

# Towards Dialogue Based Shared Control of Navigating Robots<sup>\*</sup>

Robert J. Ross, Hui Shi, Tillman Vierhuff,  
Bernd Krieg-Brückner, John Bateman<sup>†</sup>

FB3 Mathematik und Informatik, Universität Bremen,  
`robertr,shi,tv,bkb@informatik.uni-bremen.de`

<sup>†</sup>FB10 Sprach und Literaturwissenschaften, Universität Bremen,  
`bateman@uni-bremen.de`  
<http://www.sfbtr8.uni-bremen.de/project/I3/>

**Abstract.** Establishing a clean relationship between a robot’s spatial model and natural language components is a non-trivial task, but is key to designing verbally controlled, navigating service robots. In this paper we examine the issues involved in the development of dialogue controlled navigating robots. In particular, we treat our robots as so-called *Shared Control Systems*, where robot and user cooperate to achieve a shared goal. We begin by characterising four categories of Shared Control Problems that affect verbally controlled navigating robots. Producing solutions to these problems requires a clear methodology in the linking of ‘common-sense’ representations of space used by the robots, and the language interface. To this end, we present the *SharC Cognitive Control Architecture* as a general purpose, agent-based dialogue control system that provides a suitable framework for relating spatial information to natural language communication. To illustrate our approach, we focus in particular on natural language understanding, and show how natural language utterances may be mapped to formally modelled spatial concepts, thus helping to overcome problems in shared control.

## 1 Introduction

With increased applicability in the domestic and office domains, service robots are becoming more and more interesting for both industrial and academic research. A characteristic of service robots, as distinct from heavy industrial or exploratory robots, is that they will often operate in partially known and dynamic environments, moving between locations while performing their assigned duties. Thus, these embodied, situated robots require a working understanding of their spatial environment.

Service robots are examples of *shared-control systems* where a human operator and an automated technical system are interdependently in charge of

---

<sup>\*</sup> We gratefully acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG) through the Collaborative Research Centre SFB/TR 8 Spatial Cognition, Project I3-[SharC].

control. For this to be effective, user and robot must be able to share spatial knowledge and goal information. Since users may be technically naive, and manual interfacing may not be feasible, information should be exchanged through natural modalities. Natural language dialogues have long been acknowledged as a potentially fruitful modality in human-machine interfaces. However, practical connection of natural language and spatial information is a non-trivial task. Representations of space are often created with little concern for language oriented control, and the mapping between language, space and action is a formidable problem.

The study of cognitive control systems has led to a number of proposed architectures. On one hand, the past decade has seen the development of hybrid Robot Control Architectures [1, 2] that provide an autonomous system with multiple layers of intelligence in order to cope with both deliberative and reactive requirements. On the other hand, research based in the discourse community has led to the development of intelligent conversational systems such as Allen’s TRIPS [3], and Lemon’s WITAS [4], providing complex discourse models to mediate control of an autonomous system between user and automaton.

In this paper we give a detailed description of the *SharC Cognitive Control Architecture*, and show its relevance in addressing so-called *Shared Control Problems* in the navigating robots domain. We begin in Section 2 by reviewing the state of the art in dialogue-capable navigating robots. This is followed by Section 3 which addresses *Shared Control Systems*, discussing four classes of problems related to verbal interaction with spatially aware systems. Section 4 then introduces the SharC architecture, as a general framework for bridging language systems and spatial representations. We discuss the architecture both from a conceptual and an implementation perspective, giving outlines of relevant components. Since the matching of a natural language utterance to an interpretation is of utmost importance to user-friendly robotics, Section 5 examines our natural language understanding model in more detail – showing how a user utterance can be mapped to a natural language independent action and spatial representation. This is followed by summary and a discussion of future work.

## 2 Related Work

Work on the linguistic control of robots dates back to the pioneering research of Shakey [5] and SHRDLU [6]. More recent work has either focused on the integration of natural language components with sophisticated service robot design [7–9, 1], or have developed models of dialogue management and control from a more theoretic perspective [10, 11, 4].

One of the more recent examples of complete language enabled service robots is Jijo-2 from Matsui et. al. [7]. Jijo-2 was an office assistant robot designed for the Japanese market. As such, Jijo-2’s development was more concerned with the integration of a complete system than with the improvement of any one piece of language technology. Notable characteristics of its language systems include: inference of under-specified referents and zero pronouns using the attentional

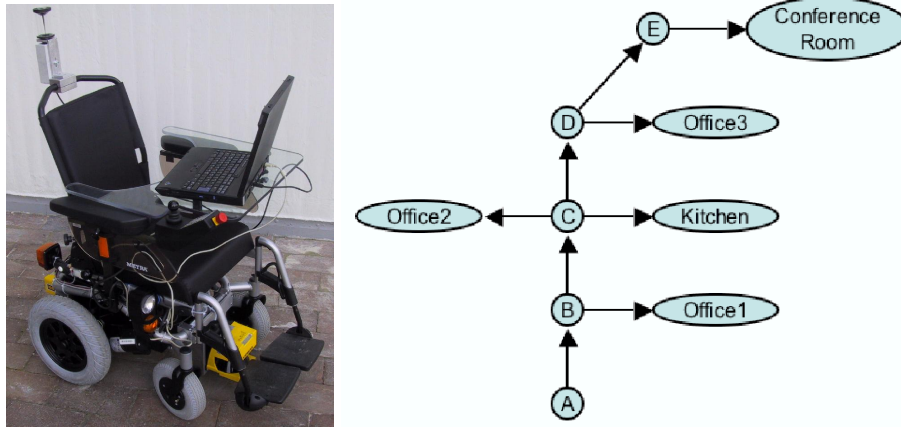
states; context-sensitive construction of semantic frames from fragmented utterances; a modular speech recogniser that swapped recognition vocabularies at runtime. A state based *Dialogue Manager* was used to control both dialogue progress and overall robot action. Dialogue decisions were partially decided upon by the output of a frame like language analyser. The language analyser and speech recognition systems were based around a restrictive grammar; while this improved speech recognition accuracy, it limited overall robustness to partial and ungrammatical utterances which are common in natural spoken language. Also, there was no attempt to recover or process ambiguous utterances through dialogues or other processes.

The relationship between verbal movement commands and action abstractions in human-robot interaction has recently been studied by Bugmann et. al. [10, 12]. The Instruction Based Learning for Mobile Robots (IBL) project has used a miniature remote-brained robot in a model town to build a model of corpus-based instructions for mobile robots. Research has focused on the production of a number of primitive movement actions that map directly to phrases used by humans to direct robots in spatial environments. While an important part of IBL’s research has been on the detection and analysis of errors in the analysis process, treatment of how to deal with such errors is only lightly considered, with a mention of the probable importance of confirmation dialogues being noted without any concise treatment of how such a dialogue confirmation would be implemented.

### 3 Shared Control Problems in Spatially Aware Assistants

In today’s shared-control systems, such as intelligent service robots, human operators are no longer continuously in control of the technical system. Instead, they monitor the behaviour of the automation, making command level decisions, and sometimes taking over control of the system in unforeseen critical situations. In our navigating robots domain, natural language communication is an important modality in maintaining shared control (e.g., [13, 14]). The shared control of navigating robots will present a distinct range of problems related to the exchange of spatial knowledge with intelligent robots. In this section we discuss a number of shared control problem types that can occur in dialogue based spatially aware systems. The categories presented are not intended to be a complete taxonomy of shared control issues, but serve instead to illustrate the types of problems faced in this domain.

Our discussions here, and approaches presented later in the paper, are made with reference to our chief experimental scenario: the shared control of Rolland, the Bremen Autonomous Wheelchair [15]. Rolland, depicted in Figure 1, makes use of laser range finding sensors to construct spatial representations that can later be used in shared control interactions between user and wheelchair. While such capabilities are modest, they are sufficient for the exploration of shared control problems such as those presented in the rest of this section.



**Fig. 1.** (a) Rolland – the autonomous wheelchair (b) A user-level route graph – internal space representation used by Rolland (nodes and edges denote decision points and route segments respectively)

### 3.1 Linguistic Ambiguity

One of the well-known problems in human language technology is so-called *linguistic ambiguity*: some natural language inputs cannot be allocated a unique semantic representation. In the context of human-robot communication for navigation, such problems must be solved before later stages of processing can reasonably be addressed. Linguistic ambiguities may occur at any level of natural language processing, the most prominent examples being attachment of prepositional phrases and quantifier scopes. The latter can be modelled using minimal recursion semantics [16], however other ambiguities are more difficult to handle. As an example, consider the following short dialogue of an ambiguity at the lexical level between the user and Rolland when approaching a crossroad:

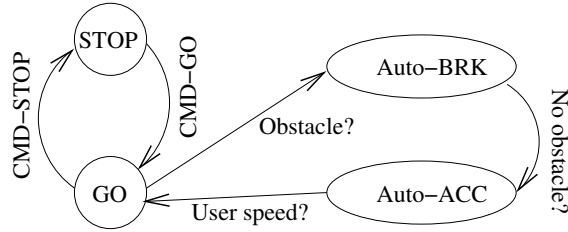
|          |             |
|----------|-------------|
| Rolland: | Drive left? |
| User:    | Right!      |

The user's answer is ambiguous, it could either be interpreted to acknowledge or to correct the robot's suggestion; hence, there are two conflicting interpretations which cannot be resolved by linguistic knowledge only. Consequently, for this kind of ambiguity, a clarification dialogue must be raised in order to negotiate the user's intention.

### 3.2 Mode Confusion

For several years, research in the human factors and aviation psychology communities has focused on the issue of *mode awareness* and *mode confusion* (see for example [17–19]). A mode represents a set of system behaviours defined by

transitions between system states. Mode confusions occur if the human operator loses track of the mode transitions performed by the automation. Consider a scenario in which Rolland is driving down a corridor when a person suddenly steps into its path. Upon seeing the colleague, the user may decide to stop and talk for a moment; hence uttering “please halt”. However, unbeknownst to the user, Rolland did not actually respond to the user’s utterance, but decided to come to a stop of its own accord – having viewed the colleague as an obstacle. Thus, when the colleague moves on, the user will be surprised that the wheelchair continues on its path, despite the user not having instructed it to continue.



**Fig. 2.** A simple model of wheelchair behaviour

Figure 2 presents an abstracted model of Rolland automation. In normal circumstances, a stop command, but not a halt command, will cause the robot to move to state STOP that can only be left through a direct command CMD-GO to proceed. Conversely, the detection on an obstacle during the movement state will cause the automation to move to the brake state (Auto-BRK), which will automatically be exited and Rolland will accelerated in state Auto-ACC when the obstacle is no longer present.

In recent approaches [20, 21], formal methods are used to systematically detect and avert these conflict situations in which the human operator assumes the technical system to be in a different situation than it actually is. After detecting the general categories of situations where such ‘mismatches’ may occur, audio or visual feedback from the system can be used to notify the user when state initiated behaviours have occurred, and indeed to confirm when a command from the user has been recognised.

Formalisation of machine models has made extensive progress in the last 10 years. There are a number of successful examples of applying formal methods for handling complex industrial systems (e.g., [22]). The challenge here is the construction of the user’s mental models, discussion of which goes beyond the scope of the present paper.

### 3.3 Spatial Representation Disparity

Routes are a concept commonly encountered when dealing with human spatial navigation. When planning a trip from one point to another, a number of in-

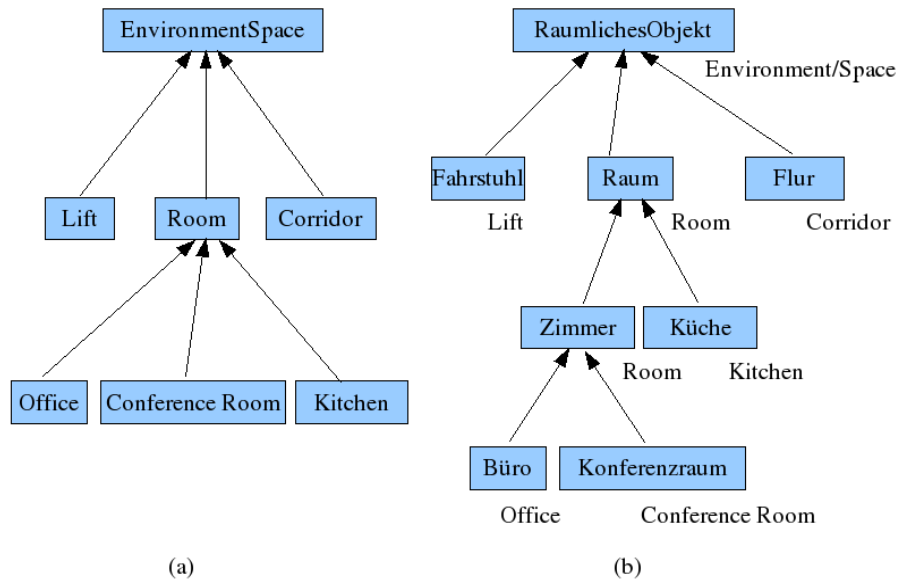
intermediate points can usually be identified, thus allowing the total trip to be broken down into a number of different segments. The routes we take are often physically defined in our environment, e.g. by determining the sequence of cities and exits when driving in a particular direction. The set of navigational strategies found in artificial moving agents, such as service robots, mirrors the complexity of those employed by human beings. The Route Graph ([23, 24]) is a simple model describing key elements for route based navigation as part of an agents' general knowledge, in which a *route* is a concatenation of *route segments* from one place to another; and a *place* is a tactical decision point. Figure 1 (b) gives a part of a route graph for an office building. Such a representation can be used as the internal representation for a navigating robot such as an autonomous wheelchair. A to E are positions in a corridor.

While the route graph shown above can be viewed as the robot's internal representation of its environment, users will often have their own internal representation, or mental model, of their surroundings. If the user is mistaken in the understanding of the office space layout, or, indeed, if the robot's representation is not accurate, then a command issued by the user may conflict with what the robot 'believes to be true'. To illustrate this, suppose that, at position (D) in Figure 1 B, the user orders Rolland to "Follow the corridor, and turn left at the end", then clearly this is not actually possible based on what Rolland believes. Further dialogue is necessary to resolve such conflicts in order to coordinate the user's and the robot's representation of the environment.

### 3.4 Spatial Concept Disparity

The ontological modelling of space is considered to be particularly necessary for facilitating qualitative spatial reasoning in general, and for ontologically grounding the spatial expressions found in natural language. For example, in the so-called 'perspectivalist' approach of Smith and colleagues (e.g., [25]), objects, events and locations such as 'rooms', 'corridors', 'robot movements' and 'at the end of the corridor' are considered as real as quantum flows. Cognitively, there is little doubt that the kinds of everyday objects, events and places found in commonsense views of the world play an important role in all aspects of cognitive behaviour. From the robotics point of view, well formed ontologies of conceptual knowledge provide the robot with a common-sense viewpoint that can be related to other artificial or human agents. Inevitably, there are mismatches between the conceptual information held by the robotic agent, and information held by the user. Identification of such 'mismatches' is undoubtedly vital to developing a robot which does what the user expects it to do.

To illustrate this, we will consider an example involving the perceived categorisation of real world objects. Such an example is a very simplistic view of the role of ontology in cognitive robotics, but serves to illustrate the issues involved. Figure 3 (a) shows the common-sense taxonomy that might be used directly or indirectly by a robot in its representation of and reasoning about the outside world. Figure 3 (b) depicts a typical taxonomy of rooms for a German speaking user. On examination we see that there is a fundamental difference between the



**Fig. 3.** Two contrasting views of commonsense reality

classification of spatial areas as viewed by a German user, typical in a domain ontology.

To see the effect of such a conceptual difference, consider the simple example where a user at position (A) of the route graph (see Figure 1 (b) ) instructs Rolland with “Ich möchte zum dritten Zimmer rechts” (meaning “I’d like to go to the third room on the right”). If the user’s mental model is that of Figure 3 (b) then it is likely that she/he intended to go to the conference room. However, if Rolland has a flat conceptualisation such as that shown in 3 (a), then Rolland should decide to drive the user to the kitchen.

As another example we once again examine the route graph of Figure 1 (b) and consider what it means if an English speaking user were to say “go right” at point (D). At D the corridor both veers to the right and has a clear right turn. Thus, the user’s utterance can be seen as either underspecification, or that the user has a very clear understanding that “go right” means that the robot should veer to the right. In either case, pragmatics dictate that the analysis of the user’s command should at least identify that there are two possible interpretations to this utterance (at least as far as the robot is concerned), and that further dialogue, task oriented reasoning, or dialogue history is required to explicitly determine the underlying intent of the user. To that extent, this situation can actually be viewed as an advanced form of linguistic ambiguity where the utterance, although specifying a clear meaning (turning right) cannot

be mapped to an action, since multiple interpretations are possible at that point. Once again, a dialogue can be initiated to clear up such a misunderstanding.

## 4 The SharC Cognitive Control Architecture

In the above examples, we saw that clarification dialogues and contextual knowledge can be used to circumvent many shared control problems. It is therefore important that robot control systems be developed that provide a well defined cognitive interface to behaviours and sensory capabilities, thus facilitating interaction with users in a natural manner. In this section we are going to introduce the *SharC Cognitive Control Architecture* to bridge these human-robot interaction issues, with the more traditional deliberative and reactive requirements. The approach taken to the architecture's construction has been discussed elsewhere [26]. Here we will detail the components used within the architecture, showing their relevance to implemented shared control tasks such as those described above.

The SharC architecture, shown in Figure 4, splits system control amongst a number of deliberative agents. Each agent encapsulates a central component – often inherited from legacy applications) – with a Belief-Desire-Intention (BDI) abstraction. These abstractions are made in the AgentFactory Agent Programming Language (AF-APL) [27], and allow each agent to perform high-level introspective reasoning in a similar manner to hybrid architecture design. But, by distributing control amongst a number of agents, the architecture also provides robustness and scalability gains. The SharC Cognitive Control Architecture, presented here, is based on a more abstract MultiAgent Architecture for Robot Control (MARC) [28].

The SharC architecture is a high-level control architecture for Rolland, as opposed to Rolland's lower level automation architecture, which has previously been described in [15]. The automation control architecture, which addresses low-level automation and safety issues, is encapsulated in one SharC agent. Although primarily developed for the Rolland platform, our agent oriented approach will allow us to easily migrate SharC to other platforms as needed.

Figure 4 presents the SharC architecture for Rolland. Rounded blocks represent complete control agents that encapsulate a system component. Arrows between the agents show primary information flow. All information exchange is via messages rather than more tightly coupled method calls. This provides a loosely coupled distributed system which can be implemented across a number of different machines. Where possible, we have based the agents on off-the-shelf components. This code re-use approach was essential in precuring the tools for speech synthesis and recognition. However with integrating legacy components, there is always a risk that some components may not behave as expected. In such cases it is important that the overall architecture should be robust to fault. SharC's AF-APL based agent design is ideally suited to such occurrences.

The architecture is being developed for both German and English use. This bilingual requirement is facilitated with linguistic components that perform map-



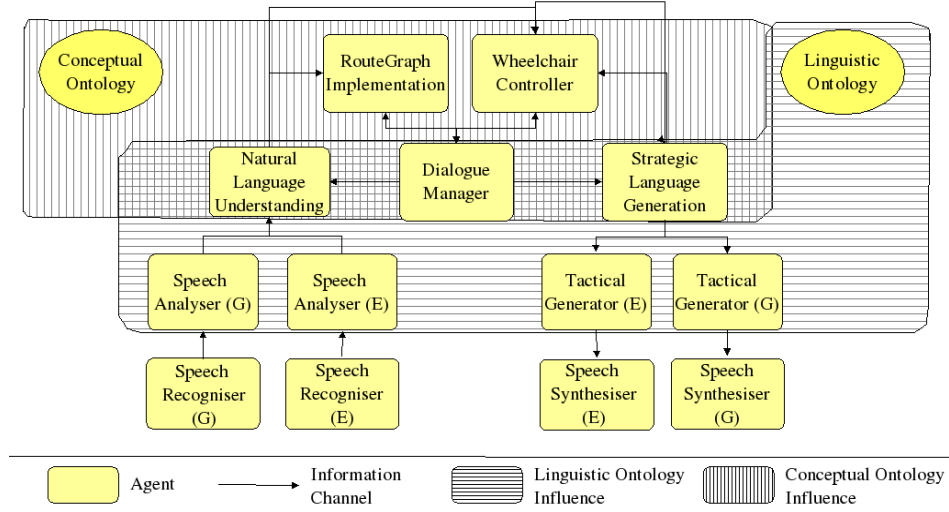


Fig. 4. The SharC Architecture for Rolland

ping from either German or English to internal representations and vice versa. Key to this mapping is the use of formally specified Linguistic and Domain Ontologies [29,30]. These two bodies of knowledge provide the agents with a common ontological viewpoint, based on which they can also reason about the environment and internal states. Two hatched regions in Figure 4 show where these Spatial and Linguistic Ontologies are principally used. The vertically hatched region depicts the conceptual ontology that provides SharC agents with a common-sense style of spatial and action knowledge. It is used in the definition of Rolland’s internal map representation, the RouteGraph. The horizontally hatched region shows the influence of the Linguistic Ontology over the SharC architecture. Concepts from the Linguistic Ontology, including the Generalized Upper Model [31], form the cornerstone of SharC’s handling of natural language. As can be seen from the ontological overlaps, the SharC architecture can be split between a natural language independent, internal representation, and a language dependent section. The task of natural language generation and understanding is to mediate between these different viewpoints. In the following we will briefly outline each of the SharC agents, as well as the relationship between them. This is followed in Section 5 with a more detailed description of our language understanding approach and its use of ontology.

#### 4.1 Natural Language Synthesis and Recognition

*Automatic Speech Recognition* (ASR) is the process of extracting lexical representations of a users’ utterances from acoustic input. Despite considerable improvements in the last decade, ASR systems still encounter many problems

including: interference by ambient or electronic noise; speaker-dependent pronunciation characteristics such as dialects; phonetic ambiguities; determination of word boundaries; unknown words; and multi-user interference (so-called Cocktail Party Effect). Despite these problems, commercially available ASR systems can provide a reasonable degree of reliability, as long as they are provided with a description of which words and combinations of words are likely to occur. For our wheelchair application, we wish to have both German and English communication available. Based on the recognition rates of available recognisers, we decided to employ the Nuance<sup>1</sup> speech recognition system for both English and German recognition.

Conversely, the *Speech Synthesis* process generates acoustic signals (audible speech) from a lexical input representation (string of text). As with ASR, there are a number of commercially available systems both for German and English. While *Concatenative Speech Systems* are often employed in telephone exchanges and public address systems, they are dependent on a vocabulary of input utterances having been created and joined together. While this approach is practical for systems based on numeric output (such as those mentioned), they are not practical for systems that require a large number of dynamic sentence constructions, particularly in the experimental domain where additions to the vocabulary are often required. For these reasons, we employ the MARY speech synthesis system [32] for both German and English synthesis.

## 4.2 Natural Language Analysis

While keyword-spotting can provide control levels acceptable to some limited voice controlled applications, the communication of task and spatial information to mobile robots requires a more sophisticated analysis approach. As mentioned before, in the highly dynamic scenario of robot navigation, humans are likely to produce ungrammatical sentences or even sentence fragments. Furthermore, while ASR systems have grown more reliable, recognition mistakes are always possible. Hence, language analysis must be robust enough to produce at least some meaningful representation of the user's intention, from which contextual resolution and confirmation dialogues can be used to further interpret meaning.

An important requirement related to analysis robustness is incrementality – allowing for immediate processing of partial input, and, thus, providing interpretations as early as possible. Incremental handling of ambiguities and inconsistencies requires that syntactic and semantic analysis be done simultaneously, or at least alternately. This approach is supported by the cognitive observation that humans integrate all available information as soon as possible (*rule-to-rule hypothesis*, [33]). Our analysis approach achieves this by integrating both syntactic and semantic information into one formalism, utilising the well-known notion of *unification* [34]. Furthermore, we are currently augmenting the formalism with a probabilistic approach inspired by Graded Unification [35], thus allowing the language analyser to achieve robustness against minor mismatches

---

<sup>1</sup> [www.nuance.com](http://www.nuance.com)

at either level. In the SharC context, natural language processing will be done in English and German, hence the parsing algorithm must be flexible enough to deal with different grammars.

Taking into account the requirements of incrementality and lexicalisation, we opted for a variant of *Combinatory Categorical Grammar* (CCG) [36]. The key concept of CCG is to regard linguistic entities as functions, and to provide a formal model of how these functions can be applied and composed with each other, thereby determining the syntactic roles of phrases. The same formalism can be easily extended to cover semantic roles in the same way, thus enabling the desired incremental integration of different information. Concerning expressivity, CCG falls into the class of mildly context-sensitive grammars which seem to be sufficient to capture any natural language construct while still being tractable [37]. CCG defines a set of abstract reduction rules together with a lexicon of possibly complex categories for each word. In our grammar, the set of rules used for processing has been slightly altered in order to include some phenomena of German word order [38].<sup>2</sup>

At the semantics level, the implications of all sorts of ambiguities must be dealt with correctly, which calls for information about how syntactic and semantic categories behave and combine with each other. This leads to the notion of a linguistic ontology that defines the interactions of linguistic entities. The semantic side of processing is aligned to the concepts defined in a linguistic ontology. Because of the alternative use of German and English, a language-independent approach is needed, that allows interfacing between natural language components and domain ontologies. We adopt the Generalized Upper Model [39] that meets these requirements. During analysis, a semantic structure is constructed according to the *Sentence Planning Language* (SPL) [40] which was originally designed for language generation using the Generalized Upper Model, but is equally suitable as a representation language for communication between natural language components in general. Furthermore, to account for complex quantifications, we are currently considering moving this representation the way of minimal recursion semantics [16]. Ambiguities that persist after the analysis of a sentence is complete are delegated to the *Natural Language Understanding* agent in order to consider contextualisation for resolution, or to have a clarification dialogue raised.

While natural language analysis provides a semantic representation of a user's input, it cannot by itself decide on actions to be performed; nor can it decide whether sufficient or appropriate information has been provided to initiate a particular action. Within the SharC architecture, the *Natural Language Understanding* component processes the output of language analysis to build complete queries for particular domain components. The Semantic Interpreter makes use of a slot-filling strategy to build requests to domain components including the *RouteGraph* and *Wheelchair Controller*. The Natural Language Understanding, making use of the *Dialogue Manager* for reference management and dialogue

---

<sup>2</sup> This work is partially based on results contributed by the Collaborative Research Centre "Situating Artificial Communicators" at the University of Bielefeld.

history, can initiate clarification dialogues when necessary. In Section 5, the interaction between the language analysis, understanding and domain components is discussed in greater detail with an example from the wheelchair control domain.

### 4.3 Natural Language Generation

In order to produce dialogue contributions, including spatial descriptions and announcements to a user of potential problems in interpretation, and so on, we have decided also to employ general purpose natural language generation technology. The functionality of these components involves mapping between a semantic specification and annotated text strings suitable as input to the adopted speech synthesis component. The functionality aimed at within our system demands flexible solutions to the generation task for two principal reasons.

First, generated dialogue utterances must be appropriate for their particular contexts of use within the unfolding dialogue between user and robotic system. This cannot be achieved with canned text or restricted template generation, since an essential property of naturally produced dialogue-contributions is that they show in their design just how an interlocutor's statement is being interpreted. This is one of the main methods by which smooth dialogic interaction is achieved: possible misinterpretations can be signalled very early and the corresponding dialogue partner can provide further information or corrected information in order to keep the interaction on track. Building in such implicit feedback signals of interpretations can only be done with a fully flexible generation component that has extensive grammatical competence in the languages being produced.

Second, we are also exploring empirical methods by which particular selections of lexical items and grammatical constructions can influence the linguistic behaviour of a user so as to channel that behaviour along predictable lines. This channelling involves both purely linguistic properties, such as keeping the forms of language used within the interpretative capabilities of the system, and the selection of spatial perspectives. The latter is again potentially of significant value for improving the perceived robustness and utility of the complete system. If the user is implicitly directed towards spatial description strategies that align well with the functionalities supported by the perceptive systems of the robot, then the result is an increase in successful interactions. Previous work [41] has shown that mismatches in the expectations of the user concerning spatial strategies and the actual spatial mechanisms employed by a robot can cause severe communication problems and task failures. The flexible generation of interactionally appropriate utterances tailored specifically to avoiding such misperceptions promises to improve on this substantially.

To meet these requirements we are employing a general purpose generation system that already has reached a robust and mature stage of development, the KPML multilingual generation system developed over the past 10 years and currently maintained at Bremen [42]. This system also has the advantage that its semantic input has strong semantic typing and these types are drawn exclusively from the linguistic ontology that we are employing within the SharC architecture

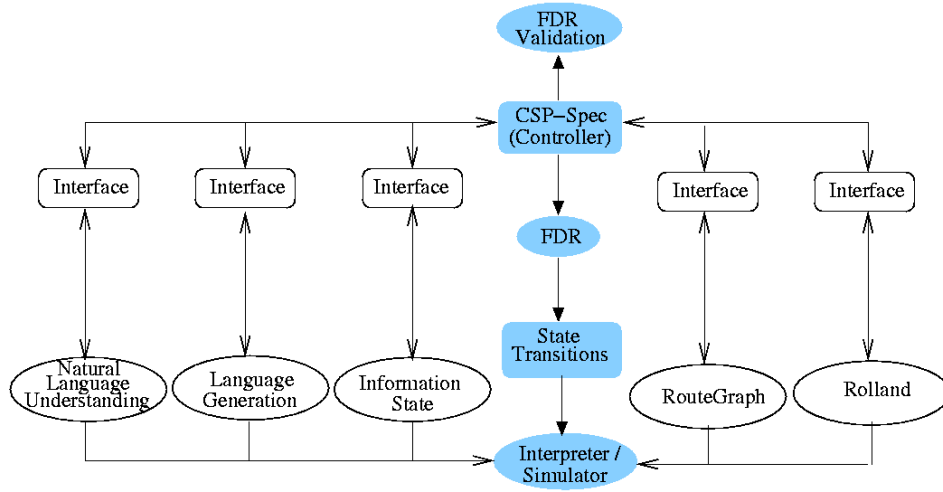
overall, the Generalized Upper Model. The general purpose orientation of the system and the grammars that it supports also allows extensive control of the precise phrasing adopted for any particular semantic content. This is essential for ensuring the precise interactional appropriateness of the dialogue contributions produced.

The basis of the generation process within KPML is a deterministic traversal of grammatical resources written within the linguistic framework of systemic-functional linguistics [43], a linguistic theory with a long tradition both of computational instantiation and of attending to interactional and other non-propositional aspects of meaning [44]. Substantial computational grammars for several languages are available within this framework, making it a natural choice for an initially bilingual system such as is envisaged for SharC. Systemic-functional grammars are used extensively in natural language generation because of their orientation to organising linguistic resources around communicative function and intentions rather than autonomous structure. The deterministic implementation makes the use even of very large grammars unproblematic. Even relatively long (for dialogue) utterances, such as running descriptions of scenes or way finding explanations, are readily produced in real-time, once the semantic specifications are available. The output of the generation process is a sentence structure that can also include functionally-motivated annotations for guiding international choice by the speech synthesis component—which is another prerequisite for achieving natural dialogic interaction [45].

#### 4.4 Dialogue Management

Mode Confusions, such as those introduced in Section 3.2, are a considerable issue in the deployment of service robotics. The dialogue manager described below enables us to construct a formal model of a robotic or complex service system at a high level. This model can then be compared and contrasted against the user’s perceived model of the system’s operation. Within the context of natural language interaction, the Dialogue Manager is responsible for controlling the flow of dialogue between user and robot – deciding at a high level what questions should be asked of the user, what information is required, and which information should be passed onto domain specific components such as the *RouteGraph* or *Wheelchair Controller*. The Dialogue Manager is also responsible for maintaining a history of user and robot dialogue acts, thus providing an essential resource for the evaluation of otherwise ambiguous references.

Since Shared Control systems are often embedded in safety-critical devices, such as aircraft, power plants or service robots, it is crucial that safety-critical requirements be accounted for in Dialogue Manager design. Experience with safety-critical systems shows that the quality of such systems can be significantly improved through the application of formal methods. Since dialogue management plays a central role in the shared-control of the whole system, we have chosen to apply the well developed method *Communicating Sequential Processes* (CSP) to model the component. In fact, in [21] CSP is used to model and detect the



**Fig. 5.** A formal framework for dialogue management

mode confusion problems discussed in Section 3.2. Once detected, a mode confusion situation can be avoided through generating some proper dialogues by the dialogue manager.

CSP has been designed to describe systems of cooperating processes such as reactive systems. In general, processes proceed by engaging into events, where synchronisation of such events is required. This, rather than assignments to shared state variables, is the fundamental means of interaction between agents. CSP can be seen as a very abstract, highly readable and easily maintainable language to specify finite state automata. It is not a very strong specification language, indeed it lacks the ability of a more abstract temporal logic to specify liveness properties, but it is executable and comes with good support, there is extensive experience with it (e.g., [21, 46, 47, 22]). Nevertheless, our approach is not restricted to CSP. Other formal methods such as SPIN [48], Kronos [49, 50], SMV [51] and so forth, could also be applied.

A formal method based framework for implementing theories of information state is shown schematically in Figure 5. This framework splits control over a number of different elements and interfaces, including: the Dialogue Management control module that incorporates a CSP specification and a validation tool to perform verification using the model-checker FDR; a set of interfaces for integrating and communicating with information states, natural language input and output, and other domain specific components, e.g. RouteGraph [23] and wheelchair control for Rolland; and, finally, a state machine module including a interpreter for state transitions with development tools including a simulator using the graphical editor *da Vinci* [52] to view state transition graphs (generated by FDR) dynamically.

## 4.5 Domain Components

The components outlined above constitute the linguistic elements of the SharC architecture. These, along with the Generalized Upper Model provide a general purpose dialogue framework for natural language based shared control systems. In principle, this general design can be applied to a number of different application domains. Here we will discuss the domain components used for the Rolland navigation demonstrator. This is important in illustrating where spatial concepts are relevant to the general architecture implementation.

As indicated earlier, the SharC architecture is being primarily developed for the autonomous wheelchair Rolland (depicted in Figure 1), but is intended for use beyond this single experimental platform. Rolland’s low-level control is encapsulated within an *Wheelchair Controller* agent. The automation control regulates low-level control issues including basic trajectory management, movement and obstacle-avoidance behaviours, and low-level safety-critical issues. The automation control’s interface is defined via a set of primitive movement actions.

The *Wheelchair Controller* agent is not responsible for maintaining a representation of the robot’s behaviour. Instead, such representations are maintained within the RouteGraph component. This component, based broadly on the Route Graph described in Section 3.3 and [24], provides a spatial representation along with query and information update algorithms. The component is a dynamic data structure which may be updated and queried either by the *Wheelchair Controller* or the user through the natural language interface.

## 5 Interpreting Navigational Instructions

The control architecture presented above provides a general framework for the mapping of linguistic representations to domain specific concepts. In this section we further detail the relationship between linguistic and domain knowledge in the architecture, and show how this yields a generalised model of natural language understanding and action invocation. While our approach is illustrated via the Rolland use scenario, i.e., spatially aware service robots, the approach can be generalised to other application domains.

Rather than applying ad-hoc representations of spatial concepts, or, alternatively, spatial representations influenced solely by linguistic concerns, our approach is to apply well defined conceptual ontologies in the creation and description of spatial representations. It is our hope that by adopting this rigorous approach, we can simplify the process of mapping from natural language to internal representation, and vice versa.

In recent years a number of so-called “upper” ontologies have been developed, most notably the OpenCyc upper ontology [53], the Suggested Upper Merged Ontology (SUMO) [54], and the Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [55]. Upper ontologies are an attempt to formally specify and constrain knowledge at the *ontological level* of knowledge-based systems [56]. Here principles are set out for the kinds of formal properties that

we demand of ‘concepts’ and the kinds of formal properties that we demand of ‘roles’ – this is then intended to lead to more consistent modelling decisions being taken for domains as a whole and, as a consequence, more reliable and re-usable representations. This level of representation is no longer arbitrary: ontology is very much constrained. Using these upper ontologies as a formal basis, domain ontologies can then be developed for multiple software systems, e.g., autonomous agents, to take advantage of shared knowledge to negotiate in some context.

Within the context of navigating robots, domain knowledge is most often required to model representation of space, motion, and action. As outlined in [57], the cognitive robotics literature is now becoming more endowed with formally modeled representations of spatial knowledge. The basic calculi of regional representation [58] have recently been augmented with more focused models of route type information [24] and spatial relationships [59]. While such formal representations are currently heterogeneous in nature, there is a drive towards a unified approach using upper spatial ontologies, below which more specified domain ontologies can be developed for individual mobile robots.

Such a well defined representation of spatial concepts should then make it possible to implement more concrete and extensible systems that relate language to spatial knowledge. To illustrate the application of this approach within the SharC architecture, Figure 6 schematically depicts the composition of ontologies within the SharC architecture, along with a number of components and knowledge sources that are constrained by those ontologies. In general, we see the linguistic ontology can actually be partitioned into two overlapping linguistic ontologies for English and German respectively; in addition, we see that the conceptual ontology can be viewed as a construction from three distinct ontologies, all constrained by a formal upper ontology. The details of the relationships between these ontologies is discussed further in [24, 57]; here, we describe the general approach and explicit flow of information. Consider the processing of the following request from a user while the wheelchair is situated at point (D) in Figure 1 (b) : *go right*. Assuming clean language recognition, an analyser, making use of a suitable linguistic ontology, can produce a shallow semantic representation such as the following<sup>3</sup>:

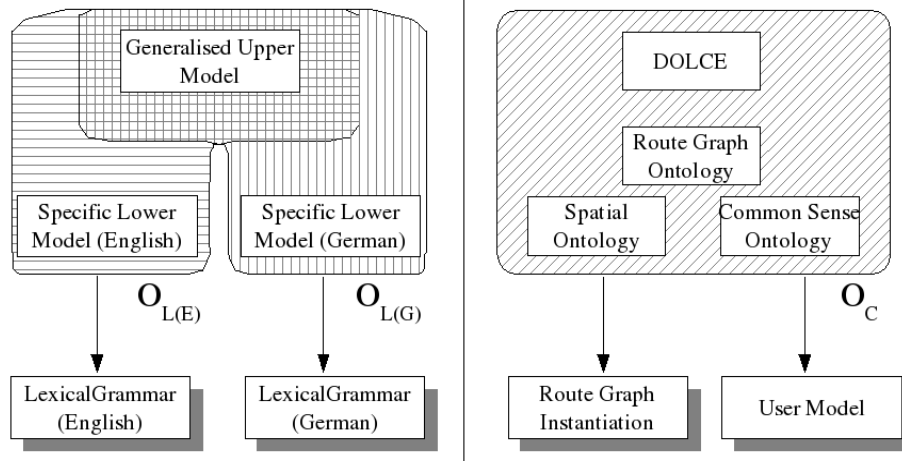
```
[syn: [type: sentence, mood: imperative],
  sem: [type: motion-process,
        actor: hearer,
        spatial-direction: right,
        speechact: command
      ]
]
```

where **syn** and **sem** denote syntactic and semantic information about the utterance respectively. Here, the linguistic ontology defines the semantic types and

---

<sup>3</sup> For simplicity, we illustrated the example with a relatively neutral structure, rather than using the actual, more complex, analysis output – the details of which are not relevant here.





**Fig. 6.** Relationships between conceptual information

terms, including: *motion-process*, and *command* and *right*. Although analysis provides an initial semantic model of user input, it does nothing to determine the effect of the user input on the complete system. To do this, the semantic input must be examined with respect to constraints and requirements of domain specific components such as the route graph. This mapping and examination of user input is vital to the task of relating to coherent spatial concepts, and is generally the responsibility of natural language understanding.

The natural language understander, along with the dialogue manager, interact with domain components through abstract conceptual interfaces. These interfaces, adhering to the types, relations, and predicates defined in the conceptual ontology provide a consistent and natural language independent abstraction of low level functionality [27]. Table 1 presents an excerpt from the Route Graphs interface. The interface defines a number of different query types along with valid parameters for these queries; parameters, along with implicit return values, are statements of first order knowledge. It should be noted that while such abstract protocols define an conceptual interface into a component, they do not by necessity constrain the component implementation to be based on the conceptual ontology.

The dialogue manager, in conjunction with the natural language understander, performs contextual resolution, and also implements a procedural knowledge based frame filling system. Queries to the RouteGraph or Robot Controller, composed under the constraints of the conceptual ontology, can then be passed to the domain components as appropriate.

In our example, the resolution of ‘hearer’ to Rolland is relatively straightforward, whereas the results of a *go right* request are entirely dependent on the spatial reasoning ontologies adopted for the route graph. For example, one ap-

**Table 1.** A sample of the conceptual interface used by dialogue management and natural language understanding to interact with the route graph component. All parameters are formalised as first order sentences

| Action             | Parameters           | Description  |
|--------------------|----------------------|--|
| isKnownLocation    | place                | Determine if place is known  |
| addPlanInformation | place<br>description | Add annotation information about place                             |
| findRoute          | place<br>place       | Determine the Route from one place to another                      |
| isValidRoute       | route                | Determine whether route is valid with respect to current knowledge |

proach would be to decide that all corridors leading from  $15^\circ$  to  $75^\circ$  relative the forward axis of the wheelchair’s motion correspond to “right”. In such a case, dialogue based resolution would most likely be required to confirm the user’s intentions. If other domain information – or user modelling – cannot be used to determine which of the two possibilities is most accurate, then the *Dialogue Manager* must initiate a clarification dialogue with the user.

Our approach is intended to produce a degree of natural language independent human-robot interaction, thus augmenting a future localisation process. Such an approach is in contrast to either a *Canned Speech* approach, which is most common in the localisation industry, or indeed the development of internal robotic representations which are directly linked to a specific natural language; this so called *Corpus Based Robotics* approach of Bugmann et al [10] may however prove fruitful in the provision of a corpus of instruction types that are often used – to be incorporated in the production of a natural language independent internal representation of action.

## 6 Summary and Future Work

In this paper, we reported on our initial investigations into the problems of verbally controlled navigational robots. In particular, we presented a clarification of the types of Shared Control Problems that can be encountered in our application domain. To address these problems we have presented a partially implemented framework for testing dialogue based control methods. Although initially designed for a semi-autonomous wheelchair, the ontology-centric, agent-oriented nature of the architecture should make application to other hardware and domain examples feasible. Internally, this architecture has made extensive use of state-of-the-art language technology components to implement a clean separation between the robot’s language interface, and the representations which underpin its internal reasoning systems.

In future work, we will be applying the architectural framework to investigate many shared control problems, looking in particular at how formal ontology definitions can be used to identify and solve ambiguities on-line, and then using dialogues with the user to solve issues that could not be handled otherwise. In immediate work, we will be investigating the direct translation of route instructions to a suitable representation for the robot's own spatial representations.

## References

1. Konolige, K., Myers, K.L., Ruspini, E.H., Saffiotti, A.: The Saphira Architecture: A Design for Autonomy. *Journal of experimental & theoretical artificial intelligence: JETAI* **9** (1997) 215–235
2. Peters II, R.A., Wilkes, D., Gaines, D., Kawamura, K.: A software agent based control system for human-robot interaction. In: *Proc Second International Symposium on Humanoid Robotics*, Tokyo, Japan (1999)
3. Allen, J., Ferguson, G., Stent, A.: An architecture for more realistic conversational systems. In: *Proceedings of Intelligent User Interfaces 2001 (IUI-01)*, Sante Fe, New Mexico, Association for Computing Machinery (2001) 1–8
4. Lemon, O., Gruenstein, A., Peters, S.: Collaborative activities and multi-tasking in dialogue systems. *Traitement Automatique des Langues (TAL)* **43** (2002) 131–154 Special issue on dialogue.
5. Nilsson, N.J.: A mobile automaton: An application of artificial intelligence techniques. In Walker, D.E., Norton, L.M., eds.: *Proceedings of the 1st International Joint Conference on Artificial Intelligence*, Washington, D. C., William Kaufmann (1969) 509–520
6. Winograd, T.: *Understanding Natural Languages*. 2 edn. Academic Press, New York (1976)
7. Matsui, T., Asoh, H., Fry, J., Motomura, Y., Asano, F., Kurita, T., Hara, I., Otsu, N.: Integrated Natural Spoken Dialogue System of Jijo-2 Mobile Robot for Office Services. In: *AAAI/IAAI*. (1999) 621–627
8. Bischoff, R., Jain, T.: Natural Communication and Interaction with Humanoid Robots. In: *Proceedings of The Second International Symposium on Humanoid Robots*, Tokyo, Japan (1999)
9. Roy, N., Pineau, J., Thrun, S.: Spoken Dialog Management for Robots. In: *Proceedings of the ACL 2000*. (2000)
10. Bugmann, G., Klein, E., Lauria, S., Kyriacou, T.: Corpus-Based Robotics: A Route Instruction Example. In: *Proceedings of IAS-8*. (2004)
11. Porzel, R., Gurevych, I.: Towards context adaptive utterance interpretation. In: *Proceedings of the 3rd SIGDial Workshop on Discourse and Dialogue*. Association for Computational Linguistics, Philadelphia. (2002)
12. Bugmann, G.: Challenges in Verbal Instruction of Domestic Robots. In: *Proceeding of the ASER'03, the 1st International Workshop on Advances in Service Robotics*. (2003)
13. Stopp, E., Blocher, A.: Spatial information in instructions and questions to an autonomous system. Technical report, Universität des Saarlandes (1997)
14. Fong, T., Thorpe, C., Baur, C.: Collaboration, dialogue, and human-robot interaction. In: *10th Int'l Symp. of Robotics Research*. (2001)
15. Lankenau, A., Röfer, T.: A versatile and safe mobility assistant. *IEEE Robotics and Automation Magazine* **7** (2001) 29 – 37

16. Copestake, A., Flickinger, D.P., Sag, I.A.: Minimal recursion semantics: An introduction. Manuscript, Stanford University: CSLI (1999)
17. Starter, N., Woods, D.: How in the world did we ever get into that mode? mode error and awareness in supervisory control. **37(1)** (1995) 5–19
18. Leveson, N., et al.: Analyzing software specifications for mode confusion potential. In Johnson, C.W., ed.: Workshop on Human Error and System Development. Volume 97-2 of GAAG-TR., Glasgow Accident Analysis Group (1997) 132–146
19. Degani, A., Heymann, M.: Pilot-autopilot interaction: A formal perspective. In Abbott, K., S.J.J., Boy, G., eds.: International Conference on Human-Computer Interaction in Aeronautics: HCI-Aero 2000. (2000) 157–168
20. Rushby, J.: Using model checking to help discover mode confusions and other automation surprises. **75(2)** (2002) 167–177
21. Brederke, J., Lankenau, A.: A rigorous view of mode confusion. In: SAFECOMP 2002, 21st Int. Conf. on Computer Safety, Reliability and Security. Volume 2434 of LNCS., Springer Verlag (2002)
22. Shi, H., Peleska, J., Kouvaras, M.: Combining methods for the analysis of a fault-tolerant system. In: Pacific Rim International Symposium on Dependable Computing (PRDC), IEEE Computer Society (1999) 135–142
23. Werner, S., Krieg-Brückner, B., Herrmann, T.: Modelling navigational knowledge by route graphs. In Freksa, C., H.C., Wender, K., eds.: Spatial Cognition II. Volume 1849 of Lecture Notes in Artificial Intelligence., Springer Verlag (2000) 295–317
24. Krieg-Brückner, B., Frese, U., Lüttich, K., Mandel, C., Mossakowski, T., Ross, R.J.: Specification of route graphs via an ontology. (2004) In Proceedings of Spatial Cognition 2004, Chiemsee, Germany.
25. Smith, B., Grenon, P.: The cornucopia of formal-ontological relations. (Technical report)
26. Krieg-Brückner, B., Shi, H., Ross, R.: A safe and robust approach to shared-control via dialogue. (Submitted to the Chinese Journal of Software, Beijing, China)
27. Ross, R., Collier, R., O'Hare, G.: [af-apl]: Bridging principles & practices in agent oriented languages. In: Proc. The Second International Workshop on Programming Multiagent Systems Languages and Tools (PROMAS 2004). Held at AAMAS'04, New York, USA (2004)
28. Ross, R.: MARC - applying multiagent systems to service robot control. Master's thesis, University College Dublin (2004)
29. Bateman, J., Farrar, S.: Spatial ontology baseline. SFB/TR8 internal report I1-[OntoSpace]: D2, Collaborative Research Center for Spatial Cognition, Universität Bremen, Germany (2004)
30. Mossakowski, T., Lüttich, K., Krieg-Brückner, B.: Ontologies for the semantic web in casl. Submitted to WADT 04 (2004)
31. Bateman, J.A., Kasper, R.T., Moore, J.D., Whitney, R.A.: A general organization of knowledge for natural language processing: the PENMAN upper model. Technical report, USC/Information Sciences Institute, Marina del Rey, California (1990)
32. Schröder, M., Trouvain, J.: The German text-to-speech synthesis system MARY: A tool for research, development and teaching. In: Proceedings of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis, August 29 - September 1, Perthshire, Scotland (2001) 131–136
33. Just, M., Carpenter, P.: The psychology of reading and language comprehension. Allyn and Bacon (1987)
34. Shieber, S.M.: An Introduction to Unification Based Grammar. Center for the Study of Linguistics and Information (1986)

35. Kim, A.: Graded unification: A framework for interactive processing. (1994)
36. Steedman, M.: The Syntactic Process. MIT Press (2000)
37. Joshi, A., Vijay-Shanker, Weir, D.: The convergence of mildly context-sensitive grammar formalisms. In Sells, P., Shieber, S., Wasow, T., eds.: Foundational Issues in Natural Language Processing, Cambridge, MA (1991)
38. Vierhuff, T., Hildebrandt, B., Eikmeyer, H.J.: Effiziente Verarbeitung deutscher Konstituentenstellung mit der Combinatorial Categorical Grammar. *Linguistische Berichte* **194** (2003) 213–237
39. Bateman, J.A., Henschel, R., Rinaldi, F.: Generalized upper model 2.0: documentation. Technical report, GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany (1995)
40. Kasper, R., Whitney, R.: SPL: A sentence plan language for text generation. Technical Report forthcoming, Information Sciences Institute (1989) 4676 Admiralty Way, Marina del Rey, California 90292-6695.
41. Moratz, R., Fischer, K., Tenbrink, T.: Cognitive modelling of spatial reference for human-robot interaction. *International Journal On Artificial Intelligence Tools* **10** (2001)
42. Bateman, J.A.: Enabling technology for multilingual natural language generation: the KPML development environment. *Journal of Natural Language Engineering* **3** (1997) 15–55
43. Halliday, M.A.K.: An Introduction to Functional Grammar. Edward Arnold, London (1985) (2nd. edition 1994).
44. Matthiessen, C.M.I.M., Bateman, J.A.: Text generation and systemic-functional linguistics: experiences from English and Japanese. Frances Pinter Publishers and St. Martin's Press, London and New York (1991)
45. Teich, E., Hagen, E., Grote, B., Bateman, J.A.: From communicative context to speech: integrating dialogue processing, speech production and natural language generation. *Speech Communication* **21** (1997) 73–99
46. Buth, B., Kouvaras, M., Peleska, J., Shi, H.: Deadlock analysis for a fault-tolerant system. In Johnson, M., ed.: Algebraic Methodology and Software Technology (AMAST). Volume 1349 of Lecture Notes in Computer Science., Springer Verlag (1997) 60–75
47. Buth, B., Peleska, J., Shi, H.: Livelock analysis for a fault-tolerant system. In Haeberer, A.M., ed.: Algebraic Methodology and Software Technology (AMAST). Volume 1548 of Lecture Notes in Computer Science., Springer Verlag (1998) 124–135
48. Holzmann, G.: Design and Validation of Computer Protocols. Prentice-Hall (1991)
49. Daws, C., Yovine, S.: Two examples of verification of multirate timed automated with kronos. In: 1995 IEEE Real-Time Systems Symposium., IEEE Computer Society (1995) 66–77
50. Henzinger, T., Nicollin, X., Sifakis, J., Yovine, S.: Symbolic model checking for real-time system. **111** (1994) 193–244
51. McMillan, K.: Symbolic Model Checking: An Approach to the State Explosion Problem. Kluwer Academic Publishers (1993)
52. daVinci: (<http://www.tzi.de/davinci>)
53. Cycorp: Opencyc 0.7.0. Technical report (2004)
54. Niles, I., Pease, A.: Toward a standard upper ontology. In Welty, C., Smith, B., eds.: Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001), Ogunquit, Maine, Association for Computing Machinery (2001)

55. Masolo, C., Borgo, S., Gangemi, A., Guarino, N., Oltramari, A., Schneider, L.: The WonderWeb library of foundational ontologies: preliminary report. WonderWeb Deliverable D17, ISTC-CNR, Padova, Italy (2002)
56. Guarino, N.: The ontological level. In Casati, R., Smith, B., White, G., eds.: *Philosophy and the Cognitive Sciences*. Hölder-Pichler-Tempsky, Vienna (1994)
57. Bateman, J., Farrar, S.: Modelling models of robot navigation using formal spatial ontology. (2004) In *Proceedings of Spatial Cognition 2004*, Chiemsee, Germany.
58. Randell, D., Cui, Z., Cohn, A.: A spatial logic based on regions and connection. In: *Proceedings of the 3rd. International Conference on Knowledge Representation and Reasoning*, San Mateo, Morgan Kaufmann (1992) 165–176
59. Moratz, R., Wallgrün, J.O.: Spatial reasoning about relative orientation and distance for robot exploration. In Kuhn, W., Worboys, M., Timpf, S., eds.: *Spatial Information Theory: Foundations of Geographic Information Science. Conference on Spatial Information Theory (COSIT)*. Lecture Notes in Computer Science, Springer-Verlag; D-69121 Heidelberg, Germany; <http://www.springer.de> (2003) 61–74