grammar (CG) theory by Steedman (1999). OpenCCG provides a unified framework for parsing and realization. Given a string-based natural language utterance, it constructs a representation of its syntactic as well as semantic structure and the relation between syntactic units and their meaning.[5]

The basic grammatical unit of CCG is the *category*. Categories may be atomic (*primitive categories*) or complex (*functions*). Complex categories reflect the combinatoric potential of a word, i.e., the way in which it takes other sentence constituents as arguments. The position where a word expects specific constituents to combine with is expressed in complex categories using functors.

Definition 13 (CCG categories and functors (Steedman and Baldridge, 2007))**.**

> Categories (i.e., primitive categories as well as functions) can be combined using functors, called *slashes*: $\backslash$ and $/$. These functors determine the directionality of a function, and the order in which it takes its arguments.
>
> If X and Y are categories, then (X/Y) and $(X\backslash Y)$ are also categories. Outermost parentheses can be omitted.
>
> The leftmost category is always the resulting category after combining a constituent with its arguments. In this "result leftmost" notation, the rightward-combining functor is written X/Y, and, conversely, the leftward-combining functor is written $X\backslash Y$. This means that, here, X is always the range of the function, while Y is its domain. X and Y may be primitive categories or functions. ■

The approach of CCG is "fully lexicalized" (Steedman and Baldridge, 2007) as the way in which constituents combine to form more complex constituents is only driven by the categories of the lexical entries, which are atomic constituents. In the lexicon, a word is represented as a lexical entry that is assigned a category. In traditional CCG, each word has its own category. OpenCCG allows for the use of *lexical families*. These map a category to a set of lexical entries that share the same syntactic behavior.

[5] We use CCG because it was chosen as the grammar formalism for the natural language communication system used in the research projects this thesis contributed to (Kruijff et al., 2010). Therefore the choice for a specific grammar formalism was predetermined to CCG. In fact, any grammar formalism that comprises a semantic analysis and that is able to derive syntactic as well as semantic representations, and that supports generation of natural language surface forms from semantic representations could be used for the methods presented in this thesis.

An example for a simple category is N for nouns. Adjectives, in contrast, have a complex category N/N. They take an N as their argument and their resulting category has the same combinatory properties as any normal N. Finally, verbs belonging to the lexical family of transitive verbs take an NP to their right as their argument and yield again a complex category that takes another noun phrase to its left to produce a sentence: $(S\backslash NP)/NP$.

Functional categories can combine with their arguments according to a number of *combinatory rules*. The basic combinators are *forward application* and *backward application*. OpenCCG extends these by adding the associativity-inducing rules for *forward* and *backward type raising* and *forward and backward harmonic composition* and rules for *forward and backward crossed composition*, which induce permutation. The incremental application of rules until no more rules can be applied yields a grammatical *derivation* of an input string.

Definition 14 (Combinatory rules of CCG (Steedman and Baldridge, 2007))**.** In the following list, the symbols for the rules are given on the left. The left-hand side of a rule (before the $\Rightarrow$) shows the categories and their linear order that must be matched in the current derivation step. The right-hand side of the rule specifies the category that replaces the matched categories when a rule is applied.

Forward and backward application:

$$(>) \quad X/Y\,Y \Rightarrow X$$
$$(<) \quad Y\,X\backslash Y \Rightarrow X$$

Forward and backward type raising:

$$(>\mathbf{T}) \quad X \Rightarrow Y/(Y\backslash X)$$
$$(<\mathbf{T}) \quad X \Rightarrow Y\backslash(Y/X)$$

Forward and backward harmonic composition:

$$(>\mathbf{B}) \quad X/Y\;Y/Z \Rightarrow X/Z$$
$$(<\mathbf{B}) \quad Y\backslash Z\;X\backslash Y \Rightarrow X\backslash_{\diamond}Z$$

Forward and backward crossed composition:

$$(>\mathbf{B}_{\times}) \quad X/Y\;Y\backslash Z \Rightarrow X\backslash Z$$
$$(<\mathbf{B}_{\times}) \quad Y/Z\;X\backslash Y \Rightarrow X/Z$$

■

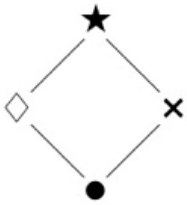


Figure 8.2: Hierarchy of CCG modal markers.

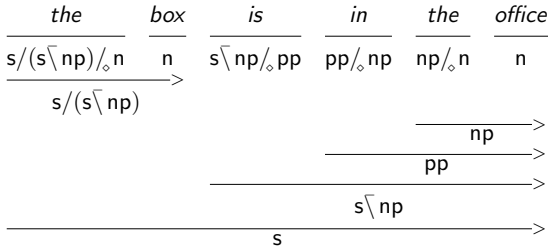
These additional rules, however, can lead to ungrammatical derivations. To overcome this problem, modal markers ($m \in \{\star, \Diamond, \times, \bullet\}$, cf. Figure 8.2) are introduced to control accessibility to combinators, as presented in (Baldridge and Kruijff, 2003). All connectives are marked with such a modal marker ($\backslash_m$ and $/_m$) and the combinators are then restricted to specific modally marked, *modalized*, connectives. The modal markers are ordered hierarchically. A combinator that is restricted by a modal marker may be applied to categories that contain only combinators of the same or weaker modality. The strongest modality is $\star$. Only forward and backward application ($>$ and $<$) are controlled by this modality. Type raising and harmonic composition ($>\mathbf{T}$, $<\mathbf{T}$ and $>\mathbf{B}$, $<\mathbf{B}$ resp.) are controlled by $\Diamond$, which is only accessible to $\bullet$, the weakest modality. $\bullet$ controls all combinators. Similarly, the modality $\times$ controls crossed composition ($>\mathbf{B}_\times$ and $<\mathbf{B}_\times$) and is also only accessible to $\bullet$. $\star$, the strongest modality, is accessible to all other modalities. Like this, lexical entries can be assigned modalized categories to have a detailed control over permitted derivations. By adjusting lexical categories, Multi-Modal CCG (and thus OpenCCG) can reflect language-specific properties in a universal, strictly lexicon-driven way. Table 8.1 shows the modalized combinatory rules from Definition 14. OpenCCG offers another functor type: $\backslash\!\!\!\backslash$ and $/\!\!\!/$. These slashes are *inhibited* combinators. They serve to construct complex categories that are not applied as functions. Example (67) shows a syntactic derivation[6] for a typical sentence in our domain.

Table 8.1: The modalized OpenCCG combinators.

Name	**Symbol**	**Rule**
Forward application	$>$	$X/_\star Y \; Y \Rightarrow X$
Backward application	$<$	$Y \; X\backslash_\star Y \Rightarrow X$
Forward type raising	$>\mathbf{T}$	$X \Rightarrow Y/_\diamond(Y\backslash_\diamond X)$
Backward type raising	$<\mathbf{T}$	$X \Rightarrow Y\backslash_\diamond(Y/_\diamond X)$
Forward harmonic composition	$>\mathbf{B}$	$X/_\diamond Y \; Y/_\diamond Z \Rightarrow X/_\diamond Z$
Backward harmonic composition	$<\mathbf{B}$	$Y\backslash_\diamond Z \; X\backslash_\diamond Y \Rightarrow X\backslash_\diamond Z$
Forward crossed composition	$>\mathbf{B}_\times$	$X/_\times Y \; Y\backslash_\times Z \Rightarrow X\backslash_\times Z$
Backward crossed composition	$<\mathbf{B}_\times$	$Y/_\times Z \; X\backslash_\times Y \Rightarrow X/_\times Z$

[6] This and all the other syntactic derivations in this work were constructed using the "moloko.v6" grammar by Trevor Benjamin and Geert-Jan M. Kruijff.

(67) Syntactic derivation for "the box is in the office."



8.1.3 Hybrid Logic Dependency Semantics

Besides the syntactic derivation, the OpenCCG parser constructs at the same time a *semantic representation* of the utterance in Hybrid Logic Dependency Semantics (HLDS) (Kruijff, 2001)). It substitutes the λ-calculus typically used in traditional Categorial Grammar. The approach of combining HLDS with CCG has been presented by Baldridge and Kruijff (2002). HLDS offers a dependency-based, compositional representation of different sorts of semantic meaning: *propositional content* and *intentional content*. HLDS also offers an extended modal logic framework preserving the advantages of standard modal logic, i.e., decidability and a convenient complexity (Areces, 2000).

The most prominent feature of hybrid logic is the introduction of *nominals* as an additional basic formula. Nominals allow for explicitly referring to states, a property that standard modal logic is lacking (Blackburn, 2000). Moreover, a new operator @, the satisfaction operator, is introduced. It can be used to form formulas in the same way as the common boolean operators. A formula $@_i p$ serves to express that a formula p holds at the state referred to by i. Furthermore, nominals can be typed with the ontological sorts of the states they refer to. In Example 68, the nominal $be1$ is typed as $ascription$. We represent the linguistically realized meaning of an utterance in an HLDS *logical form*. An LF is a conjunction of elementary predications (EPs), cf. Definition 15, anchored by the nominal that identifies the head's proposition, cf. Definition 16.

Definition 15 (HLDS elementary predications (Baldridge and Kruijff, 2002))**.**
$@_{idx:sort}$(**prop**): represents a proposition **prop** with ontological sort $sort$ and index idx,
$@_{idx1:sort1}\langle Rel\rangle(idx2 : sort2)$: represents a relation Rel from index $idx1$ to index $idx2$,
and $@_{idx:sort}\langle Feat\rangle$(**val**): represents a feature $Feat$ with value **val** at index idx. ■

Definition 16 (HLDS logical forms (Baldridge and Kruijff, 2002))**.**
Let EP be an elementary predication, then EP is a logical form.
Let $LF1$ and $LF2$ be logical forms, then $LF1 \wedge LF2$ is also a logical form.

In general, a logical form can take the following form
$@_{h:sort_h}(\mathbf{prop_h} \wedge \langle\delta_i\rangle(d_i : sort_{d_i} \wedge \mathbf{dep}_i) \wedge \langle Feat\rangle(\mathbf{val}))$,
where $\langle\delta_i\rangle$ represents a dependency relations between the head h and its dependents, identified by the nominals d_i. ■

The connection between the syntactic representation of an utterance and its corresponding semantics is established by application of a *linguistic linking theory*: i) in HLDS, nominals denote *discourse referents* of their heads, and ii) in a dependent part of a logical form, the nominal denotes the discourse referent of the respective syntactic dependent. In OpenCCG, the *logical form* of a word is defined at the categorial level. The logical form of a word expresses its own nominal and defines the semantic roles that its syntactic dependents have by coindexing the dependent nominals of the target category with the head nominals of the arguments. Subsequent derivation steps then apply unification to compositionally build a complex logical form of the head category that contains the full logical forms of its dependents. Example (68) shows the HLDS logical form of the sentence "the box is in the office" that is constructed from the syntactic derivation in Example (67).

(68) HLDS logical form of the utterance "the box is in the office."

$$
\begin{aligned}
@_{be1:ascription}(&\mathbf{be} \wedge \\
&\langle Mood\rangle\mathbf{ind} \wedge \\
&\langle Tense\rangle\mathbf{pres} \wedge \\
&\langle Copula-Restr\rangle(box1 : thing \ \wedge \mathbf{box} \wedge \\
&\quad \langle Delimitation\rangle\mathbf{unique} \wedge \\
&\quad \langle Num\rangle\mathbf{sg} \wedge \\
&\quad \langle Quantification\rangle\mathbf{specific}) \wedge \\
&\langle Copula-Scope\rangle(in1 : m-location \ \wedge \mathbf{in} \wedge \\
&\quad \langle Anchor\rangle(office1 : e-place \ \wedge \mathbf{office} \wedge \\
&\quad\quad \langle Delimitation\rangle\mathbf{unique} \wedge \\
&\quad\quad \langle Num\rangle\mathbf{sg} \wedge \\
&\quad\quad \langle Quantification\rangle\mathbf{specific})) \wedge \\
&\langle Subject\rangle(box1 : thing))
\end{aligned}
$$

8.1.4 Utterance planning and surface realization

Production covers the entire path from handling dialogue goals to speech synthesis. The dialogue system in which our approach is embedded can itself produce goals (e.g., to handle communicative phenomena like greetings), and it accepts goals from a higher level planner. Once there is a goal, an *utterance content planner*[7] produces a content representation for achieving that goal, which the realizer then turns into one or more surface forms to be synthesized.

A *dialogue goal* specifies a goal to be achieved, and any content that is associated with it. A typical example is to convey an answer to a user: the goal is to tell, the content is the answer. Content is given as a conceptual structure or *proto-LF* abstracting away from linguistic specifics, similar to the a-modal structures we produce for comprehension. Based on this, it incrementally transforms the proto-LF into one or more logical forms that express the propositional and intentional meaning in a contextually appropriate way. This includes generation of referring expressions, which is the topic of the work presented herein.

Content planning turns this proto-LF into a complete LF which matches the specific linguistic structures defined in the grammar we use to realize it. "Turning into" means extending the proto-LF with further semantic structure. This may be non-monotonic in that parts of the proto-LF may be rewritten, expanding into locally connected graph structures.

The OpenCCG realizer constructs a string-based, syntactically well-formed surface structure of the resulting HLDS logical form. The last step of the production process consists of providing the string-based output of OpenCCG to a text-to-speech module.

8.2 Context Determination in Hierarchically Ordered Domains

Imagine the situation in Figure 8.1 did not take place somewhere on campus, but rather inside building 3B. It would have made little or no sense for the robot to say that "the IT help desk is on the 1st floor in building 3B." To avoid confusion, an utterance like "the IT help desk is on the 1st floor" would have been appropriate. Likewise, if the IT help desk happened to be located on another site of the university, the robot would have had to identify its locations as being, e.g., "on the 1st floor in building 3B on the new campus". It is obvious that the physical and spatial situatedness of the dialogue participants plays an important role when determining which related parts of space come into consideration as potential distractors. The examples illustrate that the hierarchical

[7] For the work presented in this thesis, we make use of the utterance planner by Kruijff (2005).

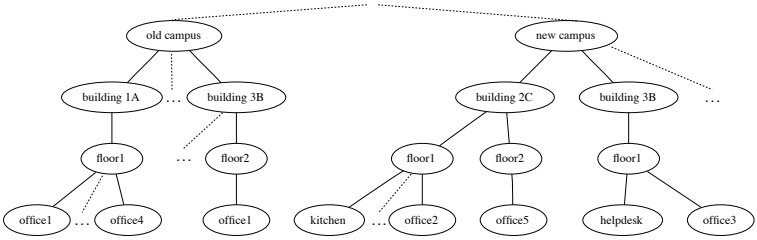


Figure 8.3: Example for a hierarchical representation of space.

representation of space that humans adopt (cf. Section 3.1.1) reflects upon the choice of an appropriate context when producing referential descriptions that involve attention-directing information.

Large-scale space can thus be viewed as a hierarchically ordered domain (see Figure 8.3 for an illustration). To keep track of the correct referential context in such a domain, we propose a general principle of situated *topological abstraction* (TA) for context extension. In the following section, we present the model in more detail and instantiate it with two algorithms that can be used for the GRE and RRE tasks, respectively.

This model is similar to Ancestral Search by Paraboni et al. (2007). However, their approach suffers from the shortcoming that their GRE algorithm treats spatial relationships as one-place attributes. For example a spatial containment relation that holds between a room entity and a building entity ("the library in the Cockroft building") is given as a property of the room entity (BUILDING NAME = COCKROFT), rather than a two-place relation (`in(library,Cockroft)`).Thereby they avoid recursive calls to the GRE algorithm. In principle, recursive calls to the algorithm are necessary if an intended referent is related to another entity that must be identified to the hearer through a definite description.

We believe that this imposes an unnecessary restriction onto the design of the knowledge base. Moreover, it makes it hard to separate the process of context determination from the actual GRE algorithm. In order to be compatible with the many existing GRE algorithms, and also to be useful for the RRE task, we propose an algorithm for situated context determination. It can be applied to the input knowledge bases of existing GRE approaches, and can determine the part of the knowledge base against which to perform the RRE task. We show how, in particular, the spatial knowledge representation for autonomous agents introduced in Part I can be used as knowledge base for generating and resolving referring expressions.

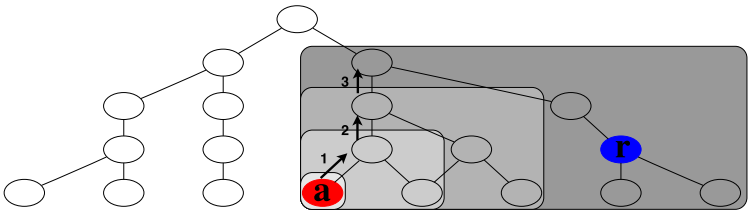


Figure 8.4: Illustration of the TA principle: starting from the attentional anchor (a), the smallest sub-hierarchy containing both a and the intended referent (r) is formed incrementally.

TA relies on two parameters. One involves the location of the *intended referent* 'r'. The other parameter is the *attentional anchor* 'a'. For single expressions the attentional anchor corresponds to the "position of the speaker and the hearer in the domain" Paraboni et al. (2007) – the so-called *utterance situation*, see Section 9.2 and (Poesio, 1993). For longer discourses about large-scale space, we will propose a model for attentional anchor-progression and evaluate it against real-world data in Chapter 9.

8.2.1 Context determination through topological abstraction

To establish a correct referential context for references to entities in large-scale space, we propose a general principle of *topological abstraction* (TA) for context extension, which is rooted in what we call the *attentional anchor* a.

TA is designed for a multiple spatial abstraction hierarchy. Such a spatial representation decomposes space into into parts that are related through a tree or lattice structure in which edges denote a containment relation (cf. Figure 8.3). The attentional anchor a corresponds to the current focus of attention, and it thus forms the nucleus of the context to be generated. In the basic case, a corresponds to the hearer's physical location. During a longer discourse, however, a can also move along the "spatial progression" of the most salient discourse entity. In Chapter 9 we explain this principle of *anchor-progression* in more detail.

If the intended referent is outside the current context, TA extends the context by incrementally ascending the spatial abstraction hierarchy until the intended referent is in the resulting sub-hierarchy (cf. Figure 8.4). Below we describe two instantiations of the TA principle, a TA algorithm for reference generation (TAA1, cf. Algorithm 3) and a TA algorithm for reference resolution (TAA2, cf. Algorithm 4). They differ only minimally, namely in their use of an intended referent r (in case of GRE), or a logical description $desc(x)$ of the referent (in case of RRE) to determine the conditions for entering and exiting the TA loop.

Context determination for GRE

TAA1 (cf. Algorithm 3) constructs a set of entities dominated by the attentional anchor a (including a itself). If this set contains the intended referent r, it is taken as the current utterance context set. Else TAA1 moves up one level of abstraction and adds the set of all descendant nodes to the context set. This loop continues until r is in the thus constructed set. At that point TAA1 stops and returns the constructed context set.

Algorithm 3 TAA1 (for reference generation).

Input: attentional anchor a, intended referent r
Output: the smallest sub-hierarchy containing a and r

Initialize context: $C := \emptyset$
$C := C \cup \{a\} \cup \mathsf{topologicalDescendants}(a)$
if $r \in C$ **then**
  return C
else
  Initialize abstraction queue: $Q := [a]$
  while $\mathsf{size}(Q) > 0$ **do**
    $n := \mathsf{pop}(Q)$
    for each $p \in \mathsf{topologicalParents}(n)$ **do**
      $\mathsf{push}(Q, p)$
      $C := C \cup \{p\} \cup \mathsf{topologicalDescendants}(p)$
    end for
    if $r \in C$ **then**
      return C
    end if
  end while
  return failure
end if

TAA1 is formulated to be neutral to the kind of GRE algorithm that it is used for. It can be used with the original IA by Dale and Reiter (1995), augmented by a recursive call if a relation to another entity is selected as a discriminatory feature. It could in principle also be used with the standard approach to GRE involving relations (Dale and Haddock, 1991), but we agree with Paraboni et al. (2007) that the mutually qualified references that it can produce[8] are not easily resolvable if they pertain to circumstances where a confirmatory search is

[8] An example for such a phenomenon is the expression "the ball on the table" in a context with several tables and several balls, but of which only one is on a table. Humans find such REs natural and easy to resolve in visual scenes.

costly (such as in large-scale space). More recent approaches to avoiding infinite loops when using relations in GRE make use of a graph-based knowledge representation (Krahmer et al., 2003; Croitoru and van Deemter, 2007). TAA1 is compatible with these approaches, as well as with the salience based approach of Krahmer and Theune (2002).

Context determination for reference resolution

Analogous to the GRE task, a dialogue system must be able to resolve verbal descriptions by its users to symbols in its knowledge base. In order to avoid over-generating possible referents, we propose TAA2 (cf. Algorithm 4) which tries to select an appropriate referent from a relevant subset of the full knowledge base.

It is initialized with a given semantic representation $desc(x)$ of the referential expression in a format compatible with the knowledge base. We show how

Algorithm 4 TAA2 (for reference resolution).

Input: attentional anchor a, referential description $desc(x)$

Output: set of possible referents in the smallest sub-hierarchy containing a and at least one referent satisfying $desc(x)$

Initialize context: $C := \emptyset$
Initialize possible referents: $R := \emptyset$
$C := C \cup \{a\} \cup \mathsf{topologicalDescendants}(a)$
$R := desc(x) \cap C$
if $R \neq \emptyset$ **then**
  return R
else
  Initialize abstraction queue: $Q := [a]$
  while $\mathsf{size}(Q) > 0$ **do**
    $n := \mathsf{pop}(Q)$
    for each $p \in \mathsf{topologicalParents}(n)$ **do**
      $\mathsf{push}(Q, p)$
      $C := C \cup \{p\} \cup \mathsf{topologicalDescendants}(p)$
    end for
    $R := desc(x) \cap C$
    if $R \neq \emptyset$ **then**
      return R
    end if
  end while
  return failure
end if

this is accomplished in our framework in Section 8.3.1. Then, an appropriate entity satisfying this description is searched for in the knowledge base. Similarly to TAA1, the description is first matched against the current *context set* C consisting of a and its child nodes. If this set does not contain any instances that match $desc(x)$, TAA2 increases the context set along the spatial abstraction axis until at least one possible referent can be identified within C.

8.3 Implementation

Our approach for resolving and generating spatial referring expressions has been integrated with the dialogue functionality in a cognitive system for the mobile robot presented in Chapter 5 (Kruijff et al., 2010). The robot is endowed with a *conceptual spatial map*, as presented in Chapter 3, which represents knowledge about places, objects and their relations in an OWL-DL ontology. In this specific implementation, the Jena reasoning framework[9] with its built-in OWL reasoning and rule inference facilities is used. Internally, Jena stores the facts of the *conceptual map* as RDF triples, which can be queried through SPARQL queries, cf. Section 4.2.4. Figure 8.5 shows a subset of such a knowledge base.

In the following, we use this example scenario to illustrate our approach to generating and resolving spatial referring expressions in the robot's dialogue system. We assume that the interaction takes place at the reception on the ground floor (i.e., FLOOR0), so that for TAA1 and TAA2 a=RECEPTION.

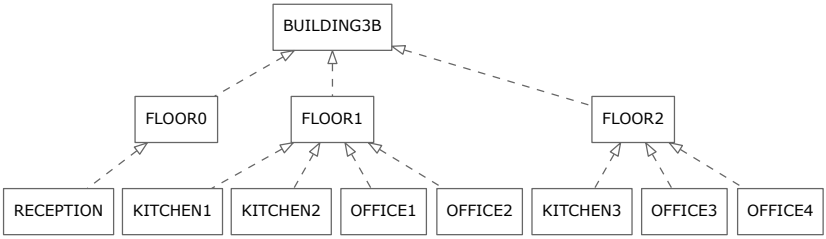
8.3.1 The comprehension side

In situated dialogue processing, the robot needs to build up an interpretation for an utterance which is linked both to the dialogue context and to the (referenced) situated context. Here, we focus on the meaning representations.

We represent meaning as a logical form (LF) in HLDS (cf. Section 8.1.3). The LF can be viewed as a directed acyclic graph (DAG), with labeled edges, and nodes representing propositions. Each proposition has an ontological sort, and a unique index. The representations are built compositionally, parsing the word lattices provided by speech recognition with a Combinatory Categorial Grammar (cf. Section 8.1.2), using the approach of Lison and Kruijff (2008). Reversely, we use the same grammar to realize strings (cf. Section 8.3.2) from these meaning representations (White and Baldridge, 2003).

An example is the meaning we obtain for "the big kitchen on the first floor" in Figure 8.6a. In the resulting logical form, elementary predications are folded under a single scope of @, as shown in Figure 8.6b. It illustrates how each propositional meaning gets an index, similar to situation theory. "kitchen" gets

[9] `http://jena.sourceforge.net` [last accessed on 2010-04-15]



(a) Topological abstraction hierarchy in an example ABox. Arrows represent the immediate topological containment relation.

```
Kitchen rdfs:subClassOf Room .      Office rdfs:subClassOf Room .
contains owl:inverseOf in .         in rdf:type owl:ObjectProperty .
kitchen1 rdf:type Kitchen .         reception rdf:type Reception .
office1 rdf:type Office .           floor1 rdf:type Floor .
kitchen2 size big .
bob rdf:type Person .               bob name "Bob" .
bob owns office1 .
floor1 contains kitchen1 .          floor2 contains office3 .
floor1 ordNum "1" .                 floor2 ordNum "2" .
```

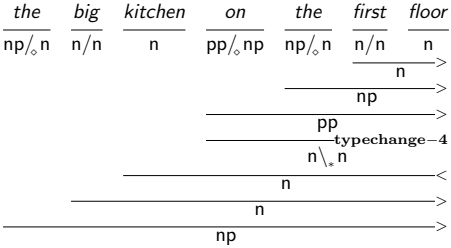
(b) Some RDF triples in the conceptual map (ABox, TBox, and RBox). XSD type definitions are left out for ease of reading.

Figure 8.5: Subset of a conceptual map for an office environment.

one, and also modifiers like “big,” “on” and “one.” This enables us to single out every aspect for possible contextual reference.

Next, we resolve contextual references, and determine the possible dialogue move(s) the utterance may express. Contextual reference resolution determines how we can relate the content in the utterance meaning, to the preceding dialogue context. If part of the meaning refers to previously mentioned content, we associate the identifiers of these content representations; else, we generate a new identifier. Consequently, each identifier is considered a dialogue referent. The details of this step and the next one below are outside the scope of this work – the interested reader is referred to (Kruijff et al., 2010) for more information.

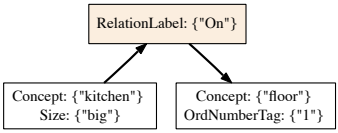
Once we have a representation of utterance meaning in dialogue context, we build a further level of representation to facilitate connecting dialogue content with models of the robot’s situation awareness. This next level of representation is essentially an a-modal abstraction over the linguistic aspects of meaning, to provide an a-modal conceptual structure (Jacobsson et al., 2008). Abstraction



(a) CCG derivation for "the big kitchen on the first floor."

$$@_{l1:e-place}(\textbf{kitchen} \wedge$$
$$\langle Delimitation\rangle \textbf{unique} \wedge$$
$$\langle Num\rangle \textbf{sg} \wedge \langle Quantification\rangle \textbf{specific} \wedge$$
$$\langle Modifier\rangle(b1 : q - size \wedge \textbf{big}) \wedge$$
$$\langle Modifier\rangle(o1 : m - location \wedge \textbf{on} \wedge$$
$$\langle Anchor\rangle(f1 : thing \wedge \textbf{floor} \wedge$$
$$\langle Delimitation\rangle \textbf{unique} \wedge$$
$$\langle Num\rangle \textbf{sg} \wedge \langle Quantification\rangle \textbf{specific} \wedge$$
$$\langle Modifier\rangle(n1 : number - ordinal \wedge \textbf{1}))))$$

(b) Logical form for "the big kitchen on the first floor."



(c) A-modal directed acyclic graph.

```
SELECT ?x0 ?x1 WHERE { ?x0 rdf:type Kitchen .
  ?x0 size big . ?x1 rdf:type Floor .
  ?x1 ordNum "1" . ?x0 on ?x1 . }
```

(d) Corresponding SPARQL query. In the knowledge base of Figure 8.5b, `?x0` resolves to `kitchen2`.

Figure 8.6: Syntactic derivation, logical form, a-modal DAG and corresponding SPARQL query for "the big kitchen on the first floor."

is a recursive translation of DAGs into DAGs, whereby the latter (conceptual) DAGs are typically flatter than the linguistic DAGs (Figure 8.6c), as illustrated in Figure 8.6c. The final step in resolving an RE is to construct a query to the robot's knowledge base. In our implementation we construct a SPARQL query from the a-modal DAG representations (see Figure 8.6d). This query corresponds to the logical description of the referent $desc(r)$ in TAA2. TAA2 then incrementally extends the context until at least one element of the result set of $desc(r)$ is contained within the context.

8.3.2 The production side

The utterance planning (cf. Kruijff (2005) for a description of the method employed here) is agenda-based, and uses a planning domain defined as a (systemic) grammar network alike the approach of Bateman (1997). A grammar network is a collection of systems that define possible sequences of operations to be performed on a node with characteristics matching the applicability conditions for the system. A system's decision tree determines which operations are to be applied. Decisions are typically context-sensitive, based on information about the shape of the (entire) LF, or on information in context models (dialogue or otherwise). While constructing an LF, the planner cycles over its nodes, and proposes new agenda items for nodes which have not yet been visited. An agenda item consists of the node, and a system which can be applied to that node. A system can explicitly trigger the generation of an RE for the node on which it operates. It then provides the dialogue system with a request for an RE, with a pointer to the node in the (provided) LF. The dialogue system resolves this request by submitting it to the GRE module. The GRE module produces an LF with the content for the RE. The planner then gets this LF and integrates it into the overall LF.

For example, say the robot in our previous example is to answer the question "where is Bob?" We receive a *dialogue goal* (cf. Section 8.1.4) to inform the user, specifying the goal as an *assertion* related to the previous dialogue context as an answer. The content is specified as an *ascription* e of a property to a target entity. The target entity is t which is specified as a person called "Bob" already available in the dialogue context, and thus familiar to the hearer. The property is specified as topological inclusion ($\langle TopIn \rangle$) within the entity p, the reference to which is to be produced by the GRE algorithm – hence the type "rfx" and the "$RefIndex$" which is the address of the entity (cf. Example 69). The content planner makes a series of decisions about the type and structure of the utterance to be produced. As it is an assertion of a property ascription, it decides to plan a sentence in indicative mood and present tense with "be" as the main verb. The reference to the target entity makes up the copula restriction

($\langle Copula - Restr \rangle$), and a reference to the ascribed property is in the copula scope ($\langle Copula - Scope \rangle$). This yields the expansion of the goal content in Example 70.

(69) Logical form specifying an assertion about Bob's location.

$$
\begin{aligned}
@_{d:dvp}(&c - goal \wedge \\
&\langle SpeechAct \rangle \textbf{assertion} \wedge \\
&\langle Relation \rangle \textbf{answer} \wedge \\
&\langle Content \rangle (e : ascription \wedge \\
&\quad \langle Target \rangle (t : person \wedge \textbf{Bob} \wedge \\
&\quad\quad \langle InfoStatus \rangle \textbf{familiar}) \wedge \\
&\quad \langle TopIn \rangle (p : rfx \wedge RefIndex)))
\end{aligned}
$$

(70) Expansion of the LF in Example 69. The assertion can be realized as a copula construction.

$$
\begin{aligned}
@_{e:ascription}(&\textbf{be} \wedge \\
&\langle Tense \rangle \textbf{pres} \wedge \\
&\langle Mood \rangle \textbf{ind} \wedge \\
&\langle Copula - Restr \rangle (t : person \wedge \textbf{Bob} \wedge \\
&\quad \langle InfoStatus \rangle \textbf{familiar}) \wedge \\
&\langle Subject \rangle (t : person) \wedge \\
&\langle Copula - Scope \rangle (prop : m - location \wedge \\
&\quad \textbf{in} \wedge \langle Anchor \rangle (p : rfx \wedge RefIndex)))
\end{aligned}
$$

The next step consists in calling the GRE algorithm to produce an RE for the entity p. In our NLP system we use a slightly modified implementation of the incremental algorithm (Dale and Reiter, 1995). The context set C is determined using TAA1. The preferred attributes list is specified as P=[type, name, topoIncluded, ordNum, ownedBy, size]. It is crucial that the spatial relation topoIncluded occur early on in order to ensure that the RE contains *navigational information* in case the intended referent is outside the immediate environment of the hearer. Let's assume that Bob is currently in KITCHEN3. In our example (a =RECEPTION) the GRE algorithm then produces the following logical form. The LF is then returned to the planner and inserted into the proto-LF created so far (cf. Example 71). The planner then makes further decisions about the realization, expanding this part of the LF to the result in Example 72.

(71) LF generated by the GRE algorithm – "the kitchen on the second floor."

$$@_{p:entity}(\textbf{kitchen} \wedge \langle Unique\rangle \textbf{true} \wedge \langle TopOn\rangle(f : entity \wedge \textbf{floor} \wedge \langle Unique\rangle \textbf{true} \wedge \langle OrdNum\rangle(n : number \wedge 2)))$$

(72) Full HLDS LF for the NP in Example 71.

$$@_{p:entity}(\textbf{kitchen} \wedge \langle Delimitation\rangle \textbf{unique} \wedge \langle Num\rangle \textbf{sg} \wedge \langle Quantification\rangle \textbf{specific} \wedge \langle Modifier\rangle(o1 : m - location \wedge \textbf{on} \wedge \langle Anchor\rangle(f : thing \wedge \textbf{floor} \wedge \langle Delimitation\rangle \textbf{unique} \wedge \langle Num\rangle \textbf{sg} \wedge \langle Quantification\rangle \textbf{specific} \wedge \langle Modifier\rangle(t1 : number - ordinal \wedge 2))))$$

Once the planner is finished, the resulting overall LF is provided to a CCG realizer (White and Baldridge, 2003), turning it into a surface form ("Bob is in the kitchen on the second floor"). This string is synthesized to speech using the MARY text-to-speech software (Schröder and Trouvain, 2003).

8.4 Summary and Outlook

In this chapter, we have presented an approach to generating and resolving referring expressions (GRE and RRE) for dialogues that are situated in large-scale space. It extends existing GRE and RRE algorithms with algorithms for spatial context determination. These algorithms are based on a general principle of topological abstraction (TA) for spatial domains. Besides an overview of related research on GRE we have introduced the relevant formalisms underlying the approach, i.e., Combinatory Categorial Grammar (CCG) and Hybrid Logics Dependency Semantics (HLDS), as well as a short introduction to the remainder of the natural language generation process. We have concluded with an implementation of this approach in a dialogue system for autonomous mobile robots. In the next chapter, we will extend this approach to the generation and resolution of multiple consecutive references to entities in large-scale space.

Chapter 9

Anchor-Progression in Situated Discourse about Large-Scale Space

Summary
In this chapter, we present an approach to producing and understanding referring expressions to entities in large-scale space during a discourse. The approach builds upon the principle of topological abstraction presented in Chapter 8. Here, we address the general problem of establishing reference from a discourse-oriented perspective. To this end, we propose anchor-progression and anchor-resetting mechanisms to track the origin of the TA algorithms throughout the discourse that model the way attention-directing information unfolds during the course of a discourse. We present an empirical production experiment that evaluates the utility of the proposed methods with respect to situated instruction-giving in small-scale space on the one hand, and large-scale space on the other. We conclude with a discussion of an implementation of the approach and give examples of its performance with respect to the domain of the production experiment.

I gratefully acknowledge the support of Christopher Koppermann (preparation and supervision of the experiment, annotation and evaluation of the data), Fai Greeve (data annotation), as well as the helpful discussions with Geert-Jan M. Kruijff and Ivana Kruijff-Korbayová, cf. (Zender et al., 2010).

9.1 Motivation and Background

GRE algorithms can nowadays be applied in a variety of real systems. This has led to a shift of focus towards systemic approaches to reference. This does not only involve other intra-linguistic processes, such as discourse planning, sentence aggregation, lexical choice, and surface realization. It also involves the *extra-linguistic* challenge of knowledge base construction and maintenance,

and the need to interface such a knowledge base with the natural-language processing system. Such systemic approaches also gave rise to the broader view that reference is established during discourse – as opposed to the limited view that a single, isolated RE must be all-encompassing in order to be successful. Establishing reference is thus not only a task to be solved by an isolated GRE algorithm. Reference is established during the course of a discourse. It is not sufficient to determine which information needs to be realized in an utterance, but also when and where. The challenge that we address here is how the focus of attention can move over the course of a discourse if the domain is larger than the currently visible scene. Let us illustrate this with an example. The two sentences (translated to English) are taken from the data that we gathered in our production experiment (see Section 9.3).

(73) "Go to the living room and take the ball. Then go to the bathroom and put the ball into the box. Then take the ball from the floor and put it in the study into the box on the table."

(74) "Go to the bathroom, take the ball, go to the study and put the ball into the box. Take the other ball, go to the living room, put the ball into the box on the table."

The sentences are meant as part of a set of instructions,which are given in a different place and before the recipient of orders is supposed to execute them. As can be seen, the noun phrase "the ball" occurs quite often – sometimes as exophoric reference, sometimes as anaphoric reference. The referent of the exophoric "the ball" differs in each utterance. What exactly it refers to is determined by the previous context: the first "ball" in Example 73 refers to "the ball in the living room," whereas its first occurrence in Example 74 refers to "the ball in the bathroom."[1]

In this chapter, we identify *attention-direction* and *context determination* as crucial steps towards the generation and resolution of references to entities in *large-scale space*. We employ the principle of *topological abstraction* presented in Chapter 8 for determining an appropriate spatial context for referring expressions, and discuss principles that determine the origin of the TA algorithms along the course of a discourse. We propose the mechanisms of *anchor-progression* and *anchor-resetting*, which model the way attention-directing information unfolds during the course of a discourse. We then present an empirical experiment that evaluates the utility of the proposed methods with respect

[1] As we have already shown in the previous chapter, the appropriateness of the circumscriptions "the ball in the X" is of course also dependent on the situations in which they are used.

to situated instruction-giving in small-scale space on the one hand, and large-scale space on the other. Finally, we provide an example of the behavior of the implemented methods with respect to the domain of the production experiment.

9.1.1 Existing corpora

There exists a great amount of empirical research on the production and resolution of referring expressions in visual scenes or shared small-scale work spaces. Empirical research on referential behavior in large-scale space has been overlooked for the most part.

We want to investigate the different forms referring expressions to entities in large-scale space can have during a discourse. There exist many corpora of referring expressions to entities in small-scale visual scenes, such as, the GRE3D3 corpus (Viethen and Dale, 2008b,a), and the Drawer data set (Viethen and Dale, 2006). These corpora provide insights in the different processes involved in the production of referring expressions. They do not, however, cover the specifities of references in large-scale space.

Other corpora address situated task-oriented natural language in large-scale spatial settings. The OSU Quake 2004 corpus (Byron and Fosler-Lussier, 2006) and the SCARE corpus (Stoia et al., 2008) are recordings of experiments performed using "first person graphics" 3D games. The drawback of these corpora, however, is that the process of establishing reference develops in a task-oriented situated dialogue while the participants are exploring their virtual 3D environment. This elicits phenomena of *interactive alignment* (Garrod and Pickering, 2004; Pickering and Garrod, 2006) and *conceptual pacts* (Brennan and Clark, 1996), which among other things, include the use of "risky references" (Carletta and Mellish, 1996). This can then be followed by interactive *repair* processes, and indefinite descriptions to introduce new referents to the shared context. It has been shown that two interlocutors who are faced with a situation that is new to them, will spend quite an amount of time and effort to collaboratively establish mutual reference. This involves the development of shorter, sometimes even *idiosyncratic* verbal descriptions over the course of such a dialogue (Clark and Wilkes-Gibbs, 1986). For several reasons these phenomena are very prominent in the aforementioned corpora. For one, the individual recorded conversations are rather short (15 minutes on average per session in the SCARE corpus, 9–35 minutes per session in the OSU corpus). And, secondly, the participants were embodied and situated in a virtual world that was new to them. All in all, this leads to an over-representation of verbal behaviors that serve the purpose of building up *common ground* (Stalnaker, 2002).

The GIVE challenge (Koller et al., 2007; Byron et al., 2009) follows a similar approach as the OSU and SCARE experiments. Participants embody an

avatar in a 3D environment. They have to navigate their large-scale environment following the orders of an NLG system that acts as instruction-giver. The overall task is that the participants have to complete a treasure-hunt task based on the system's instructions. Most referring expressions (that is, definite exophoric noun phrases) in such scenarios are generated *in-situ*, treating the local visual scene as a small-scale spatial context. Expressions like the ones we are interested in are conceivable in the GIVE challenge (e.g., "now press the yellow button in the dining room again."). However, at least with respect to the objective measures of the challenge (i.e., success rate and completion time of the treasure hunt), these might not be the most efficient ones. Embodied motion within the domain, visual salience, and especially short-term (spatial) memory effects determine which objects qualify as referents and distractors.

The aim of the TRAINS project (Allen et al., 1995) was to develop a spoken dialogue system that a user can interact with in order to negotiate train schedules. One of its achievements was the collection of a corpus of recordings and transcriptions of several hours of spoken human-human dialogues in the TRAINS domain. The domain is characterized by its orientation towards collaborative task planning. The data gathering was performed by having one human speaker play the role of the system, while the other participant acted as the system's user (Poesio, 1993). Their joint task was to develop plans for routing and scheduling freight trains, based on identical copies of a map that each of them had been given. Table 9.1 shows a fragment of user instructions from the recorded corpus.

29.1	okay,
29.2	great
29.3	while this is happening,
29.4	take engine E1 to Dansville,
29.5	pick up *the boxcar*,
29.6	and come back to Avon

Table 9.1: Transcript of user instructions, reproduced from (Poesio, 1993).

Many of these experiments trace back to the HCRC Map Task experiments (Anderson et al., 1991), which yielded a large corpus of instruction giver-instruction follower dialogues. The experimental setting was collaborative route replication using incomplete and differing maps of pseudo-large-scale space. The map was not meant as a depiction of a realistic large-scale domain, but rather the map was the domain itself, rendering the situation effectively to a small-scale space.

Taking into account the shortcomings of existing corpora with respect to the verbal phenomena we want to investigate, we conducted an empirical data gathering experiment. Our experiment, like many of the more recent experiments on establishing mutual reference, draws inspiration from the original Map Task experiments. The design of our experiment is aimed at controlling memory effects and common knowledge of the domain, and specifically at eliciting exophoric definite noun phrases.

9.2 A Model for Attentional Anchor-Progression

The approaches to topological abstraction for distractor-set formation presented in the previous chapter rely on two parameters. For one, obviously, the location of the *intended referent* plays a role. The second parameter is the *attentional anchor*. As discussed in Section 8.2, for single expressions it is determined by the utterance situation, i.e., the physical position where the utterance is made. For longer discourses about large-scale space the challenge of determining the attentional anchor remains.

Poesio (1993) observes that users interacting with the TRAINS-92 system make use of short non-anaphoric definite descriptions (e.g., "the boxcar", see the transcript in Table 9.1 on the preceding page) to felicitously refer to a specific one, even though the overall domain contains several boxcars (one located in Dansville, two in Bath, and one at Elmira).

The TRAINS domain is represented by a map, which is visually presented as a whole to the user, and which is assumed to be fully known to both the system and the user. Poesio (1993) interprets the referring expression "the boxcar" as a "*visible situation use* of a definite NP," which is defined in terms of *Situation Theory* (cf., e.g., (Devlin, 2006)). Essentially, a *situation* is a part of the world consisting of a "set of objects and facts about these objects" (Devlin, 1991), where, in turn, these facts comprise properties of the objects and the relationships that hold between them. Two basic distinctions can be made for situations. Temporally characterized situations are *events* and *episodes*. Conversely, there are situations that are characterized rather by their spatial properties, rather than temporally. Devlin (2006) identifies visual *scenes* as such a kind of situation.

When producing and, conversely, understanding an utterance, its interpretation in *situation semantics*depends on three situations: the *utterance situation* (defined as "the context in which the utterance is made and received"), the *resource situation*, which can become available in various ways, and the *focal situation* (Devlin, 2006). As factors that can make a situation available as resource situation, Devlin (2006) lists:

1. *being perceived by the speaker,*

2. *being the objects of some common knowledge about the world,*

3. *being the way the world is,*

4. *being built up by previous discourse.*

For the cases we are interested in, i.e., situated discourse about large-scale space, especially the second and fourth factor are relevant.

Although the previously mentioned NP "the boxcar" is, strictly speaking, underspecific with respect to the whole domain, the speaker can nevertheless make a felicitous reference. The NP must thus be interpretable with respect to a resource situation in which it is a unique description of its intended referent. Poesio (1993) hence argues for the need of a "*situation forming principle*, which states under which conditions a conversational participant will assume that a piece of information is part of that situation." More precisely, he claims that there must be "*principles for anchoring resource situations*" in the course of a discourse. An important determining factor of resource situations is the current *focus of attention*. The *mutual* focus of attention of the interlocutors can be felicitously used as resource situation (the so-called *situation of attention*, which Poesio (1993) explains in terms of shared visual attention). A second principle for determining the resource situation is via the current discourse topic. This can then lead to a *shift of attention* (Grosz, 1977) induced by the "movement" of the referents in the domain of discourse. For instance, the resource situation for "the boxcar" is the part of the map that was the terminal location of the instruction in the previous sentence (i.e., "Dansville," cf.(Poesio, 1993)). According to him, this movement then determines the updated current mutual focus of attention because the hearer can direct his visual attention accordingly.

Based on these observations, we claim that what is true for visual scenes also holds for situated discourse about entities large-scale space (cf. Section 3.1.1). Parallel to the focus shift in visual attention, we extend this notion to mental shifts of attention during a discourse about large-scale space. We show how such a principle can account for "movement" of the attentional anchor required for situated context determination in large-scale space presented in the previous chapter.

In order to account for the determination of the attentional anchor, we hence propose a model that we call *anchor-progression*. The model assumes that each reference to an extra-linguistic entity in large-scale space serves as *attentional anchor* for the subsequent reference. Formally speaking, each *exophoric* referring expression sets a new anchor. This excludes pronominal anaphora as well as other "short" descriptions that pick up an existing referent from the linguistic context, as, e.g., addressed in the salience-based GRE approach of Krahmer and

Theune (2002). The attentional anchor and the intended referent are then passed to the respective TA algorithm, i.e., TAA1 (see Algorithm 3 on page 163) for reference generation or TAA2 (see Algorithm 4 on page 164) for reference resolution. Taking into account the verbal behavior observed in the experiment, cf. Section 9.3, we also propose a refined model of *anchor-resetting*. In this model, for each new turn (e.g., a new instruction), the attentional anchor is reset to the position of the interlocutors. This model leads to the inclusion of navigational information for each first referring expression in a turn, and thus makes it easier for the hearer to follow.

9.3 Data Gathering Experiment

We are interested in the way the disambiguation strategies change when producing expressions during a discourse about large-scale space versus discourse about small-scale space. In our experiment, we hence gathered a corpus of spoken instructions in two different situations: *small-scale space* and *large-scale space*. We use the gathered data to evaluate the utility of the *anchor-progression/resetting* model. We specifically evaluate it against the traditional (*global*) model in which the intended referent must be singled out against all entities in the domain. Analyzing the data gathered in the small-scale space scenes with respect to the global model establishes the baseline for other experiments and well-studied GRE approaches for visual scenes.

9.3.1 Design considerations

As discussed earlier, small-scale space and large-scale space differ significantly. Large-scale space is a space that cannot be fully perceived from a single viewpoint – whereas small-scale space is defined by its immediate observability. This poses a fundamental problem when designing comparable stimuli for both conditions.

There is an inherent difficulty to conducting situated experiments in large-scale space. In a realistic physical environment with which the participants are familiar, the factors influencing the participants' behavior are hard, if not impossible, to control. That is why experiments are usually conducted in specificly instrumented, dedicated environments in order to be able to record the participants as unobtrusively as possible. Most participants are thus unfamiliar with the experimental environment. Memory effects as well as different spatial reasoning capabilities in the participants are likely to overshadow the observed verbal behavior. Embodiment in a virtual 3D world has a similar disadvantage, because the participants' mental map of the environment is typically very brittle.

A common practice for the study of human language processing is the use of drawings to depict small-scale scenes, e.g., using the *visual world paradigm* for correlating eye-movement and utterance processing (Cooper, 1974; Knoeferle et al., 2005). Production and resolution of referring expressions has also been extensively studied using drawings or other artificial renderings of small-scale scenes, such as the work done by Funakoshi et al. (2004), Kelleher (2007), or Viethen and Dale (2008b) to name a few.

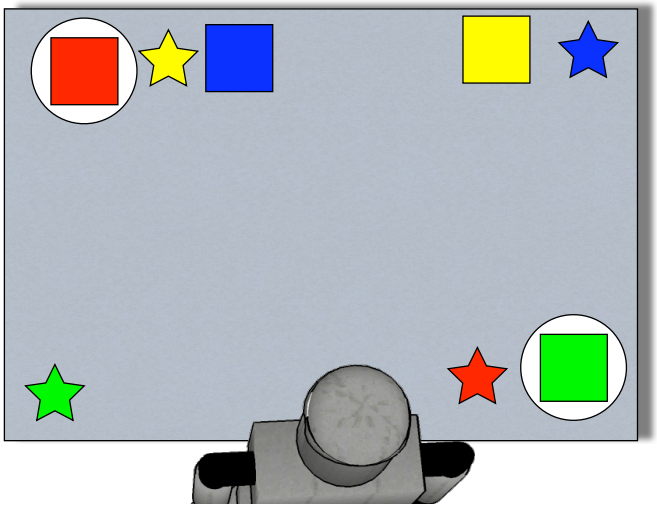
In order to study the differences in language use for small-scale and large-scale environments, we adopt the well-studied approach of using drawings of table-top scenes as the comparison standard. For the large-scale counterparts we draw inspiration from the Map Task experiments, as well as from more recent work by Hois and Kutz (2008). The large-scale scenes are depicted by a floor-plan like depiction of a domestic indoor environment. Hois and Kutz (2008) report on an experiment with a bird's-eye view of an office represented in a traditional floor-plan style, which succeeded in situating the participants' imagination in a room. In contrast to their study, however, we do not want to address the problems of spatial orientation for spatial calculi and their natural language realizations. We hence need to exclude perspectivization induced by spatial orientation of the objects as a factor for verbalization. In our experiment, we therefore depict the target objects in an upright fashion. This violates the strict bird's-eye perspective most people are used to from realistic floor plans. However, it has the advantage of emphasizing the hierarchical structure of the scene, rather than its exact interior design.

Strictly speaking, a fully observable map of an environment violates the definition of large-scale space. However, we claim that maps, being common abstractions of mental representations of large-scale space, can stimulate the participants' *imagination* of a scene such as to induce a realistic verbal behavior.

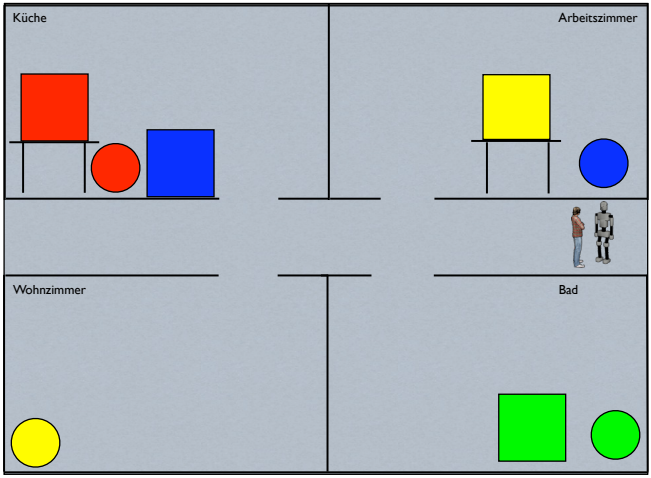
9.3.2 Stimulus design

The stimuli consist of a set of corresponding scenes depicting a table-top setting (*small-scale space*), and a domestic indoor setting (*large-scale space*), cf. Figure 9.1. For each scene in one setting, there is a scene in the other one that has the target, landmark, and distractor objects placed in a parallel fashion. As mentioned earlier, an observation from human-human dialogue production experiments is that the participants cooperate towards achieving as much *common ground* as possible about their task. In the TRAINS experiments, this included (Poesio, 1993):

1. *facts 'about the world', i.e., information obtained from the map,*



(a) Small-scale space scene: squares represent small boxes, stars represent cookies, white circles represent plates.



(b) Large-scale space scene: squares represent boxes that are either placed on the floor or on a table, circles represent balls, rooms are labeled *Küche* 'kitchen', *Arbeitszimmer* 'study', *Bad* 'bathroom', *Wohnzimmer* 'living room'.

Figure 9.1: Two of the scenes shown in the experiment.

2. *generic information about the task, such as expectations about the intentions of each conversational participant, and information about the 'rules of the game,' i.e., temporal-causal information and commonsense knowledge,*

3. *the discourse history,*

4. *the current status of the plan.*

In order to preclude this and the other aforementioned specific phenomena of collaborative, task-oriented dialogue, the participants had to instruct an *imaginary* recipient of orders. Therefore the stimuli provide a template for giving instructions to a service robot. The choice of a robot as instruction recipient was made to rule out potential social implications when imagining, e.g., talking to a child, a butler, or a friend. Moreover, it prevents the second and fourth of the above factors from playing a role: the participants cannot rely on the intelligence of the robot to figure out the 'rules of the game,' nor should they make use of meta-instructions that relate to the overall task status.

The small-scale setting shows a similar perspective on the scene as the experiment done by Funakoshi et al. (2004), i.e., a bird's-eye view of the table-top including an illustration of the robot's position with respect to the table. The target, landmark and distractor objects consist of cookies, small boxes, and plates. The way the objects are arranged allows to refer to their location with respect to the four corners of the table. The large-scale scenes depict an indoor environment consisting of a corridor and four rooms, parallel to the four corners in the small-scale scenes. The parallel target, landmark, and distractor objects are balls, boxes, and tables. The scenes show the robot and the participant in one end of the corridor.

9.3.3 Experiment design

In order to gather more comparable data we opted for a within-participants approach. Each person participated in the *small-scale space treatment* and in the *large-scale space treatment*. To counterbalance any potential carry-over effects, half of the participants were shown the two treatments in inverse order. The treatments consisted of eight different scenes. The sequence of the scenes was varied in a principled way in order to avoid parallel learning and habituation effects between the participants of each group.

In order to make the participants produce multi-utterance discourses, they were required to refer to all the four target object pairs. The pairs could be identified by their color. The exact wording of their instructions was up to the participants.

The cover story for the experiment was that we wanted to record spoken instructions in order to improve a speech recognition system for intelligent robots. The participants were asked to imagine an intelligent service robot capable of understanding natural language and familiar with its environment. The purpose of the service robot is to help humans in household tasks. The task that the robot was to perform was to clean up a working space, i.e., a table-top and an apartment, respectively. Cleaning up meant to place target objects (cookies or balls) in boxes of the same color. An influence of visual salience on the participants' performance can be ruled out for several reasons. First of all, in each scene the same set of four colors (yellow, blue, red, green) occurs. Second, the participants had to refer to all objects in each scene, and they were free to choose their order. Moreover, part of the experiment design was that the use of color terms to identify objects verbally was discouraged. This was achieved by telling the participants that the robot is unable to perceive and understand color terms. The fact that objects of the same type always had the same size also served to exclude visual salience as a factor.

9.3.4 Experiment procedure

Each participant was placed in front of a screen and a microphone. First they were shown the general instructions on the screen. Then they were presented the specific instructions for the first treatment, followed by three practice scenes that were showing stimuli of the same kind than the experimental scenes but with a lower complexity. After that the participants were given the opportunity to rest or ask clarifying questions before they were presented the eight scenes of the first treatment. After one more opportunity for a short pause, the instructions and practice scenes for the second treatment were shown, again allowing them to ask for clarification before starting with the experimental scenes.

During the practice runs and the experiments, the participants would utter their orders to the imaginary robots into the microphone, followed by a self-paced keyword that would allow the experimenter to know when to proceed to the next scene. Whenever participants asked clarifying questions the experimenter would repeat the appropriate part of the experiment's instructions to them. The experimenter was operating the computer that the screen was attached to and hit the forward button to advance to the next scene whenever the participants uttered the keyword.

Participants The experiment consisted of a pilot study with ten participants and the main experiment with 33 participants (19 female, 14 male students). The participants were paid for their efforts. Their median age was 22 (19–53 years). All of them were native speakers of German. One male participant had a color

vision deficiency. He reported that he was able to discriminate the target objects based on their shade, rather than hue. Due to his above average performance with respect to accuracy and reasonable completion time of the task it was not necessary to discard his session. The data of three other participants had to be discarded because they did not behave according to the instructions. The individual experiments took between 20 and 35 minutes.

9.3.5 Annotation

The recorded spoken instructions were first manually transcribed. Then the transcriptions were automatically transformed to a machine-readable XML-based mark-up format encoding the different parameters of the experiment (age and gender of the participant, order and type of treatments, order of scenes per treatment). These XML files were then imported into the UAM CorpusTool annotation software.[2] Occurrences of the linguistic phenomenon we are interested in, i.e., referring expressions, were then manually annotated. Samples of the annotations were cross-checked by a second annotator.

The annotation part consisted of several tasks. First of all, referring expressions were marked as 'refex' segments. Only definite noun phrases (NPs) qualify as 'refex' segments. If a turn contained an indefinite NP to introduce a new referent, the whole turn was discarded.[3] Only exophoric references were marked as 'refex'. This excludes pronouns and mentions of already introduced referents. The segmentation was done in a shallow manner, i.e., complex NPs were not decomposed into their constituents. The 'refex' segment thus spanned across the head noun and its determiner, and all other modifiers, such as adverbials, adjectives, dependent propositional phrases, and relative clauses.

The next step in the annotation process consisted of coding the 'refex' segments with respect to a set of features. These features encode the amount of semantic information that the segments contain, and under which disambiguation model – *global (G)*, *anchor-progression (A)*, or *anchor-resetting (R)*, the latter only for the large-scale treatment – this information can be used for singling out the described referent. We distinguish three types of semantic specificity with respect to each model according to the terms introduced by Engelhardt et al. (2006). A 'refex' is coded as an *over-description* with respect to a model $M \in \{G, A, M\}$ ($over_M$) if it contains redundant information according to the respective model M. Coding as an *under-description* ($under_M$) means that the 'refex' segment is ambiguous with respect to the model. *Minimal descriptions* with respect to the model (min_M) contain just enough information to uniquely

[2] `http://www.wagsoft.com/CorpusTool/` — Thanks to Mick O'Donnell for his support.

[3] This happened relatively seldom: only 18 turns out of the total 1,907 turns produced by the 30 evaluated participants contained an indefinite description and were thus discarded.

identify the referent. If the participants made an error with respect to the instructions of the experiment, the respective 'refex' was coded as error. Example 75 and Example 76 show annotated examples taken from the data.

(75) Example from the small-scale space scene in Figure 9.1a:

1. *nimm [das plätzchen unten links]*$_{min_{G,A}}$ *, leg es [in die schachtel unten*
   take the cookie bottom left , put it into the box bottom
   rechts auf dem teller]$_{over_{G,A}}$
   right on the plate
   'take the cookie on the bottom left, put it into the bottom right box on the plate'
2. *nimm [das plätzchen unten rechts]*$_{min_G,over_A}$ *, leg es [in die schachtel oben*
   take the cookie bottom right , put it into the box top
   links auf dem teller]$_{min_{G,A}}$
   left on the plate
   'take the cookie on the bottom right, put it into the top left box on the plate'
3. *nimm [das plätzchen oben links]*$_{min_G,over_A}$ *, leg es [in die schachtel oben*
   take the cookie top left , put it into-the box top right
   rechts]$_{min_{G,A}}$
   'take the top left cookie, put it into the top right box'
4. *nimm [das plätzchen oben rechts]*$_{min_G,over_A}$ *, leg es [in die schachtel oben*
   take the cookie top right , put it into the box top
   links]$_{under_{G,A}}$
   left
   'take the top right cookie, put it into the top left box'

(76) Example from the large-scale space scene in Figure 9.1b:

1. *geh [ins wohnzimmer]*$_{min_{G,A,R}}$ *und nimm [den ball]*$_{under_G,min_{A,R}}$ *und*
   go into-the living-room and take the ball and
   bring ihn [ins arbeitszimmer]$_{min_{G,A,R}}$ *, leg ihn [in die kiste auf dem*
   bring it into-the work-room , put it into the box on the
   tisch]$_{under_G,over_{A,R}}$
   table
   'go to the living room and take the ball and bring it to the study; put it into the box on the table'
2. *und nimm [den ball]*$_{under_{G,R},min_A}$ *und bring ihn [in die küche]*$_{min_{G,A,R}}$ *und*
   and take the ball and bring it into the kitchen and
   leg ihn [in die kiste auf dem boden]$_{under_G,min_{A,R}}$
   put it into the box on the floor
   'and take the ball and bring it to the kitchen and put it into the box on the floor'

3. *und dann nimmst du [den ball in der küche]*$_{min_{G,R},over_A}$ *und legst ihn [in die*
and then take you the ball in the kitchen and put it into the
kiste auf dem tisch]$_{under_G,min_{A,R}}$
box on the table

   'and then you take the ball in the kitchen and you put it into the box on the table'

4. *und dann gehst du [ins bad]*$_{min_{G,A,R}}$ *und nimmst [den ball der dort*
and then go you into-the bathroom and take the ball that there
liegt]$_{min_G,over_{A,R}}$ *und legst ihn [in die kiste die dort steht]*$_{min_G,over_{A,R}}$
lies and put it into the box that there stands

   'and then you go to the bathroom and you take the ball that lies there and you put it into the box that stands there'

9.4 Results

The collected corpus consists of 30 annotated sessions, each composed of two treatments (small-scale space and large-scale space). Each treatment comprises eight scenes with four sub-goals (termed *turns*) each. In total, the corpus contains 4,589 annotated referring expressions, out of which 83 are errors (i.e., confusion of target objects and other errors with respect to the task of the experiment). With the exception of the calculation of the error rate, we only consider non-error 'refex' segments as the universe. The small-scale treatment contains 1,902 'refex', with a mean number of 63.4 and a standard deviation of σ=1.98 per participant. This corresponds to the expected number of 64 referring expressions to be uttered: 8 scenes $\times$ 4 target object pairs (i.e., cookie and box). The large-scale treatment contains 2,604 'refex'. On average the participants produced 86.8 correct referring expressions (σ=18.19). As can be seen in Example 76, this difference results from the participants' referring to intermediate way-points that introduce new spatial contexts in addition to the target objects.

Overall, the participants had no difficulties completing the two treatments of the experiment. For both, the error rates are low: 1.78% on average (σ=3.36%) in the small-scale treatment, and 1.80% on average (σ=2.98%) for the large-scale treatment. A paired sample t-test of both scores for each participant shows that there is no significant difference between the error rates in the treatments (t=-0.019, df=29, p=0.985). This supports the claim that both treatments were of equal difficulty for the participants. In addition, a multivariate analysis of variance shows that there is no significant effect of treatment-order for the verbal behavior under study. This rules out potential carry-over effects. Figure 9.2 and Table 9.2 show the mean frequencies of over-descriptive, minimally descriptive, and under-descriptive referring expressions with respect to the models in both treatments.

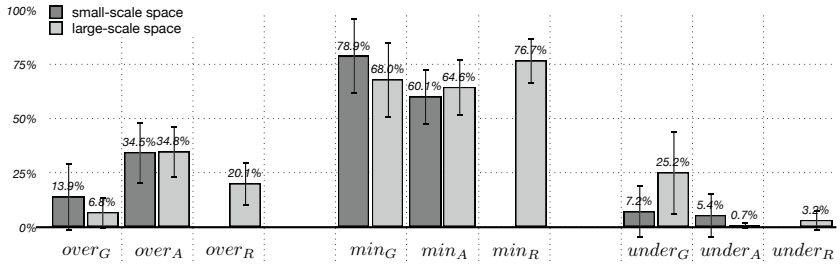


Figure 9.2: Bar chart visualization of Table 9.2.

Table 9.2: Mean frequencies (with standard deviation in italics) of minimal (min), over-descriptions ($over$), and under-descriptions ($under$) with respect to the models (A, R, G) in both treatments (LSS and SSS).

	$over_G$	$over_A$	$over_R$
small-scale space	13.94% *15.85%*	34.45% *14.37%*	
large-scale space	6.81% *7.53%*	34.75% *12.13%*	20.06 % *10.10%*

	min_G	min_A	min_R
small-scale space	78.90% *17.66%*	60.11% *13.13%*	
large-scale space	68.04% *17.87%*	64.55% *13.13%*	76.73% *10.66%*

	$under_G$	$under_A$	$under_R$
small-scale space	7.16% *12.07%*	5.43% *10.50%*	
large-scale space	25.16% *19.48%*	0.69% *1.72%*	3.21% *5.06%*

As can be seen, evaluating the participants' referring expressions in the small-scale space treatment with respect to the *global* model yields the expected results: about 13.9% of the referring expressions contain redundant information ($over_G$). This is comparable to the results of Viethen and Dale (2006) who report a rate of about 21% of over-descriptive referring expressions. In contrast to their experiment, however, the small-scale scenes in our experiment did not provide the possibility for producing more-than-minimal referring expressions for every target object. The large standard deviation of the frequency of $over_G$ referring expressions (σ=15.9%) illustrates that there is a huge variety in the participants' verbal behavior. $under_G$ referring expressions – hence unsuccessful and ambiguous references – occur with a frequency of 7.2% in the data of the small-scale space treatment. This is considerably less than the 16% reported by Viethen and Dale (2006). Moreover, among the participants of our experiment there is one outlier with a rate of 56% $under_G$ referring expressions. This is due to the participant's inconsistent use of the equivocal prepositional phrase *vor dir* 'in front of you', which we annotated as ambiguous. Excluding this outlier results in a mean frequency of 5.5% of $under_G$ 'refex'.

Although $under_A$ has a slightly lower mean frequency than $under_G$ for the small-scale scenes, this difference is not significant (t=2.018, df=29, p=0.053). The significantly (t=9.806, df=29, $p<0.001$) higher mean frequency of min_G (78.9%, σ=17.7%) than min_A (60.1%, σ=13.1%), however, shows that *global* is a much more accurate model for the verbal behavior in the small-scale space treatment. This observation is supported by the significantly (t=-13.745, df=29, $p<0.001$) lower mean frequency of $over_G$ (13.9%, σ=15.9%) than $over_A$ (34.5%, σ=14.4%).

For the large-scale space treatment, on the other hand, the *global* model does not fit the data well. A mean frequency of 25.2% $under_G$ referring expressions means that an RRE algorithm would fail to resolve the intended referent in approximately 1 out of 4 cases. The high standard deviation σ=19.5% and the high median of 29% illustrate that for some participants the model fits even worse.

With only 0.7% $under_A$ referring expressions (σ=1.7%) on average the *anchor-progression* assumption models the gathered data from the large-scale space treatment significantly better (t=6.776, df=29, $p<0.001$). Still, the model yields a high rate of $over_A$ referring expressions (mean frequency of 34.8%, σ=12.1%). In comparison, the *anchor-resetting* model yields a significantly (t=-10.348, df=29, $p<0.001$) lower amount of over-descriptions $over_R$ (20,1%, σ=10.1%). The mean frequency of under-descriptions $under_R$ (3.2%, σ=5.1%) is significantly (t=2.765, df=29, p=0.010) higher than for $under_A$, but still below what the *global* model generates in the small-scale space treatment. With

a mean frequency of 76.7% (σ=10.7%) minimal descriptions, *anchor-resetting* models the data better than both *global* and *anchor-progression*. For the referring expressions in large-scale space min_R is in the same range as min_G for the referring expressions in small-scale space.

9.5 Discussion

Overall, the two proposed models (*anchor-progression* and *anchor-resetting*) yield a high mean frequency of over-descriptions in the large-scale space data. This could be a side-effect of the experiment design. The participants might have been inclined to make more frequent use of redundant information because of the imagined intelligence level of the robot. At the same time, 21% of the referring expressions in the small corpus collected by Viethen and Dale (2006) – which did not involve speaking to an artificial system like a robot – are over-descriptions. This number is in the same range as the 20.1% of over-descriptions that the *anchor-resetting* model yields for large-scale space.

However, since this means that the human-produced referring expressions sometimes contain more information than minimally necessary, this does not negatively affect the performance of an RRE algorithm. For a GRE algorithm, however, a more cautious approach might be desirable. One measure for this can be the principle of *anchor-resetting*. In order to reassure the hearer of the current anchor, mentioning attention-directing information from the current physical location to the location of the anchor can be useful after a sequence of minimal descriptions. Whereas an algorithm has little difficulty in keeping track of long sequences of transitions between symbols in its knowledge base, the linguistic *performance* of humans deviates from their *competence* because of the nature of human memory and cognition (Chomsky, 1957).

We thus suggest that the *anchor-progression* model is suitable for the RRE task because it yields the least amount of unresolvable under-descriptions, whereas for the GRE task, the *anchor-resetting* model is more appropriate. It strikes a balance between producing short descriptions and supplementing additional, helpful information by providing navigational information at the beginning of each turn. This allows the hearer to follow the spatial progression with little effort. Note that the resolution and generation of anaphora and other expressions that pick up already introduced referents are outside the proposed models and must be handled separately.

Another factor that might increase the inclusion of redundant information when referring to entities outside the visual context is the inherent uncertainty involved in knowledge about large-scale space. Typically, considerable portions of such a dialogue serve the construction of a common agreement about

some particular state of affairs underlying the topic under discussion. While in dialogue people sometimes make "risky" utterances, in the specific setting of one-way instruction-giving potential under-descriptions cannot be tolerated because the robot cannot "collaborate" on the construction of reference. The inclusion of redundant information might thus answer the purpose of increasing the likelihood of identifying the correct referent in case the conversation partner has incomplete or divergent knowledge.

9.6 Implementation

The algorithms presented in the previous chapter, the knowledge representations and reasoning mechanisms from Part I, and the *anchor-progression* and *anchor-resetting* models from Section 9.2 have been implemeted in a natural language generation system. Here, we sketch the design of the implementation and illustrate what the algorithms generate with respect to the domain of the large-scale space treatment. The specificity of the generated referring expressions corresponds to min_A and min_R, respectively.

9.6.1 Knowledge base design

The knowledge base is represented by an OWL-DL ontology that is augmented with a set of custom rules reflecting the kind of reasoning a human would perform on the explicitly given information to infer additional knowledge.

The TBox consists of five major classes: Scene, Room, Corner, LandmarkObject, and TargetObject (see Figure 9.3). Scene is used as a top-level spatial concept that contains all the individuals of the respective scene. The five subclasses of Room correspond to the four labeled target rooms (Study, Kitchen, Livingroom, and Bathroom), and Corridor (i.e., the initial position of the robot and the person). The individuals in the domain correspond to the TargetObjects Ball and Box for the LargeScaleScenes. The LandmarkObjectss that can be used to further describe the target objects are instances of Table. Additionally, the Floor of a Room can be used to disambiguate the referents. Figure 9.3 shows the part of the TBox and ABox corresponding to the large-scale space scene no. 12 in Figure 9.1b. The figure only shows the direct asserted classes for the individuals. The ontology asserts a number of facts about the individuals in the scenes. Further knowledge is derived through OWL-DL based reasoning, and through the use of custom rules. These rules perform a kind of *closed-world reasoning* in that they rely on the absence of facts in order to introduce new facts. The rules are listed in Figure 9.6.

The `on_in_rule` takes care of the fact that any object that is `on` another object is also `in` the same room – this allows us to underspecify the asserted

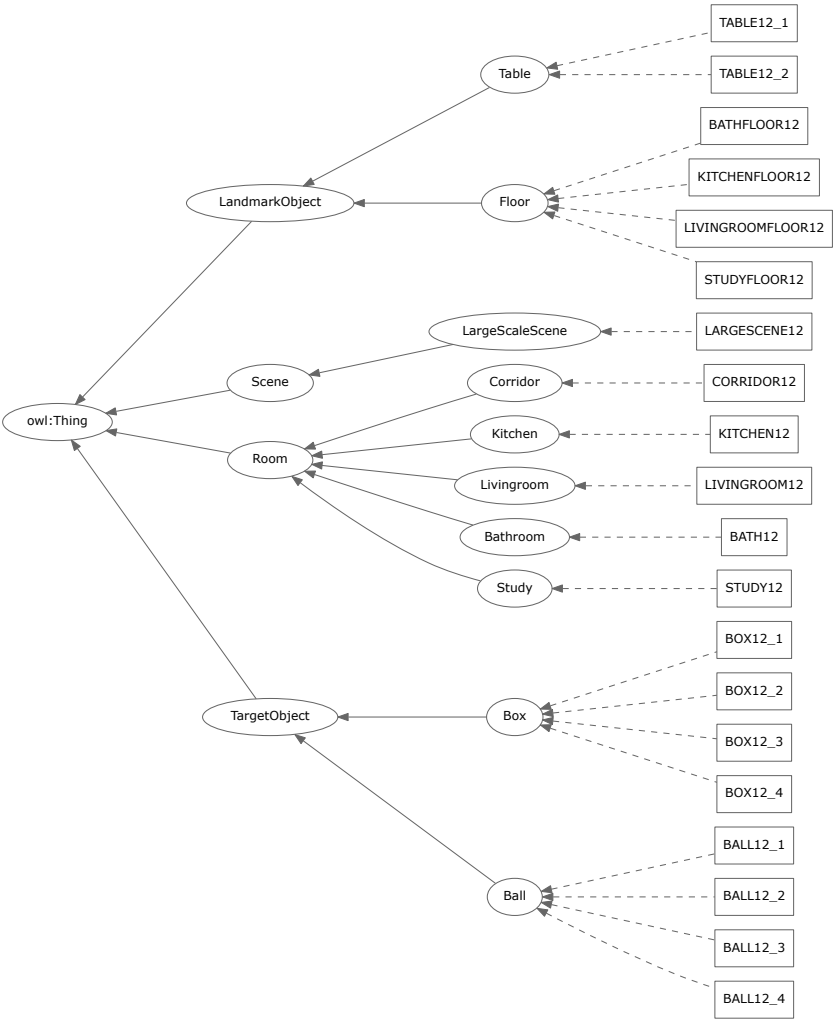


Figure 9.3: The TBox of the large-scale space treatment and an ABox instantiating one scene.

roles between the individuals in the domain. For example, the knowledge base contains the asserted facts that `box12_3` is `on` `table12_1`, and that `table12_1` is `in` `kitchen12`. The rule engine then infers the additional fact that `box12_3` is also `in` `kitchen12`. The other rule (`on_floor_rule`) encodes another piece of commonsense knowledge, namely that everything that is not on a table must be on the floor of respective room it is located in.[4]

[4]The rule is of course a simplification. But it serves the purpose of not having to assert for every object that it is placed on the floor. This information is neglected in most cases, but it is used in cases where, within one room, a box must be discriminated from another box that is on a table.

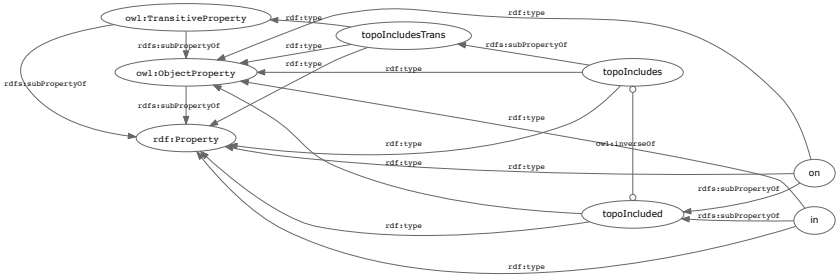


Figure 9.4: RBox of the ontology used in the implementation.

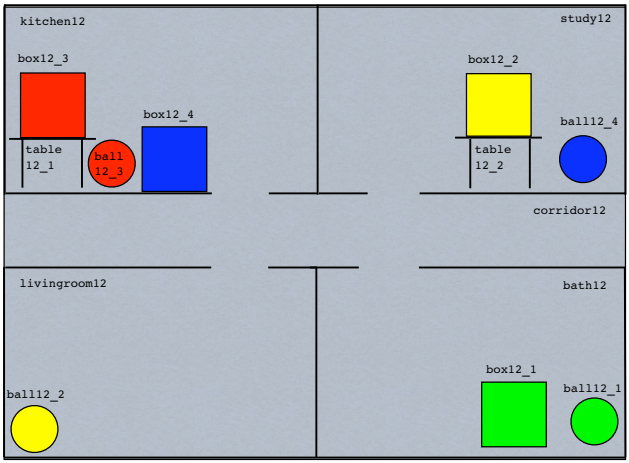


Figure 9.5: Depiction of the individuals in scene no. 12.

```
[on_in_rule:   [on_floor_rule:
   (?x on ?y),    noValue(?x rdf:type Floor),
   (?y in ?z)     (?x anchor:in ?y), (?z rdf:type:Floor),
   ->             (?z in ?y), noValue(?x on ?o)
   (?x in ?z)]    -> (?x on ?z)]
```

Figure 9.6: Custom Jena rules for the large-scale scene.

9.6.2 Generation of referring expressions

Discourse planning and surface realization are outside the scope of this thesis. We thus present only the LFs of the referring expressions that are generated by our implemented approach. The decision of which objects to refer to are based on the utterances produced by the participants of the experiment. In a full instruction-giving system, this decision could, for instance, be made by an automated planning module such as the one presented by Brenner and Nebel (2009) with an appropriately designed planning domain. The decision between anaphoric and non-anaphoric REs could then be made by a GRE algorithm that is aware of the discourse context, like Algorithm 2 in Section 8.1.1.

The GRE algorithm presented in the previous chapter is used to generate the LFs of the referring expressions, which can be realized using OpenCCG (see Section 8.1.2). Apart from the knowledge base (illustrated in Figures 9.3 and 9.5) and the functions to interface with it, the GRE algorithm requires an ordered list of preferred attributes. For the implementation here we chose P=[type, topoIncluded]. The modified algorithm for finding the best spatial relation first determines the best landmark object, and then chooses the right sub-role (i.e., in or on, cf. Figure 9.4). The contrast set is determined according to TAA1 (cf. Algorithm 3) on the basis of the attentional anchor and the intended referent. For the generation task, we have used both the *anchor-progression* as well as the *anchor-resetting* models. The attentional anchor progresses along the chosen model, whereas the intended referents are given in a fixed sequence. Initially the anchor corresponds to the *utterance situation*, i.e., `corridor12`.

The instructions consist of putting the green ball (`ball12_1`) into `box12_1` (the green box) then to go to the living room (`livingroom12`), and to put the yellow ball (`ball12_2`) into the yellow box (`box12_2`). The next instruction turn is to take `ball12_4` (the blue ball) go to the kitchen (`kitchen12`), and put it into the blue box (`box12_4`). Finally, the red ball (`ball12_3`) is supposed to be put into the red box (`box12_3`). The instructions are given in full sentences for the sake of presentation. The differences between the anchor-progression and the anchor-resetting models are highlighted in yellow.

(77) "take *the ball in the bathroom* (r=ball12_1, a=corridor12) and put it into *the box* (r=box12_1, a=ball12_1)"

$@_{e59:entity}(\textbf{ball} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopIn\rangle(e60 : entity \wedge \textbf{bathroom} \wedge \langle Unique\rangle\textbf{true}))$ (ball12_1)

$@_{e62:entity}(\textbf{box} \wedge \langle Unique\rangle\textbf{true})$ (box12_1)

---

(78) "go to *the livingroom* (r=livingroom12, a=box12_1), take *the ball* (r=ball12_2, a=livingroom12) and put it into *the box in the study* (r=box12_2, a=ball12_2)"

$@_{e64:entity}(\textbf{livingroom} \wedge \langle Unique\rangle\textbf{true})$ (livingroom12)

$@_{e66:entity}(\textbf{ball} \wedge \langle Unique\rangle\textbf{true})$ (ball12_2)

$@_{e68:entity}(\textbf{box} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopIn\rangle(e69 : entity \wedge \textbf{study} \wedge \langle Unique\rangle\textbf{true}))$ (box12_2)

---

(79) "take *the ball* (r=ball12_4, a=box12_2), go to *the kitchen* (r=kitchen12, a=ball12_4), and put it into *the box on the floor* (r=box12_4, a=kitchen12)"

$@_{e71:entity}(\textbf{ball} \wedge \langle Unique\rangle\textbf{true})$ (ball12_4)

$@_{e73:entity}(\textbf{kitchen} \wedge \langle Unique\rangle\textbf{true})$ (kitchen12)

$@_{e75:entity}(\textbf{box} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopOn\rangle(e76 : entity \wedge \textbf{floor} \wedge \langle Unique\rangle\textbf{true}))$ (box12_4)

---

(80) "take *the ball* (r=ball12_3, a=box12_4) and put it into *the box on the table* (r=box12_3, a=ball12_3)"

$@_{e78:entity}(\textbf{ball} \wedge \langle Unique\rangle\textbf{true})$ (ball12_3)

$@_{e80:entity}(\textbf{box} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopOn\rangle(e81 : entity \wedge \textbf{table} \wedge \langle Unique\rangle\textbf{true}))$ (box12_3)

(81) "take *the ball in the bathroom* (r=ball12_1, a=corridor12) and put it into *the box* (r=box12_1, a=ball12_1)"

$@_{e83:entity}(\textbf{ball} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopIn\rangle(e84 : entity \wedge \textbf{bathroom} \wedge \langle Unique\rangle\textbf{true}))$ (ball12_1)

$@_{e86:entity}(\textbf{box} \wedge \langle Unique\rangle\textbf{true})$ (box12_1)

---

(82) "go to *the livingroom*, (r=livingroom12, a=corridor12) take *the ball* (r=ball12_2, a=livingroom12) and put it into *the box in the study* (r=box12_2, a=ball12_2)"

$@_{e88:entity}(\textbf{livingroom} \wedge \langle Unique\rangle\textbf{true})$ (livingroom12)

$@_{e90:entity}(\textbf{ball} \wedge \langle Unique\rangle\textbf{true})$ (ball12_2)

$@_{e92:entity}(\textbf{box} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopIn\rangle(e93 : entity \wedge \textbf{study} \wedge \langle Unique\rangle\textbf{true}))$ (box12_2)

---

(83) "take *the ball in the study*, (r=ball12_4, a=corridor12) go to *the kitchen*, (r=kitchen12, a=ball12_4) and put it into *the box on the floor* (r=box12_4, a=kitchen12)"

$@_{e95:entity}(\textbf{ball} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopIn\rangle(e96 : entity \wedge \textbf{study} \wedge \langle Unique\rangle\textbf{true}))$ (ball12_4)

$@_{e98:entity}(\textbf{kitchen} \wedge \langle Unique\rangle\textbf{true})$ (kitchen12)

$@_{e100:entity}(\textbf{box} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopOn\rangle(e101 : entity \wedge \textbf{floor} \wedge \langle Unique\rangle\textbf{true}))$ (box12_4)

---

(84) "take *the ball in the kitchen* (r=ball12_3, a=box12_4) and put it into *the box on the table* (r=box12_3, a=ball12_3)"

$@_{e103:entity}(\textbf{ball} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopIn\rangle(e104 : entity \wedge \textbf{kitchen} \wedge \langle Unique\rangle\textbf{true}))$ (ball12_3)

$@_{e106:entity}(\textbf{box} \wedge \langle Unique\rangle\textbf{true} \wedge \langle TopOn\rangle(e107 : entity \wedge \textbf{table} \wedge \langle Unique\rangle\textbf{true}))$ (box12_3)

9.7 Conclusions

In this chapter, we have presented an approach to the challenges of generating and resolving referring expressions to entities in large-scale space. The issues we have addressed include the determination of an appropriate part of the domain as referential context, and the way exophoric references can shift the focus of attention in the course of a discourse. We make use of the principle of topological abstraction for context determination in the GRE and RRE tasks. The presented mechanisms of anchor-progression and anchor-resetting account for the motion of the focus of attention across multiple utterances. We have also reported on a production experiment for evaluating the proposed models. The evaluation shows that simple global context models fail for situated discourse about large-scale space. The gathered data support the claim that the anchor-progression and anchor-resetting models are a more accurate account of human verbal behavior in such discourses. We have also presented an implementation of the proposed models integrated with the GRE approach of Chapter 8. We have illustrated the output of the models for one large-scale space scene from the experiment.

Conclusions

Chapter 10

Summary and Outlook

Summary
In this chapter, we recapitulate the work presented in this thesis. We describe an ongoing effort to transfer the proposed robotics-oriented models to autonomous virtual agents that act in an online virtual 3D world. We conclude with a discussion of open issues.

10.1 Recapitulation

The work presented in this thesis addresses the fundamental questions how machines, such as robots and other autonomous agents, can acquire a mental representation of their environment that allows them to (a) act and navigate in it, and (b) communicate about it with humans in natural language. The kinds of environments under discussion are structured, human-oriented, large-scale spatial environments – i.e., environments that cannot be apprehended as a perceptual whole, such as indoor domestic environments, or building ensembles.

To this end, we have proposed a multi-layered conceptual spatial map that ranges from low-level specialized sensor-based maps for robotic applications over topological and categorical abstraction steps to a conceptual model that divides space into rooms whose concepts are based on salient objects. The conceptual map layer is implemented as an ontology-based knowledge base, which allows for different kinds of reasoning, including Description Logic-based inference, nonmonotonic maintenance of symbolic representations of spatial units, and prototypical default reasoning.

The proposed models have been implemented in several integrated autonomous mobile cognitive robotic systems developed within the EU-funded research projects "CoSy" and "CogX." The systems highlight how a mobile robot can build up spatial representations, both autonomously (i.e., in a curiosity-driven way) and semi-autonomously (i.e., through a process called human-augmented mapping), and how the conceptual spatial knowledge can be used for goal-directed action planning and execution.

These models are then used as knowledge bases for situated natural language dialogue about entities in large-scale spatial environments. By that, the presented work goes beyond situated natural language interaction about an agent's immediate surroundings (i.e., *small-scale space*), such as table-tops or single room spaces. The approach allows an agent to successfully generate and resolve natural language expressions that refer to entities in large-scale space. The approach is backed by observations from an empirical spoken language production experiment. The thesis concludes with a discussion of ongoing work to transfer the models made for intelligent mobile robots to autonomous virtual agents that act in an online virtual 3D world.

In **Chapter 2**, we presented the scientific background that the work in this thesis builds upon. After a short overview of research on cognitive systems, we presented an introduction to autonomous agents, including some background in robotics, in particular autonomous and intelligent mobile robots, and virtual worlds. We then discussed relevant aspects of the study of embodied cognition, human categorization and conceptualization. An introduction to ontology-based knowledge representations concluded the chapter.

In **Chapter 3**, we identified structuring of space and categorization of large-scale space as two important aspects of spatial understanding. In order to enable an autonomous agent to engage in a situated dialogue about its environment, it needs to have a human-compatible spatial understanding, whereas autonomous behavior, such as navigation, requires the agent to have access to low-level spatial representations. Addressing these two challenges, we presented an approach to *multi-layered conceptual spatial mapping*. The description of our approach is embedded in a discussion of relevant research in human spatial cognition and mobile robot mapping.

In **Chapter 4**, we focused on the *conceptual map layer* of the multi-layered spatial. We showed how Description Logics can be used to perform inference on a human-compatible symbolic conceptualization of space. We further proposed methods for prototypical default reasoning and belief revision to extend the capabilities of autonomous agents.

In **Chapter 5**, we introduced the EXPLORER robot system. The EXPLORER implements the approach to multi-layered conceptual spatial mapping in an integrated robotic system. The mobile robot base is equipped with different sensors for map building, place and object recognition, and user interaction. We illustrated how the multi-layered map can be acquired interactively in a so-called guided tour scenario. We furthermore presented a method for human- and situation-aware people following that makes use of the higher-level information of the multi-layered conceptual spatial map, thus increasing the perceived level of intelligence of the robot.

In **Chapter 6**, we presented an extension of the EXPLORER system. The presented implementation makes use of PECAS, a cognitive architecture for intelligent systems, which combines fusion of information from a distributed, heterogeneous architecture, with an approach to continual planning as architectural control mechanism. We showed how the PECAS-based EXPLORER system implements the multi-layered conceptual spatial model. Moreover, we showed how – in the absence of factual knowledge – prototypical default knowledge derived from a Description Logic-based ontology can be used for goal-directed planning for situated action in large-scale space.

In **Chapter 7**, we presented an approach in which a conceptual map is acquired or extended autonomously, through a closely-coupled integration of bottom-up mapping, reasoning, and active observation of the environment. The approach extends the conceptual spatial mapping approach, and allows for a nonmonotonic formation of the conceptual map, as well as two-way connections between perception, mapping and inference. The approach has been implemented in the integrated mobile robot system DORA. It uses rule- and DL-based reasoning and nonmonotonic inference over an OWL ontology of commonsense spatial knowledge, together with active visual search and information gain-driven exploration. It has been tested in several experiments that illustrate how a mobile robotic agent can autonomously build its multi-layered conceptual spatial representation, and how the conceptual spatial knowledge can influence its autonomous goal-driven behavior.

In **Chapter 8**, we presented an approach to the task of *generating and resolving referring expressions to entities in large-scale space*. It is based on the spatial knowledge base presented in the previous chapters. Existing algorithms for the generation of referring expressions try to find a description that uniquely identifies the referent with respect to other entities that are in the current context. The kinds of autonomous agents we are considering, however, act in large-scale space. One challenge when referring to elsewhere is thus to include enough information so that the interlocutors can extend their context appropriately. To this end, we present the principle of *topological abstraction* (TA) as a method for context construction that can be used for both generating and resolving referring expressions – two previously disjoint aspects. We showed how our approach can be embedded in a bi-directional framework for natural language processing for conversational robots.

In **Chapter 9**, we presented an approach to producing and understanding referring expressions to entities in large-scale space during a discourse. The approach builds upon the principle of topological abstraction (TA) presented in Chapter 8. Here, we addressed the general problem of establishing reference from a discourse-oriented perspective. To this end, we proposed *anchor-*

progression and *anchor-resetting* mechanisms to track the origin of the TA algorithms throughout the discourse that model the way attention-directing information unfolds during the course of a discourse. We presented an empirical production experiment that evaluates the utility of the proposed methods with respect to situated instruction-giving in small-scale space on the one hand, and large-scale space on the other. We concluded with a discussion of an implementation of the approach and gave examples of its performance with respect to the domain of the production experiment.

10.2 Ongoing Work: Transfer of the Spatial Model to a Virtual Agent

As sketched earlier (cf. Section 2.3 and Chapter 3), virtual online worlds can also be viewed as *human-oriented environments*. Autonomous virtual agents acting in such a 3D world and autonomous mobile robots operating in the real world thus encounter similar challenges. They both have an *embodied* shared presence (cf. Section 2.1) with the humans they interact with. *Situated* interactions with humans in their environment require them to be aware of the spatial situation they are part of.

We are currently investigating a transfer of the spatial model and natural language processing methods presented in this work to a virtual NPC agent for the 3D virtual world *Twinity*.[1] According to the multi-layered conceptual spatial mapping approach, the lower layers of the spatial model must be instantiated with interfaces to and abstractions over the world representation of the Twinity engine, whereas the conceptual layer must only be adapted to reflect the concepts that are present in the domain. Figure 10.1 shows the basic system architecture used for interfacing our virtual agent control with the Twinity world server. The implementational details of this infrastructure is described in (Klüwer et al., 2010).

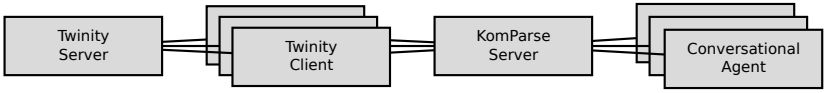


Figure 10.1: System architecture for virtual agents in the Twinity world.

In Twinity, the world consists currently of three cities that are based on real cities (i.e., Berlin, Singapore, and London).[2] These cities contain a street network based on the actual city map. Buildings contain apartments and other indoor spaces (termed "LocalSpace"). Our spatial model will focus on LocalSpaces. Being human-oriented environments, the LocalSpaces are not crisply divided into individual rooms or other areas that would be accessible through the Twinity interface. At the present time, the Twinity server does not provide much environment information. It only provides information about which items (i.e., objects, pieces of furniture, decoration etc.) are present in the LocalSpace and

[1] This section describes a currently ongoing integration of our approach with the KomParse system developed by Peter Adolphs, Tina Klüwer, Feiyu Xu, and Xiwen Cheng, with the help of Torsten Huber and Weijia Shao. See also `http://komparse.dfki.de/` [last accessed 2010-05-20]

[2] `http://www.twinity.com/` [last accessed on 2010-05-05]

their x-y-coordinates. Since the positions of walls are currently not available through the interface, we determined the outlines of the walls in a few apartments by hand. This then yields a basic spatial segmentation of LocalSpaces into rooms. In order to automatically and autonomously acquire room models, we are investigating the utility of the home tour scenario for a human-augmented mapping (cf. Section 5.3.5) of novel environments. Figure 10.2 shows a user-furnished apartment for which we have manually created a metric map.

Using a topological computation on the basis of the boundaries of the rooms, we can then compute in which rooms contain which items in the LocalSpace. Together with the information which rooms (and possibly which floors, in case of multistory LocalSpaces) are in a given apartment, we can then derive a basic containment hierarchy (cf. Section 3.1.1) for a LocalSpace and its constituents (illustrated in Figure 10.3).

Figure 10.2: A modern apartment in *Twinity*.

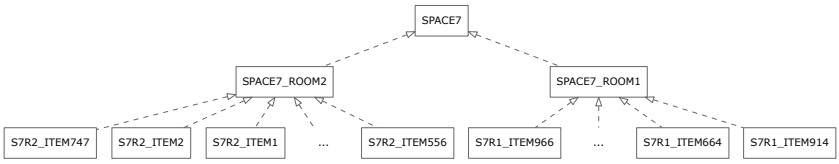


Figure 10.3: Containment hierarchy of the individuals in ABox $\mathcal{A}_{Twinity}$.

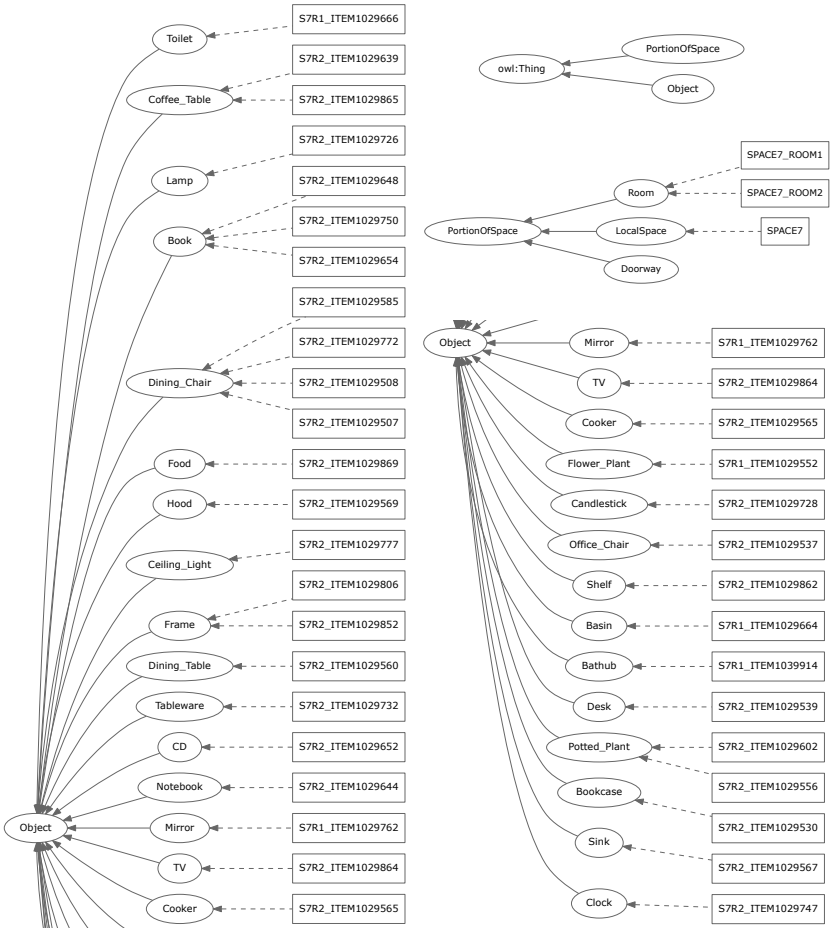
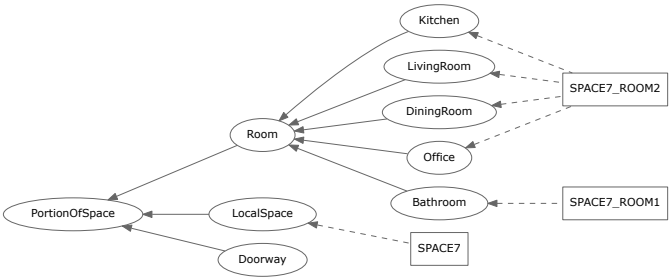
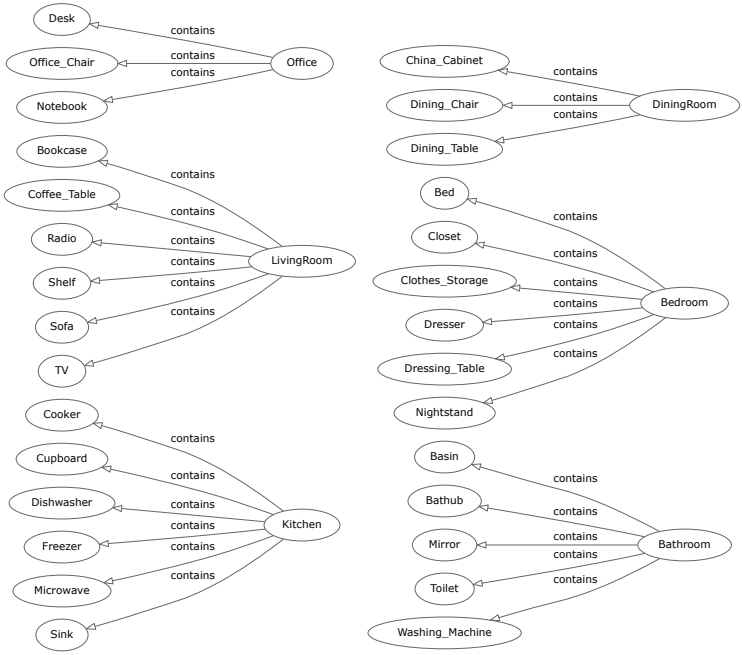


Figure 10.4: Part of the TBox $\mathcal{T}_{Twinity}$ instantiated in $\mathcal{A}_{Twinity}$ (see Figure 10.3).



(a) Inferred concepts for individuals in $\mathcal{A}_{Twinity}$ using the definitions from (b).



(b) Room concepts defined by contained objects.

Figure 10.5: Object-based room concept inference for $\mathcal{T}_{Twinity}$.

One advantage of the Twinity world over realistic robotic applications is that object detection does not have to rely on computer vision. Recognition of object categories as well as detection of arbitrary objects are hard problems for computer vision that have not yet been solved. At least for the items that are provided by the Twinity engine (e.g., obtainable through in-game stores) the categories are known – for items that are based on user-built custom 3D models, this is not necessarily the case. For the 'official' Twinity items, there exists a rich ontology, as described in (Klüwer et al., 2010). This makes it easy to integrate the Twinity object ontology with the indoor common sense ontologies presented in this work. Figure 10.4 shows those concepts from the TBox $\mathcal{T}_{Twinity}$ that are instantiated by the individuals from the example environment (see Figure 10.2).

The high amount of objects in the apartment, however, highlights a shortcoming of our approach. Due to different factors, e.g., presumably, the way in which players control their avatars, and the perspective that players have of the environment, leads players to have apartments that contain large, open spaces. The requirements of an apartment in a virtual world are very different from apartments in the real world. Rather than providing privacy and facilities for daily chores, virtual apartments serve mainly representative functions. Giving artistic form to a part of a virtual world is probably the predominating motivation behind players' decorating and furnishing such apartments. Instead of individual rooms dedicated to specific tasks, these large rooms are then conceptually divided into smaller areas where objects that support a common purpose are grouped. Figure 10.2 shows the greater part of the apartment of $\mathcal{A}_{Twinity}$. In such a case, the objects-in-room based inference from Chapter 4 (see also the simplified definitions shown in Figure 10.5b) yields a too diverse conceptualization and a too coarse spatial segmentation, as shown in Figure 10.5a.

For such environments, the notion of room as functional unit is too coarse. We are currently investigating the possibility of using a clustering approach like the one by Viswanathan et al. (2009) for determining spatial segments based on object clusters. In the example apartment (see Figure 10.2), the concept "kitchen" applies to a part of the overall room only – the part in which the kitchen range, the freezer and the sink are located – whereas the portion of space in front of the desk with the notebook could be referred to as "office."

10.3 Open Issues

There are several aspects of the work presented in this thesis that offer opportunities for improvements and extensions in future research. We want to conclude this thesis with identifying three such aspects, and presenting ideas and suggestions for how to address them.

10.3.1 Ontology design and commonsense knowledge

Typically, ontologies are designed and manually created by experts, often domain experts or experts in formal knowledge representation and formalization. This engineering-oriented approach towards ontology building has the disadvantage that the thus created ontologies tend to reflect their creators' view of the world. Instead of real *common sense*, it reflects the common sense of one person or a small group of people.

One of the open challenges with respect to the conceptual spatial knowledge base presented in this work is thus to automatically acquire an ontology that reflects common sense in a broader view. Such an approach should be based on a large amount of gathered data, such as from observing and analyzing people's homes and other indoor environments. However, given the current limitations of computer vision (as mentioned in the previous section), and considering safety, security, robustness, and privacy issues, it is currently not feasible to deploy autonomous mobile robot's in people's homes in order to acquire more realistic, commonsense models.

Intelligent robotics, though, is not the only discipline that has a demand for large amounts of realistic training data. Currently, a number of efforts are in progress that try to gather large amounts of labeled and segmented images for training computer vision algorithms on. We suggest exploiting these data bases for establishing correlations between object occurrences and specific room concepts that could help make our approach more generally applicable. One such effort is the *LabelMe*[3] project by MIT CSAIL (Russell et al., 2008). LabelMe contains, as of today, 183,407 digital images (out of which 58,926 are annotated with the objects they contain) of indoor and outdoor scenes. (Viswanathan et al., 2010) present an approach to visual place classification based on detected object occurrences that makes use of the LabelMe statistics.

Another effort towards collecting commonsense knowledge from many people is the *Open Mind Indoor Common Sense* (OMICS) project[4] by Honda Research Institute USA Inc.. The approach of the OMICS project is to ask people on the web to fill in the blanks in different sentences about everyday

[3] `http://labelme.csail.mit.edu/` [last accessed 2010-05-20]
[4] Freely available at `http://openmind.hri-us.com/`

situations. One table of the OMICS database contains the associations between typical objects and their typical locations that people have reported. The knowledge is rather unstructured – in fact only object-location pairs can be extracted – and is not immediately ready for automated reasoning. Its advantage is its focus on domestic indoor environments, which makes it valuable for our approach. It is conceivable to make use of the indoor objects and locations reported by users in order to restrict the search space when computing correlations between rooms and objects in the LabelMe dataset. Moreover, the information in the OMICS locations table can be used as a different method for generating prototypical assumptions (see Section 4.3.1 and Section 6.2.3).

Finally, in a virtual world like Twinity (see Section 10.2), NPC avatars encounter fewer restrictions regarding safety and security. Even privacy is much less of an issue. We are thus planning to use a guided home tour scenario to gather statistical data about object presence and room conceptualization for user-maintained LocalSpaces in Twinity. Although the collected data might not scale to the real world, the approach can yield a valid conceptualization within the virtual world, and secondly the approach can inform and foster extensions of robot-based exploration and human-augmented mapping.

10.3.2 Belief revision and belief update

As explained in Section 4.3, our focus in the conceptual map layer is to maintain a consistent and faithful model of the *current state* of the environment. Autonomous agents are faces with two fundamentally different circumstances that require them to adapt their representation of the current state-of-affairs. For one, in a dynamic environment things move, new events happen – change is the rule rather than the exception. An agent must be aware of these changes and accommodate them in his knowledge base. Secondly, the perception of an agent might be noisy and, in case of large-scale spatial environments, incomplete. There is hence an inherent uncertainty in interpretations of sensor input, which might lead to false assumptions and erroneous derivations. As soon as the agent notices a previous error, it has to revise its belief about the state of the world. The first process, accommodation of changed information, is also called *belief update*, whereas the latter, error-recovery, is a case of *belief revision* (Russell and Norvig, 2003).

In our approach, this distinction is not made because the aim of the conceptual map is to provide an accurate account of the current state-of-affairs at any point. In contrast, an agent that is supposed to have a memory of its past experiences, i.e., an agent that is supposed to have a notion of time, must be able to differentiate between the two cases. Krieger (2010a) presents an account to representing individuals as *perdurants* that have *time slices*, which specify an

extension in time. This allows to express diachronic (e.g., relationships that change over time) as well as synchronic knowledge, thus extending the knowledge base with a notion of dynamics. An approach to also take the agent's changing belief into account could make use of creating time-indexed ABoxes. Together with the perdurant/time-slice approach, this would provide the agent with the possibility to reason about its own changing belief about a changing world.

10.3.3 Restrictive and attributive information

We presented a model for situated generation and resolution of referring expressions that makes use of a spatial knowledge base. In our algorithms as well as our experiment, we assume that the interlocutors, i.e., the agent and its user(s), are familiar with the environment they are talking about. A topic that we have not addressed in this work is the ability to talk about entities that are not (yet) known.

For natural language processing, this poses challenges in both directions. The agent must be able to understand that an expression refers to an unknown entity, or an unknown property of the referent – either unknown to itself, or unknown to the hearer – and adapt its verbal behavior accordingly. If a referent (or some of its properties) can be unknown to the agent, the agent must be able to deal with the possibility that for the reference resolution process not all information is to be used restrictively. Some of the information conveyed by the speaker can be used attributively, i.e., conveying additional information that augments the agent's knowledge about the referent. In such a case, the agent should employ an active clarification strategy in which it negotiates which information is restrictive (i.e., meant to single out the referent among potential distractors) and which is attributive (i.e., meant to provide new information about the referent). An open issue for further research is hence to extend approaches to situated clarification and tutoring like the ones by Kruijff et al. (2008), Vrečko et al. (2009) and Skočaj et al. (2010), which are targeted at fully observable scenes, to large-scale space scenarios, which are not fully observable by the agent and its interlocutors. If, on the other hand, the agent must use information that is not known to the hearer, it needs to be able to notice that the hearer might have misunderstood a reference. In realistic settings, in which partial knowledge about the environment being talked about must be assumed for all interlocutors, however, misunderstanding and the need for clarification might arise in any of the interlocutors. Ultimately, hence, the agent must be able to engage in the collaborative process of building common ground during the course of a situated dialogue (Clark and Wilkes-Gibbs, 1986; Janíček, 2010).

List of Figures

Bibliography

James F. Allen, Lenhart K. Schubert, George Ferguson, Peter Heeman, Chung Hss Hwang, Tsuneaki Kato, Marc Light, Nathaniel G. Martin, Bradford W. Miller, Massimo Poesio, and David R. Traum. The TRAINS project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, 7(1):7–48, 1995.

Anne H. Anderson, Miles Bader, Ellen G. Bard, Elizabeth Boyle, Gwyneth M. Doherty, Simon C. Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Cathy Sotillo, Henry S. Thompson, and Regina Weinert. The HCRC Map Task corpus. *Language and Speech*, 34:351–366, 1991.

Grigoris Antoniou. *Nonmonotonic Reasoning*. The MIT Press, Cambridge, MA, USA, 1997.

Grigoris Antoniou and Frank van Harmelen. *A Semantic Web Primer, 2nd Edition*. Cooperative Information Systems. The MIT Press, Cambridge, MA, USA, 2008.

Douglas E. Appelt. Planning english referring expressions. *Artificial Intelligence*, 26(1):1–33, 1985.

Carlos Areces. *Logic Engineering. The Case of Description and Hybrid Logics*. PhD thesis, University of Amsterdam, Amsterdam, The Netherlands, October 2000.

Kai O. Arras, Óscar Martínez Mozos, and Wolfram Burgard. Using boosted features for the detection of people in 2D range data. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA 2007)*, Rome, Italy, April 2007.

Nicholas Asher and Alex Lascarides. *Logics of Conversation*. Cambridge University Press, Cambridge, UK; New York, NY, USA, 2003.

Franz Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys*, 23(3):345–405, 1991.

Franz Baader. Description logic terminology. In Franz Baader, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter Appendix 1. Cambridge University Press, Cambridge, UK; New York, NY, USA, 2003.

Franz Baader and Werner Nutt. Basic description logics. In Franz Baader, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter 2. Cambridge University Press, Cambridge, UK; New York, NY, USA, 2003.

Franz Baader, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press, Cambridge, UK; New York, NY, USA, 2003.

William Sims Bainbridge. The scientific research potential of virtual worlds. *Science*, 317(5837):472–476, July 2007.

Jason Baldridge and Geert-Jan M. Kruijff. Coupling CCG and Hybrid Logic Dependency Semantics. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 319–326, Philadelphia, PA, USA, July 2002.

Jason Baldridge and Geert-Jan M. Kruijff. Multi-modal combinatory categorial grammmar. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, pages 211–218, Budapest, Hungary, April 2003.

Richard Bartle. *Designing Virtual Worlds*. New Riders, 2003.

John A. Bateman. Enabling technology for multilingual natural language generation: the KPML development environment. *Journal of Natural Language Engineering*, 3(1):15–55, 1997.

John A. Bateman. Using aggregation for selecting content when generating referring expressions. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics (ACL'99)*, pages 127–134, Morristown, NJ, USA, 1999. Association for Computational Linguistics.

John A. Bateman, Renate Henschel, and Fabio Rinaldi. Generalized Upper Model 2.0: documentation. Technical report, GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany, 1995.

Andrea Bauer, Klaas Klasing, Georgios Lidoris, Quirin Mühlbauer, Florian Rohrmüller, Stefan Sosnowski, Tingting Xu, Kolja Kühnlenz, Dirk Wollherr, and Martin Buss. The Autonomous City Explorer: Towards natural human-robot interaction in urban environments. *International Journal of Social Robotics*, 1(2):127–140, 2009.

Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, and Lynn Andrea Stein. OWL Web Ontology Language reference. `http://www.w3.org/TR/owl-ref/`, February 2004. [Last accessed on 2010-04-26].

Dave Beckett and Art Barstow. N-Triples W3C RDF Core WG internal working draft. `http://www.w3.org/2001/sw/RDFCore/ntriples/`, September 2001. [Last accessed on 2010-04-21].

Patrick Beeson, Matt MacMahon, Joseph Modayil, Aniket Murarka, Benjamin Kuipers, and Brian Stankiewicz. Integrating multiple representations of spatial knowledge for mapping, navigation, and communication. In *Interaction Challenges for Intelligent Assistants*, Papers from the AAAI Spring Symposium, Stanford, CA, USA, 2007. AAAI.

Patrick Beeson, Joseph Modayil, and Benjamin Kuipers. Factoring the mapping problem: Mobile robot map-building in the Hybrid Spatial Semantic Hierarchy. *International Journal of Robotics Research*, 29(4):428–459, 2010.

Brent Berlin and Paul Kay. *Basic Color Terms: Their Universality and Evolution*. University of California Press, Berkeley, CA, USA, 1969.

Patrick Blackburn. Representation, reasoning, and relational structures: a hybrid logic manifesto. *Journal of the Interest Group in Pure Logic*, 8(3): 339–365, 2000.

Anna M. Borghi. Object concepts and action. In Diane Pecher and Rolf A. Zwaan, editors, *Grounding Cognition – The Role of Perception and Action in Memory, Language and Thinking*. Cambridge University Press, Cambridge, UK; New York, NY, USA, 2005.

Johan Bos, Ewan Klein, and Tetsushi Oka. Meaningful conversation with a mobile robot. In *Proceedings of the Research Note Sessions of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, pages 71–74, Budapest, Hungary, 2003.

Susan E. Brennan and Herbert H. Clark. Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 22(6):1482–1493, 1996.

Michael Brenner and Bernhard Nebel. Continual planning and acting in dynamic multiagent environments. *Journal of Autonomous Agents and Multiagent Systems*, 19(3):297–331, 2009.

Dan Brickley and R. V. Guha. RDF vocabulary description language 1.0: RDF Schema. `http://www.w3.org/TR/rdf-schema/`, October 2004. [Last accessed on 2010-03-23].

Roger Brown. How shall a thing be called? *Psychological Review*, 65(1): 14–21, 1958.

Wolfram Burgard, Armin B. Cremers, Dieter Fox, Dirk Hähnel, Gerhard Lakemeyer, Dirk Schulz, Walter Steiner, and Sebastian Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1–2): 3–55, October 1999.

Pär Buschka and Alessandro Saffiotti. Some notes on the use of hybrid maps for mobile robots. In *Proceedings of the 8th International Conference on Intelligent Autonomous Systems (IAS)*, Amsterdam, The Netherlands, March 2004.

Donna K. Byron and James F. Allen. What's a reference resolution module to do? redefining the role of reference in language understanding systems. In *Proceedings of the 4th Discourse Anaphora and Anaphor Resolution Colloquium (DAARC2002)*, pages 80–87, 2002.

Donna K. Byron and Eric Fosler-Lussier. The OSU Quake 2004 corpus of two-party situated problem-solving dialogs. In *Proceedings of the 15th Language and Resources and Evaluation Conference (LREC'06)*, 2006.

Donna K. Byron, Alexander Koller, Kristina Striegnitz, Justine Cassell, Robert Dale, Johanna Moore, and Jon Oberlander. Report on the first NLG challenge on generating instructions in virtual environments (GIVE). In *Proceedings of the 12th European Workshop on Natural Language Generation*

(ENLG 2009), Athens, Greece, March 2009. Association for Computational Linguistics.

Jean Carletta and Christopher S. Mellish. Risk-taking and recovery in task-oriented dialogue. *Journal of Pragmatics*, 26(1):71–107, 1996.

Kai-Uwe Carstensen, Christian Ebert, Cornelia Ebert, Susanne Jekat, Ralf Klabunde, and Hagen Langer, editors. *Computerlinguistik und Sprachtechnologie – Eine Einführung*. Spektrum Akademischer Verlag, Heidelberg, Germany, 3rd edition, 2010.

Noam Chomsky. *Syntactic Structures*. Mouton, The Hague / Paris, 1957.

Howie Choset, Kevin M. Lynch, Seth Hutchinson, George Kantor, Wolfram Burgard, Lydia E. Kavraki, and Sebastian Thrun. *Principles of Robot Motion: Theory, Algorithms and Implementations*. The MIT Press, Cambridge, MA, USA, 2005.

Eric L. Chown. Making predictions in an uncertain world: Environmental structure and cognitive maps. *Adaptive Behavior*, 7(1):17–33, December 1999.

Eric L. Chown. Gateways: An approach to parsing spatial domains. In *Proceedings of the International Conference on Machine Learning Workshop on Machine Learning of Spatial Knowledge*, pages 1–6, Palo Alto, California, 2000.

Eric L. Chown, Stephen Kaplan, and David Kortenkamp. Prototypes, location, and associative networks (PLAN): Towards a unified theory of cognitive mapping. *Cognitive Science*, 19(1):1–51, 1995.

Henrik Iskov Christensen, Geert-Jan M. Kruijff, and Jeremy L. Wyatt, editors. *Cognitive Systems*, volume 8 of *Cognitive Systems Monographs*. Springer Verlag, Berlin/Heidelberg, Germany, 2010.

Herbert H. Clark and Deanna Wilkes-Gibbs. Referring as a collaborative process. *Cognition*, 22:1–39, 1986.

Anthony G. Cohn and Shyamanta M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46:1–29, 2001.

Roger M. Cooper. The control of eye fixation by the meaning of spoken language : A new methodology for the real-time investigation of speech perception, memory, and language processing. *Cognitive Psychology*, 6(1):84–107, 1974.

Kenny R. Coventry and Simon C. Garrod. *Saying, Seeing and Acting – The Psychological Semantics of Spatial Prepositions*. Essays in Cognitive Psychology. Psychology Press, 2004.

Jacob W. Crandall and Michael A. Goodrich. Experiments in adjustable autonomy. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, 2001.

Madalina Croitoru and Kees van Deemter. A conceptual graph approach to the generation of referring expressions. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, January 2007.

Robert Dale and Nick Haddock. Generating referring expressions involving relations. In *Proceedings of the Fifth Meeting of the European Chapter of the Association for Computational Linguistics*, Berlin, Germany, April 1991.

Robert Dale and Ehud Reiter. Computational interpretations of the Gricean Maxims in the generation of referring expressions. *Cognitive Science*, 19(2): 233–263, 1995.

Keith Devlin. *Logic and Information*. Cambridge University Press, Cambridge, UK; New York, NY, USA, 1991.

Keith Devlin. Situation theory and situation semantics. In Dov M. Gabbay and John Woods, editors, *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of the History of Logic*, pages 601–664. Elsevier, 2006.

Albert Diosi, Geoffrey Taylor, and Lindsay Kleeman. Interactive SLAM using laser and advanced sonar. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, Barcelona, Spain, April 2005.

Keith S. Donnellan. Reference and definite descriptions. *Philosophical Review*, 75(3):281–304, 1966.

Max J. Egenhofer and M. Andrea Rodríguez. Relation algebras over containers and surfaces: An ontological study of a room space. *Spatial Cognition and Computation*, 1(2):155–180, 1999.

Mica R. Endsley and Daniel J. Garland, editors. *Situation Awareness Analysis and Measurement*. Laurence Erlbaum Associates, Mahwah, NJ, 2000.

Paul E. Engelhardt, Karl G.D. Bailey, and Fernanda Ferreira. Do speakers and listeners observe the Gricean Maxim of Quantity? *Journal of Memory and Language*, 54(4):554–573, 2006.

Deborah Estrin, David Culler, Kris Pister, and Gaurav Sukhatme. Connecting the physical world with pervasive networks. *IEEE Pervasive Computing*, 1 (1):59–69, 2002. ISSN 1536-1268.

Juan-Antonio Fernández and Javier González. *Multi-Hierarchical Representation of Large-Scale Space – Applications to Mobile Robots*, volume 24 of *International Series on Microprocessor-Based and Intelligent Systems Engineering*. Kluwer Academic Publishers, Dordrecht / Boston / London, 2001.

John Folkesson, Patric Jensfelt, and Henrik I. Christensen. Vision SLAM in the measurement subspace. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, Barcelona, Spain, April 2005.

Food and Agriculture Organization of the United Nations (FAO). Domain ontologies: Agricultural information management standards (AIMS). `http://aims.fao.org/website/Domain-Ontologies/`, 2010. [Last accessed on 2010-03-23].

Charles L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial Intelligence*, 19(1):17 – 37, 1982.

Stan Franklin and Art Graesser. Is it an agent, or just a program?: A taxonomy for autonomous agents. In Jörg P. Müller, Michael J. Wooldridge, and Nicholas R. Jennings, editors, *Intelligent Agents III. Agent Theories, Architectures, and Languages (ECAI'96 Workshop (ATAL))*, volume 1193 of *Lecture Notes in Computer Science*, pages 21–35. Springer Verlag, Berlin/Heidelberg, Germany, 1997.

Gottlob Frege. Über sinn und bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, pages 25–50, 1892.

Udo Frese and Lutz Schröder. Closing a million-landmarks loop. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, pages 5032–5039, 2006.

Scott M. Freundschuh and Madhu Sharma. Spatial image schemata, locative terms and geographic spaces in children's narrative. *Cartographica*, 32(2): 36–49, 1996.

Jannik Fritsch, Marcus Kleinehagenbrock, Sebastian Lang, Gernot A. Fink, and Gerhard Sagerer. Audiovisual person tracking with a mobile robot. In *Proceedings of the International Conference on Intelligent Autonomous Systems*, pages 898–906, Amsterdam, The Netherlands, March 2004.

Kotaro Funakoshi, Satoru Watanabe, Naoko Kuriyama, and Takenobu Tokunaga. Generation of relative referring expressions based on perceptual grouping. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA, 2004. Association for Computational Linguistics.

Cipriano Galindo, Alessandro Saffiotti, Silvia Coradeschi, Pär Buschka, Juan-Antonio Fernández-Madrigal, and Javier González. Multi-hierarchical semantic maps for mobile robotics. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-05)*, pages 3492–3497, Edmonton, Canada, August 2005.

Cipriano Galindo, Juan-Antonio Fernández-Madrigal, and Javier González. *Multiple Abstraction Hierarchies for Mobile Robot Operation in Large Environments*, volume 68 of *Studies in Computational Intelligence*. Springer Verlag, Berlin/Heidelberg, Germany, 2007.

Dorian Gálvez López. Combining object recognition and metric mapping for spatial modeling with mobile robots. Master's thesis, Royal Institute of Technology, Stockholm, Sweden, July 2007.

Dorian Gálvez López, Kristoffer Sjöö, Chandana Paul, and Patric Jensfelt. Hybrid laser and vision based object search and localization. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA 2008)*, pages 2636–2643, Pasadena, CA, USA, May 2008.

Peter Gärdenfors. Belief revision: An introduction. In Peter Gärdenfors, editor, *Belief Revision*. Cambridge University Press, Cambridge, UK; New York, NY, USA, 1992.

Peter Gärdenfors. *Knowledge in Flux – Modeling the Dynamics of Epistemic States*. The MIT Press, Cambridge, MA, USA, 1988.

Simon C. Garrod and Martin J. Pickering. Why is conversation so easy? *Trends in Cognitive Sciences*, 8(1):8–11, January 2004.

Héctor González-Banos and Jean-Claude Latombe. A randomized art-gallery algorithm for sensor placement. In *Proceedings of the Seventeenth Annual*

Symposium on Computational Geometry, pages 232–240, Medford, MA, USA, 2001.

Paul H. Grice. Logic and conversation. In Peter Cole and Jerry L. Morgan, editors, *Syntax and Semantics: Vol. 3, Speech Acts*, pages 43–58. Academic Press, New York, NY, USA, 1975.

Horst-Michael Gross, Hans Joachim Böhme, Christof Schröter, Steffen Müller, Alexander König, Christian Martin, Matthias Merten, and Andreas Bley. ShopBot: Progress in developing an interactive mobile shopping assistant for everyday use. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics 2008 (SMC 2008)*, pages 3471–3478, Singapore, October 2008.

Barbara J. Grosz. *The Representation and Use of Focus in Dialogue Understanding*. PhD thesis, Stanford University, 1977.

Thomas R. Gruber. Toward principles for the design of ontologies used for knowledge sharing. In Nicola Guarino and Roberto Poli, editors, *Formal Ontology in Conceptual Analysis and Knowledge Representation*. Kluwer Academic Publishers, Deventer, The Netherlands, 1993.

Rolf Grütter and Bettina Bauer-Messmer. Combining OWL with RCC for spatioterminological reasoning on environmental data. In *Proceedings of OWL: Experiences and Directions (OWLED 2007)*, Innsbruck, Austria, June 2007.

Axel Haasch, Sascha Hohenner, Sonja Hüwel, Marcus Kleinehagenbrock, Sebastian Lang, Ioannis Toptsis, Gernot A. Fink, Jannik Fritsch, Britta Wrede, and Gerhard Sagerer. BIRON – the Bielefeld robot companion. In Erwin Prassler, Gisbert Lawitzky, Paolo Fiorini, and Martin Hägele, editors, *Proceedings of the 2nd International Workshop on Advances in Service Robotics*, pages 27–32, Stuttgart, Germany, May 2004. Fraunhofer IRB Verlag.

Edward T. Hall. *The Hidden Dimension*. Doubleday, Garden City, NY, USA, 1966.

Marc Hanheide, Nick Hawes, Jeremy L. Wyatt, Moritz Göbelbecker, Michael Brenner, Kristoffer Sjöö, Alper Aydemir, Patric Jensfelt, Hendrik Zender, and Geert-Jan M. Kruijff. A framework for goal generation and management. In *Proceedings of the AAAI Workshop on Goal-Directed Autonomy*, Atlanta, GA, USA, July 2010. AAAI.

Nick Hawes and Jeremy L. Wyatt. Engineering intelligent information-processing systems with CAST. *Advanced Engineering Informatics*, 24: 27–39, 2010.

Nick Hawes, Aaron Sloman, Jeremy L. Wyatt, Michael Zillich, Henrik Jacobsson, Geert-Jan M. Kruijff, Michael Brenner, Gregor Berginc, and Danijel Skočaj. Towards an integrated robot with multiple cognitive functions. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, pages 1548–1553, Vancouver, Canada, July 2007.

Nick Hawes, Michael Brenner, and Kristoffer Sjöö. Planning as an architectural control mechanism. In *Proceedings of the 4th ACM/IEEE International Conference on Human-Robot Interaction (HRI 2009)*, pages 229–230, La Jolla, CA, USA, March 2009a. ACM.

Nick Hawes, Hendrik Zender, Kristoffer Sjöö, Michael Brenner, Geert-Jan M. Kruijff, and Patric Jensfelt. Planning and acting with an integrated sense of space. In Alexander Ferrein, Josef Pauli, Nils T. Siebel, and Gerald Steinbauer, editors, *HYCAS 2009: 1st International Workshop on Hybrid Control of Autonomous Systems – Integrating Learning, Deliberation and Reactive Control*, pages 25–32, Pasadena, CA, USA, July 2009b.

Nancy L. Hazen, Jeffery J. Lockman, and Herbert L. Pick, Jr. The development of children's representations of large-scale environments. *Child Development*, 49(3):623–636, September 1978.

Frederik W. Heger, Laura M. Hiatt, Brennan Sellner, Reid Simmons, and Sanjiv Singh. Results in sliding autonomy for multi-robot spatial assembly. In *Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS 2005)*, Munich, Germany, September 2005.

James F. Herman and Alexander W. Siegel. The development of cognitive mapping of the large-scale environment. *Journal of Experimental Child Psychology*, 26:389–406, 1978.

Ralf Hinkel and Thomas Knieriemen. Environment perception with a laser radar in a fast moving robot. In *Procedings of the Symposium on Robot Control (SYROCO '88)*, pages 68.1–68.7, October 1988.

Stephen C. Hirtle and John Jonides. Evidence for hierarchies in cognitive maps. *Memory and Cognition*, 13:208–217, 1985.

Joana Hois and Oliver Kutz. Natural language meets spatial calculi. In Christian Freksa, Nora S. Newcombe, Peter Gärdenfors, and Stefan Wölfl, editors, *Learning, Reasoning, and Talking about Space (SC'08)*, volume VI of *Spatial Cognition*, pages 266–282. Springer Verlag, Berlin/Heidelberg, Germany, 2008.

Honda Research Institute USA Inc. Open Mind Indoor Commonsense. `http://openmind.hri-us.com/`. [Last accessed on 2010-03-23].

Helmut Horacek. An algorithm for generating referential descriptions with flexible interfaces. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics (ACL-97)*, pages 206–213, Morristown, NJ, USA, 1997. Association for Computational Linguistics, Association for Computational Linguistics.

International Federation of Robotics (IFR) Statistical Department. *World Robotics 2009 Service Robots*. VDMA Verlag, Frankfurt a.M., Germany, 2009.

Hiroshi Ishiguro, Tetsuo Ono, Michita Imai, Takeshi Maeda, Takayuki. Kanda, and Ryohei Nakatsu. Robovie: An interactive humanoid robot. *Industrial Robot: An International Journal*, 28(6):498–504, 2001.

Henrik Jacobsson, Nick Hawes, Geert-Jan M. Kruijff, and Jeremy L. Wyatt. Crossmodal content binding in information-processing architectures. In *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI 2008)*, Amsterdam, The Netherlands, March 2008.

Miroslav Janíček. Continuous planning for communicative grounding in situated dialogue. Unpublished master's thesis, Charles University Prague, Prague, Czech Republic, 2010.

Kathryn J. Jeffery and Neil Burgess. A metric for the cognitive map: Found at last? *Trends in Cognitive Sciences*, 10(1), January 2006.

Wolfgang Kainz, Max J. Egenhofer, and Ian Greasley. Modeling spatial relations and operations with partially ordered sets. *International Journal of Geographical Information Systems*, 7(3):215–229, 1993.

Michael Karg, Kai M. Wurm, Cyrill Stachniss, Klaus Dietmayer, and Wolfram Burgard. Consistent mapping of multistory buildings by introducing global constraints to graph-based SLAM. In *Proceedings of the 2010 IEEE*

International Conference on Robotics and Automation (ICRA 2010), pages 5383–5388, Anchorage, AK, USA, May 2010.

Yarden Katz and Bernardo Cuenca Grau. Representing qualitative spatial information in OWL-DL. In *Proceedings of OWL: Experiences and Directions (OWLED 2005)*, Galway, Ireland, November 2005.

Kazuhiko Kawamura, Stephen M. Gordon, Palis Ratanaswasd, Erdem Erdemir, and Joseph F. Hall. Implementation of cognitive control for a humanoid robot. *International Journal of Humanoid Robotics*, 2008.

John D. Kelleher. Integrating visual and linguistic salience for reference resolution. In Norman Creaney, editor, *Proceedings of the 16th Irish conference on Artificial Intelligence and Cognitive Science (AICS '05)*, pages 159–168. The University of Ulster, September 2005.

John D. Kelleher. Attention driven reference resolution in multimodal contexts. *Artificial Intelligence Review*, 25:21–35, 2007.

John D. Kelleher and Josef van Genabith. Exploiting visual salience for the generation of referring expressions. In *Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*, Miami Beach, FL, USA, May 2004.

Shanker Keshavdas. Grid based SLAM using Rao-Blackwellized particle filters. Unpublished master's thesis, Heriot Watt University, Edinburgh, UK, May 2009.

Marcus Kleinehagenbrock, Sebastian Lang, Jannik Fritsch, Frank Lömker, Gernot A. Fink, and Gerhard Sagerer. Person tracking with a mobile robot based on multi-modal anchoring. In *Proceedings of the IEEE International Workshop on Robot and Human Interactive Communication (ROMAN 2002)*, Berlin, Germany, September 2002.

Tina Klüwer, Peter Adolphs, Feiyu Xu, Hans Uszkoreit, and Xiwen Cheng. Talking NPCs in a virtual game world. In *Proceedings of the System Demonstrations Section at ACL 2010*, Uppsala, Sweden, July 2010.

Pia Knoeferle, Matthew Crocker, Martin Pickering, and Christoph Scheepers. The influence of the immediate visual context on incremental thematic role-assignment: evidence from eye-movements in depicted events. *Cognition*, 95(1):95–127, 2005.

Alexander Koller, Johanna Moore, Barbara Di Eugenio, James Lester, Laura Stoia, Donna K. Byron, Jon Oberlander, and Kristina Striegnitz. Shared task proposal: Instruction giving in virtual worlds. Working group report, Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation, 2007.

Vladimir Kolovski, Bijan Parsia, and Yarden Katz. Implementing OWL defaults. In *Proceedings of OWL: Experiences and Directions (OWLED 2006)*, Athens, GA, USA, November 2006.

Emiel Krahmer and Mariët Theune. Efficient context-sensitive generation of referring expressions. In Kees van Deemter and Rodger Kibble, editors, *Information Sharing: Givenness and Newness in Language Processing*, pages 223–264. CSLI Publications, Stanford, CA, USA, 2002.

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72, 2003. ISSN 0891-2017.

Bernd Krieg-Brückner, Udo Frese, Klaus Lüttich, Christian Mandel, Till Massokowski, and Robert J. Ross. Specification of an ontology for Route Graphs. In Christian Freksa, Markus Knauff, Bernd Krieg-Brückner, Bernhard Nebel, and Thomas Barkowsky, editors, *Spatial Cognition IV. Reasoning, Action, and Interaction*, volume 3343 of *Lecture Notes in Artificial Intelligence*, pages 390–412. Springer Verlag, Heidelberg, Germany, 2005.

Hans-Ulrich Krieger. A general methodology for equipping ontologies with time. In *Proceedings of LREC 2010*, 2010a.

Hans-Ulrich Krieger. A temporal extension of hayes-/ter horst-style entailment rules. In *under submission*, 2010b.

Hans-Ulrich Krieger, Bernd Kiefer, and Thierry Declerck. A framework for temporal representation and reasoning in business intelligence applications. In *AAAI 2008 Spring Symposium on AI Meets Business Rules and Process Management*, Papers from the AAAI Spring Symposium, pages 59–70. AAAI, 2008.

Geert-Jan M. Kruijff. Context-sensitive utterance planning for CCG. In *Proceedings of the 10th European Workshop on Natural Language Generation*, Abderdeen, Scotland, UK, 2005.

Geert-Jan M. Kruijff. *A Categorial-Modal Logical Architecture of Informativity: Dependency Grammar Logic & Information Structure*. PhD thesis, Charles University Prague, Prague, Czech Republic, 2001.

Geert-Jan M. Kruijff and Michael Brenner. Modelling spatio-temporal comprehension in situated human-robot dialogue as reasoning about intentions and plans. In *AAAI Spring Symposium on Intentions in Intelligent Systems*, Papers from the AAAI Spring Symposium. AAAI, 2007.

Geert Jan M. Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, and Nick Hawes. Incremental, multi-level processing for comprehending situated dialogue in human-robot interaction. In *Language and Robots: Proceedings of the Symposium*, Aveiro, Portugal, December 2007a.

Geert-Jan M. Kruijff, Hendrik Zender, Patric Jensfelt, and Henrik I. Christensen. Situated dialogue and spatial organization: What, where…and why? *International Journal of Advanced Robotic Systems*, 4(1):125–138, March 2007b.

Geert-Jan M. Kruijff, Michael Brenner, and Nick Hawes. Continual planning for cross-modal situated clarification in human-robot interaction. In *Proceedings of the 17th International Symposium on Robot and Human Interactive Communication (RO-MAN 2008)*, pages 592–597, Munich, Germany, August 2008.

Geert-Jan M. Kruijff, Pierre Lison, Trevor Benjamin, Henrik Jacobsson, Hendrik Zender, and Ivana Kruijff-Korbayová. Situated dialogue processing for human-robot interaction. In Henrik Iskov Christensen, Geert-Jan M. Kruijff, and Jeremy L. Wyatt, editors, *Cognitive Systems*, volume 8 of *Cognitive Systems Monographs*, chapter 8, pages 311–364. Springer Verlag, Berlin/Heidelberg, Germany, 2010.

Benjamin Kuipers. *Representing Knowledge of Large-Scale Space*. PhD thesis, MIT-AI TR-418, Massachusetts Institute of Technology, Cambridge, MA, USA, May 1977.

Benjamin Kuipers. The Spatial Semantic Hierarchy. *Artificial Intelligence*, 119: 191–233, 2000.

Benjamin Kuipers, Joseph Modayil, Patrick Beeson, Matt MacMahon, and Francesco Savelli. Local metrical and global topological maps in the Hybrid Spatial Semantic Hierarchy. In *Proceedings of the 2004 IEEE International*

Conference on Robotics and Automation (ICRA 2004), New Orleans, LA, USA, April 2004.

George Lakoff and Mark Johnson. *Metaphors we live by*. Chicago University Press, Chicago, IL, USA, 1980.

George Lakoff and Mark Johnson. *Philosophy in the Flesh: The Embodied Mind and Its Challenge to Western Thought*. Basic Books, New York, NY, USA, 1999.

M. Lansdale and T. Ormerod. *Understanding Interfaces*. Academic Press, London, UK, 1994.

Jean-Claude Latombe. *Robot Motion Planning*. Academic Publishers, Boston, MA, 1991.

Oliver Lemon, Anne Bracy, Alexander Gruenstein, and Stanley Peters. A multimodal dialogue system for human-robot conversation. In *Proceedings of the Second Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL 2001Proceedings of the Second Meeting of the North American Chapter of the Association of Computational Linguistics (NAACL 2001)*, Pittsburg PA, 2001.

Stephen C. Levinson. *Space in Language and Cognition – Explorations in Cognitive Diversity*. Cambridge University Press, Cambridge, UK; New York, NY, USA, 2003.

François Lévy. Weak extensions for default theories. In *Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, volume 747 of *Lecture Notes in Computer Science*. Springer Verlag, Berlin/Heidelberg, Germany, 1993.

Mattias Lindström and Jan-Olof Eklundh. Detecting and tracking moving objects from a mobile platform using a laser range scanner. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'01)*, volume 3, pages 1364–1369, Wailea Maui HI, USA, 2001.

Pierre Lison and Geert-Jan M. Kruijff. Salience-driven contextual priming of speech recognition for human-robot interaction. In *ECAI 2008*, 2008.

David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

Maryamossadat N. Mahani and Elin Anna Topp. Identifying and resolving ambiguities within joint movement scenarios in HRI. In Marc Hanheide and Hendrik Zender, editors, *Proceedings of the ICRA 2010 Workshop on Interactive Communication for Autonomous Intelligent Robots (ICAIR)*, pages 37–39, Anchorage, AK, USA, May 2010.

Dario Maio and Stefano Rizzi. Clustering by discovery on maps. *Pattern Recognition Letters*, 13(2):89–94, 1992.

Robert W. Marx. The TIGER system: Automating the geographic structure of the United States census. *Government Publications Review*, 13(2):181–201, March–April 1986.

Viviana Mascardi, Valentina Cordì, and Paolo Rosso. A comparison of upper ontologies. In *Proceedings of the Conference on Agenti e industria: Applicazioni tecnologiche degli agenti software (WOA 2007)*, Genova, Italy, September 2007.

Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. Wonderweb deliverable d18 – ontology library. IST Project 2001-33052 WonderWeb Deliverable Del 18, Laboratory For Applied Ontology – ISTC-CNR, December 2003.

John McCarthy and Patrick J. Hayes. Some philosophical problems from the standpoint of artificial intelligence. In B. Meltzer and D. Michie, editors, *Machine Intelligence 4*, pages 463–502. Edinburgh University Press, 1969.

Timothy P. McNamara. Mental representations of spatial relations. *Cognitive Psychology*, 18:87–121, 1986.

Javier Minguez and Luis Montano. Nearness diagram (ND) navigation: Collision avoidance in troublesome scenarios. *IEEE Transactions on Robotics and Automation*, 20(1):45–59, February 2004.

Francesco Mondada, Michael Bonani, Xavier Raemy, James Pugh, Christopher Cianci, Adam Klaptocz, Stéphane Magnenat, Jean-Christophe Zufferey, Dario Floreano, and Alcherio Martinoli. The e-puck, a robot designed for education in engineering. In *Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions (Robotica 2009)*, pages 59–65, May 2009.

Óscar Martínez Mozos. *Semantic Place Labeling with Mobile Robots*. Springer Tracts in Advanced Robotics (STAR). Springer Verlag, Berlin/Heidelberg, Germany, 2010. ISBN 978-3-642-11209-6.

Óscar Martínez Mozos, Cyrill Stachniss, and Wolfram Burgard. Supervised learning of places from range data using adaboost. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA 2005)*, pages 1742–1747, Barcelona, Spain, April 2005.

Óscar Martínez Mozos, Patric Jensfelt, Hendrik Zender, Geert-Jan M. Kruijff, and Wolfram Burgard. From labels to semantics: An integrated system for conceptual spatial representations of indoor environments for mobile robots. In *Proceedings of the IEEE ICRA-07 Workshop: Semantic Information in Robotics*, Rome, Italy, April 2007a.

Óscar Martínez Mozos, Patric Jensfelt, Hendrik Zender, Geert-Jan M. Kruijff, and Wolfram Burgard. An integrated system for conceptual spatial representations of indoor environments for mobile robots. In *Proceedings of the IROS 2007 Workshop: From Sensors to Human Spatial Concepts (FS2HSC)*, pages 25–32, San Diego, CA, USA, November 2007b.

Daniele Nardi and Ronald J. Brachman. An introduction to description logics. In Franz Baader, Deborah L. McGuinness, Daniele Nardi, and Peter F. Patel-Schneider, editors, *The Description Logic Handbook: Theory, Implementation, and Applications*, chapter 1. Cambridge University Press, Cambridge, UK; New York, NY, USA, 2003.

Bernhard Nebel. A knowledge level analysis of belief revision. In Ronald J. Brachman, Hector J. Levesque, and Raymond Reiter, editors, *Principles of Knowledge Representation and Reasoning: Proceedings of the 1st International Conference (KR'89)*, pages 301–311, Toronto, Canada, May 1989.

Paul M. Newman, John J. Leonard, Juan D. Tardós, and José Neira. Explore and return: Experimental validation of real-time concurrent mapping and localization. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 2002)*, pages 1802–1809, Washington, D.C., USA, 2002.

Joseph O'Rourke. *Art gallery theorems and algorithms*. Oxford University Press, Oxford, UK; New York, NY, USA, 1987.

Elena Pacchierotti, Henrik I. Christensen, and Patric Jensfelt. Embodied social interaction for service robots in hallway environments. In *Proceedings of the International Conference on Field and Service Robotics (FSR 2005)*, pages 476–487, Brisbane, Australia, July 2005. IEEE.

Ivandré Paraboni, Kees van Deemter, and Judith Masthoff. Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, 33(2):229–254, June 2007.

Diane Pecher and Rolf A. Zwaan, editors. *Grounding Cognition – The Role of Perception and Action in Memory, Language and Thinking*. Cambridge University Press, Cambridge, UK; New York, NY, USA, 2005.

Julia Peltason, Frederic H. K. Siepmann, Thorsten P. Spexard, Britta Wrede, Marc Hanheide, and Elin Anna Topp. Mixed-initiative in human augmented mapping. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA2009)*, Kobe, Japan, May 2009.

Martin J. Pickering and Simon C. Garrod. Alignment as the basis for successful communication. *Research on Language and Computation*, 4(2–3):203–228, October 2006.

Massimo Poesio. A situation-theoretic formalization of definite description interpretation in plan elaboration dialogues. In Peter Aczel, David Israel, Yasuhiro Katagiri, and Stanley Peters, editors, *Situation Theory and its Applications Volume 3*, CSLI Lecture Notes No. 37, pages 339–374. Center for the Study of Language and Information, Menlo Park, CA, USA, 1993.

Massimo Poesio and Renata Vieira. A corpus-based investigation of definite description use. *Computational Linguistics*, 24(2):183–216, 1998.

Massimo Poesio, Olga Uryupina, Renata Vieira, Mijail Alexandrov-Kabadjov, and Rodrigo Goulart. Discourse-new detectors for definite description resolution: A survey and a preliminary proposal. In *Proceedings of the ACL 2004 Workshop on Reference Resolution and Its Applications*, pages 47–54, Barcelona, Spain, 2004.

Shelley Powers. *Practical RDF*. O'Reilly Media, July 2003.

Ellen F. Prince. Toward a taxonomy of given-new information. In Peter Cole, editor, *Radical Pragmatics*, pages 223–255. Academic Press, New York, NY, USA, 1981.

Ellen F. Prince. The ZPG letter: subjects, definiteness, and information status. In Sandra Thompson and William Mann, editors, *Discourse Description: diverse analyses of a fund raising text*, pages 295–325. John Benjamins, 1992.

Andrzej Pronobis and Barbara Caputo. COLD: COsy Localization Database. *The International Journal of Robotics Research (IJRR)*, 28(5):588–594, May 2009.

Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Adrian N. Bishop, and Patric Jensfelt. A framework for robust cognitive spatial mapping. In *Proceedings of the 14th International Conference on Advanced Robotics (ICAR 2009)*, Munich, Germany, June 2009.

Andrzej Pronobis, Patric Jensfelt, Kristoffer Sjöö, Hendrik Zender, Geert-Jan M. Kruijff, Óscar Martínez Mozos, and Wolfram Burgard. Semantic modelling of space. In Henrik Iskov Christensen, Geert-Jan M. Kruijff, and Jeremy L. Wyatt, editors, *Cognitive Systems*, volume 8 of *Cognitive Systems Monographs*, chapter 5. Springer Verlag, Berlin/Heidelberg, Germany, 2010a.

Andrzej Pronobis, Kristoffer Sjöö, Alper Aydemir, Adrian N. Bishop, and Patric Jensfelt. Representing spatial knowledge in mobile cognitive systems. In *11th International Conference on Intelligent Autonomous Systems (IAS-11)*, Ottawa, Canada, August 2010b.

Eric Prud'hommeaux and Andy Seaborne. SPARQL Query Language for RDF. `http://www.w3.org/TR/rdf-sparql-query/`, January 2008. [Last accessed 2010-03-23].

Yves Raimond, Frédéric Giasson, Kurt Jacobson, George Fazekas, Thomas Gängler, and Simon Reinhardt. Music ontology specification. `http://musicontology.com/`, February 2010. [Last accessed 2010-03-23].

RDF Working Group. Resource Description Framework (RDF). `http://www.w3.org/RDF/`, October 2004. [Last accessed 2010-03-23].

Alan Rector and Guus Schreiber. Qualified cardinality restrictions (QCRs): Constraining the number of values of a particular type for a property. `http://www.w3.org/2001/sw/BestPractices/OEP/QCR/`, November 2005. [Draft of November 2, 2005; Last accessed on 2010-03-30].

Ehud Reiter and Robert Dale. A fast algorithm for the generation of referring expressions. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING-92)*, pages 232–238, Nantes, France, August 1992.

Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1-2): 81–132, 1980.

David Ribas, Pere Ridao, Juan Domingo Tardós, and José Neira. Underwater SLAM in man made structured environments. *Journal of Field Robotics*, 25 (11):898–921, December 2008.

M. Andrea Rodríguez and Max J. Egenhofer. Image-schemata-based spatial inferences: The container-surface algebra. In Stephen C. Hirtle and Andrew U. Frank, editors, *Spatial Information Theory: A Theoretical Basis for GIS (COSIT '97)*, volume 1329 of *Lecture Notes in Computer Science*, pages 35–52. Springer Verlag, Berlin, Germany, 1997.

Eleanor Rosch. Principles of categorization. In E. Rosch and B. Lloyd, editors, *Cognition and Categorization*, pages 27–48. Lawrence Erlbaum Associates, Hillsdale, NJ, USA, 1978.

Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. Labelme: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77(1–3):157–173, May 2008.

Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall Series in Artificial Intelligence. Pearson Education, Upper Saddle River, NJ, USA, second edition edition, 2003.

Marc Schröder and Jürgen Trouvain. The german text-to-speech synthesis system MARY: A tool for research, development and teaching. *International Journal of Speech Technology*, 6:365–377, 2003.

Dirk Schulz, Wolfram Burgard, Dieter Fox, and Armin B. Cremers. People tracking with a mobile robot using sample-based joint probabilistic data association filters. *International Journal of Robotics Research*, 22(2):99–116, 2003.

Murray Shanahan. A logical account of perception incorporating feedback and expectation. In *Proceedings of the Eighth International Conference on Principles and Knowledge Representation and Reasoning (KR-02)*, pages 3–13, Toulouse, France, April 2002.

Thomas C. Shermer. Recent results in art galleries. *Proceedings of the IEEE*, 80(9):1384–1399, September 1992. ISSN 0018-9219.

Candace L. Sidner, Cory D. Kidd, Christopher Lee, and Neal Lesh. Where to look: A study of human-robot engagement. In *Proceedings of the 9th International Conference on Intelligent User Interfaces (IUI '04)*, pages 78–84, 2004.

Roland Siegwart and Illah R. Nourbakhsh. *Introduction to Autonomous Mobile Robots*. The MIT Press, Cambridge, MA, USA, 2004.

Roland Siegwart, Kai Oliver Arras, Samir Bouabdallah, Daniel Burnier, Gilles Froidevaux, Xavier Greppin, Björn Jensen, Antoine Lorotte, Laetitia Mayor, Mathieu Meisser, Roland Philippsen, Ralph Piguet, Guy Ramel, Gregoire Terrien, and Nicola Tomatis. Robox at expo.02: A large scale installation of personal robots. *Robotics and Autonomous Systems*, 42:203–222, 2003.

Kristoffer Sjöö, Hendrik Zender, Patric Jensfelt, Geert-Jan M. Kruijff, Andrzej Pronobis, Nick Hawes, and Michael Brenner. The Explorer system. In Henrik Iskov Christensen, Geert-Jan M. Kruijff, and Jeremy L. Wyatt, editors, *Cognitive Systems*, volume 8 of *Cognitive Systems Monographs*, chapter 10, pages 395–421. Springer Verlag, Berlin/Heidelberg, Germany, 2010.

Danijel Skočaj, Miroslav Janíček, Matej Kristan, Geert Jan M. Kruijff, Aleš Leonardis, Pierre Lison, Alen Vrečko, and Michael Zillich. A basic cognitive system for interactive continuous learning of visual concepts. In Marc Hanheide and Hendrik Zender, editors, *Proceedings of the ICRA 2010 Workshop on Interactive Communication for Autonomous Intelligent Robots (ICAIR)*, pages 30–36, Anchorage, AK, USA, May 2010.

Michael K. Smith, Chris Wely, and Deborah L. McGuinness. OWL Web Ontology Language guide. `http://www.w3.org/TR/owl-guide/`, October 2004. [Last accessed on 2010-03-23].

Thorsten P. Spexard, Shuyin Li, Britta Wrede, Jannik Fritsch, Gerhard Sagerer, Olaf Booij, Zoran Zivkovic, Bas Terwijn, and Ben J. A. Kröse. BIRON, where are you? enabling a robot to learn new places in a real home environment by integrating spoken dialog and visual localization. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2006)*, pages 934–940, Beijing, China, 2006.

Robert Stalnaker. Common Ground. *Linguistics and Philosophy*, 25(5–6): 701–721, 2002.

Mark Steedman. Categorial grammar. In Rob Wilson and Frank Keil, editors, *The MIT Encyclopedia of Cognitive Sciences*. The MIT Press, Cambridge, MA, USA, 1999.

Mark Steedman. *The Syntactic Process*. The MIT Press, Cambridge, MA, USA, 2000.

Mark Steedman and Jason Baldridge. Combinatory categorial grammar (draft 5.0). April 2007.

Albert Stevens and Patty Coupe. Distortions in judged spatial relations. *Cognitive Psychology*, 10:422–437, 1978.

Laura Stoia, Darla Magdalena Shockley, Donna K. Byron, and Eric Fosler-Lussier. SCARE: a situated corpus with annotated referring expressions. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, May 2008. European Language Resources Association (ELRA).

Matthew Stone. On identifying sets. In *Proceedings of the First International Conference on Natural Language Generation (INLG-2000)*, pages 116–123, Morristown, NJ, USA, 2000. Association for Computational Linguistics.

Peter Frederick Strawson. On referring. *Mind*, 59(235):320–344, July 1950.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. The MIT Press, Cambridge, MA, USA, 2005.

Elin Anna Topp and Henrik I. Christensen. Tracking for following and passing persons. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2005)*, pages 70–76, Edmonton, Canada, August 2005.

Elin Anna Topp, Helge Hüttenrauch, Henrik I. Christensen, and Kerstin Severinson Eklundh. Acquiring a shared environment representation. In *Proceedings of the 1st ACM Conference on Human-Robot Interaction (HRI 2006)*, pages 361–362, Salt Lake City, UT, USA, 2006a.

Elin Anna Topp, Helge Hüttenrauch, Henrik I. Christensen, and Kerstin Severinson Eklundh. Bringing together human and robotic environment representations – a pilot study. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China, October 2006b.

Timothy Trainor. U.S. Census Bureau geographic support: A response to changing technology and improved data. *Cartography and Geographic Information Science*, 30(2):217–223, April 2003.

John K. Tsotsos. On the relative complexity of active vs. passive visual search. *Int. J. Comput. Vision*, 7(2):127–141, 1992.

Joost van de Weijer, Cordelia Schmid, Jakob Verbeek, and Diane Larlus. Learning color names for real-world applications. *IEEE Transactions on Image Processing*, 18(7):1512–1523, 2009.

Kees van Deemter. Generating vague descriptions. In *Proceedings of the First International Conference on Natural Language Generation (INLG-2000)*, pages 179–185, Mitzpe Ramon, Israel, June 2000.

Kees van Deemter. Generating referring expressions: boolean extensions of the incremental algorithm. *Computational Linguistics*, 28(1):37–52, 2002.

Shrihari Vasudevan, Stefan Gachter, Viet Nguyen, and Roland Siegwart. Cognitive maps for mobile robots – an object based approach. *Robotics and Autonomous Systems*, 55(5):359–371, May 2007.

Renata Vieira and Massimo Poesio. An empirically based system for processing definite descriptions. *Computational Linguistics*, 26(4):539–593, 2000.

Jette Viethen and Robert Dale. Algorithms for generating referring expressions: Do they do what people do? In *Proceedings of the 4th International Natural Language Generation Conference (INLG 2006)*, pages 63–70, Sydney, Australia, 2006.

Jette Viethen and Robert Dale. Generating relational references: What makes a difference? In *Proceedings of the Australasian Language Technology Association Workshop 2008*, Hobart, Australia, December 2008a.

Jette Viethen and Robert Dale. The use of spatial relations in referring expressions. In *Proceedings of the 5th International Natural Language Generation Conference (INLG 08)*, Salt Fork, OH, USA, June 2008b.

Pooja Viswanathan, David Meger, Tristram Southey, James J. Little, and Alan K. Mackworth. Automated spatial-semantic modeling with applications to place labeling and informed search. In *CRV '09: Proceedings of the 2009 Canadian Conference on Computer and Robot Vision*, pages 284–291, Washington, DC, USA, 2009. IEEE Computer Society.

Pooja Viswanathan, Tristram Southey, James J. Little, and Alan K. Mackworth. Automated place classification using object detection. In *Proceedings of the Seventh Canadian Conference on Computer and Robot Vision (CRV 2010)*, Ottawa, Canada, 2010.

Alen Vrečko, Danijel Skočaj, Nick Hawes, and Aleš Leonardis. A computer vision integration model for a multi-modal cognitive system. In *Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009)*, pages 3140–3147, St. Louis, MO, USA, October 2009.

Chieh-Chih Wang and Chuck Thorpe. Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA 2002)*, volume 3, pages 2918–2924, Washington, D.C., USA, May 2002.

Steffen Werner, Bernd Krieg-Brückner, and Theo Herrmann. Modelling navigational knowledge by Route Graphs. In Christian Freksa, Wilfried Brauer, Christopher Habel, and Karl F. Wender, editors, *Spatial Cognition II*, volume 1849 of *Lecture Notes in Artificial Intelligence*, pages 295–316. Springer Verlag, Heidelberg, Germany, 2000.

Michael White. OpenCCG: The OpenNLP CCG Library. `http://openccg.sourceforge.net/`, February 2010. [Last accessed on 2010-04-15].

Michael White and Jason Baldridge. Adapting chart realization to CCG. In *Proceedings of the 9th European Workshop on Natural Language Generation (ENLG 2003)*, Budapest, Hungary, 2003.

Ian Wright, Aaron Sloman, and Luc Beaudoin. Towards a design-based analysis of emotional episodes. *Philosophy, Psychiatry, & Psychology*, 3(2):101–126, 1996.

Jeremy L. Wyatt, Alper Aydemir, Michael Brenner, Marc Hanheide, Nick Hawes, Patric Jensfelt, Matej Kristan, Geert-Jan M. Kruijff, Pierre Lison, Andrzej Pronobis, Kristoffer Sjöö, Danijel Skočaj, Alen Vrečko, Hendrik Zender, and Michael Zillich. Self-understanding and self-extension: A systems and representational approach. *IEEE Transactions on Autonomous Mental Development*, 2(4):282–303, December 2010.

Hendrik Zender. Learning spatial organization through situated dialogue. Unpublished diploma thesis, Saarland University, Saarbrücken, Germany, August 2006.

Hendrik Zender and Geert-Jan M. Kruijff. Multi-layered conceptual spatial mapping for autonomous mobile robots. In Holger Schultheis, Thomas Barkowsky, Benjamin Kuipers, and Bernhard Hommel, editors, *Control*

Mechanisms for Spatial Knowledge Processing in Cognitive / Intelligent Systems – Papers from the AAAI Spring Symposium, Technical Report SS-07-01, pages 62–66, Menlo Park, CA, USA, March 2007a. AAAI, AAAI Press.

Hendrik Zender and Geert-Jan M. Kruijff. Towards generating referring expressions in a mobile robot scenario. In *Language and Robots: Proceedings of the Symposium*, pages 101–106, Aveiro, Portugal, December 2007b.

Hendrik Zender, Patric Jensfelt, and Geert-Jan M. Kruijff. Human- and situtaion-aware people following. In *Proceedings of the 16th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN 2007)*, pages 1131–1136, Jeju Island, Korea, August 2007a.

Hendrik Zender, Patric Jensfelt, Óscar Martínez Mozos, Geert-Jan M. Kruijff, and Wolfram Burgard. An integrated robotic system for spatial understanding and situated interaction in indoor environments. In *Proceedings of the Twenty-Second Conference on Artificial Intelligence (AAAI-07)*, pages 1584–1589, Vancouver, Canada, July 2007b.

Hendrik Zender, Óscar Martínez Mozos, Patric Jensfelt, Geert-Jan M. Kruijff, and Wolfram Burgard. Conceptual spatial representations for indoor mobile robots. *Robotics and Autonomous Systems*, 56(6):493–502, June 2008.

Hendrik Zender, Geert-Jan M. Kruijff, and Ivana Kruijff-Korbayová. A situated context model for resolution and generation of referring expressions. In *Proceedings of the 12th European Workshop on Natural Language Generation (ENLG 2009)*, pages 126–129, Athens, Greece, March 2009a. Association for Computational Linguistics.

Hendrik Zender, Geert-Jan M. Kruijff, and Ivana Kruijff-Korbayová. Situated resolution and generation of spatial referring expressions for robotic assistants. In *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI-09)*, pages 1604–1609, Pasadena, CA, USA, July 2009b.

Hendrik Zender, Christopher Koppermann, Fai Greeve, and Geert-Jan M. Kruijff. Anchor-progression in spatially situated discourse: a production experiment. In *Proceedings of the Sixth International Natural Language Generation Conference (INLG 2010)*, pages 209–213, Trim, Co. Meath, Ireland, July 2010. Association for Computational Linguistics.

Index