EU FP7 CogX
ICT-215181
May 1 2008 (52months)

DR 6.2: Adaptive dialogue strategies supporting transparency

Geert-Jan M. Kruijff, Miroslav Janiček, Ivana Kruijff-Korbayová, Hans-Ulrich Krieger, Pierre Lison, Raveesh Meena, Hendrik Zender

DFKI GmbH, Saarbrücken
⟨`gj@dfki.de`⟩

Due date of deliverable:	July 31 2010
Actual submission date:	July 24 2010
Lead partner:	DFKI
Revision:	final
Dissemination level:	PU

In Year 1, WP6 investigated how a robot could carry out a situated dialogue with a human, about items in the world it needed to learn more about. The robot was able to formulate questions against a multi-agent model of situated beliefs, what it would like to know more about. The robot was able to represent and reason with uncertainty in experience, but it was relatively fixed in the strategies it would follow to communicate with the human about resolving the uncertainty. In Year 2, WP6 investigated several issues in how to make dialogue behavior more *adaptive*. This covered several aspects: how we can achieve adaptivity in managing situated dialogue; and how we can dynamically adapt what kind of utterance is phrased in a given context, and how. Results include a suite of novel, context-sensitive methods for adapting dialogue processing, management, and planning (Task 6.3); empirically based methods for varying granularity in descriptions (Task 6.4); and, further results on context-sensitive planning and realization of utterances using varying intonation. All these methods are based on a novel probabilistic framework for representation and inference of situated, multi-agent beliefs, intentions and events, developed in WP6 and used across the board in CogX.

Executive Summary

One of the objectives of CogX is self-extension. This requires the robot to be able to actively gather information it can use to learn about the world. One of the sources of such information is dialogue. But for this to work, the robot needs to be able to establish with a human some form of mutually agreed-upon understanding, a *common ground.* The overall goal of WP6 is to develop adaptive mechanisms for situated dialogue processing, to enable a robot to establish such common ground in situated dialogue.

In Year 1, WP6 investigated how a robot could carry out a situated dialogue with a human, about items in the world it needed to learn more about. The robot was able to formulate questions against a multi-agent model of situated beliefs, indicating what it did and did not know – and what it would like to know. The robot was able to represent and reason with uncertainty in experience, but it was relatively fixed in the strategies it would follow to communicate with the human about resolving the uncertainty.

The dynamic, interactive setting of CogX in which a robot actively learns requires more than following a fixed, "universal" policy. Learning more, dynamic situations, and the changes in common ground this implies, all require the robot to *adapt* how it acts and interacts, if it is to successfully communicate with a human over time.

In Year 2, WP6 investigated several issues in how to make dialogue behavior more *adaptive.* This covered several aspects: (1) Making dialogue strategies more adaptive (Task 6.3), and (2) varying how much a robot needs to describe to be optimally transparent (Task 6.4).

Regarding the issue of adaptive dialogue management, we have begun development of a probabilistic decision model for action selection, together with a probabilistic reformulation of the processing models for continual collaborative activity we started developing in Year 1. The combination of these approaches into a single framework allows for a fine-grained modeling of how to formulate a contextually grounded dialogue act, and to adapt the strategy how this intention subsequently gets realized as a sequence of utterances. Results include a suite of novel, context-sensitive methods for adapting dialogue processing, management, and planning (Task 6.3). All these methods are based on a new probabilistic framework for situated, multi-agent models of beliefs, intentions, and events we have developed and which is used across the board in CogX.

Concerning the second issue in adaptivity, we have performed several empirical experiments to investigate how humans vary granularity in interaction with a robot, when describing objects in the kinds of small- and large-scale spatial contexts we typically encounter in CogX. These results provide the basis for methods for further development of context- and content-determination algorithms we use in content planning (Task 6.4).

Finally, in addition to our main focus on adaptivity in processing and

content determination for situated dialogue, we have obtained further results on context-sensitive planning and realization of utterances using varying intonation.

Role of situated dialogue in CogX

CogX investigates cognitive systems that self-understand and self-extend. In some of the scenarios explored within CogX such self-extension is done in a mixed-initiative, interactive fashion (e.g. the George and Dora scenarios). The robot interacts with a human, to learn more about the environment. WP6 contributes situated dialogue-based mechanisms to facilitate such interactive learning. Furthermore, WP6 explores several issues around the problems of self-understanding and self-extension in the context of dialogue processing. Dialogue comprehension and production is ultimately based in a situated, multi-agent model the robot builds up. This model captures epistemic objects like beliefs, intentions and events, in a multi-agent fashion. Such epistemic objects cover both situated and cognitive aspects, and already at this level we see forms of self-understanding and self-extension. Where this is particularly coming to the fore now in Year 2 is how these (probabilistic) models help to drive adaptation in dialogue processing itself, in how we manage what to do (selecting dialogue acts or intentions), how to do that best (strategy selection in dialogue content planning), all the way down to deciding how best to process an utterance in a given context. We thus see an important interplay between dialogue playing a supportive function in aiding self-understanding and self-extension system-wide, and how dialogue can use the same principles to adapt its own models.

Contribution to the CogX scenarios and prototypes

WP6 contributes directly to the George and Dora scenarios, in relation to work performed in WP 3 (Qualitative spatial cognition), WP 5 (Interactive continuous learning of cross-modal concepts), and WP 7 (Scenario-based integration). Adaptive dialogue management, including dialogue for clarification and verbalization, are in principle used in both scenarios. In George we illustrate the possibility for the robot to adapt how it asks about objects and properties it is uncertain about, aligning the way it takes initiative in the dialogue with the tutoring strategy the user appears to follow.

In Year 2, Dora is extended to include situated dialogue processing. We explore how a robot can use introspection of what the robot does and does not know about an area, to drive information requests to the user.

- **Robot** explores an unknown area, gradually building up rich spatial models of his environment.

- **Human** During the exploration, a human approaches the robot and asks him to find a particular object (”robot, please find the cornflakes box!”).
- **Robot** If the goal of finding the object is unclear or hasn’t been properly understood, the robot triggers additional information requests to the human (”sorry i didn’t understand you properly, did you say I should search for a cornflakes box?”), until a sufficient confidence level is reached.
- **Robot** then starts searching for the object. In the performance of its task, the robot can interact with nearby humans to retrieve additional information, such as the category of the room they are currently in (”excuse me, is this the kitchen?”).
- **Robot** Once the object has been found, the robot reports back his findings to the human, by verbalizing its newly acquired knowledge.

1 Tasks, objectives, results

1.1 Planned work

Robots, like humans, do not always know or understand everything. Situated dialogue is a means for a robot to extend or refine its knowledge about the environment. For this to work, the robot needs to be able to establish with a human some form of mutually agreed-upon understanding – they need to reach a *common ground*. The overall goal of WP6 is to develop adaptive mechanisms for situated dialogue processing, to enable a robot to establish such common ground in situated dialogue.

WP6 primarily focuses on situated dialogue for continuous learning. In continuous learning, the robot is ultimately driven by its own curiosity, rather than by extrinsic motivations. The robot builds up its own understanding of the world – its own categorizations and structures, and the ways in which it sees these instantiated in the world. While learning, the robot can solicit help from the human, to clarify, explain, or perform something. This is where transparency comes into play. The robot is acting on its own understanding, which need not be in any way similar to how a human sees the world. There is therefore a need for the robot to make clear what it is after: why the robot is requesting something from a human, what aspects of a common ground it appeals to, and how the request is related to what it does and does not know. In Year 1, the robot would follow fairly fixed "universal" strategies for trying to resolve uncertainties in understanding the world. In Year 2, WP6 investigates how to make the robot's dialogue capabilities more adaptive (Task 6.3), including the adaptation of how much information to provide in a given context (Task 6.4).

Task 6.3: Adaptive dialogue strategies *The goal is to investigate how we can use forms of reinforcement learning to adapt dialogue strategies to optimize planning content for verbalization, clarication requests and explanation on the basis of dynamic (i.e. extending, altering) categorical knowledge.*

Task 6.4: Variable granularity in content planning *The goal is to extend content planning techniques to include the use of vagueness to express properties to varying degrees of granularity.*

The intention behind Tasks 6.3 and 6.4 was to achieve more adaptive dialogue capabilities for the robot. So far, we focused on dealing with the dynamics of the contexts in which a robot acts and interacts. We looked at how these dynamics influence dialogue understanding, and production. In Year 2, we wanted to take this a few steps further. In CogX, the system *itself* is also highly dynamic. In a physical sense, surely, but also in a more cognitive sense. It is a self-extending cognitive system. What was once unknown, now becomes known. The robot is actively interacting with the world and agents therein. Through which it can learn more, *and* learn more about how *best* to learn more. Ideally, it would pick up strategies for how best to resolve uncertainties. Or, given the setting of WP6, it could learn strategies for how best to communicate with a human, under uncertainty, to help resolve uncertainty. And this is where adaptation comes in.

In §1.2 we describe how we achieved these goals.

1.2 Actual work performed

Below we succinctly describe the achievements for the individual tasks. The descriptions refer to the relevant papers and reports in the annexes, for more technical detail. In §1.3 we place these achievements in the context of the state-of-the-art.

1.2.1 Adaptativity

The goal of Task 6.3 was to develop methods for the robot to adapt its dialogue strategies: what it intends to do, how it enacts that intention, and what it communicates about. We have achieved the following:

Probabilistic situated multi-agent models *A robot needs to build up an awareness of the world around it, and the agents therein. It needs to have some idea of what can happen at some point in space and time, who believes what, and who intends to do something or already has been acting upon an intention. Uncertainty is a fundamental issue that such models need to deal with. Kruijff, Lison et al (§2.1.4, §2.1.5) describe a novel framework for formulating situated, multi-agent models of beliefs, intentions and events. The framework is based on a combination of (first-order) logic and probabilistic graphical models. It facilitates a combination of probabilistic inference using Markov Logic Networks (§2.1.4), and decidable logical reasoning (§2.1.5, §2.1.6). Its expressive power that can capture both the rich relational structure of the environment, and the uncertainty arising from the noise and incompleteness of sensory experience.*

In Year 1, we presented an initial approach to situated multi-agent belief modeling. This approach included the representation of belief content as multivariate probability distributions over ontologically rich representations. We performed inference over these distributions with a simple Bayesian network, to establish how different beliefs might be seen as correlated. It enabled us to deal with uncertainty in a principled fashion, and as such it provided an improvement over earlier ontology-based methods [12, 11].

At the same time, we could not yet exploit the rich relational structure that is inherent to the problems we are typically dealing with. Adopting

Markov Logic Networks [20], a type of statistical relational model, provides the possibility to do so. Lison et al §2.1.4 describe several applications of the framework in CogX. One application is multi-modal information fusion, now exploiting relational structure as well as ontological richness. Such fusion can be performed at arbitrary levels of abstraction, to allow for iterative belief refinement from low-level observations up to temporally smoothed, stable beliefs. Another use concerns situated reference resolution. Situated dialogue often includes expressions that refer to aspects of the world. We use inference over belief models to establish hypotheses, what aspects (i.e. beliefs about experience) a linguistic expression might be referring to. These hypotheses are probabilistic beliefs in themselves. We subsequently use available hypotheses when abductively inferring the contextually most likely interpretation for an utterance, selecting that hypothesis which contributes to establishing the best interpretation (cf. also §2.1.3).

Probabilistic models for adaptive dialogue management *The problem of uncertainty extends to interaction: to understanding, and how to act upon understanding. Lison and Kruijff (§2.1.1, §2.1.2) discuss a POMDP-based approach. They focus on the problem of action selection under uncertainty, i.e. what dialogue act or intention to adopt for continuing the dialogue. The novel aspect is that they do not assume a universal, complex policy but rather a set of smaller, modular policies that can be activated in a given context. This activation as inference is based (again) in Markov Logic Networks. The approach is integrated into the larger framework of modeling situated dialogue as continual collaborative activity, described by Kruijff and Janíček (§2.1.3). The combination of probabilistic abduction and adaptive dialogue management enables us to model how to comprehend dialogue given uncertainty, and produce forms of user-adapted dialogue.*

In Year 1, we presented a first account of how situated dialogue processing can be based on a continual model of collaborative activity. The account was based on Stone et al [25, 28], but provided several important extensions including multi-agent beliefs including uncertainty, and the possibility to revise and extend beliefs over time. At the same time, it did not allow for adaptivity, following a deterministic finite-state machine for action selection, nor could it reason over more than beliefs.

With the further developments of e.g. probabilistic situated multi-agent models, and POMDP-based action selection, we have begun work on overcoming these limitations. Lison (§2.1.1) discusses how we can formulate POMDPs over a rich state space combining dialogue models, user state, and belief models for situation awareness. To handle the high dimensionality of such action and state spaces, we developed a new mechanism for constraining the set of actions which are locally relevant in a particular situation.

This mechanism is specified using a Markov Logic Network, which allows us to exploit the rich relational structure inherent to situated dialogue. Dialogue planning is then performed on this limited set of relevant actions, and results in the selection of a new dialogue act to adopt as our next intention. Lison illustrates this on several examples from the George domain. Lison & Kruijff (§2.1.2) take this approach a step further, and consider activation of individual, modular POMDPs. Furthermore, by extending the POMDP state space with features that model particular user characteristics, e.g. observable aspects of tutoring style, we can achieve mixed-initiative behavior in the robot that tries to tailor ("optimize") interaction towards the user's style of teaching the robot. The abductive inference discussed by Kruijff & Janíček (§2.1.3) then turns the dialogue act (or intention) selected by the POMDP into a plan for realizing the intention in a contextually appropriate way relative to the situated multi-agent models the robot maintains.

Learnable controllers for adaptive dialogue processing *Dialogue processing typically applies a fixed set of processes to an input, to produce an analysis for an utterance. We have begun the development of an alternative model, preparing the grounds for Task 6.5 (Year 3). Kruijff & Krieger (§2.1.7) discuss an approach for dynamically deciding which configuration of processes should be applied to analyse an utterance. The approach is based on online learnable MDP controllers, and factored state models that capture the incrementally formed (partial) analyses at multiple levels of linguistic interpretation. Processing is directed towards a given goal state, which can be derived from a POMDP-based expectation about how a dialogue is likely to unfold. Based on how processes are known to contribute particular types of information (analyses), MDP controllers activate specific processes and form a configuration over them. The configuration specifies processes to run sequentially or concurrently, in an effort to optimize time-efficiency.*

Typically, dialogue processing adopts a model that uses a fixed set of processes that is applied to a given input, to produce an analysis. These processes may be organized as a pipeline, or in a constraint-based setup. Each process works in relative isolation (modulo the input it gets), making this type of processing is only as successful as its weakest process. The kind of incremental processing model we presented in Year 1 partly alleviates this problem, as multiple levels of analysis are unfolding in parallel. As a result processes can mutually inform or guide each other, mediated through the available partial analyses. At the same time, we are still applying a fixed set of processes, and this need not be the most efficient way of interpreting an utterance. We need to be able to parametrize what processes get applied to allow for any-time processing, up to any needed depth of linguistic information. This facilitates both a more developmental perspective, and a manner of process control that can provide for robustness and adaptivity.

In Year 2 we have begun the development of a more flexible model for managing what processes become involved in interpreting an utterance. The suggested model provides the means for online planning of process configurations, following a scheme of activation-planning-optimization that is similar to that of Lison & Kruijff (§2.1.2). The probabilistic models for activation and for the controller are learnable online. This yields interesting possibilities from the viewpoint of self-understanding and self-extension. So far, we have always considered a robot to have reasonably mature dialogue capabilities, when using interaction to learn more about the world. What if we would drop that assumption? How does language processing develop, focusing on controlling what and how is being understood, as a function of developing sensory modalities? In our approach to situated dialogue processing, we have always adopted the view that the functional understanding of language in a cognitive system reflects the level of functional (distinctive) understanding of the world. Taking this to a developing system, there is a simple hypothesis: There is an incentive to linking what you see, to a minimum understanding of what you hear. In terms of learnable controllers and models for binding language to experience, this hypothesis implies that controllers would opt for cost-efficient strategies that provide a minimally necessary and sufficient amount of linguistic information to help (categorically) distinguish what it is that you are able to experientially distinguish. If all a robot can see is colors, it would ideally understand the utterance “Look here robot this is a large blue box!” as ‘blabla blabla BLUE bla!” Initially, the robot may thus only use words as labels. As the robot is able to perceive more structure, and can thus assign more semantics, the idea is that the controllers learn to invoke processing at further levels of linguistic interpretation to provide a linguistic structure that is adequate from the viewpoint of grounding it in experience. We would like to explore these ideas further in the context of Task 6.5 in Year 3, and WP5.

1.2.2 Variable granularity

The goal of Task 6.4 was to develop methods for the robot to determine an appropriate level of information to be realized in an utterance, and, by the same token, to be able to understand such utterances given by its user.

Variable granularity for anchor-progression in situated discourse

Zender et al (§2.2.1, §2.2.2) propose methods for generating and resolving referring expressions in a situated discourse about large-scale space. They extend the work on situated resolution and generation of referring expressions presented in Year 1 to multi-utterance discourses. We present different models that determine the level of granularity of

> *spatial referring expressions, based on the way the* focus of attention *shifts along a discourse. These models are evaluated against data gathered in an empirical production experiment. In §2.2.1, Zender et al. present the production experiment. In §2.2.2, they tie the results from the production experiment together with the previous approach for bi-directional (i.e., for generation as well as resolution) context determination, and show how the proposed methods enable a dialogue system to engage in a situated dialogue about entities in a large-scale spatial environment.*

In Year 1, we considered how to appropriately refer to an object or place in the world – either locally, or in reference to large-scale spatial organization. But this considered only single utterances. Typically, dialogue provides a more dynamic way of identifying referents, gradually guiding attention and building up a context in which the referent is to be resolved. For example, when a robot is being given instructions to "go to the kitchen, take the box on the table, and bring it to the living room," it is faced with the task of resolving the referring expressions (i.e., "the kitchen," "the box on 'the table',", and "the living room" to entities in its own knowledge base. While the robot might only know one entity that satisfies the description "the kitchen," it might know of several tables or boxes that are located on tables somewhere in its environment. The interpretation of "the box on the table" as "the box on the table in the kitchen" depends on the previously given reference to "the kitchen." We call this latter reference the *attentional anchor* for the interpretation of the next reference in the discourse.

Establishing reference is 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 the issue of variable granularity across utterances in a dialogue: when, where, and how much information should be provided? 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. The examples below illustrates the issue. The two sentences (translated to English) are taken from the data that we gathered in our production experiment (see §2.2.1).

1. "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."

2. "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."

1.2.3 Variable intonation

Closely related to Tasks 6.3 and 6.4 is our work on contextually appropriate intonation for utterances in situated dialogue. Intonation is an important means in dialogue to indicate how some of the expressed content is related to the already established context, and what content the speaker intends to focus on. In other words, it is a principle means for achieving transparency. We have continued the work we started in Year 1, closely tying our approach into the overall framework for situated multi-agent modeling and situated dialogue processing.

Producing contextually appropriate intonation *Kruijff-Korbayová et al have continued to analyze the relation between empirical theories of intonation patterns and aspects of cognitive state, notably the attitudes of agreement and commitment to beliefs, and ownership of (or responsibility for) the verifiability of a belief. (See e.g. Meena's MSc thesis, available online.[1]) Based on the analysis they are developing a novel model of how a situated multi-agent model as discussed in (§2.1.5, §2.1.3) can help establish these attitudes, and thus determine intonation for production in a systematic way. Experiments are being performed to empirically test predictions made by the approach. The results of the first experiment are reported in §2.3.1.*

For example, when referring to an entity, the robot can indicate that it is aware of other (relevant) entities that share some property(ies) by accenting particular words (and not others), cf. (1). It can (also) indicate certainty vs. uncertainty (or in other terms: claim vs. relinquish dominance) by using falling vs. rising intonation, respectively, cf. (2). [2]

(1) a. the big red BALL
b. the big RED ball
c. the BIG red ball

(2) a. the ball ↓
b. the ball ↑

Context-dependent variation in the placement of pitch accent(s) based on focus/contrast was the main issue addressed in Year 1. In Year 2 we have elaborated the assignment of various types of pitch accents and boundary tones, depending on the cognitive state of the robot. For instance, to communicate a certain, uncontentious belief that the robot intends to establish as shared, the tune H*L-L% (or H-L%) is most appropriate, whereas to communicate an uncertain, contentious belief for which the robot is checking the

[1] `http://www.dfki.de/web/forschung/publikationen?pubid=4902`

[2] SMALL CAPITALS denote words carrying nuclear pitch accent.

human's commitment, it is the tune L*H-L% (or H-H%), cf. example (3a) and (3b), respectively.

(3) H: places a ball in front of R (not saying anything)
R: sees an object and forms the following belief:
B(R): KR:"a ball"
R adopts the goal to communicate its belief to H

a. R: a ball
H* L-L% (or H-L%)

b. R: a ball
L* H-L% (or H-H%)

Kruijff-Korbayová et al (§2.3.1) present the results of the first of a series of exeriments designed to verify some of the predictions the account makes with regard to intonation of clarification questions. The main goal of this experiment was to ascertain in a human-robot interaction scenario the commonly accepted view that hearers are sensitive to differences in accent placement depending on contrast between alternatives available in the context. There are two novel aspects about the study: (i) testing accent placement in clarification questions expressing a correct or an incorrect recognition hypothesis, and (ii) the application of a proper psycholinguistic experimental setup. The main hypothesis that accent placement makes a difference has been confirmed, but there are also various unexpected effects that require further investigation.

1.3 Relation to state-of-the-art

Below we briefly discuss how the obtained results relate to the current state-of-the-art. We refer the reader to the annexes for more in-depth discussions.

Moreover, we want to briefly mention dissemination activities performed in the context of the CogX project in which participants from within and outside the project discussed advances in the state-of-the-art. In May 2010, partners from BHAM and DFKI organized a workshop on "Interactive Communication for Autonomous Intelligent Robots (ICAIR)" [8] held in conjunction with the 2010 IEEE International Conference on Robotics and Automation (ICRA 2010). The workshop's theme centered around the question how to "make robots articulate what they understand, intend, and do" – thus contributing to the state-of-the-art in establishing common ground and achieving transparency in human-robot communication.

1.3.1 Adaptivity

Probabilistic models for adaptive dialogue management Uncertainty and partiality are pervasive in spoken dialogue systems. Due to speech recognition errors, linguistic or pragmatic ambiguities, the user's intentions are often difficult to decode. Furthermore, the evolution of the interaction ("what is the user going to say next?") is also typically impossible to predict for everything but the most trivial discourse domains.

In recent years, probabilistic models of dialogue combined with decision-theoretic planning have been developed to address these issues in a mathematically principled way. Most probabilistic models of dialogue management rely on the notion of dialogue *state*. The dialogue state is a variable summing up all the agent's knowledge about the dialogue history and external context which is assumed to be relevant for decision-making. Dialogue with non-deterministic transitions between states can be modelled as a *Markov Decision Process* (MDP). If in addition, we view the current dialogue state as not being directly observable (but rather inferred from observations), the dialogue can be formalised as a *Partially Observable Markov Decision Process* (POMDP). Examples of dialogue systems using MDPs and POMDPs can be found in [10, 9, 4, 30].

POMDPs combine several advantages wich make them particularly interesting for dialogue management. Besides the principled account of uncertainties mentioned above, POMDPs also rely on *decision-theoretic planning*, which is capable of forward planning over horizons of arbitrary length, and can encode complex trade-offs between competing objectives. Moreover, several reinforcement algorithms exist for the automatic optimisation of dialogue strategies based on dialogue transcripts or user simulators [23, 21]. Most of these algorithms are used offline, i.e. they generate a complete policy offline as a finite-state controller, and this policy can then be directly exploited without further planning. Alternatively, mechanisms for online planning or combinations of offline and offline planning also exist [22].

A POMDP is formally defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, \Omega, R \rangle$, where $\mathcal{S}$ is the state space; $\mathcal{A}$ is the action space; $\mathcal{Z}$ is the observation space; $T(s, a, s')$ is the transition function from state s to state s' using action a; $\Omega(z, a, s')$ is the observation function for observing z in state s' after performing action a; and $R(s, a)$ is the reward function encoding the utility for the agent of executing action a in state s.

A central assumption in POMDPs is that the state is not directly accessible and can only be inferred from observation. Such uncertainty is expressed in the *belief state* b, which is a probability distribution $b : \mathcal{S} \rightarrow [0, 1]$ over possible states. A POMDP policy is then defined over this belief space as a function $\pi : \mathcal{B} \rightarrow \mathcal{A}$ determining the action to perform for each point of the belief space.

Dialogue management can be easily cast as a POMDP problem, with the *state space* being a compact representation of the interaction, the *action space* being a set of dialogue moves, the *observation space* representing speech recognition hypotheses, the *transition function* defining the dynamics of the interaction (which user reaction is to be expected after a particular dialogue move), and the *observation function* describing a "sensor model" between observed speech recognition hypotheses and actual utterances. Finally, the *reward function* encodes the utility of dialogue policies – it typically assigns a big positive reward if a long-term goal has been reached (e.g. the retrieval of some important information), and small negative rewards for minor "inconveniences" (e.g. prompting the user to repeat or asking for confirmations).

Using a POMDP, adapting dialogue policies to specific aspects of the dialogue history or external context is mostly a matter of (1) extending the dialogue state to take these aspects into account, and (2) designing the reward function in such a way that the desirability of particular actions is made sensitive to these aspects. In order to keep the probabilistic model tractable for planning, such expansions of the state space is usually based on *factored models*. Factored models are probabilistic models where the random variable is factored into a set of separate subvariables which are assumed to be conditionally independent. An example of such adaptivity for the specific case of affective interaction is demonstrated in [4].

Our approach seeks to improve on this existing work by taking advantage of the relational structure present in most interactions. By exploiting the probabilistic belief models to handle the structural complexity of human-robot interactions, we hope to leverage the rich relational structure of the problem and efficiently abstract over large regions of the state and action spaces. In the long term, our aim is to develop a hybrid approach to adaptive dialogue management which combines the best of probabilistic and logical models of dialogue.

Learnable controllers The work on learnable controllers for dialogue processing is based on a long tradition of perceiving of the problem of control as a sequential decision making problem. Techniques typically used there are reinforcement learning, and Markov Decision Processes [27]. Our work is inspired by an approach that views the problem as an online planning problem with receding horizon control [15], in combination with factorized state spaces to exploit the structure inherent to dialogue interpretation [2, 26]. The learning techniques we are exploring are all based in reinforcement learning. We follow a model-based paradigm, particularly looking at MDPs with self-aware learning like R-MAX [3] or SLF-MAX [26]. These approaches all fit the KWIK framework [14], which models what the agent "knows what it knows" to actively drive learning in a way that fits well with the CogX focus on self-understanding and self-extension.

1.3.2 Variable granularity content planning

[18] observed that users interacting with the TRAINS-92 system make use of short non-anaphoric definite descriptions (e.g., "the boxcar") to felicitously refer to a specific one, even though the overall domain contains several boxcars. The correct referent of the utterance is determined by the previous discouse. 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* [5]. As factors that can make a situation available as resource situation, [5] 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 noun phrase "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. [18] 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 [18] 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* induced by the "movement" of the referents in the domain of discourse.

Based on these observations, we extend this approach from visual scenes to situated discourse about entities in large-scale space. 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 Year 1.

1.3.3 Variable intonation

Intonation is one of the primary means in many languages to realize the *information structure* of an utterance, and thereby its relationship to the discourse context, in terms of the discourse status of its content, the actual and attributed attentional states of the discourse participants, and the participants' prior and changing attitudes (knowledge, beliefs, intentions, expectations, etc.) [13]. A pioneering attempt to provide a compositional approach to the functional meaning of English intonation is [16]. Continuing in this tradition, [24] offers compositional semantics of English intonation in terms of information structure. [1], [7] and [29] provide detailed analyses concerning the meaning of various English tunes, especially focusing on boundary tones. The problem is that these accounts, although aware of one another, are separate. We have set out to make a detailed comparison and critical synthesis of their predictions from the perspective of intonation assignment. The resulting theoretical model is grounded in an implemented belief state framework, and we are in a position to experiment with the model in an integrated end-to-end system.

2 Annexes

2.1 Adaptivity

2.1.1 Lison, "Towards relational POMDPs for adaptive dialogue management." (ACL'10)

Bibliography P. Lison. "Towards relational POMDPs for adaptive dialogue management." In: Proceeding of the Student Research Workshop of the 48th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2010.

Abstract Open-ended spoken interactions are typically characterised by both structural complexity and high levels of uncertainty, making dialogue management in such settings a particularly challenging problem. Traditional approaches have focused on providing theoretical accounts for either the uncertainty or the complexity of spoken dialogue, but rarely considered the two issues in tandem. This paper describes ongoing work on a new approach to dialogue management which attempts to fill this gap. We represent the interaction as a Partially Observable Markov Decision Process (POMDP) over a rich state space incorporating both dialogue, user, and environment models. The tractability of the resulting POMDP can be preserved using a mechanism for dynamically constraining the action space based on prior knowledge over locally relevant dialogue structures. These constraints are encoded in a small set of general rules expressed as a Markov Logic network. The first-order expressivity of Markov Logic enables us to leverage the rich relational structure of the problem and efficiently abstract over large regions of the state and action spaces.

Relation to WP The paper makes it possible for the robot to adapt what dialogue actions to take, under uncertainty (Task 6.3). Together with the work on policy selection §2.1.2 it provides the basis for adapting action selection in dialogue management. Action selection is again part of the larger issue of modeling situated dialogue as a continual collaborative activity, discussed in §2.1.3.

2.1.2 Lison & Kruijff, "Policy activation for open-ended dialogue management." (Subm.)

Bibliography P. Lison & G.J.M. Kruijff. "Policy activation for open-ended dialogue management." Under submission to the AAAI 2010 Fall Symposium Dialogue with Robots.

Abstract An important difficulty in developing spoken dialogue systems for robots is the open-ended nature of most interactions. Robotic agents must typically operate in complex, continuously changing environments which are difficult to model and do not provide any clear, predefined goal. Directly capturing this complexity in a single, large dialogue policy is thus inadequate. This paper presents a new approach which tackles the complexity of open-ended interactions by breaking it into a set of small, independent policies, which can be activated and deactivated at runtime by a dedicated mechanism. The approach is currently being implemented in a spoken dialogue system for autonomous robots.

Relation to WP Together with §2.1.1 this work provides the basis for adaptation in action selection for dialogue management (Task 6.3).

2.1.3 Kruijff & Janíček, "Continual Processing of Situated Dialogue in Human-Robot Collaborative Activities." (RO-MAN 2010)

Bibliography G.J.M. Kruijff & M. Janíček. "Continual Processing of Situated Dialogue in Human-Robot Collaborative Activities." In: Proceedings of the 19th IEEE International Symposium in Robot and Human Interactive Communication (RO-MAN 2010). IEEE, 2010.

Abstract The paper presents an implemented approach of processing situated dialogue between a human and a robot. The focus is on task-oriented dialogue, set in the larger context of human-robot collaborative activity. The approach models understanding and production of dialogue to include intension (what is being talked about), intention (the goal of why something is being said), and attention (what is being focused on). These dimensions are directly construed in terms of assumptions and assertions on situated multi-agent belief models. The approach is continual in that it allows for interpretations to be dynamically retracted, revised, or deferred. This makes it possible to deal with the inherent asymmetry in how robots and humans tend to understand dialogue, and the world it is set in. The approach has been fully implemented, and integrated into a cognitive robot. The paper discusses the implementation, and illustrates it in a collaborative learning setting.

Relation to WP The paper discusses the overarching approach to modeling situated dialogue as a continual collaborative activity. The approach makes use of abductive inference to decide which beliefs and intentions to invoke when contextually comprehending or producing dialogue. The action selection mechanisms nowadays employs the POMDP-based adaptive selection. This allows for a context-senstive, adaptive way of selecting intentions and realizing them in the current (epistemic, situated) context (Task 6.3).

2.1.4 Lison et al, "Belief modelling for situation awareness in human-robot interaction." (RO-MAN 2010)

Bibliography P. Lison, C. Ehrler, and G.J.M. Kruijff. "Belief modelling for situation awareness in human-robot interaction." In: Proceedings of the 19th IEEE International Symposium in Robot and Human Interactive Communication (RO-MAN 2010). IEEE, 2010.

Abstract To interact naturally with humans, robots needs to be aware of their own surroundings. This awareness is usually encoded in some implicit or explicit representation of the situated context. In this paper, we present a new framework for constructing rich belief models of the robot's environment. Key to our approach is the use of Markov Logic as a unified representation formalism. Markov Logic is a combination of first-order logic and probabilistic graphical models. Its expressive power allows us to capture both the rich relational structure of the environment and the uncertainty arising from the noise and incompleteness of low-level sensory data. The constructed belief models evolve dynamically over time and incorporate various contextual information such as spatio-temporal framing, multi-agent epistemic status, and saliency measures. Beliefs can also be referenced and extended top-down via linguistic communication. The approach is being integrated into a cognitive architecture for mobile robots interacting with humans using spoken dialogue.

Relation to WP The paper provides a probabilistic take on the situated multi-agent models we developed in Year 1. The probabilistic models provide a proper way of modeling uncertainty in experience, and we can use structural forms of inference to reason with them. These models are the basis on which decisions on how to adapt are based (Task 6.3). §2.1.5 presents a more indepth discussion. The framework is used throughout the cognitive system in CogX to represent and reason with experience, action, and interaction.

2.1.5 Kruijff et al, "Combining Probabilistic And Logical Inference in Situated Multi-Agent Models" (Report)

Bibliography G.J.M. Kruijff, M. Janíček, P. Lison and H.-U. Krieger. "Combining Probabilistic And Logical Inference in Situated Multi-Agent Models." Report.

Abstract The paper describes work in progress on a formal system for representing, and reasoning with, situated multi-agent belief models. These models capture what a particular agent believes about the world, and what it believes about other agents. Such beliefs arise from a mixture of inferences, ranging over the agent's direct perception of the world, what it has as semantic background knowledge about the world, and what facts the agent can infer to hold over time. The model puts probabilistic and logical inference on a par, to balance logical structure with a robustness to uncertain and partial information. The paper discusses various forms of logical and probabilistic inference, and the possibilities for combining them.

Relation to WP The report presents an in-depth discussion of the situated, multi-agent framework for representing and reasoning with beliefs, intentions, and events adopted in CogX (Task 6.3).

2.1.6 Krieger, "A General Methodology for Equipping Ontologies With Time." (LREC'10)

Bibliography H.-U. Krieger. "A General Methodology for Equipping Ontologies With Time." In: Proceedings of the 7th international conference on Language Resources and Evaluation (LREC'10).

Abstract In the first part of this paper, we present a framework for enriching arbitrary upper or domain-specific ontologies with a concept of time. To do so, we need the notion of a time slice. Contrary to other approaches, we directly interpret the original entities as time slices in order to (i) avoid a duplication of the original ontology and (ii) to prevent a knowledge engineer from ontology rewriting. The diachronic representation of time is complemented by a sophisticated time ontology that supports underspecification and an arbitrarily fine granularity of time. As a showcase, we describe how the time ontology has been interfaced with the PROTON upper ontology. The second part investigates a temporal extension of RDF that replaces the usual triple notation by a more general tuple representation. In this setting, Hayes/ter Horst-like entailment rules are replaced by their temporal counterparts. Our motivation to move towards this direction is twofold: firstly, extending binary relation instances with time leads to a massive proliferation of useless objects (independently of the encoding); secondly, reasoning and querying with such extended relations is extremely complex, expensive, and error-prone.

Relation to WP The paper describes inference techniques used in computing (temporal) closures situated BIE models, as described in §2.1.5 (Task 6.3).

2.1.7 Kruijff & Krieger, "Learnable Controllers for Adaptive Dialogue Processing Management." (Subm.)

Bibliography G.J.M. Kruijff & H.-U. Krieger. "Learnable Controllers for Adaptive Dialogue Processing Management." Under submission to the AAAI 2010 Fall Symposium Dialogue with Robots.

Abstract Uncertainty is pervasive throughout processing spoken dialogue in human-robot interaction. That need not always be a problem though. The paper adopts the view that it depends on context, how much that uncertainty actually matters. The paper argues that uncertainty in input needs to be balanced off against how much actually needs to be understood to make a contextually appropriate, next move. The paper presents **work in progress** on developing mechanisms for adaptively controlling how utterances in spoken dialogue in human-robot interaction get processed "step-by-step", to deal with uncertainty in as much as necessary given an goal state. These mechanisms take the form of a learnable closed-loop controller that decides on an optimal policy or process configuration to reach a next fixed-point in a state space of (partial) analyses. The policy is planned online, adapting the processing strategy rather than using a "universal" policy.

Relation to WP The paper outlines an approach for adapting the way dialogue is processed. It proposes to use learnable controllers to adapt, at each "step" of analyzing an utterance, what processes get applied to help construct contextually relevant analyses. This is guided by a formulation of a goal state that specifies what there is to be understood about this utterance, given expectations about the way the dialogue is likely to proceed (resulting from POMDP-based action planning, §2.1.2) (Task 6.3).

2.2 Variable granularity

2.2.1 Zender et al, "Anchor-Progression in Spatially Situated Discourse: a Production Experiment." (INLG'10)

Bibliography H. Zender, C. Koppermann, F. Greeve, and G.J.M. Kruijff. "Anchor-Progression in Spatially Situated Discourse: a Production Experiment." In: Proceedings of the 6th International Natural Language Generation Conference (INLG 2010). Dublin, Ireland, July 2010

Abstract The paper presents two models for producing and understanding situationally appropriate referring expressions (REs) during a discourse about large-scale space. The models are evaluated against an empirical production experiment.

Relation to WP The paper extends our earlier work on verbalizing references to objects, places, or events outside the current situation (reported in DR6.1) to multi-utterance discourses. In such discourses, consecutive referring expressions to different entities that are located elsewhere in the interlocutors' environment must contain a different amount of information than singleton referring expressions. Consecutive referring expressions act as *attentional anchors* which evoke new mental resource situations on the part of the hearer. This allows the speaker to make use of shorter but nevertheless successfully identifying descriptions.

An example for this is a situation in which the user and the robot are in the corridor and the user gives the robot instructions how to clean up the apartment. Instead of saying "Go to the kitchen. Take the ball in the kitchen and put it into the box on the table in the kitchen," it is sufficient to say "Go to the kitchen. Take the ball and put it into the box on the table," even in cases where there exist several balls, boxes and tables elsewhere in the environment.

The granularity of the information (cf. Task 6.4) to be realized in a felicitous referring expression is thus dependent on preceding references. This paper presents two models for this dependence, called *anchor-progression* and *anchor-resetting.* While the models are described in more detail in the report in §2.2.2, this paper particularly focuses on an empirical production experiment for gathering human produced utterances to compare the models against.

2.2.2 Zender et al, "Anchor-Progression in Situated Discourse About Large-Scale Space" (Report)

Bibliography H. Zender, C. Koppermann, F. Greeve, and G.J.M. Kruijff. "Anchor-Progression in Situated Discourse About Large-Scale Space." Manuscript in preparation for journal submission.

Abstract The use of natural language processing systems is no longer limited to small, fixed, fully known and fully observable domains. In interaction with mobile robots, with non-player characters in virtual worlds, or with mobile location-based applications alike references to entities outside the currently observable scene (i.e., in *large-scale space*) are becoming more and more important. *Referring expressions* (e.g., definite noun phrases, pronouns, and proper names) are used to convey which entities in the world are being talked about. Ideally, the natural language communication with such systems is not restricted to single one-way utterances. The way successful reference between such a system and its user is established must thus be viewed from a discourse-oriented perspective. Successful reference is established by the interplay of referring expressions and the way the discourse unfolds.

In this paper we address the challenge of producing and understanding referring expressions to entities in large-scale space during a discourse. To this end, we propose a general principle of *topological abstraction* (TA) for determining an appropriate spatial context. This principle is applied to the tasks of generating and resolving referring expressions. Further, we propose *anchor-progression* and *anchor-resetting* mechanisms to track the origin of the TA algorithms throughout the discourse. Finally, we present an empirical 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.

Relation to WP This report presents a more detailed discussion of the models for tracking the attentional anchor presented in §2.2.1. The models are suitable for the tasks of generating and resolving referring expressions alike. The report describes how the models can be directly and straightforwardly used with the dialogue framework developed in the context of WP6 and the models for representing large-scale space from WP3 (see also DR6.1). The results of this work hence contribute immediately to the Dora demonstrator from WP7 because they allow a robot to produce and understand consecutive references to entities that are located elsewhere in its operating environment (such as an office floor or an apartment).

2.3 Variable intonation

2.3.1 Kruijff-Korbayová et al, "Contextually Appropriate Intonation of Clarification Requests in Human-Robot Interaction" (Report)

Bibliography I. Kruijff-Korbayová, R. Meena, and P. Pyykkönen.

Abstract It is established that assigning intonation to dialogue system output in a way that reflects contrast among entities available in the discourse context can enhance the acceptability of system utterances. Previous research has concentrated on the role of linguistic context in processing; dialogue *situatedness* and hence the role of visual context in determining the accent placement has not been studied. In this paper, we present an experimental study addressing the influence of visual context on the perception of nuclear accent placement in synthesized clarification requests. We predicted that variation in the placement of nuclear accent is perceivable and that visual context affects acceptability. We found that utterances with nuclear accent placement licenced by the visual scene are perceived as appropriate more often then utterances with nuclear accent placement not licenced by the visual scene.

Relation to WP The paper provides further insights in the contextually appropriate production of robot utterances, in mixed-initiative dialogues for cross-modal learning.

References

[1] Christine Bartels. *Towards a Compositional Interpretation of English Statement and Question Intonation.* PhD thesis, University of Massachusetts, Amherst, 1997.

[2] C. Boutilier, T. Dean, and S. Hanks. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of AI Research*, 11:1–94, 1999.

[3] R.I. Brafman and M. Tennenholtz. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

[4] Trung h. Bui, Mannes Poel, Anton Nijholt, and Job Zwiers. A tractable hybrid DDN–POMDP approach to affective dialogue modeling for probabilistic frame-based dialogue systems. *Natural Language Engineering*, 15(2):273–307, 2009.

[5] 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.

[6] Matthew Frampton and Oliver Lemon. Recent research advances in reinforcement learning in spoken dialogue systems. *Knowledge Engineering Review*, 24(4):375–408, 2009.

[7] Christine Gunlogson. *True to Form: Rising and Falling Declaratives as Questions in English.* PhD thesis, University of California at Santa Cruz, 2001.

[8] Marc Hanheide and Hendrik Zender, editors. *Proceedings of the ICRA 2010 Workshop on Interactive Communication for Autonomous Intelligent Robots (ICAIR)*, Anchorage, AK, USA, May 2010.

[9] James Henderson and Oliver Lemon. Mixture model pomdps for efficient handling of uncertainty in dialogue management. In *HLT '08: Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies*, pages 73–76, Morristown, NJ, USA, 2008. Association for Computational Linguistics.

[10] Jesse Hoey, Axel Von Bertoldi, Pascal Poupart, and Alex Mihailidis. Assisting persons with dementia during handwashing using a partially observable markov decision process. In *Proceedings of the International Conference on Vision Systems (ICVS)*, 2007.

[11] H. Jacobsson, N.A. Hawes, G.J.M. Kruijff, and J. Wyatt. Crossmodal content binding in information-processing architectures. In *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Amsterdam, The Netherlands, March 12–15 2008.

[12] G.J.M. Kruijff, J.D. Kelleher, and N. Hawes. Information fusion for visual reference resolution in dynamic situated dialogue. In E. André, L. Dybkjaer, W. Minker, H. Neumann, and M. Weber, editors, *Perception and Interactive Technologies (PIT 2006).* Spring Verlag, 2006.

[13] Ivana Kruijff-Korbayová and Mark Steedman. Discourse and information structure. *Journal of Logic, Language and Information: Special Issue on Discourse and Information Structure*, 12(3):249–259, 2003.

[14] L. Li, M.L. Littman, and T.J. Walsh. Knows what it knows: A framework for self-aware learning. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML-08)*, Helsinki, Finland, 2008.

[15] B. Morisset and M. Ghallab. Learning how to combine sensory-motor functions into a robust behavior. *Artificial Intelligence*, 172(4-5):392–412, 2008.

[16] Janet Pierrehumbert and Julia Hirschberg. The meaning of intonational contours in the interpretation of discourse. In Philip Cohen, Jerry Morgan, and Martha Pollack, editors, *Intentions in Communication*, pages 271–312. MIT Press, Cambridge, MA, 1990.

[17] Joelle Pineau, Geoffrey Gordon, and Sebastian Thrun. Anytime point-based approximations for large pomdps. *Artificial Intelligence Research*, 27(1):335–380, 2006.

[18] 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.

[19] Pascal Poupart. *Exploiting structure to efficiently solve large scale partially observable markov decision processes*. PhD thesis, Toronto, Canada, 2005.

[20] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[21] Verena Rieser. *Bootstrapping Reinforcement Learning-based Dialogue Strategies from Wizard-of-Oz data.* PhD thesis, Saarland University, October 2008.

[22] Stéphane Ross, Joelle Pineau, Sébastien Paquet, and Brahim Chaib-draa. Online planning algorithms for pomdps. *Artificial Intelligence Research*, 32(1):663–704, 2008.

[23] Jost Schatzmann, Blaise Thomson, Karl Weilhammer, Hui Ye, and Steve Young. Agenda-based user simulation for bootstrapping a POMDP dialogue system. In *NAACL '07: Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers on XX*, pages 149–152, Morristown, NJ, USA, 2007. Association for Computational Linguistics.

[24] Mark Steedman. Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 31(4):649–689, 2000.

[25] M. Stone and R.H. Thomason. Coordinating understanding and generation in an abductive approach to interpretation. In *Proceedings of DIABRUCK 2003: 7th workshop on the semantics and pragmatics of dialogue*, 2003.

[26] A.L. Strehl, C. Diuk, and M.L. Littman. Efficient structure learning in factored-state MDPs. In *Proceedings of the 22nd national conference on Artificial intelligence (AAAI'07)*, pages 645–650, 2007.

[27] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction.* Adaptive Computation and Machine Learning. The MIT Press, 1998.

[28] R.H. Thomason, M. Stone, and D. DeVault. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. In D. Byron, C. Roberts, and S. Schwenter, editors, *Presupposition Accommodation.* to appear.

[29] Marie Šafářová Nilsenová. *Rises and Falls: Studies in the semantics and pragmatics of intonation.* PhD thesis, Institute for Logic, Language and Information, Universiteit van Amsterdam, 2006.

[30] Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174, 2010.

Towards Relational POMDPs for Adaptive Dialogue Management

Pierre Lison
Language Technology Lab
German Research Centre for Artificial Intelligence (DFKI GmbH)
Saarbrücken, Germany

Abstract

Open-ended spoken interactions are typically characterised by both structural complexity and high levels of uncertainty, making dialogue management in such settings a particularly challenging problem. Traditional approaches have focused on providing theoretical accounts for either the uncertainty or the complexity of spoken dialogue, but rarely considered the two issues in tandem. This paper describes ongoing work on a new approach to dialogue management which attempts to fill this gap. We represent the interaction as a Partially Observable Markov Decision Process (POMDP) over a rich state space incorporating both dialogue, user, and environment models. The tractability of the resulting POMDP can be preserved using a mechanism for dynamically constraining the action space based on prior knowledge over locally relevant dialogue structures. These constraints are encoded in a small set of general rules expressed as a Markov Logic network. The first-order expressivity of Markov Logic enables us to leverage the rich relational structure of the problem and efficiently abstract over large regions of the state and action spaces.

1 Introduction

The development of spoken dialogue systems for rich, open-ended interactions raises a number of challenges, one of which is dialogue management. The role of dialogue management is to determine which communicative actions to take (i.e. what to say) given a goal and particular observations about the interaction and the current situation.

Dialogue managers have to face several issues. First, spoken dialogue systems must usually deal with high levels of noise and uncertainty in the spoken inputs. These uncertainties may arise from speech recognition errors, limited grammar coverage, or from various ambiguities in the linguistic or pragmatic interpretations.

Second, open-ended dialogue is characteristically complex, and exhibits rich relational structures. Natural interactions should be adaptive to a variety of factors dependent on the interaction history, the general context, and the user preferences. As a consequence, the state space necessary to model the dynamics of the environment tends to be large and sparsely populated.

These two problems have typically been addressed separately in the literature. On the one hand, the issue of uncertainty in speech understanding is usually dealt using a range of probabilistic models combined with decision-theoretic planning. Among these, *Partially Observable Markov Decision Process* (POMDP) models have recently emerged as a unifying mathematical framework for dialogue management (Williams and Young, 2007; Lemon and Pietquin, 2007). POMDPs provide an explicit account for a wide range of uncertainties related to partial observability (noisy, incomplete spoken inputs) and stochastic action effects (non-deterministic dynamics).

On the other hand, structural complexity can be addressed with logic-based approaches, based for instance on pragmatic interpretation (Thomason et al., 2006), dialogue structure (Asher and Lascarides, 2003), or collaborative planning (Kruijff et al., 2008). Such approaches are able to model sophisticated dialogue behaviours, but at the expense of robustness and adaptivity. They generally assume complete observability and provide only a very limited account (if any) of uncertainties.

We are currently developing an hybrid approach which *simultaneously* tackles the uncertainty and complexity of dialogue management, based on a POMDP framework. We present here our ongoing work on this issue. In this paper, we more specifically describe a new mechanism for dynamically constraining the space of possible actions available at a given time. Our aim is to use such mechanism to significantly reduce the search space and therefore make the planning problem globally more tractable. This is performed in two consecutive steps. We first structure the state space using *Markov Logic Networks*, a first-order probabilistic language. Prior pragmatic knowledge about dialogue structure is then exploited to derive the set of dialogue actions which are locally admissible or relevant, and prune all irrelevant ones. The first-order expressivity of Markov Logic Networks allows us to easily specify the constraints via a small set of general rules which abstract over large regions of the state and action spaces.

Our long-term goal is to develop an unified framework for adaptive dialogue management in rich, open-ended interactional settings. Such dialogue manager is to be integrated in a cognitive architecture for a mobile robot interacting with human users in an indoor environment via spoken dialogue (Hawes et al., 2007; Kruijff et al., 2010).

This paper is structured as follows. Section 2 lays down the formal foundations of our work, by describing dialogue management as a POMDP problem. We then describe in Section 3 our approach to dimensionality reduction using Markov Logic rules. Section 4 discusses some further aspects of our approach and its relation to existing work, followed by the conclusion in Section 5.

2 Background

2.1 Partially Observable Markov Decision Processes (POMDPs)

POMDPs are a mathematical model for sequential decision-making in partially observable environments. It provides a powerful framework for control problems which combine partial observability, uncertain action effects, incomplete knowledge of the environment dynamics and multiple, potentially conflicting objectives.

Via reinforcement learning, it is possible to automatically *learn* optimal or near-optimal action policies given a POMDP model combined with real or simulated user data (Rieser, 2008).

2.1.1 Formal definition

A POMDP is a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, \Omega, R \rangle$, where:

- $\mathcal{S}$ is the **state space**, which is the model of the world from the agent's viewpoint. It is defined as a set of mutually exclusive states.
- $\mathcal{A}$ is the **action space**: the set of possible actions at the disposal of the agent.
- $\mathcal{Z}$ is the **observation space**: the set of observations which can be captured by the agent.
- T is the **transition function**. It is formally defined as a function $T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, where $T(s, a, s') = P(s'|s, a)$. It is therefore the probability of reaching the state s' from the state s if action a is performed.
- Ω is the **observation function**, defined as $\Omega : \mathcal{Z} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$, with $\Omega(z, a, s') = P(z|a, s')$. It expresses the probability of observing the particular input z if the agent has performed action a and is now in state s'.
- R is the **reward function**, defined as $R : \mathcal{S} \times \mathcal{A} \rightarrow \Re$, $R(s, a)$ is a real number encoding the utility for the agent to perform the action a while in state s.

In addition, a POMDP usually includes the following two parameters h and γ:

- h is the **horizon** of the POMDP-based agent. It defines the number of look-ahead steps that are taken into account when planning.
- γ is the **discount factor**, providing a weighting scheme for non-immediate rewards.

2.1.2 Beliefs and belief update

A key idea of POMDP is the assumption that the state of the world is not directly accessible, and can only be inferred via observation. Such uncertainty is expressed in the **belief state** b, which is a probability distribution over possible states, that is: $b : \mathcal{S} \rightarrow [0, 1]$. The belief state for a state space of cardinality n is therefore represented in a real-valued simplex of dimension $(n-1)$.

This belief state is dynamically updated before executing each action. The belief state update operates as follows. At a given time step t, the agent is in some unobserved state $s_t = s \in \mathcal{S}$. The probability of being in state s at time t is written as $b_t(s)$. Based on the current belief state b_t, the agent selects an action a_t, receives a reward $R(s, a_t)$ and transitions to a new (unobserved) state $s_{t+1} = s'$, where s_{t+1} depends only on s_t and a_t. The agent then receives a new observation o_{t+1} which is dependent on s_{t+1} and a_t.

Finally, the belief distribution b_t is updated, based on o_{t+1} and a_t as follows[1].

$$b_{t+1}(s') = P(s'|o_{t+1}, a_t, b_t) \quad (1)$$

$$= \frac{P(o_{t+1}|s', a_t, b_t) P(s'|a_t, b_t)}{P(o_{t+1}|a_t, b_t)} \quad (2)$$

$$= \alpha\, \Omega(o_{t+1}, s', a_t) \sum_{s \in \mathcal{S}} T(s, a_t, s') b_t(s) \quad (3)$$

where α is a normalisation constant. An initial belief state b_0 must be specified at runtime as a POMDP parameter when initialising the system.

2.1.3 POMDP policies

Given a POMDP model $\langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, Z, R \rangle$, the agent should execute at each time-step the action which maximises its expected cumulative reward over the horizon. We define a function $\pi : \mathcal{B} \rightarrow \mathcal{A}$, called a *policy*, which determines the action to perform for each point of the belief space.

The expected reward for policy π starting from belief b is defined as:

$$J^{\pi}(b) = E\Big[\sum_{t=0}^{h} \gamma^t R(s_t, a_t) \mid b, \pi\Big] \quad (4)$$

The optimal policy π^* is then obtained by optimizing the long-term reward, starting from b_0:

$$\pi^* = \underset{\pi}{\text{argmax}}\ J^{\pi}(b_0) \quad (5)$$

The optimal policy π^* yields the highest expected reward value for each possible belief state. This value is compactly represented by the optimal value function, noted V^*, which is a solution to the Bellman optimality equation (Bellman, 1957).

Numerous algorithms for (offline) policy optimisation and (online) planning are available. For large spaces, exact optimisation is impossible and approximate methods must be used, see for instance grid-based (Thomson and Young, 2009) or point-based (Pineau et al., 2006) techniques.

2.2 POMDP-based dialogue management

Dialogue management can be easily cast as a POMDP problem:

- The *state space* is a compact representation of the interaction (information state), combined with relevant features of the situation.
- The *action space* is a set of dialogue moves.

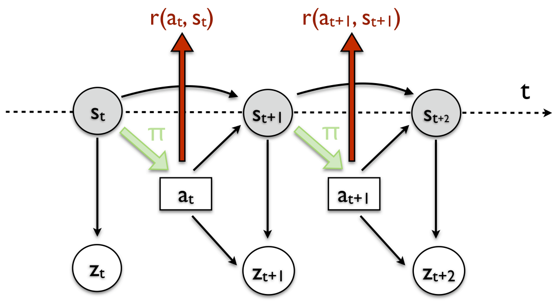


Figure 1: *Dynamic Bayesian Network* corresponding to the POMDP model. Actions are represented as rectangles to stress that they are system actions rather than observed random variables. State variables are greyed since they are hidden variables.

- The *observation space* is usually a set of possible speech recognition hypotheses (represented as a word lattice or a N-Best list), plus relevant observations from the environment.
- The *transition function* defines the local dialogue "dynamics" (which user reaction is to be expected after a particular dialogue move).
- The *observation function* describes a "sensor model" between observed speech recognition hypotheses and the real (hidden) utterance.
- Finally, the *reward function* encodes the utility of dialogue policies – it typically assigns a big positive reward if a long-term goal has been reached (e.g. the retrieval of some important information), and small negative rewards for various "annoyances" (e.g. prompting the user to repeat).

2.3 Dialogue management for human-robot interaction (HRI)

Our aim is to apply such POMDP framework to a rich dialogue domain for HRI (Kruijff et al., 2010). These interactions are typically open-ended (no predefined goal), relatively long (hundreds to thousands of turns), highly noisy (environmental noise, faulty speech recognition), and require complex state and action spaces (both the user and the robot can talk freely and perform diverse collaborative tasks and activities).

The dialogue system also needs to be *adaptive* to its user (attributed beliefs and intentions, attitude, attentional state) and to the current situation (currently perceived entities and events). As a con-

[1] As a notational shorthand, we write $P(s_t{=}s)$ as $P(s)$ and $P(s_{t+1}{=}s')$ as $P(s')$.

sequence, the state space must be expanded to include these knowledge sources.

These requirements can only be fullfilled if we address the “curse of dimensionality” which is bound to face such sophisticated state spaces. The next section provides a tentative answer.

3 Approach

3.1 Dimensionality reduction

Classical approaches to POMDP planning operate directly on the full action space and select the next action to perform based on the maximisation of the expected cumulative reward over the specified horizon. Such approaches can be used in small-scale domains with a limited action space, but quickly become intractable for larger ones, as the planning time increases exponentially with the size of the action space (for non-immediate horizons). Significant planning time is therefore wasted on actions which, from the viewpoint of a human user, are irrelevant[2]. Dismissing such irrelevant actions *before* planning would enable the agent to concentrate its computational resources on locally relevant actions.

Instead of a direct policy optimisation over the full action space, our approach formalises action selection as a *two-step* process. As a first step, a set of *relevant dialogue moves* is constructed from the full action space. The POMDP planner then computes the optimal (highest-reward) action on this reduced action space in a second step.

Such an approach is able to significantly reduce the dimensionality of the dialogue management problem by taking advantage of prior knowledge about the expected relational structure of spoken dialogue. This prior knowledge is to be encoded in a set of general rules describing the admissible dialogue moves in a particular situation.

How can we express such rules? POMDPs are usually modeled with Bayesian networks which are inherently propositional. Encoding such rules in a propositional framework requires a distinct rule for every possible state and action instance. This is not a feasible approach. We therefore need a first order (probabilistic) language able to express generalities over large regions of the state action spaces. Markov Logic is such a language.

3.2 Markov Logic Networks (MLNs)

Markov Logic combines first-order logic and probabilistic graphical models in a unified representation (Richardson and Domingos, 2006). A Markov Logic Network L is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and w_i is a real number representing the formula weight.

A Markov Logic Network L can be seen as a *template* for constructing markov networks[3]. To construct a markov network from L, one has to provide an additional set of constants $C = \{c_1, c_2, ..., c_{|C|}\}$. The resulting markov network is called a *ground markov network* and is written $M_{L,C}$. The ground markov network contains one feature for each possible grounding of a first-order formula in L, with the corresponding weight. The technical details of the construction of $M_{L,C}$ from the two sets L and C is explained in several papers, see e.g. (Richardson and Domingos, 2006).

Once the markov network $M_{L,C}$ is constructed, it can be exploited to perform *inference* over arbitrary queries. Efficient probabilistic inference algorithms such as Markov Chain Monte Carlo (MCMC) or other sampling techniques can then be used to this end (Poon and Domingos, 2006).

3.3 States and actions as relational structures

The specification of Markov Logic rules applying over complete regions of the state and action spaces (instead of over single instances) requires an explicit relational structure over these spaces.

This is realised by factoring the state and action spaces into distinct, conditionally independent parts. A state s can be expanded into a tuple $\langle f_1, f_2, ...f_n \rangle$, where each sub-state f_i is assigned a value from a set $\{v_1, v_2, ...v_m\}$. Such structure can be expressed in first-logic with a binary predicate $f_i(s, v_j)$ for each sub-state f_i, where v_j is the value of the sub-state f_i in s. The same type of structure can be defined over actions. This factoring leads to a relational structure of arbitrary complexity, compactly represented by a set of unary and binary predicates.

3.4 Relevant action space

For a given state s, the relevant action space $RelMoves(\mathcal{A}, s)$ is defined as:

$$\{a : a \in \mathcal{A} \wedge \mathtt{RelevantMove(a, s)}\} \qquad (6)$$

[2] For instance, an agent hearing a user command such as “Please take the mug on your left” might spent a lot of planning time calculating the expected future reward of dialogue moves such as “Is the box green?” or “Your name is John”, which are irrelevant to the situation.

[3] Markov networks are undirected graphical models.

The truth-value of the predicate $\texttt{RelevantMove(a, s)}$ is determined using a set of Markov Logic rules dependent on both the state s and the action a. For a given state s, the relevant action space is constructed via probabilistic inference, by estimating the probability $P(\texttt{RelevantMove(a, s)} \mid s, a)$ for each action a, and selecting the subset of actions for which the probability is above a given threshold.

Eq. 7 provides an example of such Markov Logic rule. It defines an admissible dialogue move for a situation where the user questions the agent about the feature of a particular object. The rule specifies that, if the system is in state s at time t, and if the last dialogue move is m and is a question about the feature f of object o, there is an admissible move a consisting in the assertion that the feature f of object o has the value v. The rule is universally quantified over all possible assignments of variables s, t, m, f, o and v.

Concretely, the rule states that after a question such as "what is the colour of the box?", a possible reaction would be "It is red" (assuming the box can be properly referred by the pronoun "it").

$$\begin{aligned}&[\texttt{State(s, t)} \wedge \texttt{LastUserMove(s, m)} \wedge \\ &\texttt{Question(m)} \wedge \texttt{AboutObj(m, o)} \wedge \\ &\texttt{AboutFeat(m, f)} \wedge \texttt{AssertMove(a)} \wedge \\ &\texttt{AboutObj(a, o)} \wedge \texttt{AboutFeat(a, f)} \wedge \\ &\texttt{AboutVal(a, v))}] \rightarrow \texttt{RelevantMove(a, s)}\end{aligned} \quad (7)$$

Each of these Markov Logic rules has a weight attached to it, expressing the strength of the implication. A rule with infinite weight and satisfied premises will lead to a relevant move with probability 1. Softer weights can be used to describe moves which are less relevant but still possible in a particular context. These weights can either be encoded by hand or learned from data.

3.5 Rules application on POMDP belief state

The previous section assumed that the state s is known. But the real state of a POMDP is never directly accessible. The rules we just described must therefore be applied on the belief state. Ultimately, we want to define a function $Rel : \Re^n \rightarrow \mathcal{P}(\mathcal{A})$, which takes as input a point in the belief space and outputs a set of relevant moves.

Due to the high dimensionality of the belief space, the above function must be approximated to remain tractable. One way to perform this approximation is to extract, for each point in b, a set S_m of m most likely states, and compute the set of relevant moves for each of them. We then define the global probability estimate of a being a relevant move given b as such:

$$\begin{aligned}&P(\texttt{RelevantMove(a)} \mid b, a) \approx \\ &\sum_{s \in S_m} P(\texttt{RelevantMove(a, s)} \mid s, a) \times b(s)\end{aligned} \quad (8)$$

In the limit where $m \rightarrow |S|$, the error margin on the approximation tends to zero.

4 Discussion

4.1 General comments

It is worth noting that the mechanism we just outlined does not intend to *replace* the existing POMDP planning and optimisation algorithms, but rather *complements* them. Each step serves a different purpose: the action space reduction provides an answer to the question "Is this action relevant?", while the policy optimisation seeks to answer "Is this action useful?". We believe that such distinction between relevance and usefulness is important and will prove to be beneficial in terms of tractability.

It is also useful to notice that the Markov Logic rules we described provides a "positive" definition of the action space. The rules were applied to produce an exhaustive list of all admissible actions given a state, all actions outside this list being *de facto* labelled as non-admissible. But the rules can also provide a "negative" definition of the action space. That is, instead of generating an exhaustive list of possible actions, the dialogue system can initially consider all actions as admissible, and the rules can then be used to prune this action space by removing irrelevant moves.

Which of these two options provides the optimal solution depends on two factors: the size of the dialogue domain, and the domain knowledge of the dialogue developer. As the dialogue domains grow larger, the need for a positive definition of the action space becomes more acute, as the action space is likely to grow exponentially with the domain size and become untractable. But the positive definition of the action space is also significantly more expensive for the dialogue developer. There is therefore a trade-off between the costs of tractability issues, and the costs of dialogue domain modelling.

4.2 Related Work

There is a substantial body of existing work in the POMDP literature about the exploitation of the problem structure to tackle the curse of dimensionality (Poupart, 2005; Young et al., 2010), but the vast majority of these approaches retain a propositional structure. A few more theoretical papers also describe first-order MDPs (Wang et al., 2007), and recent work on Markov Logic has extended the MLN formalism to include some decision-theoretic concepts (Nath and Domingos, 2009). To the author's knowledge, none of these ideas have been applied to dialogue management.

5 Conclusions

This paper described a new approach to exploit relational models of dialogue structure for dimensionality reduction in POMDPs. This approach is part of an ongoing work to develop a unified framework for adaptive dialogue management in rich, open-ended interactional settings. The dialogue manager is being implemented as part of a larger cognitive architecture for talking robots.

Besides the implementation, future work will focus on refining the theoretical foundations of relational POMDPs for dialogue (including how to specify the transition, observation and reward functions in such a relational framework), as well as investigating the use of reinforcement learning for policy optimisation based on simulated data.

References

N. Asher and A. Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

R. Bellman. 1957. *Dynamic Programming*. Princeton University Press.

N. A. Hawes, A. Sloman, J. Wyatt, M. Zillich, H. Jacobsson, G.-J. M. Kruijff, M. Brenner, G. Berginc, and D. Skocaj. 2007. Towards an integrated robot with multiple cognitive functions. In *Proc. AAAI'07*, pages 1548–1553. AAAI Press.

G.J.M. Kruijff, M. Brenner, and N.A. Hawes. 2008. 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)*, Munich, Germany.

G.-J. M. Kruijff, P. Lison, T. Benjamin, H. Jacobsson, H. Zender, and I. Kruijff-Korbayova. 2010. Situated dialogue processing for human-robot interaction. In H. I. Christensen, A. Sloman, G.-J. M. Kruijff, and J. Wyatt, editors, *Cognitive Systems*. Springer Verlag. (in press).

O. Lemon and O. Pietquin. 2007. Machine learning for spoken dialogue systems. In *Proceedings of the European Conference on Speech Communication and Technologies (Interspeech'07)*, pages 2685–2688, Anvers (Belgium), August.

A. Nath and P. Domingos. 2009. A language for relational decision theory. In *Proceedings of the International Workshop on Statistical Relational Learning*.

J. Pineau, G. Gordon, and S. Thrun. 2006. Anytime point-based approximations for large pomdps. *Artificial Intelligence Research*, 27(1):335–380.

H. Poon and P. Domingos. 2006. Sound and efficient inference with probabilistic and deterministic dependencies. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, pages 458–463. AAAI Press.

P. Poupart. 2005. *Exploiting structure to efficiently solve large scale partially observable markov decision processes*. Ph.D. thesis, University of Toronto, Toronto, Canada.

M. Richardson and P. Domingos. 2006. Markov logic networks. *Machine Learningf*, 62(1-2):107–136.

V. Rieser. 2008. *Bootstrapping Reinforcement Learning-based Dialogue Strategies from Wizard-of-Oz data*. Ph.D. thesis, Saarland University, October.

R. Thomason, M. Stone, and D. DeVault. 2006. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. In Donna Byron, Craige Roberts, and Scott Schwenter, editors, *Presupposition Accommodation*. Ohio State Pragmatics Initiative.

B. Thomson and S. Young. 2009. Bayesian update of dialogue state: A pomdp framework for spoken dialogue systems. *Computer Speech & Language*, August.

Ch. Wang, S. Joshi, and R. Khardon. 2007. First order decision diagrams for relational mdps. In *IJCAI'07: Proceedings of the 20th international joint conference on Artifical intelligence*, pages 1095–1100, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

J. Williams and S. Young. 2007. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):231–422.

S. Young, M. Gašić, S. Keizer, F. Mairesse, J. Schatzmann, B. Thomson, and K. Yu. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language*, 24(2):150–174.

Policy activation for open-ended dialogue management

Pierre Lison and **Geert-Jan M. Kruijff**
Language Technology Lab
German Research Centre for Artificial Intelligence (DFKI GmbH)
Saarbrücken, Germany
{plison,gj}@dfki.de

Abstract

An important difficulty in developing spoken dialogue systems for robots is the open-ended nature of most interactions. Robotic agents must typically operate in complex, continuously changing environments which are difficult to model and do not provide any clear, predefined goal. Directly capturing this complexity in a single, large dialogue policy is thus inadequate. This paper presents a new approach which tackles the complexity of open-ended interactions by breaking it into a set of small, independent policies, which can be activated and deactivated at runtime by a dedicated mechanism. The approach is currently being implemented in a spoken dialogue system for autonomous robots.

Introduction

Human-robot interactions (HRI) often have a distinctly open-ended character. In many applications, the robot does not know in advance which goals needs to be achieved, but must discover these as it goes, during the interaction itself. The user might communicate new requests, clarify or modify existing requests, ask questions, or provide the robot with new information at any time. The robotic agent must therefore be capable of handling a wide variety of tasks, some being purely reactive (such as answering a question), some being more deliberative in nature (such as planning a complex sequence of actions towards a long-term goal).

The interaction dynamics are also significantly more difficult to predict in HRI. In classical, slot-filling dialogue applications, the domain provides strong, predefined constraints on how the dialogue is likely to unfold. Interactive robots, on the other hand, usually operate in rich, dynamic environments which can evolve in unpredictable ways. The interaction is therefore much more difficult to model and depends on numerous parameters. (Bohus and Horvitz 2009) provide a review of important technical challenges to address in such kind of open-ended interactions.

Previous work on this issue mostly focussed on techniques for enlarging the state and action spaces to directly capture this complexity. These techniques are usually coupled with mechanisms for factoring (Bui et al. 2010) or abstracting (Young et al. 2010) these large spaces to retain tractability. Applied to human-robot interactions, these approaches unfortunately suffer from two shortcomings: first, the complexity of the dialogue planning problem increases exponentially with the size of the state space, making these approaches difficult to scale. Second, from the viewpoint of the dialogue developer, maintaining and adapting dialogue policies over very large spaces is far from trivial.

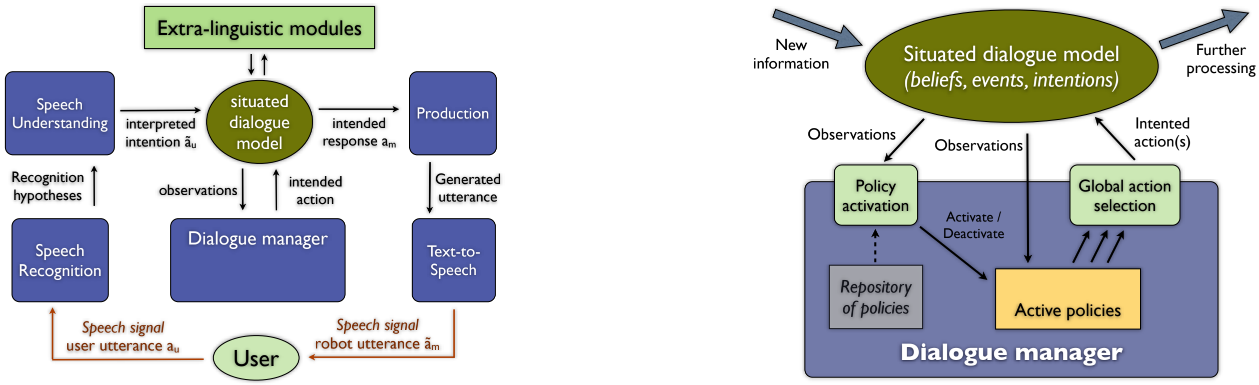
In this paper, we sketch a new approach which is specifically tailored for open-ended interactions. Instead of using one single policy operating over large spaces, the idea is to break up this complexity into a set of shorter, more predictable interactions, which can be activated and deactivated at runtime. The dialogue manager contains a repository of potential policies, and decides which policies to use at a given time via a dedicated *policy activation* mechanism. Several policies can be activated in parallel, and the dialogue manager is responsible for finding the right trade-offs between the activated policies.

Architecture

The general architecture of the dialogue system is illustrated in Figure 1. The architecture revolves around a *situated dialogue model*, which stores various epistemic objects such as beliefs, events and intentions. These epistemic objects are generic representations of the agent's knowledge (e.g. the dialogue history, but also relevant perceptual information), and are expressed as probabilistic relational structures – see (Lison, Ehrler, and Kruijff 2010) for details. The dialogue manager continuously monitors this dialogue model, and reacts upon changes by triggering new observations. These observations can in turn influence the policy activation mechanism (by activating or deactivating policies), or provide direct input to the active policies.

Approach

Instead of designing each dialogue policy by hand – a tedious task given the high levels of noise and uncertainty encountered in HRI –, we define each interaction as a *Partially Observable Markov Decision Process* (POMDP), and apply optimisation algorithms to extract a near-optimal policy for it. POMDPs are a principled mathematical framework for control problems featuring partial observability, stochastic action effects, decision-making over arbitrary horizons, in-



(a) Global schema of the spoken dialogue system.

(b) Detailed schema of the dialogue management module.

Figure 1: Architectural schema, illustrating the dialogue system as a whole (left), and the dialogue management module (right).

complete knowledge of the environment dynamics, and multiple, conflicting objectives. As such, they provide an ideal modelling tool to develop dialogue policies for HRI.

A POMDP is defined as a tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{Z}, T, \Omega, R \rangle$, where $\mathcal{S}$ is the state space; $\mathcal{A}$ is the action space; $\mathcal{Z}$ is the observation space; $T(s, a, s')$ is the transition function from state s to state s' using action a; $\Omega(z, a, s')$ is the observation function for observing z in state s' after performing action a; and $R(s, a)$ is the reward function encoding the utility for the agent of executing action a in state s.

A central idea of POMDP is the assumption that the state is not directly accessible and can only be inferred from observation. Such uncertainty is expressed in the *belief state* b, which is a probability distribution $b : \mathcal{S} \rightarrow [0, 1]$ over possible states. A POMDP policy is then defined over this belief space as a function $\pi : \mathcal{B} \rightarrow \mathcal{A}$ determining the action to perform for each point of the belief space.

Each interaction is modelled in our approach as a separate POMDP. Since these POMDPs have a small state space, a well-defined purpose and a more predictable transition function, they are much easier to model (and estimate from data) than a single, large POMDP.

Policy activation

The policy activation is based on a repository of policies. Each policy is associated with a set of *triggers*. These triggers are reactive to particular changes in the dialogue model – a dialogue policy dealing with replies to user questions will for instance be made reactive to the appearance of a new question onto the dialogue model.

The dialogue model being represented as a probabilistic relational structure, these policy triggers should exploit this rich expressivity. A possibility would be the use of approximate inference algorithms for first-order probabilistic languages such as Markov Logic Networks (Richardson and Domingos 2006) to dynamically construct the set of relevant policies for a given dialogue model.

Action selection with multiple policies

Several dialogue policies can be activated in parallel in the dialogue manager. The agent must therefore be capable of setting the right trade-offs between the various policies.

To this end, we maintain a separate belief point b_i for each activated policy p_i. We define the vector $\mathbf{b}$ as the set of these belief points. Assuming each policy also provides us directly a Q-value function $Q_i(b_i, a)$, we can then compute the best global strategy $\pi(\mathbf{b})$ by maximising the sum of Q-values over the set of activated policies:

$$\pi(\mathbf{b}) = \underset{a \in \mathcal{A}}{\operatorname{argmax}} \sum_{b_i \in \mathbf{b}} Q(b_i, a) \qquad (1)$$

The global action space $\mathcal{A}$ in Eq. (1) is defined as $\cup_i \mathcal{A}_i$. This enables us to select the action which is globally optimal with respect to the set of activated policies.

Conclusion

In this paper, we presented a first sketch of an POMDP-based approach to dialogue management which explicitly handles open-ended interactions by activating and deactivating policies at runtime. Future work will focus on implementing and evaluating the outlined approach in a real-world dialogue system for autonomous robots.

References

Bohus, D., and Horvitz, E. 2009. Dialog in the open world: Platform and applications. In *Proceedings of ICMI'09*.

Bui, T. H.; Zwiers, J.; Poel, M.; and Nijholt, A. 2010. Affective dialogue management using factored pomdps. In *Interactive Collaborative Information Systems*, volume 281 of *SCI*. Berlin: Springer Verlag. 209–238.

Lison, P.; Ehrler, C.; and Kruijff, G.-J. M. 2010. Belief modelling for situation awareness in human-robot interaction. In *Proceedings of RO-MAN 2010*. (in press).

Richardson, M., and Domingos, P. 2006. Markov logic networks. *Machine Learning* 62(1-2):107–136.

Young, S.; Gašić, M.; Keizer, S.; Mairesse, F.; Schatzmann, J.; Thomson, B.; and Yu, K. 2010. The hidden information state model: A practical framework for pomdp-based spoken dialogue management. *Computer Speech & Language* 24(2):150–174.

Continual Processing of Situated Dialogue in Human-Robot Collaborative Activities

Geert-Jan M. Kruijff, Miroslav Janíček and **Pierre Lison**
Language Technology Lab
German Research Center for Artificial Intelligence, DFKI GmbH
{gj,miroslav.janicek,pierre.lison}@dfki.de

Abstract— **The paper presents an implemented approach of processing situated dialogue between a human and a robot. The focus is on task-oriented dialogue, set in the larger context of human-robot collaborative activity. The approach models understanding and production of dialogue to include intension (what is being talked about), intention (the goal of why something is being said), and attention (what is being focused on). These dimensions are directly construed in terms of assumptions and assertions on situated multi-agent belief models. The approach is continual in that it allows for interpretations to be dynamically retracted, revised, or deferred. This makes it possible to deal with the inherent asymmetry in how robots and humans tend to understand dialogue, and the world it is set in. The approach has been fully implemented, and integrated into a cognitive robot. The paper discusses the implementation, and illustrates it in a collaborative learning setting.**

I. Introduction

Particularly in task-oriented dialogues between a human and a robot, there is usually more to dialogue than just understanding words. The robot needs to understand what is being talked about, sure. But she also needs to understand why she was told something. What the human intends her to do with the information, in the larger context of the joint activity both are involved in.

In this paper we see task-oriented dialogue as part of a larger collaborative activity, in which a human and the robot are involved. They are planning together, executing their plans. Dialogue plays a facilitatory role in this. It helps all those involved build up a common ground, and maintain it as plans are executed, and the world around them changes.

We present here an approach that models these aspects of situated, task-oriented dialogue. We provide an algorithm in which dialogue is understood, and generated, by looking at *why* something is being said (intention), *what* that something is about (intension), and *how* that helps to direct our focus (attention). Core to the algorithm is abductive reasoning. This type of reasoning tries to find the best explanation for observations. In our case, it tries to find the best explanation for why something was said (understanding), or how an intention best could be achieved communicatively (generation). Thereby abduction directly works off the situated, multi-agent belief models the robot maintains as part of its understanding of the world, and of the agents acting therein.

Our approach views dialogue from a more intentional perspective, like the work by Grosz & Sidner [6], Lochbaum et al. [10], and most recently Stone et al [14], [15], [16]. Our approach extends that of Stone et al.

Stone et al. formulate an algorithm for collaborative activity, involving abductive reasoning, that forms the basis for our approach. However, they assume that understanding and production are symmetric. What I say is how you understand it. This is optimistic for human-human dialogue, and rather unrealistic for human-robot interaction. Robots hardly ever perfectly understand what is meant. We need to allow for the robot to act upon interpretations even when they are incomplete or uncertain. And, should it turn out the robot has misunderstood what was said, roll dialogue back to a point where she can clarify and correct her understanding.

The approach enables this by introducing *assertions* into our logics. This is inspired by Brenner & Nebel's work on continual planning [3]. An assertion is a content formula that needs to be verified at a later point. In that it is different from a propositional fact, which the robot knows to be either true or false. We can introduce an assertion into an abductive inference to help find an explanation, and thus act upon it. It is just that this is then made contingent on the assertion to become true sooner or later. In this paper, we show how assertions can play a fundamental role in helping a robot and a human to achieve common ground in collaborative activity.

Below, §II provides a brief overview on intentional approaches to dialogue. §III presents the approach. We discuss situated multi-agent belief models, abductive reasoning, and the algorithm for continual processing of collaborative activity. §IV discusses the implementation, and §V illustrates it on working examples from an integrated robot system.

II. Background

Recent theories of dialogue focus on how participants can obtain common ground through alignment [11]. Agents align how they communicate content, what they pay attention to, and what they intend to do next. They base this on how they perceive each other's views on the world.

This works out reasonably well as long as we can assume a more or less common way of "looking" at things. Even when humans normally differ in what they know, can, and intend to do, there is typically a common categorical framework in which they can try to characterize the world, to arrive at a common ground. But this is where a problem arises in communication between a human, and a robot that continuously learns. Because robots tend to see things substantially different from how humans see things. Which is why mechanisms for modeling, and dealing with, such asymmetry in understanding are necessary for situated dialogue. We present here an approach providing such means.

The approach is based on an extension of Stone & Thomason's (S&T) abductive framework [14], [15], [16]. S&T model comprehension and production of dialogue as construction of abductive proofs. Abduction reasons towards an explanation consisting of a consistent context update and possible changes to attentional state. The explanation is based on factual assumptions, observations, and inferred intentions, all included at a context-sensitive cost. They thus place belief context, attention, and intention on a par. This is similar to other intentional approaches to dialogue and discourse, like Grosz & Sidner's [6]. S&T's approach arguably provides more flexibility [16] in that aspects such as reference resolution are dynamically determined through proof, rather than being constrained by hierarchical composition of a context model. For comprehension an abductive proof provides the conditions under which an agent can update her belief model and attentional model with the content for a communicated utterance, and her task model using the inferred intentions underlying the utterance. For production an abductive proof provides the conditions for executing a plan to achieve an intended context- and attentional update in another agent.

We extend S&T in several ways. We expand context [14] to incorporate the types of situated multi-agent beliefs and tasks the robot reasons with in understanding collaboration, and the world as such. We also make S&T's notion of "checkpoints" more explicit. A checkpoint is a means to establish whether assumptions are in fact warranted [16]. Checkpoints introduce a relation between the construction of an explanation, and acting on it. This suggests a similarity to the construction of a plan and the monitoring of its execution. [3] introduce a notion of assertion for continual planning. An assertion poses the availability of future observations, to enable the construction of a continual plan including actions based on such an assertion. Upon execution, assertions are checked and are points for possible plan revision.

We propose to use a similar notion. In an abductive proof, we can include assumptions, observations, and actions at varying costs to infer an explanation. They all contribute facts or outcomes from which further inferences can be drawn. An assertion is a statement whose truth we need to assume, but which we cannot prove or disprove on the current set of beliefs of the agent. Marking assertions turns these statements in an abductive proof into points that warrant explicit verification – i.e. they act as checkpoints. The notions of assertion and checkpoint provide the approach with a fundamental way for dealing with asymmetry in understanding, and resolving it to come to common ground.

III. Approach

A. Modeling multi-agent beliefs

We couch our approach to situated grounding in direct reasoning about the agents' *beliefs*. A belief is an agent's informational state that reflects her understanding of the world and the way it has been talked about. Such an understanding can be acquired through a direct observation, i.e. as a result of a sensoric input, or through communication with other agents, as is the case when engaging in a dialogue. Moreover, these beliefs can explicitly model *common beliefs*, which correspond to the beliefs that are a part of the common ground among a group of agents.

A belief is a formula $\mathsf{K}e/\sigma : \phi$ that consists of three parts: a *content formula* ϕ from a *domain logic* $\mathcal{L}_{\mathsf{dom}}$, the assignment e of the content formula to agents, which we call an *epistemic status* and the *spatio-temporal frame* σ in which this assignment is valid.

We distinguish three classes of epistemic statuses, that give rise to three classes of beliefs:

- **private** belief of agent a, denoted $\{a\}$, comes from *within* the agent a, i.e. it is an interpretation of sensor output or a result of deliberation.
- a belief **attributed** by agent a to other agents $b_1, ..., b_n$, denoted $\{a[b_1, ..., b_n]\}$, is a result of a's deliberation about the mental states of $b_1, ..., b_n$ (e.g. an interpretation of an action that they performed).
- a belief **shared** by the group of agents $a_1, ..., a_m$, denoted $\{a_1, ..., a_m\}$, is common ground among them.

A spatio-temporal frame is a contiguous spatio-temporal interval. The belief is only valid in the spatio-temporal frame σ and frames that are subsumed by σ. This way, spatio-temporal framing accounts for situatedness and the dynamics of the world. The underlying spatio-temporal structure may feature more complex spatial or temporal features.

Finally, the domain logic $\mathcal{L}_{\mathsf{dom}}$ is a propositional modal logic. We do not require $\mathcal{L}_{\mathsf{dom}}$ to have any specific form, except for it to be sound, complete and decidable.

Multiple beliefs form a *belief model*. A belief model is a tuple $\mathbf{B} = (A, S, K, F)$ where A is a set of agents, S is a set of spatio-temporal frames, K is a set of beliefs formed using A and S and $F \subseteq K$ is a set of *activated* beliefs.

Belief models are assigned semantics based on a modal-logical translation of beliefs into a poly-modal logic that is formed as a fusion of KD45$^{\mathsf{C}}_{A}$ (doxastic logic with a common belief operator [4]) for epistemic statuses, K4$_n$ for subsumption-based spatio-temporal reasoning and $\mathcal{L}_{\mathsf{dom}}$ for content formulas. This gives us a straightforward notion of belief model consistency: a belief model is consistent if and only if its modal-logical translation has a model.

The belief model keeps track of the beliefs' evolution in a directed graph called the *history*. The nodes of the history are beliefs and operations on the belief model (such as *retraction*) with (labeled) edges denoting the operations's arguments. The nodes that are beliefs and have no outcoming edges form a consistent, most recent belief model.

B. Attaining common ground

A shared belief of a group G that ϕ implies all private beliefs and all possible attributed beliefs that ϕ within that group. For example, if ϕ is common ground between the human user, h, and robot, r, then (i) implies (ii):

$$\begin{array}{ccl} & & \mathbf{B} \models \mathsf{K}\{\mathsf{r}\}/\sigma : \phi \\ \mathbf{B} \models \mathsf{K}\{\mathsf{r},\mathsf{h}\}/\sigma : \phi & \Rightarrow & \mathbf{B} \models \mathsf{K}\{\mathsf{r}[\mathsf{h}]\}/\sigma : \phi \\ & & \mathbf{B} \models \mathsf{K}\{\mathsf{h}\}/\sigma : \phi \quad * \\ & & \mathbf{B} \models \mathsf{K}\{\mathsf{h}[\mathsf{r}]\}/\sigma : \phi \quad * \\ \text{(i)} & & \qquad \text{(ii)} \end{array}$$

Since (i) and (ii) are inferentially equivalent within belief models, the relation is in fact equivalence. If (ii) holds in the belief model $\mathbf{B}$, it also satisfies (i).

However, the agents' private and attributed beliefs cannot be observed by other agents, they are not ominiscient. The beliefs above marked by asterisk (*) cannot be present in the robot's belief model. The validity of such beliefs can only be *assumed*. An invalidation of the assumptions then invalidates the premise (ii) and thus the conclusion (i). As long as they are not invalidated, agents may act upon them: they may *assume* that common ground has been attained.

But how can these assumptions be in principle mandated or falsified? Given a communication channel C, we consider a class of protocols P_C that supply the means for falsification of the assumptions. If these means are provided, then the protocol is able to reach common ground. We assume that the agents are faithful to Grice's Maxim of Quality [5], i.e. that they are truthful and only say what they believe to be true and for what they have evidence.

C. Abductive inference with assertions

1) Context in abductive inference: Our abductive framework consists of a set of modalised facts $\mathcal{F}$ and a set of rules $\mathcal{R}$. The modal contexts we utilise are the following:

- i – *i*nformation. Used to mark the information that is logically true, e.g. description of relational structures.
- e – *e*vent. Used to denote *events* which the robot is trying to understand or produce.
- γ – intention. Marks the intention of an agent's action. In the interpretation phase, it is used to mark the recognised intention. In the generation phase, it is used as a goal in order to find its best possible realisation.
- a – *a*ttentional state. Marks the formulas that are in the attention span. For beliefs, this corresponds to the notion of *foregrounded* beliefs.
- $\mathsf{k}(e)$ – epistemic status. Assigns the predicate an epistemic status (private/attributed/shared).
- DURING(σ) – spatio-temporal frame. Assigns a spatio-temporal frame to the predicate. Together with $[\mathsf{k}(e)]$, the formulas can then be translated into beliefs.

We also include two "technical" contexts that exploit the ability to bring modularity into logic programming following Baldoni *et al.* [1].

- interpret – understanding phase module.
- generate – generation phase module.

In comparison to S&T's definition of a context [14], we include specific contexts for intentions (γ), epistemic statuses ($\mathsf{k}(e)$) and spatio-temporal frames (DURING(σ)), as well as both "technical" contexts, interpret and generate. While the addition of a context for assigning epistemic statuses and spatio-temporal frames is specific for our purposes and stems from the usage of belief models to model the state of the world and common ground, the addition of the context for distinguishing intentions is more general and allows us to use intentions as an abstract layer.

2) Assertions: We propose a notion of *assertion* for abduction based on *test actions* $\langle F \rangle ?$ [2]. Baldoni *et al.* specify a test as a proof rule. In this rule, a goal F follows from a state $a_1, ..., a_n$ after steps $\langle F \rangle ?, p_1, ..., p_m$ if we can establish F on $a_1, ..., a_n$ with answer σ and this (also) holds in the final state resulting fron executing $p_1, ..., p_m$. Using the notion of context as per above, a test $\kappa : \langle F \rangle ?$ means we need to be able to verify F in context κ. If we only use axioms A, testing is restricted to observability of facts. An embedded implication $D \supset C$ establishes a *local module*: the clauses D can only be used to prove C. Formulating a test over an embedded implication $\mu : (D \supset \langle C \rangle ?)$, we make it explicit that we assume the truth of the statement but require its eventual verification in μ.

Finally, an assertion is the transformation of a test into a partial proof which assumes the verification of the test, while at the same time conditioning the obtainability of the proof goal on the tested statements. Intuitively, $\mu : \langle D \rangle ?$ within a proof $\Pi[\langle D \rangle ?]$ to a goal C turns into $\Pi[D] \rightarrow C \wedge \mu : D$. Should $\mu : D$ not be verifiable, Π is invalidated.

D. Continual collaborative acting (CCA)

1) The algorithm: Our extension of S&T's collaborative acting algorithm [16] uses assertions in abductive inference, to allow for a revision of beliefs once they were falsified. We assume their truth until such a revision occurs. This removes the need for S&T's symmetry assumption. This is represented in the VERIFIABLE-UPDATE operation, below.

Algorithm 1 Continual collaborative acting

```
$\Sigma^\pi = \emptyset$

loop {
Perception
    $e \leftarrow$ SENSE()
    $\langle c', i, \Pi \rangle \leftarrow$ UNDERSTAND$(r, Z(c) \oplus \Sigma^\pi, e)$
    $c \leftarrow$ VERIFIABLE-UPDATE$(c', i, \Pi)$

Determination and Deliberation
    $c' \leftarrow$ ACT-TACITLY$(p, c)$
    $m \leftarrow$ SELECT$(p, c')$
    $\langle i, \Pi \rangle \leftarrow$ GENERATE$(r, c', m, Z(c) \oplus \Sigma^\pi)$

Action
    ACT-PUBLICLY$(a(i))$
    $c \leftarrow$ VERIFIABLE-UPDATE$(c', i, \Pi)$
}
```

2) Verifiable update: The VERIFIABLE-UPDATE operation operates on the belief model and a structure Σ^π that we call *proof stack*. This is an ordered store of abductive proofs that contain assertions that have not been verified or falsified yet. Given the proof Π, it checks whether there is a proof Π' on the stack whose assertions can be verified using the beliefs of

Π. If there are any beliefs in Π' that were falsified, then the Π' should remain on the top: thus, the operation first pushes Π onto the stack and then Π'. The belief model update is then be based on those beliefs from Π that have been assumed in the abductive proof and the asserted beliefs beliefs from Π' that have been verified.

VERIFIABLE-UPDATE returns a consistent belief model. Should there be beliefs in the update that cannot be consistently added to the belief model, the operation retracts some beliefs from the belief model so that the model can be updated and stays as descriptive as possible. The retracted beliefs are added to the stack as assertions so that they can be corrected subsequently, or retracted altogether.

3) Grounding using CCA: If the robot (r) understands the human's (h) claim that ϕ in a frame σ, a proof containing the belief $\mathsf{K}\{\mathsf{r}[\mathsf{h}]\}/\sigma : \phi$ is added to the proof stack as an assertion. If the robot can verify ϕ, then this assertion is removed from the stack; the robot can then assume $\mathsf{K}\{\mathsf{h}\}/\sigma : \phi$ per the Maxim of Quality. Similarly, the human's acceptance of the robot's acknowledgment is a verification of an assertion of on the proof stack, on which the robot (again per Maxim of Quality) can assume the belief $\mathsf{K}\{\mathsf{h}[\mathsf{r}]\}/\sigma : \phi$.

Common ground can then be also assumed as long as these beliefs are not contradicted. Should they be contradicted, VERIFIABLE-UPDATE removes them from the belief model, and the assumption of common ground is no longer valid.

IV. Implementation

A. The architecture

The approach has been fully implemented in a cognitive robot architecture. The cognitive architecture integrates sensory and deliberative information-processing components into a single cognitive system, in a modular fashion. The continual collaborative acting (CCA) is implemented as one of these components.

The design of the system is based on the CoSy Architecture Schema (CAS) [7]. CAS is a set of rules that delimit the design of a distributed information-processing architecture in which the basic processing unit is called a *component*. Components related by their function are grouped into *subarchitectures*. Each subarchitecture is assigned a *working memory*, a blackboard which all the components within the subarchitecture may read or write to. Inter-component and inter-subarchitecture communication is achieved by writing to these working memories. The schema is implemented using the CoSy Architecture Schema Toolkit (CAST).

In our scenario, we use a robot in a table-top scenario, observing and manipulating visual objects. The goal is to build a visual categorical models of the objects in the scene. The robot can interact with a human, e.g. to ask the human for clarification when it is uncertain about its sensory interpretation of the visual input. This clarification is then used to extend or update the visual models.

The scenario involves the subarchitectures for vision [17], communication ("comsys") and binding [8]. Each subarchitecture's working memory contains specialised representations of the information processed by the associated components. The visual working memory contains regions of interest generated by a segmentor and proto-objects generated by interpreting these regions. The communication subarchitecture working memory contains logical forms generated from parsing utterances. The task of the binding subarchitecture [8] is to combine these subarchitecture-specific data representations into a common a-modal one. The binding architecture (henceforth "binder") uses Bayesian networks to derive a probability distribution over the possible combinations and builds and maintains the belief model in a bottom-up fashion.

B. The abducer

The weighted abduction algorithm as formulated by Stickel [13] and later Baldoni *et al.* is straightforward to implement within the logic programming paradigm. We have used Mercury, a purely declarative logic/functional programming language that resembles both Prolog and Haskell but which is compiled rather than interpreted [12].

The abducer rule set is currently static and is common for both the understanding and generation phases of the CCA algorithm in which the abducer is used. We use the two technical modal contexts $\mathsf{interpret}$ and $\mathsf{generate}$ as described above in order to distinguish rules that can only be applied in one of the phases.

C. The CCA component

1) Understanding an observed action: The CCA is implemented as a component within the communication subarchitecture. It is notified of any logical form corresponding to a recognized utterance together with a list of possible bindings of its referential expressions to binder unions appearing on its working memory. This is interpreted as an event observation in the perception phase of the CCA loop. Each of the possible bindings is assigned a probability by the binder. This information is used by the abducer to find the best explanation of the entire utterance.

Currently, the only action that is interpreted as an event by the CCA is a dialogue act by the user. However, the framework can accomodate events recognized by other modalities (such as vision) as well.

2) Clarification requests: If a modality (vision in our scenario) needs to find out more about a certain object from the user, it writes a *clarification request* to the comsys working memory. This is picked up by the CCA, interpreted as a tacit action within the CCA loop. It makes the robot generate a context-aware clarification question. This results in the question core to appear onto the proof stack as an assertion, thus making it a potential belief model update.

3) Verification of asserted beliefs: Modalities can verify the asserted beliefs. For instance, if the user says "the box is blue" (an assertion about the box) the vision subarchitecture is notified of the new assertion appearing on the proof stack and can check whether the information is consistent with its visual model and if not, whether the visual model can be extended or updated. If so, the subarchitecture updates the visual model and notifies the CCA component, which then (as a result of a tacit action) generates an appropriate feedback such as “yes, i can see that”.

This change then percolates into the vision working memory and triggers the binder to form an updated belief model.

4) Acting: The public action selection in our implementation is done by using a finite-state automaton that maps recognised communicative intentions to intentions to act. In the future, we would like to employ a POMDP-based action selection [18] rather than a finite-state automaton.

The action is then abductively transformed (GENERATE) to a structure that can be written to a corresponding working memory. Currently, our system only supports communicative actions using the communication subarchitecture.

V. Experimentation

We illustrate our approach on a scenario in which a robot gradually learns more about visual objects it sees (Figure 1). The interaction is mixed-initiative. Typically the robot drives the dialogue by asking more about what it does not understand. The success of such a dialogue depends strongly on whether the human and the robot can arrive at common ground. This is key in several respects. One, the robot needs to be able to consistently integrate information it gets through dialogue, into its belief models and visual models. This may concern positive information, resulting in an update of its models, or negative information. In the latter case, the robot needs to revise its belief model, unlearn the incorrect information, and then gather the correct information to learn a better model. Below, we illustrate how the robot can deal with these.

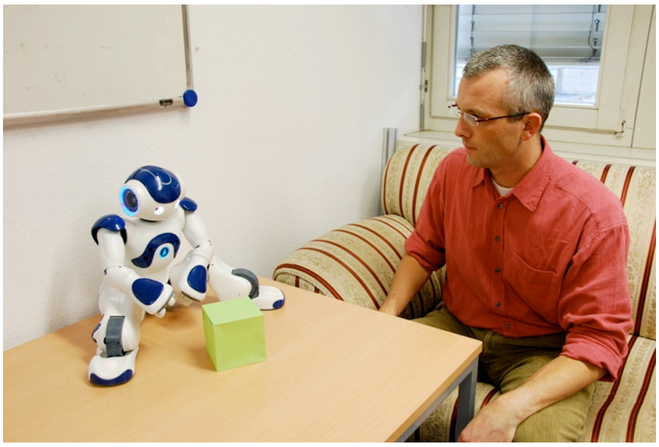


Fig. 1. The setting of the table-top scenario

A. Updating beliefs with human information

As the robot observes a new object in the visual scene, it creates a private belief, (1), about this object. The belief is explicitly connected to the a-modal representation u of the object.

$$\mathsf{K}\{\mathsf{r}\} : @_u\mathbf{object} \quad (1)$$

After the human has placed the object, he indicates what it is: “This is a box.” The robot creates a semantic representation of this utterance. It uses this information to create a belief it attributes to the human (2): The robot believes the human believes this is a box. This belief is also connected to the visual object, and thus to the robot’s private belief.

$$\begin{array}{rll} \mathsf{K}\{\mathsf{r}\} : & @_u\mathbf{object} & (1) \\ \mathsf{K}\{\mathsf{r[h]}\} : & @_u\langle \mathit{Type}\rangle\mathbf{box} & (2)\text{, assertion} \end{array}$$

The robot can use the type-information to consistently update its visual models. The vision subarchitecture thereby positively verifies the information, represented by a private belief (3) in the belief model.

$$\begin{array}{rll} \mathsf{K}\{\mathsf{r}\} : & @_u\mathbf{object} & (1) \\ \mathsf{K}\{\mathsf{r[h]}\} : & @_u\langle \mathit{Type}\rangle\mathbf{box} & (2) \\ \mathsf{K}\{\mathsf{r}\} : & @_u\langle \mathit{Type}\rangle\mathbf{box} & (3) \end{array}$$

If the robot then notifies the human of this verification, it can lift the attributed belief (2) with the private belief (3) to a shared belief (4), assuming the information to be grounded.

$$\begin{array}{rll} \mathsf{K}\{\mathsf{r}\} : & @_u\mathbf{object} & (1) \\ \mathsf{K}\{\mathsf{r,h}\} : & @_u\langle \mathit{Type}\rangle\mathbf{box} & (4) \end{array}$$

The robot infers that a box typically has a color – but it does not know what color the box is. Vision accordingly poses an information request to the architecture, which dialogue can help resolve. The request is based on a private belief of the form $\mathsf{K}\{\mathsf{r}\} : @_u\langle \mathit{Color}\rangle\mathbf{unknown}$. Stating color as an assertion means the robot needs information from the human to “verify” it, i.e. fill the gap.

$$\begin{array}{rll} \mathsf{K}\{\mathsf{r}\} : & @_u\mathbf{object} & (1) \\ \mathsf{K}\{\mathsf{r,h}\} : & @_u\langle \mathit{Type}\rangle\mathbf{box} & (4) \\ \mathsf{K}\{\mathsf{r}\} : & @_u\langle \mathit{Color}\rangle\mathbf{unknown} & (5)\text{, assertion} \end{array}$$

The human responds cooperatively, saying “It is green.” Abduction yields a proof that this information in principle could answer the question the robot just raised [9]. This gives rise to an attributed belief, with the color information: $\mathsf{K}\{\mathsf{r[h]}\} : @_u\langle \mathit{Color}\rangle\mathbf{green}$.

$$\begin{array}{rll} \mathsf{K}\{\mathsf{r}\} : & @_u\mathbf{object} & (1) \\ \mathsf{K}\{\mathsf{r,h}\} : & @_u\langle \mathit{Type}\rangle\mathbf{box} & (4) \\ \mathsf{K}\{\mathsf{r}\} : & @_u\langle \mathit{Color}\rangle\mathbf{unknown} & (5)\text{, assertion} \\ \mathsf{K}\{\mathsf{r[h]}\} : & @_u\langle \mathit{Color}\rangle\mathbf{green} & (6)\text{, assertion} \end{array}$$

If vision can now use the information in the updated belief to consistently extend its models, it verifies the assertion. The belief attains shared status.

$$\begin{array}{rll} \mathsf{K}\{\mathsf{r}\} : & @_u\mathbf{object} & (1) \\ \mathsf{K}\{\mathsf{r,h}\} : & @_u\langle \mathit{Type}\rangle\mathbf{box} & (4) \\ \mathsf{K}\{\mathsf{r,h}\} : & @_u\langle \mathit{Color}\rangle\mathbf{green} & (7) \end{array}$$

B. Revising the belief model

Now, assume that instead of not knowing the color at all, the robot hypothesizes that the box is yellow. In this case, it asks “Is the box yellow?” based on the belief $\mathsf{K}\{\mathsf{r}\} : @_u\langle \mathit{Color}\rangle\mathbf{yellow}$. If the human now replies with “No, it is not yellow,” the robot first creates a corresponding negative belief, and unlearns the classification from its visual models. The negative belief is shared. Next up, it still wants to know what color the box has. The belief model then contains both the shared negative belief (8) and the open private belief about the now unknown color (9).

$$\begin{aligned} \mathsf{K}\{\mathsf{r}\} &: @_u\mathbf{object} && (1)\\ \mathsf{K}\{\mathsf{r},\mathsf{h}\} &: @_u\langle \mathit{Type}\rangle\mathbf{box} && (4)\\ \mathsf{K}\{\mathsf{r},\mathsf{h}\} &: @_u\langle \mathit{Color}\rangle\mathrm{not}(\mathbf{yellow}) && (8)\\ \mathsf{K}\{\mathsf{r}\} &: @_u\langle \mathit{Color}\rangle\mathbf{unknown} && (9)\text{, assertion} \end{aligned}$$

The dialogue now returns to a flow similar to the above. If the human responds with "It is green," the robot can again update its belief model and visual models. The robot now holds both a negative shared belief about color $(\mathrm{not}(\mathbf{yellow}))$ and a positive shared belief about it $(\mathbf{green})$.

$$\begin{aligned} \mathsf{K}\{\mathsf{r},\mathsf{h}\} &: @_u\langle \mathit{Type}\rangle\mathbf{box} && (4)\\ \mathsf{K}\{\mathsf{r},\mathsf{h}\} &: @_u\langle \mathit{Color}\rangle\mathrm{not}(\mathbf{yellow}) && (8)\\ \mathsf{K}\{\mathsf{r},\mathsf{h}\} &: @_u\langle \mathit{Color}\rangle\mathbf{green} && (10) \end{aligned}$$

All of these beliefs are connected, being anchored to the visual referent we have been talking about. This connection provides a belief history. The robot not only has its current beliefs, it can also introspect how it got there. If the human would now ask, for example to test, whether the robot still thinks whether the object is yellow, the robot can reply "No. It is green." This makes fully transparent the chain of shared beliefs that the robot has, pertaining to the box object.

VI. Conclusions

We presented an approach to processing situated dialogue in human-robot interaction, as set in a larger collaborative activity. The approach both looks at what utterances are about, and why they are or should be uttered: Intension and intention are put on a par. The approach uses weighted abduction to drive processing. This allows for a smooth integration with probabilistic interpretation hypotheses we get from other forms of processing, e.g. binding or vision.

Currently, we are investigating how we can combine this approach with plan- and intention recognition to achieve a close integration with collaborative action planning, and with POMDP-based action selection. The latter would help us to select actions even when interpretation does not yield enough information to completely interpret an utterance.

VII. Acknowledgments

The research leading to these results has received funding from the European Union's 7th Framework Programme [FP7/2007-2013] under grant agreement No. 215181 (CogX).

References

[1] M. Baldoni, L. Giordano, and A. Martelli. A Modal Extension of Logic Programming: Modularity, Beliefs and Hypothetical Reasoning. *Journal of Logic and Computation*, 8(5):597–635, 1998.

[2] M. Baldoni, L. Giordano, A. Martelli, and V. Patti. A Modal Programming Language for Representing Complex Actions. In A. Bonner, B. Freitag, and L. Giordano, editors, *Proceedings of the 1998 JICSLP'98 Post-Conference Workshop on Transactions and Change in Logic Databases (DYNAMICS'98)*, pages 1–15, 1998.

[3] M. Brenner and B. Nebel. Continual planning and acting in dynamic multiagent environments. *Journal of Autonomous Agents and Multiagent Systems*, 2008.

[4] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[5] H. P. Grice. Logic and conversation. *Syntax and Semantics*, 3:41–58, 1975.

[6] B.J. Grosz and C.L. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[7] Nick Hawes and Jeremy Wyatt. Engineering intelligent information-processing systems with cast. *Advanced Engineering Infomatics*, 24(1):27–39, To Appear.

[8] H. Jacobsson, N.A. Hawes, G.J.M. Kruijff, and J. Wyatt. Crossmodal content binding in information-processing architectures. In *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Amsterdam, The Netherlands, March 12–15 2008.

[9] G.J.M. Kruijff and M. Brenner. Phrasing questions. In *Proceedings of the AAAI 2009 Spring Symposium on Agents That Learn From Humans*, 2009.

[10] K. Lochbaum, B.J. Grosz, and C.L. Sidner. Discourse structure and intention recognition. In R. Dale, H. Moisl, , and H. Somers, editors, *A Handbook of Natural Language Processing: Techniques and Applications for the Processing of Language as Text*. Marcel Dekker, New York, 1999.

[11] M.J. Pickering and S. Garrod. Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, 27:169–225, 2004.

[12] Zoltan Somogyi, Fergus Henderson, and Thomas Conway. Mercury: An Efficient Purely Declarative Logic Programming Language. In *Proceedings of the Australian Computer Science Conference*, pages 499–512, Feb 1995.

[13] Mark E. Stickel. A prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation. Technical Report 451, AI Center, SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, Sep 1988.

[14] M. Stone and R.H. Thomason. Context in abductive interpretation. In *Proceedings of EDILOG 2002: 6th workshop on the semantics and pragmatics of dialogue*, 2002.

[15] M. Stone and R.H. Thomason. Coordinating understanding and generation in an abductive approach to interpretation. In *Proceedings of DIABRUCK 2003: 7th workshop on the semantics and pragmatics of dialogue*, 2003.

[16] R.H. Thomason, M. Stone, and D. DeVault. Enlightened update: A computational architecture for presupposition and other pragmatic phenomena. In D. Byron, C. Roberts, and S. Schwenter, editors, *Presupposition Accommodation*. to appear.

[17] Alen Vrečko, Danijel Skočaj, Nick Hawes, and Aleš Leonardis. A computer vision integration model for a multi-modal cognitive system. In *IEEE/RSJ International Conference on Intelligent RObots and Systems*, pages 3140–3147, 2009.

[18] Steve Young, Milica Gašić, Simon Keizer, François Mairesse, Jost Schatzmann, Blaise Thomson, and Kai Yu. The Hidden Information State Model: a practical framework for POMDP-based spoken dialogue management. *Computer Speech and Language*, 24(2):150–174, 2009.

Belief Modelling for Situation Awareness in Human-Robot Interaction

Pierre Lison, Carsten Ehrler and Geert-Jan M. Kruijff

Abstract— **To interact naturally with humans, robots needs to be aware of their own surroundings. This awareness is usually encoded in some implicit or explicit representation of the situated context. In this paper, we present a new framework for constructing rich belief models of the robot's environment. Key to our approach is the use of *Markov Logic* as a unified framework for inference over these beliefs. Markov Logic is a combination of first-order logic and probabilistic graphical models. Its expressive power allows us to capture both the rich relational structure of the environment and the uncertainty arising from the noise and incompleteness of low-level sensory data. The constructed belief models evolve dynamically over time and incorporate various contextual information such as spatio-temporal framing, multi-agent epistemic status, and saliency measures. Beliefs can also be referenced and extended "top-down" via linguistic communication. The approach is being integrated into a cognitive architecture for mobile robots interacting with humans using spoken dialogue.**

I. Introduction

The situated context plays a central role in human-robot interaction (HRI). To be able to interact naturally with humans, robots needs to be aware of their own environment. This situation awareness is generally expressed in some sort of *belief models* in which various aspects of the external reality are encoded. Such belief models provide an explicit or implicit representation for the current state of the world, from the robot's viewpoint. They therefore serve as a representational backbone for a wide range of high-level cognitive capabilities related to reasoning, planning and learning in complex and dynamic environments. They are also essential for the robot to verbalise its own knowledge.

In speech-based HRI, critical tasks in dialogue understanding, management and production are directly dependent on such belief models. Examples are context-sensitive speech recognition [15], reference resolution and generation in small- [11] and large-scale space [24], spoken dialogue parsing [14] and interpretation [20], dialogue management [23], user-tailored response generation [22], and contextually appropriate intonation patterns [13]. Contextual knowledge is also a prerequisite for the dynamic adaptation of the robot's behaviour to different environments and interlocutors [3].

Belief models are usually expressed as high level symbolic representations merging and abstracting information over multiple modalities. For HRI, the incorporated knowledge might include (inter alia): entities in the visual scene, spatial structure, user profiles (intentional and attentional state, preferences), dialogue histories, and task models (what is to be done, which actions are available).

The construction of such belief models raises two important issues. The first question to address is how these high-level representations can be reliably abstracted from low-level sensory data [1], [18]. To be meaningful, most symbolic representations should be *grounded* in (subsymbolic) sensory inputs [19]. This is a difficult problem, partly because of the noise and uncertainty contained in sensory data (partial observability), and partly because the connection between low-level perception and high-level symbols is typically difficult to formalise in a general way [6].

The second issue relates to how information arising from different modalities and time points can be efficiently *merged* into unified multi-modal structures [12], and how these inputs can refine and constrain each other to yield improved estimations, over time. This is the well-known engineering problem of multi-target, multi-sensor data fusion [5].

Belief models are thus the final product of an iterative process of information *fusion*, *refinement* and *abstraction*. Typical HRI environments are challenging to model, being simultaneously *complex*, *multi-agent*, *dynamic* and *uncertain*. Four requirements can be formulated:

1) HRI environments are complex and reveal a large amount of internal structure (for instance, spatial relations between entities, or groupings of objects). The formal representations used to model them must therefore possess the expressive power to reflect this rich relational structure.
2) Interactive robots are made for multi-agent settings. Making sense of communicative acts requires the ability to distinguish between one's own knowledge (what I believe), knowledge attributed to others (what I think the others believe), and shared common ground knowledge (what we believe as a group).
3) Situated interactions are *dynamic* and evolve over time. The incorporation of spatio-temporal framing is thus necessary to go beyond the "here-and-now" and be capable of linking the present with (episodic) memories of the past and anticipation of future events.
4) And last but not least, due to the partial observability of most contextual features, it is crucial that belief models incorporate an explicit account of *uncertainties*.

Orthogonal to these "representational" requirements, crucial performance requirements must also be adressed. To keep up with a continuously changing environment, all operations on belief models (updates, queries, etc.) must be performed under soft real-time constraints.

This work was supported by the EU FP7 ICT Integrated Project "*CogX: cognitive systems that self-understand and self-extend*" (FP7-ICT- 215181).

Pierre Lison, Carsten Ehrler and Geert-Jan M. Kruijff are with the German Research Centre for Artificial Intelligence (DFKI GmbH), Language Technology Lab, Saarbrücken, Germany. {plison,carsten.ehrler,gj}@dfki.de

This paper presents ongoing work on a new approach to multi-modal situation awareness which attempts to address these requirements. Key to our approach is the use of a first-order probabilistic language, *Markov Logic* [17], as a unified representation formalism to perform various kind of inference over rich, multi-modal models of context. Markov Logic is a combination of first-order logic and probabilistic modelling. As such, it provides an elegant account of both the uncertainty and complexity of situated human-robot interactions. Our approach departs from previous work such as [9] or [18] by introducing a much richer modelling of multi-modal beliefs. Multivariate probability distributions over possible values are used to account for the partial observability of the data, while the first-order expressivity of Markov Logic allows us to consisely describe and reason over complex relational structures. As we shall see, these relational structures are annotated with various contextual information such as spatio-temporal framing (where and when is the entity assumed to exist), epistemic status (for which agents does this belief hold), and saliency (how prominent is the entity relative to others). Furthermore, performance requirements can be addressed with approximation algorithms for probabilistic inference optimised for Markov Logic [17], [16]. Such algorithms are crucial to provide an upper bound on the system latency and thus preserve its efficiency and tractability.

The rest of this paper is structured as follows. Section II provides a brief introduction to Markov Logic, the framework used for belief modelling. Section III details our approach in terms of architecture, representations, and processing operations. Section IV discusses further aspects of our approach. Section V concludes and provides directions for future work.

II. Background

Markov logic combines first-order logic and probabilistic graphical models in a unified representation [17]. A *Markov logic network* L is defined as a set of pairs (F_i, w_i), where F_i is a first-order formula and $w_i \in \mathbb{R}$ is the associated weight of that formula. A Markov logic network can be interpreted as a *template* for constructing Markov networks, which in turn can be used to perform probabilistic inference over the relational structure defined by the set of formulas F_i.

A. Markov Network

A Markov network G, also known as a *Markov random field*, is an undirected graphical model [10] for the joint probability distribution of a set of random variables $X = (X_1, \ldots, X_n) \in \mathcal{X}$. The network G contains a node for each random variable X_i. The joint probability of a Markov network is defined as such:

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}) \qquad (1)$$

where $\phi_k(x_{\{k\}})$ is a *potential function* mapping the state of a clique[1] k to a non-negative real value. Z is a normalization constant (known as *partition function*).

Alternatively, the potential function ϕ_k in (1) can be replaced by an exponentiated weighted sum over real-valued feature functions f_j:

$$P(X = x) = \frac{1}{Z} e^{\left(\sum_j w_j f_j(x)\right)} \qquad (2)$$

B. Ground Markov Network

Recall that a Markov logic network L is a set of pairs (F_i, w_i). If in addition to L we also specify a set of constants $C = \{c_1, c_2, ..., c_{|C|}\}$, one can generate a *ground Markov network* $M_{L,C}$ as follows:

1) For each possible predicate grounding over the set C, there is a binary node in $M_{L,C}$. The value of the node is true iff the ground predicate is true.
2) For every formula F_i, there is a feature f_j for each possible grounding of F_i over C. The value of the feature $f_i(x)$ is 1 if F_i is true given x and 0 otherwise. The weight of the feature corresponds to the weight w_i associated with F_i.

The graphical representation of $M_{L,C}$ contains a node for each ground predicate. Furthermore, each formula F_i defines a set of cliques j with feature f_j over the set of distinct predicates occurring in F_i. For further details see [17].

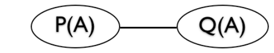


Fig. 1. Example (adapted from [17]) of a ground Markov Network $M_{L,C}$ given the Markov logic network $L = (\forall x.P(x) \vee Q(x), w)$ and $C = \{A\}$. It contains a single clique with feature f. The value of f is 1 for the three worlds $(P(A), Q(A)), (\neg P(A), Q(A)), (P(A), \neg Q(A))$. Following Eq. (3), the probability of each of these worlds is e^w/Z, where $Z = e^w + 1$. For the last world $(\neg P(A), \neg Q(A))$ the formula is false ($f = 0$) and its probability is $1/Z$ (thus tending to 0 as $w \rightarrow \infty$).

C. Inference

Once a Markov network $M_{L,C}$ is constructed, it can be exploited to perform conditional inference over the relational structure defined by L. Following (1), the joint probability distribution of a ground Markov network $M_{L,C}$ is given by

$$P(X = x) = \frac{1}{Z} \prod_i \phi_i(x_{\{k\}})^{n_i(x)} = \frac{1}{Z} e^{\left(\sum_i w_i n_i(x)\right)} \qquad (3)$$

The function $n_i(x)$ in (3) counts the number of true groundings of the formula F_i in $M_{L,C}$ given x. Due to the normalization term Z, exact inference is in general infeasible. However, efficient algorithms for probabilistic inference such as Markov Chain Monte Carlo (MCMC) can then be used to yield approximate solutions [16].

D. Learning

The weight w_i in a Markov logic network encode the “strength” of its associated formula F_i. In the limiting case, where $\lim_{w_i \rightarrow \infty}$, the probability of a world violating F_i has zero probability. For smaller values of the weight, worlds violating the formula will have a low, but non-zero probability. Weights can be learned on training samples using classical gradient-based techniques, or sampling.

[1] In graph theory, a clique is a fully connected subgraph. That is, a subset of nodes where each node is connected with each other.

III. Approach

We now describe our approach to belief modelling for situation awareness. We detail the architecture in which our system is integrated, the representations we used, and the processing components operating on them.

A. Architecture

Our approach is being developed as part of a distributed cognitive architecture for autonomous robots in open-ended environments [7]. The architecture has been applied to various scenarios such as visual learning and object manipulation in a tabletop scene [21] and exploration of indoor environments for human-augmented mapping [8].

Our approach to rich multi-modal belief modelling is implemented in a specific module called the "*binder*". The binder is directly connected to all subsystems in the architecture (i.e. vision, navigation, manipulation, etc.), and serves as a central hub for the information gathered about the environment. The core of the binder system is a shared *working memory* where beliefs are formed and refined based on incoming perceptual inputs. Fig. 2 illustrates the connection between the binder and the rest of the architecture.

B. Representation of beliefs

Each unit of information describing an entity[2] is expressed as a *probability distribution* over a space of alternative values. These values are formally expressed as propositional logical formulae. Such unit of information is called a **belief**.

Beliefs are constrained both *spatio-temporally* and *epistemically*. They include a frame stating where and when the described entity is assumed to exist, and an epistemic status stating for which agent(s) the information contained in the belief holds. Finally, beliefs are also given an *ontological category* used to sort the various belief types.

Formally, a belief is a tuple $\langle i, e, \sigma, c, \boldsymbol{\delta} \rangle$, where i is the belief identifier, e is an epistemic status, σ a spatio-temporal frame, c an ontological category, and $\boldsymbol{\delta}$ is the belief content itself. The content $\boldsymbol{\delta}$ is typically defined by a list of features. For each feature, we have a (continuous or discrete) distribution over alternative values. Fig. 3(a) provides a schematic illustration of a belief.

In addition, beliefs also contain bookkeeping information detailing the history of their formation. This is expressed as pointers to the belief ancestors (i.e. the beliefs which contributed to the emergence of this particular belief) and offspring (the ones which themselves emerged out of it).

The spatio-temporal frame σ defines a a probability distribution over the existence of the entity in a given temporal and spatial domain. The frame can for instance express that a particular visual object is thought to exist (with a given probability) in the world at a location l and in a temporal interval $[t_1, t_2]$.

The epistemic status e for an agent a can be either:

- *private*: denoted $\{a\}$, is a result of agent a's perception of the environment;
- *attributed*: denoted $\{a[b_1, ..., b_n]\}$, is a's conjecture about the mental states of other agents $b_1, ..., b_n$, usually resulting from communicative acts.
- *shared*: denoted $\{a_1, ..., a_m\}$, is information which is part of the common ground for the group[2].

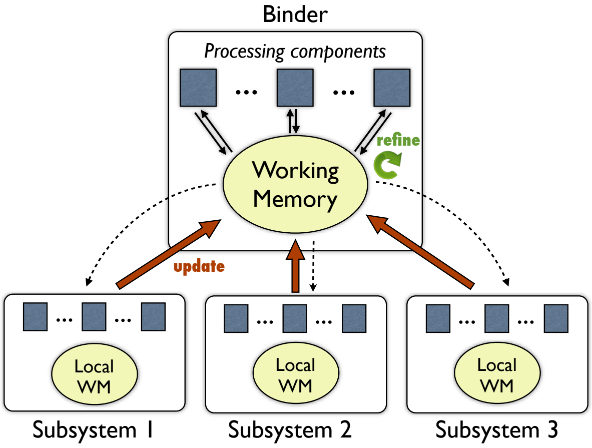


Fig. 2. Schema of the cognitive architecture in relation with the binder system and its working memory

As a brief illustration, assume a belief b_i defined as

$$\langle i, \{\text{robot}\}, \sigma_i, \mathsf{visualobject}, \boldsymbol{\delta}_i \rangle \quad (4)$$

where the spatio-temporal frame σ_i can be a normal distribution over 3D space combined with a temporal interval:

$$\sigma_i = (\mathcal{N}_3(\boldsymbol{\mu}, \boldsymbol{\Sigma}), [t_1, t_2]) \quad (5)$$

and with the content $\boldsymbol{\delta}_i$ being composed of two features:

$$\langle \text{Label} \rangle = \{(\mathsf{mug}, 0.7), (\mathsf{Unknown}, 0.3)\} \quad (6)$$
$$\langle \text{Colour} \rangle = \{(\mathsf{red}, 0.8), (\mathsf{orange}, 0.2)\} \quad (7)$$

Note that the probability distributions between features are by default assumed to be conditionally independent.

Feature values can be either discrete (as for categorical knowledge) or continuous (as for real-valued measures). A feature value can also be a pointer to another formula:

$$\langle \text{Location} \rangle \; k \quad (8)$$

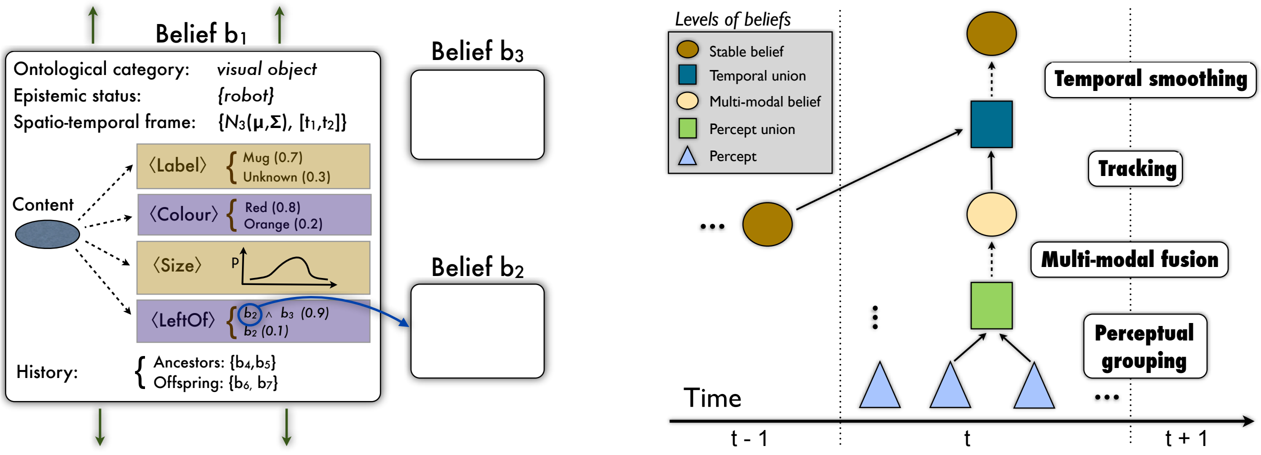
where k points to another belief. Such pointers are crucial to capture relational structures between entities.

Converting the probability distribution $\boldsymbol{\delta}$ into Markov Logic is relatively straightforward. Modal operators are translated into first-order predicates and nominals into contants. A (sub-)formula $\langle \text{Colour} \rangle$ blue with probability p_1 for a belief i is therefore expressed as:

$$w_1 \quad \texttt{Colour}(\texttt{I}_1, \texttt{I}_2) \wedge \texttt{Blue}(\texttt{I}_2) \quad (9)$$

where the weight $w_1 = \log \dfrac{p_1}{1 - p_1}$.

[2] The term "entity" should be understood here in a very general sense. An entity can be an object, a place, a landmark, a person, etc.


Belief b₁
Ontological category: visual object
Epistemic status: {robot}
Spatio-temporal frame: {N₃(μ,Σ), [t₁,t₂]}
Content
⟨Label⟩ Mug (0.7) Unknown (0.3)
⟨Colour⟩ Red (0.8) Orange (0.2)
⟨Size⟩ P
⟨LeftOf⟩ b₂ ∧ b₃ (0.9) b₂ (0.1)
History: Ancestors: {b₄,b₅} Offspring: {b₆, b₇}
Belief b₃
Belief b₂
Levels of beliefs
Stable belief
Temporal union
Multi-modal belief
Percept union
Percept
Temporal smoothing
Tracking
Multi-modal fusion
Perceptual grouping
Time
t - 1
t
t + 1

(a) Schematic view of a belief representation.

(b) Bottom-up belief formation.

Fig. 3. Rich belief modelling for HRI: representations (left) and processing (right).

C. Levels of beliefs

The beliefs constructed and refined in the binder can be of different types. The number and nature of these types depend on the application domain. We discuss here four levels which are common for cognitive robotic architectures:

1) The lowest-level type of beliefs is the *percept*, which is a uni-modal representation of a given entity in the environment. Perceptual beliefs are inserted onto the binder by the various subsystems included in the architecture. The epistemic status of a percept is private per default, and the temporal frame is constrained to the present time-point.
2) If several percepts (from distinct modalities) are assumed to originate from the same entity, they can be grouped into a *percept union*. A percept union is just another belief, whose content is the combination of all the features from the included percepts.
3) The features of a percept union can be abstracted using multi-modal fusion and yield a *multi-modal belief*.
4) If the current multi-modal belief (which is constrained to the present spatio-temporal frame) is combined with beliefs encoded in past or future spatio-temporal frames, it forms a *temporal union*.
5) Finally, the temporal unions can be refined *over time* to improve the estimations, leading to a *stable belief*, which is both multi-modal and spans an extended temporal frame.

Since beliefs can point to each other, such models are able to capture relational structures of arbitrary complexity. Beliefs can also express past or future knowledge (i.e. memories and anticipations). That is, beliefs need not be directly grounded in the "here-and-now" observations.

D. Iterative belief refinement

We now turn our attention to the way stable beliefs can be constructed bottom-up from the initial input provided by the perceptual beliefs. The formation of stable beliefs proceeds in four consecutive steps: (1) *perceptual grouping*, (2) *multi-modal fusion*, (3) *tracking* and (4) *temporal smoothing*. Fig. 3(b) provides a graphical illustration of this process.

1) Perceptual grouping: The first step is to decide which percepts from different modalities belong to the same real-world entity, and should therefore be grouped into a belief. For a pair of two percepts p_1 and p_2, we infer the likelihood of these two percepts being generated from the same underlying entity in the real-world. This is realised by checking whether their respective features *correlate* with each other.

The probability of these correlations are encoded in a Markov Logic Network. The formulae might for instance express a high compatibility between the haptic feature "shape: cylindrical" and the visual feature "object: mug" (since most mugs are cylindrical), but a very low compatibility between the features "shape: cylindrical" and "object: ball". Eq. 10 illustrates the correlation between the cylindrical shape (Cyl) and the object label "mug" (Mug).

$$
\begin{aligned}
w_i \qquad & \exists i,j \;\; \mathtt{Shape}(\mathtt{x},\mathtt{i}) \wedge \mathtt{Cyl}(\mathtt{i}) \;\; \wedge \\
& \mathtt{Label}(\mathtt{y},\mathtt{j}) \wedge \mathtt{Mug}(\mathtt{j}) \rightarrow \mathtt{Corr}_\mathtt{i}(\mathtt{x},\mathtt{y}) \qquad (10)
\end{aligned}
$$

A grouping of two percepts will be given a high probability if one or more feature pairs correlate with each other, and there are no incompatible feature pairs. This process is triggered at each insertion or update of percepts. Its outcome is a probability distribution over possible percept unions.

2) Multi-modal fusion: We want multi-modal beliefs to go beyond the simple superposition of isolated modal contents. Multi-modal information should be *fused*. In other words, the modalities should co-constrain and refine each other, yielding new multi-modal estimations which are globally more accurate than the uni-modal ones. We are not talking here about low-level fusion on a metric space, but about fusion based on conceptual structures. These approaches should be seen as complementary with each other.

Multi-modal fusion is also specified in a Markov Logic Network. As an illustration, assume a multi-modal belief B with a predicate Position(B, loc) expressing the positional coordinates of an entity, and assume the value loc can be estimated via distinct modalities a and b by way of two predicates $\texttt{Position}_{\texttt{(a)}}(\texttt{U},\texttt{loc})$ and $\texttt{Position}_{\texttt{(b)}}(\texttt{U},\texttt{loc})$ included in a percept union $\texttt{U}$.

$$w_i \quad \texttt{Position}_{\texttt{(a)}}(\texttt{U},\texttt{loc}) \rightarrow \texttt{Position}(\texttt{B},\texttt{loc}) \tag{11}$$

$$w_j \quad \texttt{Position}_{\texttt{(b)}}(\texttt{U},\texttt{loc}) \rightarrow \texttt{Position}(\texttt{B},\texttt{loc}) \tag{12}$$

The weights w_i and w_j specify the relative confidence of the modality-specific measurements.

3) Tracking: Environments are dynamic and evolve over time – and so should beliefs. Analogous to perceptual grouping which seeks to bind observations over modalities, tracking seeks to bind beliefs *over time*. Both past beliefs (memorisation) and future beliefs (anticipation) are considered. The outcome of the tracking step is a distribution over temporal unions, which are combinations of beliefs from different spatio-temporal frames.

The Markov Logic Network for tracking works as follows. First, the newly created belief is compared to the already existing beliefs for similarity. The similarity of a pair of beliefs is based on the correlation of their content (and spatial frame), plus other parameters such as the time distance between beliefs. If two beliefs $\texttt{B}_1$ and $\texttt{B}_2$ turn out to be similar, they can be grouped in a temporal union $\texttt{U}$ whose temporal interval is defined as $[\texttt{start}(\texttt{B}_1), \texttt{end}(\texttt{B}_2)]$.

4) Temporal smoothing: Finally, temporal smoothing is used to refine the estimates of the belief content *over time*. Parameters such as recency have to be taken into account, in order to discard outdated observations.

The Markov Logic Network for temporal smoothing is similar to the one used for multi-modal fusion:

$$w_i \quad \texttt{Position}_{\texttt{(t-1)}}(\texttt{U},\texttt{loc}) \rightarrow \texttt{Position}(\texttt{B},\texttt{loc}) \tag{13}$$

$$w_j \quad \texttt{Position}_{\texttt{(t)}}(\texttt{U},\texttt{loc}) \rightarrow \texttt{Position}(\texttt{B},\texttt{loc}) \tag{14}$$

IV. Extensions

A. Salience modelling

The belief formula of an entity usually contains a specific feature representing its *salience*. The salience value gives an estimate of the "prominence" or quality of standing out of a particular entity relative to neighboring ones. It allows us to drive the attentional behaviour of the agent by specifying which entities are currently in focus.

In our model, the salience is defined as a real-valued measure which combines several perceptual measures such as the object size and its linear and angular distances relative to the robot. During linguistic interaction, these perceptual measures can be completed by measures of linguistic saliency, such as the recency of the last reference to the object.

The salience being real-valued, its probability is defined as a density function $\Re \rightarrow [0,1]$.

B. Referencing beliefs

Beliefs are high-level symbolic representations available for the whole cognitive architecture. As such, they provide an unified model of the environment which can be used during interaction. An important aspect of this is *reference resolution*, which connects linguistic expressions such as "this box" or "the ball on the floor" to the corresponding beliefs about entities in the environment. Reference resolution is performed via a Markov Logic Network specifying the correlations between the linguistic constraints of the referring expression and the belief features (in particular, the entity saliency and its associated categorical knowledge).

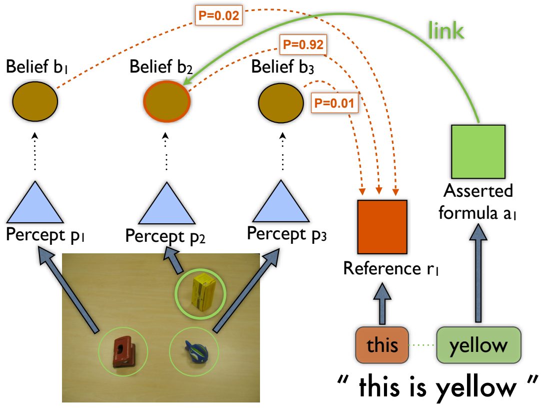


Fig. 4. An utterance such as "This is yellow" illustrates the two mechanisms of referencing and belief extension. First, the expression "this" is resolved to a particular entity. Since "this" is a (proximal) deictic, the resolution is performed on basis of saliency measures. The belief $\texttt{B}_2$ is selected as most likely referent. Second, the utterance also provides new information – namely that the object is yellow. This asserted content must be incorporated into the robot's beliefs. This is done by constructing a new belief which is linked (via a pointer) to the one of the referred-to entity.

Formula (15) illustrates the resolution of a referring expression $\texttt{R}$ containing the linguistic label "mug" to a belief $\texttt{B}$ which includes a label feature with value $\texttt{Mug}$:

$$\begin{aligned} w_i \qquad & \exists i,j \;\; \texttt{Label}(\texttt{B},\texttt{i}) \wedge \texttt{Mug}(\texttt{j}) \wedge \\ & \texttt{Ref}(\texttt{R},\texttt{j}) \wedge \texttt{Mug}(\texttt{j}) \rightarrow \texttt{Resolve}(\texttt{R},\texttt{B}) \end{aligned} \tag{15}$$

The resolution process yields a probability distribution over alternative referents, which is then retrieved by the communication subsystem for further interpretation.

C. Asserting new information

In Section III-D, we described how beliefs can be formed from percepts, bottom-up. When dealing with cognitive robots able to reflect on their own experience, anticipate possible events, and communicate with humans to improve their understanding, beliefs can also be manipulated "top-down" via high-level cognitive functions such as reasoning, planning, learning and interacting.

We concentrate here on the question of belief extension via interaction. In addition to simple reference, interacting with a human user can also provide *new* content to the beliefs. Using communication, the human user can directly extend the robot's current beliefs, in a top-down manner, without altering the incoming percepts. The epistemic status of this information is *attributed*. If this new information conflicts with existing knowledge, the agent can decide to trigger a clarification request to resolve the conflict.

Fig. 4 provides an example of reference resolution coupled with a belief extension.

D. Belief filtering

Techniques for *belief filtering* are essential to keep the system tractable. Given the probabilistic nature of the framework, the number of beliefs is likely to grow exponentially over time. Most of these beliefs will have a near-zero probability. A filtering system can effectively prune such unecessary beliefs, either by applying a minimal probability threshold on them, or by maintaining a fixed maximal number of beliefs in the system at a given time. Naturally, a combination of both mechanisms is also possible.

In addition to filtering techniques, *forgetting* techniques could also improve the system efficiency [4].

V. Conclusion

In this paper, we presented a new approach to the construction of *rich belief models* for situation awareness. These beliefs models are spatio-temporally framed and include epistemic information for multi-agent settings. Markov Logic is used as a unified representation formalism, allowing us to capture both the complexity (relational structure) and uncertainty (partial observability) of typical HRI domains.

The implementation of the approach outlined in this paper is ongoing. We are using the Alchemy software[3] for efficient probabilistic inference. The binder system revolves around a central working memory where percepts can be inserted, modified or deleted. The belief model is automatically updated to reflect the incoming information.

Besides the implementation, future work will focus on three aspects. The first aspect pertains to the use of *machine learning techniques* to learn the model parameters. Using statistical relational learning techniques and a set of training examples, it is possible to learn the weights of a given Markov Logic Network [17]. The second aspect concerns the extension of our approach to non-indexical epistemic knowledge –i.e. the representation of *events*, *intentions*, *plans*, and *general knowledge* facts. Finally, we want to evaluate the empirical performance and scalability of our approach under a set of controlled experiments.

VI. Acknowledgments

The research leading to these results has received funding from the European Union's 7th Framework Programme [FP7/2007-2013] under grant agreement No. 215181 (CogX). The authors wish to thank Miroslav Janíček and Hendrik Zender for useful discussions.

References

[1] L. W. Barsalou. Abstraction in perceptual symbol systems. *Philosophical Transactions of the Royal Society of London: Biological Sciences*, 358:1177–1187, 2003.

[2] H. H. Clark and E. F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989.

[3] F. Doshi and N. Roy. Spoken language interaction with model uncertainty: an adaptive human-robot interaction system. *Connection Science*, 20(4):299–318, 2008.

[4] S. T. Freedman and J. A. Adams. Human-inspired robotic forgetting: Filtering to improve estimation accuracy. In *Proceedings of the 14th IASTED International Conference on Robotics and Applications*, pages 434–441, 2009.

[5] D. L. Hall and J. Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, August 2002.

[6] S. Harnad. The symbol grounding problem, 1990.

[7] N. Hawes and J. Wyatt. Engineering intelligent information-processing systems with cast. *Advanced Engineering Infomatics*, To Appear.

[8] N. Hawes, H. Zender, K. Sjöö, M. Brenner, G.-J. M. Kruijff, and P. Jensfelt. Planning and acting with an integrated sense of space. In *Proceedings of the 1st International Workshop on Hybrid Control of Autonomous Systems – Integrating Learning, Deliberation and Reactive Control (HYCAS)*, pages 25–32, Pasadena, CA, USA, July 2009.

[9] H. Jacobsson, N.A. Hawes, G.-J. M. Kruijff, and J. Wyatt. Crossmodal content binding in information-processing architectures. In *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, Amsterdam, The Netherlands, March 12–15 2008.

[10] D. Koller, N. Friedman, L. Getoor, and B. Taskar. Graphical models in a nutshell. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2007.

[11] G-.J. M. Kruijff, J.D. Kelleher, and N. Hawes. Information fusion for visual reference resolution in dynamic situated dialogue. In *Perception and Interactive Technologies (PIT 2006)*. Spring Verlag, 2006.

[12] G.-J. M. Kruijff, John D. Kelleher, and N. Hawes. Information fusion for visual reference resolution in dynamic situated dialogue. In *Perception and Interactive Technologies: International Tutorial and Research Workshop, PIT 2006*, volume 4021 of *Lecture Notes in Computer Science*, pages 117 – 128, Kloster Irsee, Germany, June 2006. Springer Berlin / Heidelberg.

[13] I. Kruijff-Korbayová, S. Ericsson, K. J. Rodríguez, and E. Karagjosova. Producing contextually appropriate intonation in an information-state based dialogue system. In *EACL '03: Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, pages 227–234, Morristown, NJ, USA, 2003. Association for Computational Linguistics.

[14] P. Lison. Robust processing of situated spoken dialogue. In *Von der Form zur Bedeutung: Texte automatisch verarbeiten / From Form to Meaning: Processing Texts Automatically*. Narr Verlag, 2009. Proceedings of the GSCL Conference 2009 , Potsdam, Germany.

[15] P. Lison and G.-J. M. Kruijff. Salience-driven contextual priming of speech recognition for human-robot interaction. In *Proceedings of ECAI 2008*, Athens, Greece, 2008.

[16] H. Poon and P. Domingos. Sound and efficient inference with probabilistic and deterministic dependencies. In *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*, pages 458–463. AAAI Press, 2006.

[17] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62(1-2):107–136, 2006.

[18] D. Roy. Semiotic schemas: A framework for grounding language in action and perception. *Artificial Intelligence*, 167(1-2):170–205, 2005.

[19] D. Roy and E. Reiter. Connecting language to the world. *Artificial Intelligence*, 167(1-2):1–12, 2005.

[20] M. Stone and R.H. Thomason. Context in abductive interpretation. In *Proceedings of EDILOG 2002: 6th workshop on the semantics and pragmatics of dialogue*, 2002.

[21] A. Vrečko, D. Skočaj, N. Hawes, and A. Leonardis. A computer vision integration model for a multi-modal cognitive system. In *IEEE/RSJ International Conference on Intelligent RObots and Systems*, pages 3140–3147, 2009.

[22] M. Walker, S. Whittaker, A. Stent, P. Maloor, J. Moore, M. Johnston, and G. Vasireddy. Generation and evaluation of user tailored responses in multimodal dialogue. *Cognitive Science*, 28(5):811–840, 2004.

[23] J. Williams and S. Young. Partially observable markov decision processes for spoken dialog systems. *Computer Speech and Language*, 21(2):231–422, 2007.

[24] H. Zender, G.-J. M. Kruijff, and I. 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 2009.

[3] Cf. `http://alchemy.cs.washington.edu/`

Combining Probabilistic And Logical Inference in Situated Multi-Agent Models

[Report on Work in Progress]

Geert-Jan M. Kruijff, Miroslav Janíček, Pierre Lison & Hans-Ulrich Krieger
German Research Center for Artificial Intelligence
DFKI GmbH, Saarbrücken (Germany)

***Abstract*—The paper describes work in progress on a formal system for representing, and reasoning with, situated multi-agent belief models. These models capture what a particular agent believes about the world, and what it believes about other agents. Such beliefs arise from a mixture of inferences, ranging over the agent's direct perception of the world, what it has as semantic background knowledge about the world, and what facts the agent can infer to hold over time. The model puts probabilistic and logical inference on a par, to balance logical structure with a robustness to uncertain and partial information. The paper discusses various forms of logical and probabilistic inference, and the possibilities for combining them.**

I. Introduction

A robot continuously builds up beliefs about the world, and about the agents it is working with. It bases these beliefs in its experience. This experience naturally covers the here-and-now. What the robot currently sees, hears, plans to do. At the same time, experience needs to go beyond that. Recording past experience, the robot can reason about what once was – and what might still be, even when the robot isn't looking that way right now. And, looking beyond the moment, the robot can create expectations about what the future might bring. Either based on what has been, or what can be expected to be the case given general "world" knowledge the robot has.

How can we capture all that in a belief model for a robot?

A belief model, or simply a collection of beliefs, is essentially a dynamic model. Every time the robot forms new beliefs, or alters ones that it already entertains, the belief model needs to be updated. This update takes the model from some state t to a new state $t+1$.This new state reflects the robot's beliefs about the world.

Typically, the robot updates its beliefs on the basis of perceptual input it gets, or deliberative steps it takes in e.g. dialogue processing or action planning. Experiencing the world, and acting on it, are the main drives for maintaining and updating a robot's belief model.

What we would like to achieve is that, within bounds, the belief model represents "all" the robot could possibly know about the aspects of the world it has held beliefs about. By this we mean both a sense of temporal continuity or persistence, and a sense of completion. By persistence we mean that the robot can infer whether what it believed earlier is still the case, currently. Even when the robot does not have any current experience to confirm or disconfirm that. By semantic completion we mean that when a robot creates a belief based in experience, it can expand that belief by making further inferences about that aspect of reality by using its domain knowledge. There is a certain appeal to a model like that. At each point, it represents "all there is to know about experience" relative to the robot's domain knowledge and inference capabilities. And that means that any process acting on, working with, that model only needs to inspect the model to make its decisions, i.e. without needing to request further information from other processes.

Just seeing this logically would be nice, but there is a problem we need to face here. Namely, there is an inherent uncertainty to a robot's experience, reflected in the beliefs and inferences it can draw from them. A robot never knows for sure that the object in front of it is a mug – with some likelihood, yes, but it could also be a box. Or the room it believes to be a bedroom is actually a kitchen. A robot is never certain. Any computation for dynamically updating belief models needs to take these uncertainties into account. In this paper we discuss how we can deal with this uncertainty, in combination with complementary forms of logical inference.

§II discusses the situated multi-agent models of beliefs, intentions and events we employ. These models are relational, probabilistic models. Content takes the form of probabilistic distributions over ontologically sorted, relational structures. Beliefs, intentions, and events can also be related. In §III we consider different forms of probabilistic inference over these models, and in §IV we discuss the possibility of combining logical inference to compute deductive closures, with probabilistic inference for filtering.

II. Situated Multi-Agent Models

We would like to provide a cognitive system with an awareness of the world it find itself in. And in which it is acting and interacting with other agents. Often with the express purpose to cooperate, to learn more. This naturally requires the model to be *situated* in the world, but we need more. A model needs to provide the means for the robot to form an understanding of itself relative to world, and to other agents. What they might know (or not), or might be able to do (or not). Any potential asymmetry there is a potential source for self-extension. And any successful resolution to that end ultimately relies on the possibility of forming a mutual understanding.

In this section we describe the modeling framework we adopt. We focus primarily on the representational aspects of the framework. Inference mechanisms over these belief models are presented in more detail in the next sections.

A *situated multi-agent model* is an epistemic construct. It is something internal to a robot. It is a reflection of the world it is situated in, mediated by its experience. In that sense, it is always the robot's *model* of the world, as certain as its experience can ever be. It is never the world itself.

Within this construct, we distinguish three different types of epistemic objects: *Beliefs*, *Intentions*, and *Events*. As epistemic objects they situate particular information relative to one or more agents.

Definition 1 (Epistemic object): An *epistemic object* is a tuple $\langle \sigma, e \rangle$ with σ a frame, and e an epistemic status.

Definition 2 (Epistemic status): The *epistemic status* $e \in \mathcal{E}$ of an object indicates for which agent(s) the information in the object holds. The set of possible statuses $\mathcal{E}$ is defined by construction from a set of agents $\mathcal{A}$ and three (operator) types:

- *private*, denoted $\mathsf{K}\{a\}$. Private beliefs come from within the agent a. These beliefs are a direct or indirect result of agent a's experience of the environment.
- *attributed*, denoted $\mathsf{K}\{a[b_1, ..., b_n]\}$. Attributed beliefs are beliefs which are ascribed to other agents. They are a's conjecture about the cognitive states of other agents $b_1, ..., b_n$.
- *shared*, denoted $\mathsf{K}\{a_1, ..., a_m\}$. Shared beliefs contain information which is assumed to be part of the common ground for $a_1, ..., a_m$.

Shared epistemic status subsumes both private and attribute epistemic status. A shared belief $\mathsf{K}\{a, b\}$ therefore also implies the two private beliefs $\mathsf{K}\{a\}$ and $\mathsf{K}\{b\}$ and the two attributed beliefs $\mathsf{K}\{a[b]\}$ and $\mathsf{K}\{b[a]\}$.

Definition 3 (Frame): The frame σ of an object represents the spatial or spatiotemporal frame for which the epistemic object is thought to hold. Let $\mathcal{S}$ be the frame domain, i.e. $\forall \sigma : \sigma \in \mathcal{S}$.

Typically, we consider a frame to be a contiguous spatiotemporal interval. The object is only valid in this interval, and any frames σ' that are subsumed by σ. This way, framing can account for the situatedness and the dynamics of the world. Under uncertainty, the spatio-temporal frame defines a probability distribution over the existence of the entity in a given temporal and spatial domain.

A **belief** is an epistemic object that represents a statement about a state of the world. This state can be now, in the past, or in the future; somewhere. This is captured by the frame of the belief. It is contributed to one or more agents, represented by its epistemic status. We represent the statement itself as a mixture between logical and probabilistic information, namely as a distribution over logical formulas. These formulas form a graph structure that indicates (local) variations in how an experience can be interpreted. Referential aspects of the experience are captured directly in the formulas, e.g. which area or what object a belief is about. In addition, the belief itself has a (possibly sorted) identifier. This identifier makes it possible for formulas to construct relational structures over beliefs.

Definition 4 (Belief): A belief is an epistemic object, represented as a tuple $\langle \sigma, e, \boldsymbol{\delta}, h \rangle$. σ and e are the frame and epistemic status of the belief, respectively. $\boldsymbol{\delta}$ is the content of the belief, and h is the history of the belief. The content $\boldsymbol{\delta}$ is typically defined by a list of features. For each feature, we have a (continuous or discrete) distribution over alternative values. h provides a revisioning record of changes to the belief, making it possible to roll-back to a previous version of the belief.

The distribution $\boldsymbol{\delta}$ defines the possible content values for each feature defined in the belief. The feature values can be either discrete (as for categorical knowledge) or continuous (as for real-valued measures). Discrete values are generally expressed as (propositional) logical formulae. A feature value can also specify a *reference* to another belief, allowing us to capture the relational structure of the environment we want to model. The resulting relational structure can be of arbitrary complexity.

Discrete probability distributions can be expressed as a set of pairs $\langle \varphi, p \rangle$ with φ a formula, and p a probability value, where the values of p must satisfy the usual constraints for probability values. For continuous distributions, we generally assume a known distribution (for instance, a normal distribution) combined with the required parameters (e.g. its mean and variance). The distributions for the features contained in the belief are assumed to be conditionally independent.

Instead of seeing a belief as a tuple, we can also think of a belief functionally. A belief is a function from epistemic statuses, frames, and logical formulas to a probability – or a probability distribution, if a set of alternative formulas is considered.

Definition 5 (Belief as function): Given $\mathcal{E}$, $\mathcal{S}$ and a logical domain $\mathcal{L}$ defining possible formulas, a belief can be defined as a function $\mathcal{E} \times \mathcal{S} \times \mathcal{L} \rightarrow [0...1]$ if strictly one formula is selected from $\mathcal{L}$. If multiple formulas from $\mathcal{L}$ are allowed, the function maps to a PDF.

And we can decompose a belief as function even further, if we consider $\mathcal{L}$ as the range of a mapping from (uncertain) perceptual structures. This provides us then with a complex yet continuous functional characterization from experience to beliefs. The interesting aspect of seeing a belief from a functional perspective is that we can turn this definition into a probabilistic characterization of the belief *space*.

An **event** is a statement about dynamics that can make a transition from one state into another state possible. This might be a simple agent-initiated action, or an expectation about the dynamics of an environment.

Definition 6 (Event): An *event* is an epistemic object, represented as a tuple $\langle \sigma, e, \boldsymbol{\tau} \rangle$. σ is the frame of the event, whereas e is the set of agents (including the world as agentive force) that bring about a transition. $\boldsymbol{\tau}$ is a probabilistic distribution over possible transitions.

Functionally, an event defines a transition function from (beliefs about) frames to (beliefs about) frames. It is this functional understanding that we use in the notion of intention.

An **intention** is a statement that relates a set of beliefs about an initial state, to a deliberately brought about change ("action"), to yield another state. Which is again captured by a set of beliefs. In other words, an intention is a relation between beliefs about a state, to another set of beliefs about another state, brought about by an event that encodes a (usually physical) action. Achieving the intention thus relies both on the ability to bring about the event, and on the ability to perceive (enough) of the world to be able to form the required beliefs about the resulting state.

Definition 7 (Intention): An *intention* is an epistemic object, represented as a tuple $\langle \sigma, e, \boldsymbol{\iota} \rangle$. σ is the frame of the event, whereas e is the set of agents for which the intention holds. $\boldsymbol{\iota}$ is a probabilistic distribution over alternative intents. Each intent is a tuple $\langle \mathbf{pre}, \mathbf{event}, \mathbf{post}, p \rangle$ with $\mathbf{pre}$ a set of beliefs about the state that is the *precondition*, $\mathbf{post}$ is a set of beliefs about the state that is to ensue as *post-condition*, and $\mathbf{event}$ is the event that is to bring about the transition between the pre- and post-conditions. p is the probability of the intent.

III. Hybrid Inference Methods

In this section we discuss hybrid methods for performing abductive and deductive inference on situated multi-agent models. These methods combine a logical type of inference with probabilistic models to deal with uncertainty and incompleteness.

A. Probabilistic abduction

Abduction is a method of backward logical reasoning that allows inferring *explanations* of observations (facts). Formally, given a theory T, a rule $(T \vdash) A \rightarrow B$ and a fact B, abduction allows inferring A as an explanation of B. B can be deductively inferred from $A \cup T$. If $T \not\vdash A$, then we say that A is an *assumption*. Naturally, as there may be many possible explanations for a given observation, a mechanism for selecting the best explanation is required in practical applications.

There are many ways to do this. For instance, one may only allow the assumption of some facts, and prefer proofs with the minimal number of assumptions. This is a direct application of Occam's razor on the "surface form" of the proofs. However, in general, this syntactic criterion does not always lead to a single best answer. Proof selection techniques therefore need to look at the *meaning* of the assumed facts, in order to select the most plausible explanation.

In our current work, we are employing logic programming as a backbone of our abductive inference. The method for selecting the most plausible explanations, while probabilistic, is based on a cost-based mechanism called *weighted abduction*.

1) Proof procedure: We extend Hobbs and Stickel's logic programming approach to weighted abduction [1], [2] by a contextual aspect following Baldoni *et al* [3]. We further extend the approach with the notion of *assertion* [4] in order to be able to reason about information not present in the knowledge base, thereby addressing the need for reasoning under the open-world assumption. Weights in the system are assigned probabilistic semantics following Charniak and Shimony [5].

Formally, inference in our system makes use of four ingredients: *facts*, *rules*, *disjoint declarations* and *assumability functions*.

- Facts are modalised formulas of the form
$$\mu : A$$
where μ is a (possibly empty) sequence of modal *contexts*, and A is an atomic formula, possibly containing variables. Contexts help separating unrelated facts, restricting the search space.
- Rules are modalised Horn clauses, i.e. formulas of the form
$$(\mu_1 : A_1/t_1) \wedge ... \wedge (\mu_n : A_n/t_n) \rightarrow (\mu_H : H)$$
where each of the $\mu_i : A_i$ and $\mu_H : H$ are modalised formulas. Each antecedent is annotated by t_i, which determines the way the antecedent is manipulated and is one of the following:
  - *true* – the antecedent has to be proven, i.e. either it is a fact, or a head of some rule;
  - *assumable(f)* – the antecedent is assumable under function f;
  - *assertion* – the antecedent is asserted, i.e. the validity is assumed, but will eventually have to be proved.
- Assumability functions are partial functions f, $f : \mathcal{F} \rightarrow \mathbb{R}_0^+$, where $\mathcal{F}$ is the set of modalised formulas. Assumability functions assign weights to modalised formulas.
- A disjoint declaration is a statement of the form
$$disjoint([\mu : A_1, ..., \mu : A_n])$$
which specifies that at most one of the modalised formulas $\mu : A_i$ may be used in the proof. A_i and A_j cannot be unified for all $i \neq j$.

A *proof state* is a sequence of marked modalised formulas (called *queries* in this context)

$$Q_1[n_1], ... Q_m[n_m]$$

The markings n_i are one of the following:

- *unsolved(f)* – the query is yet to be proved, assumable under assumability
- *proved* – the query is proved or in the process of being proved; function f
- *assumed(f)* – the query is assumed under f;
- *asserted* – the query is asserted

The proof procedure starts from a single query marked as unsolved, iteratively rewriting the proof state by manipulating the leftmost unsolved query Q_l. First, the query has to pass constraints imposed by disjoint declarations. If it does, it is either proved (using facts or rules), assumed under an assumability function, or eliminated if any of the queries to the right is unifiable with Q_l. In other words, each query is proved or assumed at most once.

The initial query Q is proved when there is no unsolved query in the proof state. The final proof state Π_Q is then the proof of Q.

2) Weights and probabilities: In weighted abduction, weights assigned to assumed queries are used to calculate the overall proof cost. The proof with the lowest cost is the best explanation. However, the weights are not assigned any semantics, and often a significant effort by the writer of the rule set is required to achieve expected results [1].

Charniak and Shimony [5] showed that by setting weights to $-\log$ of the prior probability of the query, the resulting proofs can be given probabilistic semantics.

Suppose that query Q_k can be assumed true with some probability $P(Q_k \text{ is true})$. Then if Q_k is assumable under assumability function f such that $f(Q_k) = -\log(P(Q_k \text{ is true}))$, and under the independence assumption, we can represent the overall probability of the proof $\Pi = Q_1[t_1], ..., Q_n[t_n]$ as

$$P(\Pi) = e^{\sum_{k=1}^{n} c(Q_k)}$$

where

$$c(Q_k) = \begin{cases} f(Q_k) & \text{if } m_i = \textit{assumed}(f) \\ 0 & \text{otherwise} \end{cases}$$

The best explanation Π_{best} of a query Q is then

$$\Pi_{best} = \underset{\Pi \text{ proof of } Q}{\arg\max} \; P(\Pi)$$

Exact inference in this system is NP-complete, and so is approximate inference given a threshold [5]. However, it is straightforward to give an anytime version of the algorithm – simply by performing iterative deepening depth-first search [6] and memoizing the most probable proof so far.

3) Situated belief models and abduction: The abductive inference allows smooth integration with multi-agent belief models as described in §II. All logical information in beliefs, events and intentions (e.g. epistemic statuses) is represented as a set of modalised formulas. To demonstrate how probabilistic information is modelled in the abductive inference, we discuss the treatment of beliefs in detail. For events and intentions, an analogous process applies.

Recall that per Definition 4, a belief content is a probability distribution over possible content values. Every point in this distribution is assigned a unique identifier. These identifiers are then translated into a disjoint declaration, effectively promising to fix the choice of a content value in a proof by the proof procedure. The content itself is then represented as a set of rules with antecedents corresponding to the identifier of the content value, with the antecedent assumable under probability given by the distribution.

For example, a content value distribution

$$\begin{aligned}\delta = \{ & (\langle \mathit{Color}\rangle \mathit{blue} \wedge \langle \mathit{Shape}\rangle \mathit{small}) : 0.5, \\ & (\langle \mathit{Color}\rangle \mathit{red} \wedge \langle \mathit{Shape}\rangle \mathit{small}) : 0.3, \\ & (\langle \mathit{Color}\rangle \mathit{green} \wedge \langle \mathit{Shape}\rangle \mathit{smal}) : 0.2\}\end{aligned}$$

translates to a disjoint declaration

$$disjoint([cont(b, i_1), cont(b, i_2), cont(b, i_3)])$$

and rules

$$\begin{aligned} cont(b, i_1)/assumable(p(0.5)) &\rightarrow val(b, color(blue)) \\ cont(b, i_1)/assumable(p(0.5)) &\rightarrow val(b, shape(small)) \\ cont(b, i_2)/assumable(p(0.3)) &\rightarrow val(b, color(red)) \\ cont(b, i_2)/assumable(p(0.3)) &\rightarrow val(b, shape(small)) \\ cont(b, i_3)/assumable(p(0.2)) &\rightarrow val(b, color(green)) \\ cont(b, i_3)/assumable(p(0.2)) &\rightarrow val(b, shape(small)) \end{aligned}$$

where $p(x) = -\log x$, and b is the identifier of the belief.

When proving $val(b, color(X))$, the proof procedure unifies it with one of the rules above, and expands the antecedent to the proof state; the antecedent is then assumed under the corresponding function. If $val(b, shape(small))$ later occurs in the proof, the disjoint declaration disallows the use of other $cont(...)$ than the one selected previously, committing to the choice. The antecedent is then factored out as it has already been assumed earlier, and no cost for assuming the valid antecedent is charged.

The entire content value is thus expanded into the proof, under the probability specified in the distribution.

4) Comparison with other approaches: Our system is similar to Poole's Probabilistic Horn abduction [7]. The main difference, apart from the proof procedure which is cost-based in our case, is that we do not include probabilities in our formulation of disjoint declarations. As we employ factoring so as to avoid double assumptions and proofs, we are able to model the semantics of disjoint declarations with probabilities.

On the other hand, having a general disjoint declaration allows us to define general rules such as simple negation,

$$disjoint([p(X), not\_p(X)])$$

or functional constraints on features, such as

$$\begin{aligned} disjoint([&val(b, color(blue), \\ &val(b, color(red)), \\ &val(b, color(green))])\end{aligned}$$

without having to specify the prior probabilities of the disjuncts.

In our rule sets for natural language understanding and generation, we need to be able to manipulate with logical structure (e.g. logical forms of utterances) efficiently. We have found that the logic-programming-based approach is quite satisfactory in this aspect, as we can employ standard Prolog programming techniques. In other approaches to probabilistic abduction such as Kate and Mooney's abduction in Markov Logic Networks [8], such tools are not available, which crucially limits their usefulness in our application.

B. Probabilistic deduction

We are exploring to use of Markov Logic Networks to perform particular types of deductive inference over situated multi-agent models. Markov logic combines first-order logic and probabilistic graphical models in a unified representation [9]. The motivation for this is to use such expressive formalism to capture the internal structure inherent to our models - something we would not able to do using e.g. (simple) Bayesian networks.

From a syntactic point of view, a *Markov logic network* L is defined as a set of pairs (F_i, w_i), where F_i is a first-order formula and $w_i \in \mathbb{R}$ is the associated weight of that formula. A Markov logic network can be interpreted as a *template* for constructing Markov networks. The structure and parameters of the constructed network will vary depending on the set of constants provided to ground the predicates of the Markov Logic formulae. Such a Markov network represents a probability distribution over possible words. It can be used to perform probabilistic inference over the relational structure defined by the formulas F_i.

In the following, we briefly review the definition of Markov networks, and then show how they can be generated from a Markov logic network L.

1) Markov Network: A Markov network G, also known as a *Markov random field*, is an undirected graphical model [10] for the joint probability distribution of a set of random variables $X = (X_1, \ldots, X_n) \in \mathcal{X}$. The network G contains a node for each random variable X_i. The nodes in the network can be grouped in a set of *cliques*. In graph theory, a clique is a fully connected subgraph – that is, a subset of nodes where each node is connected with each other. The joint probability distribution of the Markov network can then be factorised over the cliques of G:

$$P(X = x) = \frac{1}{Z} \prod_k \phi_k(x_{\{k\}}) \tag{1}$$

where $\phi_k(x_{\{k\}})$ is a *potential function* mapping the state of a clique k to a non-negative real value. Z is a normalization constant, known as *partition function*, and is defined as $Z = \sum_{x \in \mathcal{X}} \prod_k \phi_k(x_{\{k\}})$.

Alternatively, the potential function ϕ_k in (1) can be replaced by an exponentiated weighted sum over real-valued feature functions f_j:

$$P(X = x) = \frac{1}{Z} e^{\left(\sum_j w_j f_j(x)\right)} \tag{2}$$

The representation in (2) is called a *log-linear model.*

2) Constructing a Markov Network from a Markov Logic Network: Recall that a Markov logic network L is a set of pairs (F_i, w_i). If in addition to L we also specify a set of constants $C = \{c_1, c_2, ..., c_{|C|}\}$, one can generate a *ground Markov network* $M_{L,C}$ as follows [11]:

1) For each possible predicate grounding over the set C, there is a binary node in $M_{L,C}$. The value of the node is true iff the ground predicate is true.
2) For every formula F_i, there is a feature f_j for each possible grounding of F_i over C. The value of the feature $f_i(x)$ is 1 if F_i is true given x and 0 otherwise. The weight of the feature corresponds to the weight w_i associated with F_i.

The graphical representation of $M_{L,C}$ contains a node for each ground predicate. Furthermore, each formula F_i defines a set of cliques j with feature f_j over the set of distinct predicates occurring in F_i.

Following (1) and (2), the joint probability distribution of a ground Markov network $M_{L,C}$ is then given by:

$$P(X = x) = \frac{1}{Z} \prod_i \phi_i(x_{\{k\}})^{n_i(x)} = \frac{1}{Z} e^{\left(\sum_i w_i n_i(x)\right)} \tag{3}$$

The function $n_i(x)$ in (3) counts the number of true groundings of the formula F_i in $M_{L,C}$ given x.

3) Example of Markov Logic Network: Consider a simple Markov Logic network L made of three unary predicates, `Tube(x)`, `Rolls(x)`, and `Box(x)`, and two formulae:

$$w_1 \qquad \texttt{Tube(x)} \rightarrow \texttt{Rolls(x)} \tag{4}$$

$$w_2 \qquad \texttt{Box(x)} \rightarrow \neg\texttt{Rolls(x)} \tag{5}$$

The formulae encode the fact that most tubes roll, while most boxes don't. Since these two rules admit a few exceptions (some tubes may be square, and some boxes might be of a form that affords rolling), they are specified as soft constraints with finite weights w_1 and w_2.

Assuming a particular object `A`, we can construct a ground Markov network $M_{L,\{A\}}$ over this single constant following the procedure we just outlined. The network $M_{L,\{A\}}$ defines a probability distribution over a set of 2^3 possible worlds (since we have three unary predicates which can be true or false, and one constant).

The probability of the world $x = (\texttt{Tube(A)}, \neg\texttt{Rolls(A)}, \neg\texttt{Box(A)})$ can then be directly computed using (3). The ground Markov Network countains two features (one for each formula). In the case of world x, the first formula is violated, while the second is not. This means that $n_1(x) = 0$ and $n_2(x) = 1$. This gives us the probability $P(X = x) = \frac{1}{Z} e^{(w_1 \times 0 + w_2 \times 1)} = \frac{1}{Z} e^{w_2}$, where the partition function $Z = 4e^{w_1 + w_2} + 2e^{w_1} + 2e^{w_2}$. Notice that the partition function Z grows exponentially with the weights, and will tend to infinity for large values of w_1 or w_2. If we increase the value of w_1 while keeping the value of w_2 constant, the probability $P(X = x)$ will thus approach 0.

4) Inference: Once a Markov network $M_{L,C}$ is constructed, it can be exploited to perform conditional or MPE inference over the relational structure defined by L. A Markov Logic Network can be used to answer arbitrary queries such as "What is the probability that formula F_1 holds given that formula F_2 does?'. Such query can be translated as:

$$\begin{aligned} P(F_1|F_2, L, C) &= P(F_1|F_2, M_{L,C}) && (6)\\ &= \frac{P(F_1 \wedge F_2|M_{L,C})}{P(F_2|M_{L,C})} && (7)\\ &= \frac{\sum_{x \in \mathcal{X}_{P_1} \cap \mathcal{X}_{P_2}} P(X = x|M_{L,C})}{\sum_{x \in \mathcal{X}_{P_2}} P(X = x|M_{L,C})} && (8) \end{aligned}$$

where $\mathcal{X}_{P_i}$ represent the set of worlds where the formula F_i holds.

Exact inference in Markov Networks is a #P-complete problem [10] and is thus untractable. However, several anytime algorithms for probabilistic inference such as weighted MAX-SAT or Markov Chain Monte Carlo (MCMC) can then be used to yield approximate solutions [12], [13], [14]. Given the requirements of our application domain, and particularly the need to operate under soft real-time constraints, such approximation methods are crucial.

5) Learning: The weight w_i in a Markov logic network encodes the "strength" of its associated formula F_i. In the limiting case, where $\lim_{w_i \to \infty}$, the probability of a world violating F_i has zero probability. For smaller values of the weight, worlds violating the formula will have a low, but non-zero probability.

But how are these weights specified? In most cases, weights are learned based on training samples extracted from a relational database. Several machine learning algorithms for parameter learning can be applied to this end, from classical gradient-based techniques to more sophisticated algorithms specifically designed for statistical relational learning [15], [16].

In addition to weights learning, it is in theory also possible to learn the *structure* of a Markov Logic problem, either partially (by adding additional clauses to the network or refining the existing ones), or completely (by learning a full network from scratch). Structure learning is usually performed with algorithms borrowed from Inductive Logic Programming [17], [18].

6) Experiments: We performed some preliminary experiments with Markov Logic Networks for inference over belief content. We started our experiments with the problem of visual reference resolution, which is a relatively simple and well-defined task. Visual reference resolution is the part of the dialogue interpretation process which is concerned with linking particular linguistic expressions such as "the ball" or "the red object" to their visual counterpart in the real world.

These experiments were performed using the open-source tool Alchemy combined with a mechanism for serialising belief contents into a Markov Logic representation. Practically, such operation is realised by converting each individual formulae contained in the belief distribution into a distinct Markov Logic formula, with weights corresponding to the logarithmic equivalent of the formula probability in the distribution. The serialised beliefs are then combined with a sequence of (problem-specific) Markov Logic rules for the inference.

In the case of reference resolution, such rules will specify particular correlations between the visual characteristics of the objects in the scene and the linguistic characteristics of the referring expression. One (simplified) example of such rule is

$$w_i \quad \mathtt{Resolve(x)} \land \mathtt{LingColor(B, Red)} \Rightarrow \mathtt{Color(x, Red)}$$

where `x` denotes an arbitrary visual belief and `B` the belief corresponding to the referring expression.

The results we can currently report on these experiments are negative results. As explained in the previous sections, Markov Logic Networks need to be compiled into ground Markov Networks as a preprocessing step for inference. This means that in the general case, the number of ground clauses in this network grows exponentially with the number of beliefs, rendering such inference intractable beyond a few beliefs and inference rules. The intractability of the inference is due to the use of full logical structures in the belief content. The conversion of such structure into Markov Logic necessitates the use of universal and existential quantifiers in the inference rules, which quickly result in a combinatorial explosion of the ground network size.

As a consequence, the solution we are using in our reference resolution component consists in "propositionalising" the belief content, by collapsing the logical structure of the belief formula into a single atomic symbol, and performing inference over this simplified representation. This reduces the inference to a form akin to inference over a Bayesian network. We are currently investigating how to overcome the problems with online inference using Markov Logic, (which was originally designed for offline inference over "certain" database content). One possibility might be to constraint the space over which the inference is to be drawn, by first of all exploiting logical structure to its full extent using efficient logical reasoners. For this we are currently considering an approach based on the mechanisms discussed in the next section.

IV. Hybrid Model Update Using Completion

In this section we present the formal aspects of the approach to computing a completion over a model. As already indicated earlier, the basic idea is to compute an update of a belief model using a mixture of logical and probabilistic reasoning. Given a model, and a set of inference rules, we compute a closure over the model with the beliefs to be updated. Each closure is computable as a sequence over sets of extensions. At each step in this sequence, we choose one single extension, using discriminate probabilistic inference. We repeat this compute-choose-extend cycle until a fixpoint is reached, after which we compute the probability of the resulting belief model and the probabilities of the beliefs it is made up of. Below we first discuss the closure computation, then the basis for hybrid model update.

A. Closure Computation And Forward Chaining

Logical inference within the individual models is performed by *HFC* [19], a rule-based forward chainer that was originally implemented for reasoning and querying with OWL-encoded ontologies [20] over RDF triples [21].[1]

Usually, bottom-up forward chaining is employed to carry out (all possible) inferences at compile time, so that querying information reduces to an indexing problem at runtime. The process of making implicit information explicit is often called *materialization* or computing the *deductive closure* of a set of ground atoms A w.r.t. a set R of universally-quantified implications $B \rightarrow H$ (if-then rules). Bottom-up here means that one starts from the ground atoms to which the rules are applied, contrary to top-down approaches which start with a goal (the head H) and potentially hypothesize intermediate goals that can hopefully be satisfied by ground atoms finally (Prolog's strategy). The body and the head of a rule consist

[1] Due to decidability issues, OWL, or description logic in general, restricts itself to unary and binary predicates, so-called *classes* and *roles* (OWL is a instance of the decidable two-variable PL1 fragment). OWL relation instances, such as r : Robot or (r, a) : in are represented in RDF through a uniform data structure, the RDF triple: `subject predicate object`. The above instances then translate into `r rdf:type Robot` and `r in a`. We will often use the more common relational notation *Robot*(r) and *in*(r, a).

of a set of clauses, interpreted *conjunctively*. In *HFC*, clause arguments are either constants $c \in C$ or variables $v \in V$.

Closure computation can be characterized as the computation of the *least fixpoint* of a certain *monotonic* function T_R. A fixpoint is reached in case there exists some number n, such that

$$T_R^{n+1}(A) = T_R^n(A)$$

where the n-fold composition of T_R is defined as follows:

- $T_R^0(A) := A$
- $T_R^{n+1} := T_R(T_R^n(A))$

In order to define T_R for our setting, let $\theta = \{v_1/c_1, \ldots, v_n/c_n\}$ be a *ground substitution* for the variables in the body B of a rule and $\Theta(C)$ the set of all ground substitutions w.r.t. C [22]. We define $B\theta$ as the set of ground atoms obtained from B by simultaneously replacing each occurrence of v_i by constant c_i $(1 \leq i \leq n)$.

This leads us to the following definition of T_R:

$$T_R(A) = \bigcup_{(B \rightarrow H) \in R} \; \bigcup_{\theta \in \Theta(C)} \{H\theta \mid B\theta \subseteq A\}$$

Due to the use of set union, the following inclusion does always hold, a requirement that we use later to efficiently realize a practical fixpoint computation ($n \in \mathbf{N}$):

$$T_R^n(A) \subseteq T_R^{n+1}$$

Since set union is a monotonic operation, T_R is also clearly monotonic, and thus a well-defined least fixpoint exists. This, of course, does not tell us that the fixpoint can be reach in finitely-many steps. Finitely-reachable fixpoints, however, are guaranteed by the following *sufficient* condition: if the set of constants C does not change, e.g., *new* constants are *not* generated during the fixpoint computation, only a finite number of ground atoms can be generated. For the RDF/OWL case, this number is bound by $|C|^3$, since the data model is the RDF triple, where C refers to the union of the sets of XSD atoms and URI references found in the ontology (= TBox + RBox + ABox).

Forward chaining, as we used it here, can be seen as *model building* over the Herbrand interpretation of a function-free definite program (Horn logic as used in Prolog). In general, model builders are systems that try to construct a finite model for a given theory (usually, a set of first-order formulae) [23]. Forward chaining is also related to the Datalog query and rule language for deductive databases. Datalog calls T_R the *elementary production principle* which can be shown to be sound and complete (as is the case for T_R).

Given the definition of T_R, a naïve (but not very efficient) implementation is relatively straightforward:

```
input $R$: set of if-then rules, $A$: set of RDF triples (= ground atoms here)
repeat
  $A' := A$
  for each $(B \rightarrow H) \in R$
    for each binding $b \in \text{match}(B, A')$
      $A := A \cup \{\text{instantiate}(H, b)\}$
until $A' = A$
```

In a naïve implementation, the second *for* loop is usually realized by a nesting of n for loops (or a heavily-recursive procedure), where $n = |B|$.

In order to make forward chaining scalable, *HFC* applies several optimization techniques that are realized as a sequence of filter stages, leading to a filter rate of more than 99%. I.e., less than 1% of possible matching candidates are actually computed and used for instantiating the RHS of a rule. This possibility comes as a side product of the fact that closure computation is a monotonic operation. Consider, for instance a rule $r = (b_1\, b_2 \rightarrow H)$ and assume that r is currently applied in iteration n of the closure computation. Due to the monotonicity argument, matching candidates M^n from A for the LHS variables of rule r at iteration n can be decomposed into those which are brand new at n and those which come from iteration $n-1$: $M^n = N \uplus M^{n-1}$. Since bindings for the variables of individual clauses are actually tables, computing a binding for all LHS variables effectively reduces to a *natural join* $\bowtie$ known from data base theory. Given the distinction *new vs. old* already mentioned, we can compute all possible bindings for $b_1\, b_2$ from the individual bindings, given N and M^{n-1}:

$$\begin{aligned} M^n(b_1\, b_2) = \quad & N(b_1) \bowtie N(b_2) \cup \\ & N(b_1) \bowtie M^{n-1}(b_2) \cup \\ & M^{n-1}(b_1) \bowtie N(b_2) \end{aligned}$$

This optimization massively speeds up forward chaining, since useless bindings, leading to already instantiated tuples, are no longer generated. In our case here, $M^{n-1}(b_1) \bowtie M^{n-1}(b_2)$ is not computed, and those bindings are by far the largest, when closure generation n increases. This techniques not only applies to individual clauses, but also to larger parts, so called (LHS) clusters.

Other optimizations are also applied in *HFC*, for instance:

- bindings are shared over "similar" clause between different rules;
- the LHSs of rules are reordered to faster compute matching candidates;
- equivalence relations instances on the LHS and the RHS of rules (e.g., `owl:sameAs`) are efficiently handled through rule rewriting and a union-find structure;
- the processing of individual rules is parallelized at each fixpoint iteration step;
- efficient data structures, such as open-address hash tables, integer arrays for tuples, specialized sets with strategy objects to support binding/table projection, etc., are used.

It is worth noting that the forward chainer operates in a monotonic and certain conjunctive search space: neither do we delete any information (remember the monotonicity assumption used in the closure computation above), nor do we attach probabilities to asserted facts. As stated above, the information in the body and the head of a rules is always interpreted conjunctively. Disjunction is realized through efficient and lazy model duplication, whereas negation can be partly implemented through special negated types and by reformulating rules, as the below LTL example will show.

We apply forward chaining to perform logical inference at runtime w.r.t multivariate probability distributions over beliefs/incoming sensor data. *HFC* is initially equipped with some axiomatic knowledge and regularly queries new information within a situation-awareness loop, adds this information to its current state (the ABox) and computes an extended closure, given the old closure and the new information. Since the size of relevant *new* information (the *deltas*) between different closure computations is relatively small, a new fixpoint is usually computed extremely fast, requiring only a few iteration steps.

HFC efficiently handles ABoxes with millions of facts and provides means to work with extended *copies* of a given ABox in parallel, an important feature that we later employ when performing *Viterbi search over non-probabilistic ABoxes* (see section).

HFC has implemented several extensions that are not available in comparable systems, such as OWLIM [24]:

- replacement of triples by more general tuples,
- possibility to add arbitrary tests to the LHS of a rule,
- possibility to add arbitrary actions to the RHS of a rule,
- incorporation of aggregation rules,
- incorporation of metric linear time into OWL.

Rules in such systems usually serve a two-fold purpose:

1) to implement OWL entailment, and thus consistency; e.g.,

```
?s owl:sameAs ?o
?s owl:differentFrom ?o
->
?s <rdf:type> <owl:Nothing>
?o <rdf:type> <owl:Nothing>
```

2) to provide custom functionality; e.g., to move from a point-based sensor-oriented representation to an extendable interval-based encoding:

```
?s ?p ?o ?t
->
?s ?p ?o ?t ?t
```

Due to experiences we have gained in several projects [25], we have opted to go for more general tuples (instead of using encoding schemes, such as reification), thus making *HFC* able to address two further important areas of functionality:

3) to coalesce information over time, e.g.,

```
?s ?p ?o ?b1 ?e1
?s ?p ?o ?b2 ?e2
->
?s ?p ?o ?b ?e
@test
IntervalNotEmpty ?b1 ?e1 ?b2 ?e2
@action
?b = Min2 ?b1 ?b2
?e = Max2 ?e1 ?e2
```

4) to reformulate LTL safety conditions (r = robot, a = area), such as $\mathbf{G}(\mathit{explore}(r) \wedge \mathit{risky}(a) \Rightarrow \neg \mathit{in}(r,a))$:

```
?r explore ?b1 ?e
?a rdf:type Risky ?b2 ?e
?r in ?a ?b3 ?e
->
DO SOMETHING / MOVE ROBOT OUT / ...
```

In order to make the entailment rules for RDFS [26] and OWL [27] also sensitive to time, we have extended them by further temporal arguments, expressing durations within a calendar treatment of time. Here is an example of a rule that talks about functional object properties in OWL:

```
?p rdf:type owl:FunctionalProperty
?p rdf:type owl:ObjectProperty
?x ?p ?y ?b1 ?e1
?x ?p ?z ?b2 ?e2
->
?y owl:sameAs ?z
@test
?y != ?z
IntervalNotEmpty ?b1 ?e1 ?b2 ?e2
```

The `IntervalNotEmpty` predicate in the test section (`@test`) guarantees that we only *identify* `?y` and `?z` if the temporal extents [`?b1`, `?e1`] and [`?b2`, `?e2`] have a *non-empty intersection*. Thus a single overlapping observation leads to a *total* identification of `?y` and `?z`, so the `sameAs` statement need not be equipped with temporal information (and this is not desired). If both observations, however, do talk about different *non-intersecting* times, it makes perfect sense that `?y` and `?z` need not be equal, even though `?p` is a *functional* property.

B. *Hybrid model update*

We first present a logical approach to model update. After that we discuss how the properties of this approach make it possible to include probabilistic filtering, to guide completion computation.

Logically speaking, computing the closure over a belief (as per Definition 4) given a domain logic $\mathcal{L}_{dom}$ means we are computing a model in the model space of $\mathcal{L}_{dom}$. This model space is structured, following the (typically hierarchical) structure of $\mathcal{L}_{dom}$. We use this structure to guide closure computation. In this section we focus here on the computation of closures from a single belief.

Given a belief $\mathbf{B} = \langle \sigma, e, \boldsymbol{\delta}, h \rangle$. $\mathbf{B}$ provides us with grounded information about an instance. The content $\boldsymbol{\delta}$ provides the alternative interpretations within that information. For our current purposes, we assume $\boldsymbol{\delta} = \langle \delta_1, \delta_2, ..., \delta_n \rangle$ with $\delta_2, ..., \delta_n$ conditionally independent, and dependent on ($\mathcal{L}_{dom}$-implied by) δ_1. For example, if we have a belief about a tabletop object, we consider *color* and *shape* to be independent of each other, but dependent on what type of object we are dealing with.

The simplest (though most expensive) way to compute completions over $\mathbf{B}$ is by compiling out logically possibly combinations of variant interpretations in $\boldsymbol{\delta}$, in a *breadth-first* manner. We do so through graph construction. Each node in the graph we annotate with a set of logical statements. A statement is conjunction of interpretations selected from $\boldsymbol{\delta}$, one for each $\delta_i \in \boldsymbol{\delta}$. Edges between nodes are labelled with δ_j's. The dependency of a δ_j on δ_1 can result in a partitioning of this set. This is the case if there is an interpretation in δ_1 that would be inconsistent with one or choices in δ_j. We then create multiple edges for δ_j, one for each partition.

Figure 1 provides an illustration of Algorithm 1. Given a belief about i with $\boldsymbol{\delta} = \langle type = \langle mug : 0.6, box : 0.4 \rangle, color = \langle red : 0.5, blue : 0.5 \rangle, shape = \langle square : 0.4, round : 0.6 \rangle \rangle$.

Algorithm 1 BREADTH-FIRST COMPLETION FOR B

```
Require: A belief B = $\langle \sigma, e, \boldsymbol{\delta}, h \rangle$
Require: $\boldsymbol{\delta} = \langle \delta_1, \delta_2, ..., \delta_n \rangle$
Require: $\delta_2, ..., \delta_n$ mutually independent, dependent on $\delta_1$
Require: instant identifier $i$ referred to by B
Graph G = node $n_0(\{i\})$
for $\delta_i \in \boldsymbol{\delta}$ do
  Let $S_{i-1}$ be the statements at previous node $n_{i-1}$ along
  the path back to $n_0$
  $S_i = \emptyset$
  for $Formula\ f \in \delta_i$ do
    $s = \emptyset$
    $X_f = closure(f)$
    for each $f' \in S_{i-1}$ s.t. $f' \wedge X_f \rightarrow \top$,
    add $f' \wedge X_f$ to $s$
    $S_i = S_i \cup \{s\}$
  end for
  for $\{s\} \in S_i$ do
    create a node $n_i(\{s\})$
    connect $n_{i-1}$ to $n_i(\{s\})$ with an edge labelled $\delta_i$
  end for
end for
```

We assume for the sake of illustration that closure is just an identity function, and that mugs are round and boxes are square. The algorithm then computes the graph as in Figure 1. This yields two logically consistent models, which can be straightforwardly translated into two beliefs.

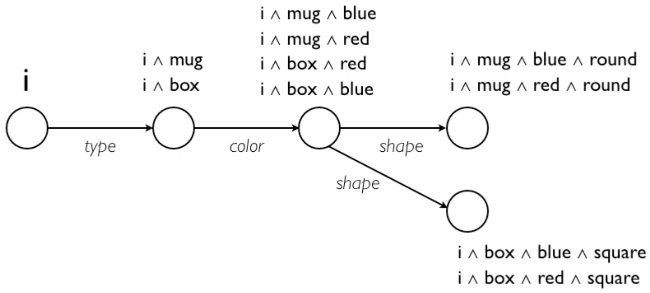


Fig. 1. Breadth-first graph for computing possible models (id closure)

There are several interesting properties we can note for Algorithm 1.

Property 1: If closure is the identity function $closure(f) = f$, then Algorithm 1 computes $\mathcal{L}_{dom}$-consistent models $\mathfrak{M} = \mathcal{M}_1, ...\mathcal{M}_n$ over **B** such that for each $\mathcal{M}_i \in \mathfrak{M}$ it holds that $\mathcal{M}_i \subseteq \mathcal{M}(\mathbf{B})$. .

Property 1 highlights that Algorithm 1 computes consistent, disjoint subspaces in model space. If closure is the identity function, these subspaces are entirely contained in the model space spawn from **B**. Even for the more general case, it holds that the branches in **G** as computed by Algorithm 1 partition model space.

Property 2: For each model M_{n_i} at a node n_i in a graph **G** it holds that it is a consistent model.

Proof: Follows from the definition of Algorithm 1: Models are only built through consistent extension. ■

From Property 2 it immediately follows that each model is contained in the extension(s) it helps construct.

Property 3: Given a graph **G**. Given nodes $n_i...n_k$ that are on a unique path from n_k back to n_0. Let $n_i < n_j$ mean that n_i comes before n_j on the path starting in n_0. For each $n_i < n_k$ it holds that $\mathcal{M}_{n_i} \subset \mathcal{M}_{n_k}$.

Finally, branching induces partitioning of the model space.

Property 4: If nodes n_i and n_j in **G** are on different branches, i.e. $\exists n_k \in \pi(n_j)\ s.t.\ n_k \notin \pi(n_i)$, then $\mathcal{M}_{n_i} \cup \mathcal{M}_{n_j} \rightarrow \perp$.

Proof: The minimal submodel of $\mathcal{M}_{n_i}$ that leads to an inconsistency with $\mathcal{M}_{n_j}$ can be constructed as follows. Assume n_i, n_j are on different branches, i.e. $\exists n_k \in \pi(n_j)\ s.t.\ n_k \notin \pi(n_i)$. Let $\pi(n_{k-1}$ be the longest path starting in n_0 s.t. $\pi(n_{k-1}) \subset \pi(n_i)$ and $\pi(n_{k-1}) \subset \pi(n_j)$. Let n_k be the node immediately following up on n_{k-1} on $\pi(n_i)$. Because n_k is the first node after the split with $\pi(n_j)$, $\mathcal{M}_{n_k}$ is a minimal model s.t. $\mathcal{M}_{n_k} \subseteq \mathcal{M}(n_i)$ and $\mathcal{M}_{n_k} \cup \mathcal{M}_{n_j} \rightarrow \perp$. ■

To construct a hybrid approach, we exploit Property 4. The importance of Property 4 is that it provides the basis for efficient filtering. The mutual independence of $\delta_2, ..., \delta_n$ enables us to compute extensions using a δ_i in arbitrary order. After each step in Algorithm 1 resulting in a node n_i we can then make a decision: continue with a submodel of $\mathcal{M}(n_i)$, or even just a subset of branches. Because the algorithm partitions the model space, any filtering reduces the model space along the dimensions considered by the δ's leading up to n_i. Property 3 and Property 4 ensure that we only continue building consistent models that will not expand (through closure computation) to include submodels implied by material we filtered out.

V. CONCLUSIONS

We discussed the main representational aspects of the situated multi-agent models we have developed for CogX, and presented several forms of inference over these representations. Characteristic for all forms of inference is that they can deal with the uncertainty inherent to these models, while at the same time exploiting logical structure. As we have found, this has its limits. Powerful statistical relational models are not yet capable of dealing with the types of uncertainty in observations ("soft evidence") that is typical for robotic applications. This currently requires a simplification of the model for the purpose of online inference, due to representational overhead that arises from having to explicitly encode uncertainty as additional rules. We are looking into possible solutions to this, along complementary lines. One line of research concerns the reformulation of MLN inference to allow for soft evidence, and perform inference in an any-time fashion. Another line concerns the restriction of the search space over which inference ranges. For this, we are considering the approach described in §IV.

The approaches discussed in this paper are part of the integrated systems developed for CogX. Completion reasoning is used in conceptual mapping in Dora. Based on (uncertain) beliefs, we infer possible interpretations of what an area might be. We associate probabilities with the resulting closure(s), to reflect typicality: "a kitchen typically has a coffee machine, (with probability p)." These probabilities can then be combined (outside the closure computation) with the actual evidence, to finalize the categorization. The probabilistic abductive and -deductive inferences are used in situated dialogue processing.

References

[1] J. R. Hobbs, M. E. Stickel, D. Appelt, and P. Martin, "Interpretation as abduction," AI Center, SRI International, Menlo Park, CA, USA, Tech. Rep. 499, Dec 1990.

[2] M. E. Stickel, "A Prolog-like inference system for computing minimum-cost abductive explanations in natural-language interpretation," AI Center, SRI International, Menlo Park, CA, USA, Tech. Rep. 451, Sep 1988.

[3] M. Baldoni, L. Giordano, and A. Martelli, "A modal extension of logic programming: Modularity, beliefs and hypothetical reasoning," *Journal of Logic and Computation*, vol. 8, no. 5, pp. 597–635, 1998.

[4] G. Kruijff, M. Janicek, and P. Lison, "Continual processing of situated dialogue in human-robot collaborative activities," in *Proceedings of the 19th IEEE International Symposium in Robot and Human Interactive Communication*. Viareggio, Italy: IEEE, September 2010.

[5] E. Charniak and S. E. Shimony, "Probabilistic semantics for cost based abduction," in *AAAI-90 proceedings*, 1990.

[6] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach (2nd ed.)*. Prentice Hall, 2003.

[7] D. Poole, "Probabilistic horn abduction and bayesian networks," *Artificial Intelligence*, vol. 64, no. 1, pp. 81–129, 1993.

[8] R. Kate and R. Mooney, "Probabilistic abduction using Markov Logic Networks," in *Proceedings of the IJCAI-09 Workshop on Plan, Activity, and Intent Recognition (PAIR-09)*, Pasadena, CA, July 2009.

[9] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[10] D. Koller, N. Friedman, L. Getoor, and B. Taskar, "Graphical models in a nutshell," in *Introduction to Statistical Relational Learning*, L. Getoor and B. Taskar, Eds. MIT Press, 2007.

[11] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[12] H. Poon and P. Domingos, "Sound and efficient inference with probabilistic and deterministic dependencies," in *AAAI'06: Proceedings of the 21st national conference on Artificial intelligence*. AAAI Press, 2006, pp. 458–463.

[13] S. Riedel, "Improving the accuracy and efficiency of MAP inference for markov logic," 2008, pp. 468–475.

[14] P. Singla and P. Domingos, "Lifted first-order belief propagation," in *AAAI'08: Proceedings of the 23rd national conference on Artificial intelligence*. AAAI Press, 2008, pp. 1094–1099.

[15] D. Lowd and P. Domingos, "Efficient weight learning for markov logic networks," in *PKDD 2007: Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases*. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 200–211.

[16] T. N. Huynh and R. J. Mooney, "Max-margin weight learning for markov logic networks," in *ECML PKDD '09: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 564–579.

[17] S. Kok and P. Domingos, "Learning the structure of markov logic networks," in *ICML '05: Proceedings of the 22nd international conference on Machine learning*. New York, NY, USA: ACM, 2005, pp. 441–448.

[18] L. Mihalkova and R. J. Mooney, "Bottom-up learning of markov logic network structure," in *ICML '07: Proceedings of the 24th international conference on Machine learning*. New York, NY, USA: ACM, 2007, pp. 625–632.

[19] H.-U. Krieger, "A temporal extension of hayes-/ter horst-style entailment rules," in *submitted for publication*, 2010.

[20] S. Bechhofer, F. van Harmelen, J. Hendler, I. Horrocks, D. L. McGuinness, P. F. Patel-Schneider, and L. A. Stein, "OWL Web Ontology Language Reference," W3C, Tech. Rep., 2004, 10 February.

[21] F. Manola and E. Miller, "RDF primer," W3C, Tech. Rep., 2004, 10 February.

[22] J. Lloyd, *Foundations of Logic Programming*, 2nd ed. Springer, 1987.

[23] P. Blackburn and J. Bos, *Representation and Inference for Natural Language*, ser. CSLI Studies in Computational Linguistics. Stanford: CSLI Publications, 2005.

[24] A. Kiryakov, D. Ognyanov, and D. Manov, "OWLIM – a pragmatic semantic repository for OWL," in *Proceedings of the International Workshop on Scalable Semantic Web Knowledge Base Systems*, 2005, pp. 182–192.

[25] H.-U. Krieger, "Where temporal description logics fail: Representing temporally-changing relationships," in *KI 2008: Advances in Artificial Intelligence*, ser. Lecture Notes in Artificial Intelligence, vol. 5243. Springer, 2008, pp. 249–257.

[26] P. Hayes, "RDF semantics," W3C, Tech. Rep., 2004, w3C Recommendation, http://www.w3.org/TR/rdf-mt/.

[27] H. J. ter Horst, "Combining RDF and part of OWL with rules: Semantics, decidability, complexity," in *Proceedings of the International Semantic Web Conference*, 2005, pp. 668–684.

A General Methodology for Equipping Ontologies With Time[*]

Hans-Ulrich Krieger

German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, D-66123 Saarbrücken, Germany
krieger@dfki.de

Abstract

In the *first* part of this paper, we present a framework for enriching arbitrary upper or domain-specific ontologies with a concept of time. To do so, we need the notion of a time slice. Contrary to other approaches, we directly interpret the original entities as time slices in order to (i) avoid a duplication of the original ontology and (ii) to prevent a knowledge engineer from ontology rewriting. The diachronic representation of time is complemented by a sophisticated time ontology that supports underspecification and an arbitrarily fine granularity of time. As a showcase, we describe how the time ontology has been interfaced with the PROTON upper ontology. The *second* part investigates a temporal extension of RDF that replaces the usual triple notation by a more general tuple representation. In this setting, Hayes/ter Horst-like entailment rules are replaced by their temporal counterparts. Our motivation to move towards this direction is twofold: firstly, extending binary relation instances with time leads to a massive proliferation of useless objects (independently of the encoding); secondly, reasoning and querying with such extended relations is extremely complex, expensive, and error-prone.

1. Introduction

The first part of this paper presents a framework for enriching arbitrary upper or domain-specific ontologies with a concept of time. The work reported here is part of an EU-funded project called MUSING which is dedicated to the investigation of semantic-based business intelligence solutions.

Temporal information in MUSING is based on a diachronic representation of time, on top of which temporal reasoning services are defined (Krieger et al., 2008a). Since ontological knowledge in MUSING is encoded in OWL (McGuinness and van Harmelen, 2004), extending binary relations with an additional time argument is not that easy, due to the fact that OWL (or description logic in general) only provides unary and binary relations.

In order to equip ontologies with time, we need the notion of a time slice, as explained, e.g., in (Sider, 2001). Contrary to (Welty and Fikes, 2006), we directly interpret the original entities as time slices in order to (i) avoid a duplication of the original ontology and (ii) to prevent a knowledge engineer from ontology rewriting.

We will see that this reinterpretation makes it easy to extend an upper/domain ontology with time. The diachronic representation of time is complemented by a sophisticated time ontology that supports underspecification and an arbitrarily fine granularity of time.

MUSING makes use of a general upper-base ontology called PROTON (http://proton.semanticweb.org) that has been extended mostly by the MUSING partners from STI (formerly DERI), Innsbruck. As a showcase, we describe how the time ontology has been interfaced with PROTON. The OWL implementation of the methodology reported here (plus a general time ontology) can be obtained freely from the author.

Even though our approach keeps the original ontology, it leads to a massive proliferation of "container" objects, due to the fact that the underlying data structure is still the RDF *triple* (Klyne and Carroll, 2004). Furthermore and very important, reasoning and querying with such a representation is extremely complex, expensive, and error-prone. It is worth noting that all other approaches, as presented in section 3., do suffer from the same disadvantage.

In order to overcome this problem, we propose to add some kind of temporal annotation to an RDF triple, realized as further temporal arguments (starting and ending time). We describe an extension of Hayes/ter Horst-like RDFS/OWL entailment rules that are "sensitive" to temporal information. We show that only lightweight reasoning capabilities are needed when working with such information.

The work reported in the second part of this paper is an outcome of the lessons learned from the MUSING project and is actively used and extended in the CogX project, whose aim is to develop a unified theory of self-understanding and self-extension with a convincing instantiation and implementation of this theory in a robot.

The representation of generalized tuples, reasoning with them and querying them is realized through *HFC*, a forward chainer developed at DFKI that scales up to millions of tuples, which is reasonable fast and expressive enough to formulate the extended entailment rules.

2. A Motivating Example

The problem with so-called *synchronic* relationships is that they all refer to only *one*, potentially hidden point/period in/of time. Here is an example:

Tony Blair was born on May 6, 1953.

Assuming a RDF-based representation, an information extraction system might compute the following set of triples:

```
tb rdf:type Person .
tb hasName "Tony Blair" .
tb dateOfBirth "1953-05-06" .
```

However, most relationships are *diachronic*, i.e., they embody the possibility to vary with time. Take, for instance,

[*]The research described here has been financed by the European Integrated projects **MUSING** (Multi-Industry Semantic-Based Business Intelligence, http://musing.eu/) and **CogX** (Cognitive Systems that Self-Understand and Self-Extend, http://cogx.eu) under contract numbers FP6-027097 and FP7 ICT-215181. I would like to thank the three referees for their valuable comments.

the following example:

> *Christopher Gent was Vodafone's chairman until July 2003. Later, Chris became the chairman of GlaxoSmithKline with effect from 1st Jan 2005.*

When applying the synchronic representation scheme from above, however, the resulting RDF graph mixes up the association between the fact and the temporal extend (two out of four possibilities are wrong):

```
cg isChairman vf .
cg isChairman gsk .
cg hasTime [????-??-??,2003-07-??]  .
cg hasTime [2005-01-01,????-??-??]  .
```

No longer is it clear whether `[????-??-??,2003-07-??]` belongs to `vf` or `gsk` (same holds for `[2005-01-01, ????-??-??]`).

3. Approaches to Diachronic Representation

Several well-known techniques of extending binary relations with additional arguments have been proposed in the literature. (Welty and Fikes, 2006) mention three of them and add a fourth one (4D or perdurantist view; see below), which we reinterpret w.r.t. an upper or domain ontology. This reinterpretation is the basis for representing temporal information in MUSING and one of the topics of this paper, since it opens a way to enrich arbitrary ontologies with the concept of time, without any ontology rewriting.

3.1. Equip Relations With a Temporal Argument

This approach has been pursued in temporal databases and the logic programming community. A binary relation, such as *hasCeo* between a company c and a person p becomes a ternary relation with a further temporal argument t (we limit ourself to one further argument encoding an interval, instead of two, representing the starting and ending time of an interval):

$$\mathit{hasCeo}(c,p) \longmapsto \mathit{hasCeo}(c,p,\underline{t})$$

Unfortunately, OWL and description logic in general only support unary (classes) and binary relations (properties) in order to guarantee decidability of the usual inference problems. Thus, forward chainers (such as OWLIM and Jena) as well as description logic reasoners (e.g., Racer or Pellet) are unable to handle such descriptions.

We note here that this approach is clearly the *silver bullet* of representation, since it is the easiest and most natural one, although a direct interpretation is incompatible with RDF and currently available reasoners. We will favor this kind of representation in the second part.

3.2. Apply a Meta-Logical Predicate

McCarthy & Hayes' situation calculus, James Allen's interval logic, and the knowledge representation formalism KIF use the meta-logical predicate *holds*. Hence, our *hasCeo* relation becomes

$$\mathit{hasCeo}(c,p) \longmapsto \underline{\mathit{holds}}(\mathit{hasCeo}(c,p),\underline{t})$$

McCarthy & Hayes call a statement whose truth value changes over time a *fluent* (McCarthy and Hayes, 1969). Thus the extended ternary relation from the previous subsection is a *relational* fluent. The *holds* expression here, however, embodies a *functional* fluent, meaning that $\mathit{hasCeo}(c,p)$ is assumed to yield a situation-dependent value. Such kinds of relations are not possible in OWL, since description logics limit themselves to subsets of function-free first order logic.

3.3. Reify the Original Relation

Reifying a relation instance leads to the introduction of a new object and four additional new relationships. In addition, a new class needs to be introduced for each reified relation, plus accessors to the original arguments. Furthermore and very important, relation reification loses the original relation, requiring a modification of the original ontology. Coming back to our *hasCeo* example, we get something like this (*HasCeo* is the newly introduced class):

$$\begin{aligned}&\underline{\mathit{hasCeo}(c,p,t)} \;\longmapsto\; \exists e\,.\\&\quad \mathit{type}(e,\mathit{HasCeo}) \wedge \mathit{hasTime}(e,t) \wedge\\&\quad \mathit{company}(e,c) \wedge \mathit{person}(e,p)\end{aligned}$$

3.4. Encode the 4D View in OWL

(Welty and Fikes, 2006) have presented an implementation of the 4D or perdurantist view in OWL, using so-called time slices (Sider, 2001), encoding the time dimension of spacetime.[1] Relations from the original ontology no longer connect the original entities, but instead connect time slices that belong to those entities. A time slice is merely a container for storing time. For a given ontology, such a representation requires a lot of rewriting:

$$\begin{aligned}&\underline{\mathit{hasCeo}(c,p,t)} \;\longmapsto\; \exists \mathit{ts}_1,\mathit{ts}_2\,.\\&\quad \mathit{type}(\mathit{ts}_1,\mathit{TimeSlice}) \wedge \mathit{hasTimeSlice}(c,\mathit{ts}_1) \wedge\\&\quad \mathit{type}(\mathit{ts}_2,\mathit{TimeSlice}) \wedge \mathit{hasTimeSlice}(p,\mathit{ts}_2) \wedge\\&\quad \mathit{hasTime}(\mathit{ts}_1,t) \wedge \mathit{hasTime}(\mathit{ts}_2,t) \wedge\\&\quad \mathit{hasCeo}(\mathit{ts}_1,\mathit{ts}_2)\end{aligned}$$

3.5. Reinterpret the 4D View

In MUSING, we have reinterpreted the perdurantist/4D view in that we have reinterpreted the original entries from the ontology. The basic idea can be summarized in the following slogan:

> *What has been an entity becomes a time slice.*

In the example above, c and p are no longer entities, but instead time slices of an entity (a perdurant), that explain the behavior of an entity within a certain extension or point in time (e.g., that c is a time slice talking about a company or p a time slice, dealing with a person).

This reinterpretation does not need any ontology rewriting and makes it easy to equip arbitrary upper/domain ontologies with the concept of time. Coming back to our example, we have

$$\mathit{hasCeo}(c,p,\underline{t}) \;\longmapsto$$

[1] In the 4D view, all entities (the *perdurants*) only exist for some period of time. Given this view, it does not matter whether we are talking about an accidental, perhaps infinitely-small event (say, the shooting of a pistol) or a very long time interval (e.g., the lifetime of our universe). Entities under this view are often referred to as *spacetime worms* (Sider, 2001), since a four-dimensional trajectory identifies a perdurant in time and space.

$$\mathit{hasCeo}(c, p) \land \mathit{hasTime}(c, t) \land \mathit{hasTime}(p, t) \land \mathit{hasTimeSlice}(C, c) \land \mathit{hasTimeSlice}(P, p)$$

Note that the former binary predicate *hasCeo* is still available and unchanged. But the argument classes, viz., Company and Person have been equipped with an additional relation called *hasTime*, defined on class *TimeSlice*, as we will see later. Given this representation, everything that is defined on c, such as the CEOship, the name, the address, or the number of employees of this company, is assumed to co-occur during time period t. I.e., different facts speaking about the same time interval of the same individual in the first place of the relation need *not* to be encoded in different time slices. Furthermore, the original entities p and c are linked to perdurants P and C which, however, only need to be created once.

The 4D reinterpretation is easier than Welty&Fike's original formulation, viewed from the standpoint of complexity. Let us have a look at the domain (D) and range (R) of the above *hasCeo* property, using abstract description logic syntax:

- **Welty & Fikes (2006)**
  (D) $\exists \mathsf{hasCeo} . \top \sqsubseteq \forall \mathsf{hasTimeSlice}^{-} . \mathsf{Company}$
  (R) $\top \sqsubseteq \forall \mathsf{hasCeo} . (\forall \mathsf{hasTimeSlice}^{-} . \mathsf{Person})$
- **4D reinterpretation**
  (D) $\exists \mathsf{hasCeo} . \top \sqsubseteq \mathsf{Company}$
  (R) $\top \sqsubseteq \forall \mathsf{hasCeo} . \mathsf{Person}$

As we have already noticed, this reinterpretation also makes it easy to interface arbitrary ontologies with existing time ontologies. We will see this in a moment.

4. The Perdurant Ontology

Given the above discussion, this section now presents the basic ontology for perdurants and time slices used in the MUSING project that is, however, directly applicable to other applications and projects that deal with changing relationships over time in RDF. Here is the overall picture:

Perdurant: hasTimeSlice
TimeSlice: timeSliceOf, hasTime
Time

Let us describe the three top-level classes that are only necessary. Objects whose properties change over time are called *perdurants*, as already explained above. Those objects possess a number of *time slices*, hence we need a property hasTimeSlice in order to access their time slices. A time slice specifies an extension in time through the functional property hasTime and is associated with a perdurant via timeSliceOf, the inverse relation to hasTimeSlice. A time slice "contains" those properties whose values stay constant over the specified period of time. The range of hasTime is exactly an object of class Time which will be described in a moment:

$$\top \sqsubseteq \leq 1 \mathsf{hasTime} \sqcap \forall \mathsf{hasTime.Time}$$

We note here that this simple ontology is completely open to the choice of the *time ontology* (and open to the upper/domain ontology that is equipped with a concept of time). Thus it will be possible to interface the perdurant ontology with popular time ontologies, such as Hobbs&Pan's OWL Time (Hobbs and Pan, 2004). This is achieved through the above mentioned class Time, a simple placeholder that is interfaced with the corresponding class in the time ontology. Similarly, the placeholder TimeSlice needs to be interfaced with the corresponding concept(s) in the upper/domain ontology. This will be shown in section 5.

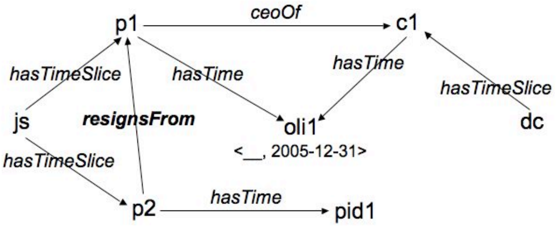
4.1. Flexible Semantic Representation

Let us focus on a natural language example and its (simplified) representation to see how things go together:

> *DaimlerChrysler's CEO Schrempp announces that he will resign by 31st December 2005.*

Consider that an information extraction system has find out that Jürgen Schrempp and DaimlerChrysler are named entities. Consequently, we introduce two perdurants *js* and *dc* for these entities (assuming that they have not already been introduced).

The fact that Schrempp was CEO of DC until 31st December 2005 is expressed by a time slice p_1 (of type Person) that contains an instance oli_1 of class OpenLeftInterval, whereas his resignation is encoded in a time slice p_2 (again of type Person) that is temporally anchored in an instance pid_1 which is of class ProperInstantDay, having value "2005-12-31".



Notice that Schrempp did not resign from DC, but instead resigned from DC's ceoship. Thus property resignsFrom points to p_1 that expresses Schrempp's ceoship with DaimlerChrysler.

4.2. Advantages of the Approach

Firstly, properties that do not change over time (e.g., birthdate) can be relocated from TimeSlice to Perdurant (no duplication of information). Time-varying information instead is kept in a series of time slice. If several properties of a perdurant are constant over the *same* period of time, we do not need several time slices.

Secondly, the subtypes of TimeSlice specify the *behavior* of a *perdurant* within a certain time interval (e.g., whether a perdurant *acts* as a company, a person, etc.). We will see in a moment how this can be achieved.

Thirdly, since hasTimeSlice is typed to TimeSlice, different slices of the same perdurant need *not* to be of the same type. For instance, the perdurant SRI might have a time slice for Company as well as a slice for AcademicInstitution, i.e., a perdurant can act in different ways.

Fourthly, representing modalities, such as *believe* can be achieved relatively easy. Representing space and movements in space can be modeled similarly.

Finally, Allen's 13 temporal topological interval relations (Allen, 1983) can be naturally extended to time slices.

5. Extending Ontologies With Time

As promised, we now describe how we have interfaced the 4D and the time ontology with an upper/domain ontology, in our case PROTON (http://proton.semanticweb.org).

Before going into the details, let us remark that our global ontology consists of concepts and properties that implement a 4D perdurantist view, but also deals with time in general, building on instants and intervals (and their subclasses). So we get the following picture for the merged ontology PROTime:

$$\begin{tikzcd}
 & \text{4D} \arrow[d] & \\
\text{Time} \arrow[r] & \text{PROTime} & \text{[Allen]} \arrow[l] \\
 & \text{PROTON} \arrow[u] &
\end{tikzcd}$$

The 4D reinterpretation which we have presented so far says that the *original* entities should be regarded as *time slices*. To do so, one need to identify the most general classes in PROTON (or in another arbitrary upper/domain ontology) that are supposed to be extended by a temporal dimension—actually, we are interested in the domain/range classes of the time-varying properties. There is such a single, most general class in PROTON: psys:Entity. Thus we only need a single axiom, employing `owl:equivalentClass`:

$$\mathsf{fourd{:}TimeSlice} \equiv \mathsf{psys{:}Entity}$$

In general, a new integrated ontology is constructed as follows:

1. **always use 4D**
   Perdurant: hasTimeSlice
   TimeSlice: timeSliceOf, hasTime
   Time
2. **choose time**
   an arbitrary time ontology (e.g., OWL Time)
3. **choose upper/domain ontology**
   the original ontology (e.g., PROTON)
4. **choose Allen (optional)**
   Allen relations over time slices

plus an equivalence statement of the above kind.

Note that the class Time in the 4D ontology is a simple placeholder used in $\mathsf{hasTime} \subseteq \mathsf{TimeSlice} \times \mathsf{Time}$. When interfacing 4D with an arbitrary time ontology, one needs to say what is meant by Time, in our case:

$$\mathsf{fourd{:}Time} \equiv \mathsf{time{:}TemporalEntity}$$

We will describe TemporalEntity and the time ontology in the next section.

In case there will be several maximal incompatible classes $c_1, \ldots, c_n$ that need to be extended by a temporal dimension, the above axiom clearly becomes

$$\mathsf{fourd{:}TimeSlice} \equiv c_1 \sqcup \ldots \sqcup c_n$$

6. The Time Ontology

In this section, we will describe the time ontology that we have employed in MUSING. We have opted against OWL Time (Pan, 2007), a rich first-order axiomatization of time, since we have decided to model temporal underspecification in natural language and granularity of time through a subtyping hierarchy. The ontology described here, however, is fully compatible with OWL Time through the use of the class TemporalEntity as well as its subclasses Instant and Interval. Here is the overall picture:

```
TemporalEntity
  Instant
    NegativeInfinity
    PositiveInfinity
    ProperInstantYear: year
      ProperInstantMonth: month
        ProperInstantDay: day
          ProperInstantHour: hour
            ProperInstantMinute: minute
              ProperInstantSecond: second
                .....
  Interval: begins, ends
    OpenLeftInterval
      ClosedInterval
        Forever
        .....
    OpenRightInterval
      ClosedInterval
        Forever
        .....
```

OWL *classes* start with uppercase letter characters; *properties* are written in lower case. Thus

Interval: begins, ends

means that properties begins and ends are defined on class Interval. *Indentation* expresses subtyping/subclassing. Subtyping also means that properties defined on superclasses are also available in subclasses. Hence, the properties year and month are also accessible in class ProperInstantDay.

Let us quickly describe the most top-level classes. We distinguish between two exhaustive partitioning and disjoint subclasses of TemporalEntity: Instant and Interval.

$$\mathsf{TemporalEntity} \equiv \mathsf{Interval} \sqcup \mathsf{Instant}$$

$$\mathsf{Interval} \sqsubseteq \neg\, \mathsf{Instant}$$

Instant is used to describe infinitely short events (i.e., instants), whereas Interval identifies measurable periods of time. Thus, Interval possesses two properties: begins and ends, both returning an instant. All classes above are expressed as OWL axioms.

We now give a more complex example—the definition of ClosedInterval:

$$\begin{aligned}
&\mathsf{ClosedInterval} \equiv \\
&\quad \mathsf{OpenLeftInterval} \sqcap \mathsf{OpenRightInterval} \sqcap \\
&\quad {=}1\,\mathsf{begins} \sqcap {=}1\,\mathsf{ends} \sqcap \\
&\quad \exists\, \mathsf{begins.Instant} \sqcap \exists\, \mathsf{ends.Instant}
\end{aligned}$$

This definition says that begins and ends must be specified exactly once. begins and ends must furthermore be assigned an instance of (at least) type Instant.

ProperInstantYear, PositiveInfinity, and NegativeInfinity are declared as being mutually disjoint:

$$\mathsf{ProperInstantYear} \sqsubseteq \neg\, \mathsf{NegativeInfinity}$$

$\mathsf{ProperInstantYear} \sqsubseteq \neg\ \mathsf{PositiveInfinity}$
$\mathsf{PositiveInfinity} \sqsubseteq \neg\ \mathsf{NegativeInfinity}$

Actually, saying that begins takes exactly one value is done in the direct superclass OpenRightInterval (same for ends and class OpenLeftInterval). begins and ends are being declared as functional on the very general Interval class. Functionality clearly means that a value need not to be present (as can be seen, e.g., for property ends in class OpenRightInterval):

$\leq 1 \mathsf{begins} \sqsubseteq \mathsf{Interval}$
$\leq 1 \mathsf{ends} \sqsubseteq \mathsf{Interval}$

begins and ends furthermore take objects of type Instant as values:

$\top \sqsubseteq \forall \mathsf{begins.Instant}$
$\top \sqsubseteq \forall \mathsf{ends.Instant}$

Given NegativeInfinity and PositiveInfinity, the definition for the time period Forever is easy:

$$\begin{array}{l} \mathsf{Forever} \equiv \\ \quad \mathsf{ClosedInterval} \sqcap \\ \quad \exists \mathsf{begins.NegativeInfinity} \sqcap \\ \quad \exists \mathsf{ends.PositiveInfinity} \end{array}$$

ClosedInterval has further subclasses that we only mention here:

- ClosedInterval
  - Day
    - Monday, Thuesday, ...
    - SpecialDay
      - Christmas, NewYearsEve
  - Month
    - January, February28, February29, ...
  - Quarter
    - FirstQuarter, SecondQuarter, ...
  - Season
    - Spring, Summer, ...
  - Year
    - Year365, Year366

Let us finally focus in this section on the definition of two of these classes in order to flesh out this framework, viz., Day and NewYearsEve:

$$\begin{array}{l} \mathsf{Day} \equiv \\ \quad \mathsf{ClosedInterval} \sqcap \\ \quad \exists\ \mathsf{begins.ProperInstantDay} \sqcap \\ \quad \exists\ \mathsf{ends.ProperInstantDay} \end{array}$$

$$\begin{array}{l} \mathsf{NewYearsEve} \equiv \\ \quad \mathsf{SpecialDay} \sqcap \\ \quad \exists \mathsf{begins.}(\exists \mathsf{month.}\{12\} \sqcap \exists\ \mathsf{day.}\{31\}) \sqcap \\ \quad \exists \mathsf{ends.}(\exists \mathsf{month.}\{12\} \sqcap \exists\ \mathsf{day.}\{31\}) \end{array}$$

It is worth noting that even though we have specified a value for properties month and day, the definition of NewYearsEve misses the value for year. But this is correct and only get assigned in examples such as *New Year's Eve 2007* which will be modeled as an instance of class NewYearsEve, having value 2007 for property year. Otherwise, such an expression is underspecified w.r.t. to the value of year, as in the sentence *Over New Year's Eve I have visited the Eiffel Tower.*

Further subclasses of Instant and ClosedInterval help to deal with the *granularity* of time and the *underspecifiction* of time in natural language. We will address this in the next section.

7. Granularity and Underspecification

Granularity of time, i.e., the degree of how finely time is measured and the *temporal underspecification* of natural language expressions are closely related topics. Consider, for instance, the following example:

> *In 1995, Edzard Reuter handed over the CEOship of Daimler Benz AG to Schrempp.*

and assume that a year is the smallest amount of time that we want to measure. Thus the starting point for enriching the RDF triple

```
js ceoOf db .
```

is 1995 and this temporal information will be encoded via an instance of class ProperInstantYear–remember, we measure things no finer than a year. Since ProperInstantYear only possesses the property year and since this year is known, 1995 is a *fully specified* temporal expression, according to the measure we have applied.
Independent of the degree of measurement, one can clearly ask what is meant by *1995* here. Within the above context, 1995 probably does not refer to the instant 1995-01-01T00:00:00, assuming we would measure even seconds. Instead, 1995 expresses the fact that there *exists* an *interval* that *starts* somewhere in 1995 in which Schrempp started his CEOship with Daimler Benz. Since the temporal end point of the above fact is *not* known at this moment (but the starting point) and since the time of Schrempp's CEOship is probably not infinitely small, we encode this interval information in an instance of class OpenRightInterval.
This very simple example shows that temporal underspecification happens to appear on two levels:

1. instances of Instant might be underspecified in case not every property (year, month, day, ...) has been given a value;

2. instances of Interval might be underspecified in case its properties begins and/or ends have not been given a value *or* in case begins and/or ends are assigned a value (instances of Instant), this value is underspecified.

The recursive part of this definition for temporal underspecification is applied in the following sentence:

> *Between 1995 and 2005, Schrempp was the CEO of DC.*

Now assume our fineness of time is measured in terms of days, thus we generate two instances of ProperInstantDay that fill the slots begins and ends of an instance of ClosedInterval. Even though this interval is closed, its beginning and end points are underspecified, hence this closed interval is regarded as being underspecified. If we, however, had measured time in terms of years, the above natural language description would have led to a totally specified closed interval. It should be clear that further textual information might close an open-left/open-right interval. Textual information might even make a partially underspecified instant or interval total.

The above examples are fully compatible with the property restrictions imposed on begins, ends, year, month, day, etc., viz., being functional properties (0 or 1 value). In case we want to enforce a property to be instantiated, e.g., that begins and ends are "present" on ClosedInterval, we have applied a local number restriction on this specific class (see description logic axioms above).

We finally note that our approach to underspecification is a result of the subclass hierarchy of proper instants which applies a more finer measuring system when moving down the classes. An alternative, albeit less satisfying approach to underspecification would apply 0/1 cardinality constraints to the properties year, month, etc. in order to "switch them off/on", depending on the predefined granularity of time.

8. An Application

Let us focus on an application that uses the above time ontology and the methodology to represent temporally changing information: *imprint monitoring*. The monitoring system described in (Federmann and Declerck, 2010) extracts imprints (and other information) from a large number of companies on a regular temporal basis. Imprints specifies, e.g., the name of a company, the postal address, its legal form, authorized executives, etc. This information and its change over time is interesting for rating agencies (such as Creditreform).

In case the imprint of a perdurant `perd` changes at time `t` (w.r.t. information recorded in the ontology), the latest time slice `old` of `perd` is closed, using `t` (actually its time interval `oldint`). A new time slice `new` (of type `Company`) is also added to the ontology, storing the new imprint. Since `new` contains the latest information whose temporal ending point is unknown, `t` is stored as the starting point of an `OpenRightInterval`. Not only new triples are build up here, but also new individuals/URIs: besides `new`, an interval object `newint` is generated. More formally, we construct the following RDF triples:

```
old fourd:hasTime oldint .
oldint rdf:type time:ClosedInterval .
oldint time:ends t .
perd fourd:hasTimeSlice new .
new rdf:type Company .
new fourd:hasTime newint .
newint rdf:type time:OpenRightInterval .
newint time:begins t .
.....     // add imprint info to new
```

Information from the ontology can be queried using the SPARQL query language, as is used to obtain the latest time slice. The ontology, the reasoning and querying services are realized by the CROWL system (Combining Rules and OWL). CROWL consists of several publicly available reasoners (viz., Pellet (Sirin et al., 2007), OWLIM (Kiryakov, 2006), and Jena (Reynolds, 2009)), running in a fixpoint loop, and is extended by a template language to implement complex aggregation rules (Krieger et al., 2008b).

9. Problems: An Example

As we indicated in the introduction, even though our approach keeps the original ontology, it leads to a massive proliferation of objects, making reasoning and querying unnecessarily complex, expensive, and error-prone. This is due to the underlying data structure, the RDF triple, and the approaches presented in section 3. do suffer from the same problem.

Let us present an example to see how complexity builds up, even for a relatively easy task. This example will then be used in the next section when a solution is presented. The task we want to achieve is the following:

> *Compute maximal intervals, given a property, e.g.,* `ceoOf`*, between time slices* `?p` *and* `?c`.

Such queries often arise in practice when temporally-anchored facts need to be extended by further incoming information. Our approach, as described in section 3.5., would require a "lengthy" Jena-like heuristic rule to solve this task, impossible to formulate in OWLIM or Pellet, since it employs two aggregates, as realized by the functions `Min2` and `Max2`:

```
?p rdf:type fourd:Perdurant
?p fourd:hasTimeSlice ?ts1
?p fourd:hasTimeSlice ?ts2
?ts1 ceoOf ?obj1
?ts1 rdf:type ?tstype
?obj1 fourd:timeSliceOf ?q
?obj1 rdf:type ?objtype
?ts2 ceoOf ?obj2
?obj2 fourd:timeSliceOf ?q
?ts1 fourd:hasTime ?i1
?ts2 fourd:hasTime ?i2
?i1 time:begins ?b1
?i1 time:ends ?e1
?i2 time:begins ?b2
?i2 time:ends ?e2
->
?ts rdf:type ?tstype
?p fourd:hasTimeSlice ?ts
?ts ceoOf ?obj
?obj fourd:timeSliceOf ?q
?obj rdf:type ?objtype
?ts fourd:hasTime ?i
?obj fourd:hasTime ?i
?i rdf:type time:ClosedInterval
?i time:begins ?min
?i time:ends ?max
?i time:ends ?max
@test
?ts1 != ?ts2
@action
?min = Min2 ?b1 ?b2
?max = Max2 ?e1 ?e2
```

Independent of the underlying approach, we immediately feel that such a rule is hard to manage and expensive, both in terms of time (when matching clauses) and space (when introducing new objects/URIS, bound to `?ts`, `?obj`, `?i`, `?min`, and `?max`.

10. A Solution

The solution we propose in this section has been realized in the reasoning engine *HFC*, developed at DFKI. The idea here is to move from RDF triples to tuples in order to extend relation instances with further (temporal) arguments, as already described in section 3.1.

To achieve this goal, we also need to conservatively extend RDFS and OWL entailment rules, as originally described in (Hayes, 2004) and (ter Horst, 2005), i.e., to make these rules sensitive to temporal information. Here are three instantiated examples that show how things are supposed to work.

Assuming that *hasCeo* is the *inverse* of *ceoOf* and that our ontology has been populated with the fact that Jürgen Schrempp was DC's CEO from 1995 until 2005, represented as $\mathit{ceoOf}(\mathit{js}, \mathit{dc}, 1995, 2005)$, we would then like to deduce that $\mathit{hasCeo}(\mathit{dc}, \mathit{js}, 1995, 2005)$ also holds.

The fact that Angelina Jolie was married with Billy Bob Thornton from 2000 until 2003 is represented by $\mathit{marriedWith}(\mathit{aj}, \mathit{bbt}, 2000, 2003)$. Given that *marriedWith* is a *symmetric* property, the following should also be the case: $\mathit{marriedWith}(\mathit{bbt}, \mathit{aj}, 2000, 2003)$.

Given that my office is part of the DFKI building, i.e., $\mathit{contains}(\mathit{dfki}, \mathit{room{+}1.26}, 1990, 2010)$ and that my old office chair was replaced in 2002, i.e., $\mathit{contains}(\mathit{room{+}1.26}, \mathit{chair42}, 2002, 2010)$, we are allowed to infer that new chair is (at least) inside the DFKI since 2002 ($\mathit{contains}(\mathit{dfki}, \mathit{chair42}, 2002, 2010)$), due to the *transitivity* of the containment relation.

Such behavior can be formalized through temporally-extended entailment rules, quite similar to the "untensed" version described in (Hayes, 2004) and (ter Horst, 2005). As we indicated above, the temporal arguments are attached to the original triples, thus we end up in quintuples, assuming that we have a starting and ending time. Here are some examples:

- *?p is inverse of ?q*
```
?p owl:inverseOf ?q
?s ?p ?o ?t1 ?t2
->
?o ?q ?s ?t1 ?t2
```

- *?p is a symmetric property*
```
?p rdf:type owl:SymmetricProperty
?s ?p ?o ?t1 ?t2
->
?o ?p ?s ?t1 ?t2
```

- *?p is a transitive property*
```
?p rdf:type owl:TransitiveProperty
?x ?p ?y ?t1 ?t2
?y ?p ?z ?t3 ?t4
->
?x ?p ?z ?t5 ?t6
@action
?t5 = Max2 ?t1 ?t3
?t6 = Min2 ?t2 ?t4
```

- *copy subject for owl:sameAs*
```
?x owl:sameAs ?y
?x ?p ?z ?t1 ?t2
->
?y ?p ?z ?t1 ?t2
```

- *enforce domain restriction*
```
?p rdfs:domain ?dom
?s ?p ?o ?t1 ?t2
->
?s rdf:type ?dom
```

- *universal instantiation*
```
?i rdf:type ?c ?t1 ?t2
?c rdfs:subClassOf ?d
->
?i rdf:type ?d ?t1 ?t2
```

Note that only relation instances from the ABox are (usually) extended with temporal information—at the moment, we do not think that terminological knowledge needs to be equipped this way (e.g., that the domain/range restrictions of a property or the subtype relation between two classes only hold for some period of time).

Let us now come back to the example from the previous section that tries to build a contiguous interval from its two input intervals. Here is the new version:

```
?p ceoOf ?c ?b1 ?e1
?p ceoOf ?c ?b2 ?e2
->
?p ceoOf ?c ?min ?max
@test
?b1 != ?b2
?e1 != ?e2
@action
?min = Min2 ?b1 ?b2
?max = Max2 ?e1 ?e2
```

This is clearly much simpler and extremely intuitive: the only two clauses in the antecedent deal with the CEOship of a person with a company at different times and the single consequent extends the CEOship to a larger time span.

Such a rule can even be generalized to arbitrary properties which persist through time in a similar way. Assuming that such properties are characterized as subproperties of `ContinuousProperty`, the above rule becomes

```
?r rdfs:subPropertyOf ContinuousProperty
?p ?r ?c ?b1 ?e1
?p ?r ?c ?b2 ?e2
->
?p ?r ?c ?min ?max
.....
```

Even though the old rule is extremely complex, both rules only "look" at two intervals. Now, assuming that we want to glue n intervals together, both rules require $n-1$ iterations to compute the maximal interval. The number of rule applications is even larger: $(n-1) \times 2 \times \sum_{i=1}^{n-1} i$.

In order to overcome this last obstacle, we need *aggregation rules* that differ from ordinary rules in that variables do not bind only one individual at a time, but all individuals, satisfying the left-hand side constraints and the tests. This is quite similar to aggregates as used in query languages (e.g., in SQL), except that the queried information is used to instantiate further tuples which are then added to the ontology.

HFC provides us with such aggregation rules. The above rule even becomes more simple; the important point, however, is that one rule application immediately yields the maximal interval. Note the different arrow sign => to indicate that the below rule aggregates information through `MinN` and `MaxN`:

```
?p ceoOf ?c ?b ?e
=>
?p ceoOf ?c ?min ?max
@action
?min = MinN ?b
?max = MaxN ?e
```

11. Conclusion

In this paper, we have presented two approaches that are able to enrich arbitrary ontologies with a concept of time.

The first approach implements a 4D or perdurant view on temporally-changing information, complemented by a sophisticated time ontology that permits temporal underspecification. This approach keeps the original ontology and does not leave the territory of RDF. This approach was used in the MUSING project.

The lessons, we learned in MUSING, have led us to a second approach that is much simpler, more expressive, and more efficient, but requires to move from RDF triples to general tuples. Temporal information here is directly attached to the relation instance. We have indicated how RDFS and OWL entailment rules can be conservatively extended to make them sensitive to temporal information. This approach is currently employed in the CogX project.

12. References

James F. Allen. 1983. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

Christian Federmann and Thierry Declerck. 2010. Extraction, merging, and monitoring of company data from heterogeneous sources. In *Proceedings LREC 2010.*

Patrick Hayes. 2004. RDF semantics. Technical report, W3C.

Jerry Hobbs and Feng Pan. 2004. An ontology of time for the Semantic Web. *ACM Transactions on Asian Language Processing (TALIP)*, 3(1):66–85.

Atanas Kiryakov. 2006. OWLIM: balancing between scalable repository and light-weight reasoner. Presentation of the Developer's Track of WWW2006.

Graham Klyne and Jeremy J. Carroll. 2004. Resource description framework (RDF): Concepts and abstract syntax. Technical report, W3C. 10 February.

Hans-Ulrich Krieger, Bernd Kiefer, and Thierry Declerck. 2008a. A framework for temporal representation and reasoning in business intelligence applications. In *AAAI 2008 Spring Symposium on* AI Meets Business Rules and Process Management, pages 59–70. AAAI.

Hans-Ulrich Krieger, Bernd Kiefer, and Thierry Declerck. 2008b. A hybrid reasoning architecture for business intelligence applications. In *8th International Conference on Hybrid Intelligent Systems, HIS-2008*, pages 843–848. IEEE.

John McCarthy and Patrick J. Hayes. 1969. 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.

Deborah L. McGuinness and Frank van Harmelen. 2004. OWL Web Ontology Language Overview. Technical report, W3C. 10 February.

Feng Pan. 2007. *Representing Complex Temporal Phenomena for the Semantic Web and Natural Language.* Ph.D. thesis, University of Southern California.

Dave Reynolds. 2009. Jena 2 inference support (version 1.40). http://jena.sourceforge.net/inference/index.html.

Theodore Sider. 2001. *Four Dimensionalism. An Ontology of Persistence and Time.* Oxford University Press.

Evren Sirin, Bijan Parsia, Bernardo Cuenca-Grau, Aditya Kalyanpur, and Yarden Katz. 2007. Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*, 5(2).

Herman J. ter Horst. 2005. Combining RDF and part of OWL with rules: Semantics, decidability, complexity. In *Proceedings of the International Semantic Web Conference*, pages 668–684.

Christopher Welty and Richard Fikes. 2006. A reusable ontology for fluents in OWL. In *Proceedings of Fourth International Conference on Formal Ontology in Information Systems (FOIS)*, pages 226–236.

Learnable Controllers for Adaptive Dialogue Processing Management

Geert-Jan M. Kruijff and **Hans-Ulrich Krieger**[*]
German Research Center for Artificial Intelligence
DFKI GmbH, Saarbrücken Germany
{gj,krieger}@dfki.de

Abstract

Uncertainty is pervasive throughout processing spoken dialogue in human-robot interaction. That need not always be a problem though. The paper adopts the view that it depends on context, how much that uncertainty actually matters. The paper argues that uncertainty in input needs to be balanced off against how much actually needs to be understood to make a contextually appropriate, next move. The paper presents **work in progress** on developing mechanisms for adaptively controlling how utterances in spoken dialogue in human-robot interaction get processed "step-by-step", to deal with uncertainty in as much as necessary given an goal state. These mechanisms take the form of a learnable closed-loop controller that decides on an optimal policy or process configuration to reach a next fixed-point in a state space of (partial) analyses. The policy is planned online, adapting the processing strategy rather than using a "universal" policy.

Introduction

Uncertainty is pervasive throughout all of spoken dialogue processing. This is in part due to the very nature of spoken dialogue. Utterances are typically incomplete, or even ungrammatical. Nobody speaks alike. Meaning is highly contextualized. And with that, it presents a hard problem to solve, if spoken dialogue is to succeed as (most) natural form of communication between a human and a robot.

One way of dealing with this uncertainty is to make the processes we use as robust as possible. That way we can try and deal with missing or wrong input, and still provide an analysis. Possibly, in a divide-and-conquer strategy, providing the same input to different kinds of processes in the expectation that at least one succeeds.

But this is really only part of the answer. Or, of the question, come to that. Sometimes uncertainty matters, if we are in a situation where we require high-precision understanding. For example, when a human and a robot discuss what to do next in a rescue scenario. And sometimes it doesn't, if we only need to understand enough to produce a response that keeps the conversation going. We need to balance off uncertainty against how much actually needs to be understood to make a contextually appropriate, next move.

In this paper we discuss **work in progress** on developing mechanisms for adapting how a dialogue system processes uncertain input. The aim is to be able to plan optimal processing strategies given a contextually determined target. We propose to see this as a *sequential decision making problem.* Given an utterance that needs to be processed, and looking at how a state of possible (partial) analyses develops towards "what we would like to know," what steps do we need to take next?

These mechanisms take the form of learnable closed-loop controllers. The control state space is defined in terms of partial analyses, and (lifted) representations of goal analysis states or "desired outcomes." The action space is defined as a space over statements in a system definition language. Each statement specifies a possible process configuration that leads up to a specific fixed point. A controller defines a mapping between changes in the control state, and an "action" in the action space. These actions are configurations of processes to run over a part of the input. Processes can run sequentially and/or concurrently. The selection of an action-as-policy is based on the assumption that each process has a certain expected reward, based on the cost of running that process and the outcome it is able to yield. Running the configuration is then done as open-loop controller.

Problems and design options

Following out these ideas, there are several problems we need to address.

Problem (Control state space)**.** *We have an arbitrary number of processes, and each process contributes its own type of information to the analysis of an utterance. The* state *of unfolding analysis is to be defined from the (types of) information thus provided. The issue is that these need not be independent. For example, correcting speech recognition errors typically has an impact on how well the utterance can be parsed, and reference resolution can only take place once enough semantic representation has been built. The problem here is to find a formulation that allows for a suitable factorization of the state space that enables us to (a)*

[*]The research presented here was supported by the EU FP7 IP project "Natural Human-Robot Cooperation in Dynamic Environments," (ICT-247870; http://www.nifti.eu) and by the EUF P7 IP project "Cognitive Systems that Self-Understand and Self-Extend" (ICT-215181, http://cogx.eu).

impose structure between components, (b) use this structure to guide learning and control, and (c) make it possible for processes to “discontinuously” contribute to an unfolding analysis state.

We propose to deal with this problem by using *factored models* (Boutilier, Dean, and Hanks 1999). A factored (state) model allows the explicit representation and exploitation of structure in a state space, using a vector $X = \{X_1, ..., X_n\}$ to represent a state. We assume the individual *state factors* X_i to carry further (algebraic) structure of the form $h_i \cdot ... \cdot h_j|\ c$ to indicate a history composed of $h_i \cdot ... \cdot h_j$ yielding a current class c. To deal with the issue of co-dependency between results, we allow for lifted representations in a factored model's vectors. A lifted representation basically captures what component i expects as class of representation to appear in component j. This also yields a suitable representation of fixed-points and goal states.

Problem (Action space)**.** *Given a set of processes P that can act on one or more state factors, we want to consider an action to be a* dynamic *configuration of several processes. This configuration is to advance the analysis from the current state S to a next state S'. The issue is that we want to be able to exploit sequential and/or concurrent execution of processes, rather than focusing on atomic skills. Aside from the issue of how a configuration could be specified, we face the problem that the search space over possible configurations is highly complex.*

For dynamically specifying a configuration over processes, we propose to use the system description language $\mathcal{SDL}$ (Krieger 2003). In $\mathcal{SDL}$ we can specify how a set of processes should be combined to compute from a given input I to yield an output, possibly under the condition of a fixed-point to be reached. Originally, $\mathcal{SDL}$ has been used to specify a system configuration offline. We extend $\mathcal{SDL}$ with information about opting-out on processing input (Li, Littman, and Walsh 2008). For each process we require a specification of the state factor(s) in a state vector it can apply to. These factors define the feature spaces we use later on to learn whether a process can be effectively applied on this input, to yield a desired output. Thereby, a process can opt out ($\perp$) of computing on an input. This enables us to employ an active learning strategy for exploiting and exploring a complex search space following a framework of self-aware learning (Li, Littman, and Walsh 2008).

Problem (Controllers)**.** *Given a state S and a goal state S_g we want to reach, and a set of processes P that are applicable to S. The basic problem is how to decide on a configuration of processes $P' \subseteq P$ to apply to S, to get to an updated state S' that brings us close(r) to S_g. What makes this problem more complicated is that we want to do is in an online fashion, given a finite receding horizon (Barto, Bradtke, and Singh 1995), so that we can dynamically adapt the closed-loop control policy of what configuration to execute next.*

We see this problem as one of sequential decision making. We propose to use reinforcement learning as a way of finding effective solutions (Sutton and Barto 1998). We formulate a controller as a Markov Decision Process (MDPs), particularly as a factored-state Markov Decision Process, cf. (Boutilier, Dean, and Hanks 1999; Strehl, Diuk, and Littman 2007). For each process p in a collection of processes $\mathcal{P}$, it is defined which state factor X_i it operates on (possibly non-exclusively), and which factors $X_j \neq X_i$ it can be conditioned on. At each state s, we filter $\mathcal{P}$ to a subset of applicable processes, P. For each $p \in P$ we either have an expectation $P(s'|s)$ that $X_i^s = h_i \cdot ... \cdot h_j|\ c$ extends to $X_i^{s'} = h_i \cdot ... \cdot h_j \cdot h_{j+1}|C$, or p opts out of producing an output on X_i. This filters P down to a set of positively applicable processes, P^+. We then need to construct a configuration over processes in P^+ that forms an online determined policy π to be executed. This construction-determination takes place in a modified form of the value iteration algorithm, inspired by (Morisset and Ghallab 2008). We need to solve $E(s) = max_{a \in A} Q(s, a)$, except that the “action” a in the action space A is a configuration of processes determined from P^+. We therefore break down $Q(s, a) = U(s, a) + \gamma \Sigma_{s'\ inS} P_a(s'|s) E(s')$ with $U(s, a) = R(s) - C(s, a)$ into the components that make up a, namely $p \in P^+$ and a combination of process sequencing $(p_i + p_j)$ and/or parallelization $(|p_i, .., p_j)$. The utility of a configuration a is then the sum of the utilities for the individual components c of the configuration, defined recursively as follows. For c a process $p \in P^+$, $U(s, c) = U(s, p)$; for $c = (c_1 + ... + c_j)$, $U(s, c) = \Sigma_{1 \leq i \leq j} U(s, c_i)$; and for $c = (|c_1, ..., c_j)$, $U(s, c) = (\Sigma_{1 \leq i \leq j} R(c_i)) - \underset{1 \leq i \leq j}{max} C(c_i)$. The algorithm is invoked each time a policy finishes its (open-loop) execution, and we have an updated state that does not instantiate the goal state. The stopping criterion for the iteration is defined flexibly through a fixed point of the Bellman equation $\underset{s \in S}{max} |E_n(s) - E_{n-1}(s)| < \epsilon$.

Cost-functions $C(\cdot)$ reflect processing time. We are still considering different reward functions $R(\cdot)$. This depends in part on the type of goal state we have, which may vary in its demand on precision or recall. This still leaves open several questions, though, including how distributions such as $P(s'|s)$ can be learnt relative to a specific configuration – or, rather, given the above decomposition, relative to a applying a specific process.

Problem (Learnability)**.** *How can we acquire the probability distributions for the controller in an online fashion, without having to fully explore the (complex) state spaces?*

Here we are exploring the possibility for using a framework for self-aware learning, such as various model-based approaches like R-MAX (Brafman and Tennenholtz 2002) or SLF-MAX (Strehl, Diuk, and Littman 2007) which all fit the “knows what it knows” (KWIK) framework (Li, Littman, and Walsh 2008).

Discussion

The framework presented here is still under development. We are currently working on full formalization, and the implementation of the framework for a dialogue system that can adapt the characterization of its processing goals given the expected flow of the dialogue. The aim here is to provide the possibility of adaptively deciding at what depth an utterance should be analyzed, to obtain a result that "optimally" helps to further the dialogue (in terms of speed, and level of required understanding).

References

Alami, R.; Chatila, R.; Fleury, S.; Ghallab, M.; and Ingrand, F. 1998. An architecture for autonomy. *International Journal of Robotics Research* 17(4):315–337.

Barto, A.; Bradtke, S.; and Singh, S. 1995. Learning to act using real-time dynamic programming. *Artificial Intelligence* 72(1-2):81–138.

Boutilier, C.; Dean, T.; and Hanks, S. 1999. Decision-theoretic planning: Structural assumptions and computational leverage. *Journal of AI Research* 11:1–94.

Brafman, R., and Tennenholtz, M. 2002. R-max - a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research* 3:213–231.

Kalyanakrishnan, S., and Stone, P. 2009. An empirical analysis of value function-based and policy search reinforcement learning. In *The Eighth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 749–756.

Krieger, H. 2003. *SDL*—a description language for building NLP systems. In *Proceedings of the HLT-NAACL Workshop on the Software Engineering and Architecture of Language Technology Systems, SEALTS*, 84–91.

Li, L.; Littman, M.; and Walsh, T. 2008. Knows what it knows: A framework for self-aware learning. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML-08)*.

Morisset, B., and Ghallab, M. 2008. Learning how to combine sensory-motor functions into a robust behavior. *Artificial Intelligence* 172(4-5):392–412.

Strehl, A.; Diuk, C.; and Littman, M. 2007. Efficient structure learning in factored-state MDPs. In *Proceedings of the 22nd national conference on Artificial intelligence (AAAI'07)*, 645–650.

Sutton, R., and Barto, A. 1998. *Reinforcement Learning: An Introduction*. Adaptive Computation and Machine Learning. The MIT Press.

Tesauro, G.; Jong, N.; Das, R.; and Bennani, M. 2007. On the use of hybrid reinforcement learning for automatic resource allocation. *Cluster Computing* 10:287–299.

Walsh, T.; Nouri, A.; Li, L.; and Littman, M. 2009. Learning and planning in environments with delayed feedback. *Journal of Autonomous Agents and Multi-Agent Systems* 18:83–101.

Anchor-Progression in Spatially Situated Discourse: a Production Experiment

Hendrik Zender and **Christopher Koppermann** and **Fai Greeve** and **Geert-Jan M. Kruijff**
Language Technology Lab
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
zender@dfki.de

Abstract

The paper presents two models for producing and understanding situationally appropriate referring expressions (REs) during a discourse about large-scale space. The models are evaluated against an empirical production experiment.

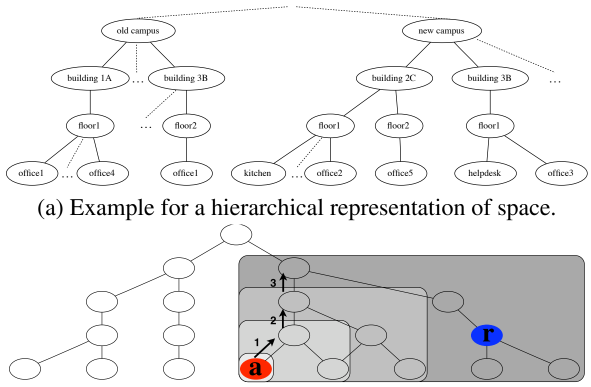
1 Introduction and Background

For situated interaction, an intelligent system needs methods for relating entities in the world, its representation of the world, and the natural language references exchanged with its user. Human natural language processing and algorithmic approaches alike have been extensively studied for application domains restricted to small visual scenes and other small-scale surroundings. Still, rather little research has addressed the specific issues involved in establishing reference to entities outside the currently visible scene. The challenge that we address here is how the focus of attention can shift over the course of a discourse if the domain is larger than the currently visible scene.

The generation of referring expressions (GRE) has been viewed as an isolated problem, focussing on efficient algorithms for determining which information from the domain must be incorporated in a noun phrase (NP) such that this NP allows the hearer to optimally understand which referent is meant. 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 GRE. Recent extensions address some of its shortcomings, such as negated and disjoined properties (van Deemter, 2002) and an account of salience for generating contextually appropriate shorter REs (Krahmer and Theune, 2002). Other, alternative GRE algorithms exist (Horacek, 1997; Bateman, 1999; Krahmer et al., 2003). However, all these algorithms rely on a given *domain of discourse* constituting the current *context* (or *focus of attention*). The task of the GRE algorithm is then to single out the intended referent against the other members of the context, which act as *potential distractors*. As long as the domains are such closed-context scenarios, the intended referent is always in the current focus. We address the challenge of producing and understanding of references to entities that are outside the current focus of attention, because they have not been mentioned yet and are beyond the currently observable scene.

Our approach relies on the dichotomy between *small-scale space* and *large-scale space* for human spatial cognition. Large-scale space is “a space which cannot be perceived at once; its global structure must be derived from local observations over time” (Kuipers, 1977). In everyday situations, an office environment, one’s house, or a university campus are large-scale spaces. A table-top or a part of an office are examples of small-scale space. 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.

It is generally assumed that humans adopt a *partially hierarchical* representation of spatial organization (Stevens and Coupe, 1978; McNamara, 1986). The basic units of such a representation are *topological* regions (i.e., more or less clearly bounded spatial areas) (Hirtle and Jonides, 1985). Paraboni et al. (2007) are among the few to address the issue of generating references to entities outside the immediate environment, and present an algorithm for *context determination* in hierar-



(a) Example for a hierarchical representation of space.

(b) 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.

Figure 1: TA in a spatial hierarchy.

chically ordered domains. However, since it is mainly targeted at producing textual references to entities in written documents (e.g., figures and tables in book chapters), they do not address the challenges of physical and perceptual situatedness. Large-scale space can be viewed as a hierarchically ordered domain. To keep track of the referential context in such a domain, in our previous work we propose the principle of *topological abstraction* (TA, summarized in Fig. 1) for context extension (Zender et al., 2009a), similar to Ancestral Search (Paraboni et al., 2007). In (Zender et al., 2009b), we describe the integration of the approach in an NLP system for situated human-robot dialogues and present two algorithms instantiating the TA principle for GRE and resolving referring expressions (RRE), respectively. It relies on two parameters: the location of the *intended referent* r, and the *attentional anchor* a. As discussed in our previous works, for single utterances the anchor is the physical position where it is made (i.e., the *utterance situation* (Devlin, 2006)). Below, we propose models for attentional anchor-progression for longer discourses about large-scale space, and evaluate them against real-world data.

2 The Models

In order to account for the determination of the attentional anchor a, we propose a model called *anchor-progression* A. The model assumes that each *exophoric* reference[1] serves as *attentional anchor* for the subsequent reference. It is based on observations on "principles for anchoring resource situations" by Poesio (1993), where the expression of movement in the domain determines the updated current mutual focus of attention. a and r are then passed to the TA algorithm. Taking into account the verbal behavior observed in our experiment, we also propose a refined model of *anchor-resetting* R, where for each new turn (e.g., a new instruction), the anchor is re-set to the *utterance situation*. R leads to the inclusion of navigational information for each first RE in a turn, thus reassuring the hearer of the focus of attention.

3 The Experiment

We are interested in the way the disambiguation strategies change when producing REs during a discourse about large-scale space versus discourse about small-scale space. In our experiment, we gathered a corpus of spoken instructions in two different situations: *small-scale space (SSS)* and *large-scale space (LSS)*. We use the data to evaluate the utility of the A and R models. We specifically evaluate them against the traditional (*global*) model G in which the indented referent must be singled out from all entities in the domain.

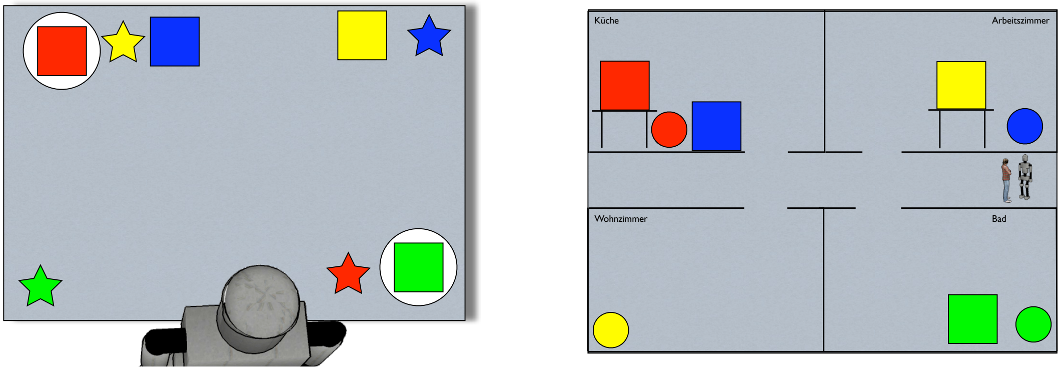
The cover story for the experiment was to record spoken instructions to help improve a speech recognition system for robots. The participants were asked to imagine an intelligent service robot capable of understanding natural language and familiar with its environment. The task of the participants was to instruct the robot to clean up a working space, i.e., a table-top (SSS) and an indoor environment (LSS) by placing target objects (cookies or balls) in boxes of the same color. The use of color terms to identify objects was discouraged by telling the participants that the robot is unable to perceive color. The stimuli consisted of 8 corresponding scenes of the table-top and the domestic setting (cf. Fig. 2). In order to preclude the specific phenomena of collaborative, task-oriented dialogue (cf., e.g., (Garrod and Pickering, 2004)), the participants had to instruct an imaginary recipient of orders. The choice of a robot was made to rule out potential social implications when imagining, e.g., talking to a child, a butler, or a friend.

The SSS scenes show a bird's-eye view of the table including the robot's position (similar to (Funakoshi et al., 2004)). The way the objects are arranged allows to refer to their location with respect to the corners of the table, with plates as additional landmarks. The LSS scenes depict an indoor environment with a corridor and, parallel to SSS, four rooms with tables as landmarks. The scenes show

[1] This excludes pronouns as well as other descriptions that pick up an existing referent from the linguistic context.

Table 1: Example from the small-scale (1–2) and large-scale space (3–4) scenes in Fig. 2.

1. *nimm [das plätzchen unten links]*$_{m_{G,A}}$ *, leg es [in die schachtel unten rechts auf dem teller]*$_{o_{G,A}}$
   ‘take the cookie on the bottom left, put it into the bottom right box on the plate’
2. *nimm [das plätzchen unten rechts]*$_{m_G,o_A}$ *, leg es [in die schachtel oben links auf dem teller]*$_{m_{G,A}}$
   ‘take the cookie on the bottom right, put it into the top left box on the plate’
3. *geh [ins wohnzimmer]*$_{m_{G,A,R}}$ *und nimm [den ball]*$_{u_G,m_{A,R}}$ *und bring ihn [ins arbeitszimmer]*$_{m_{G,A,R}}$ *, leg ihn [in die kiste auf dem tisch]*$_{u_G,o_{A,R}}$
   ‘go to the living room and take the ball and bring it to the study; put it into the box on the table’
4. *und nimm [den ball]*$_{u_{G,R},m_A}$ *und bring ihn [in die küche]*$_{m_{G,A,R}}$ *und leg ihn [in die kiste auf dem boden]*$_{u_G,m_{A,R}}$
   ‘and take the ball and bring it to the kitchen and put it into the box on the floor’



(a) Small-scale space: squares represent small boxes, stars cookies, and white circles plates.

(b) Large-scale space: squares represent boxes placed on the floor or on a table, circles represent balls, rooms are labeled.

Figure 2: Two stimuli scenes from the experiment.

the robot and the participant in the corridor.

In order to gather more comparable data we opted for a *within-participants* approach. Each person participated in the *SSS treatment* and in the *LSS treatment*. To counterbalance potential carry-over effects, half of the participants were shown the treatments in inverse order, and the sequence of the 8 scenes in each treatment was varied in a principled way. In order to make the participants produce multi-utterance discourses, they were required to refer to all target object pairs. The exact wording of their instructions was up to them.

Participants were placed in front of a screen and a microphone into which they spoke their orders to the imaginary robot, followed by a self-paced keyword after which the experimenter showed the next scene. The experiment was conducted in German and consisted of a pilot study (10 participants) and the main part (19 female and 14 male students, aged 19–53, German native speakers). The data of three participants who did not behave according to the instructions was discarded. The individual sessions took 20–35 min., and the participants were paid for their efforts.

Using the UAM CorpusTool software, transcriptions of the recorded spoken instructions were annotated for occurrences of the linguistic phenomenon we are interested in, i.e., REs. Samples were cross-checked by a second annotator. REs were marked as shallow ‘refex’ segments, i.e., complex NPs were not decomposed into their constituents. Only definite NPs representing exophoric REs (cf. Sec. 2) qualify as ‘refex’ segments. If a turn contained an indefinite NP, the whole turn was discarded. The ‘refex’ segments were coded according to the amount of information they contain, and under which disambiguation model $M \in \{G, A, R\}$ (R only for LSS) they succeed in singling out the described referent. Following Engelhardt et al. (2006), we distinguish three types of semantic specificity. A RE is an *over-description* with respect to M ($over_M$) if it contains redundant information, and it is an *under-description* ($under_M$) if it is ambiguous according to M. *Minimal descriptions* (min_M) contain just enough information to uniquely identify the referent. Table 1 shows annotated examples.

4 Results

The collected corpus consists of 30 annotated sessions with 2 treatments comprising 8 scenes with 4 turns. In total, it contains 4,589 annotated REs, out of which only 83 are errors. Except for the error rate calculation, we only consider non-error ‘refex’ segments as the universe. The SSS treat-

Table 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.

	$over_G$	$over_A$	$over_R$	min_G	min_A	min_R	$under_G$	$under_A$	$under_R$
small-scale	13.94%	34.45%		78.90%	60.11%		7.16%	5.43%	
space	*15.85%*	*14.37%*		*17.66%*	*13.13%*		*12.07%*	*10.50%*	
large-scale	6.81%	34.75%	20.06 %	68.04%	64.55%	76.73%	25.16%	0.69%	3.21%
space	*7.53%*	*12.13%*	*10.10%*	*17.87%*	*13.13%*	*10.66%*	*19.48%*	*1.72%*	*5.06%*

ment contains 1,902 'refex', with a mean number of 63.4 and a std. dev. σ=1.98 per participant. This corresponds to the expected number of 64 REs to be uttered: 8 scenes × 4 target object pairs. The LSS treatment contains 2,604 'refex' with an average of 86.8 correct REs (σ=18.19) per participant. As can be seen in Table 1 (3–4), this difference is due to the participants' referring to intermediate waypoints in addition to the target objects. Table 2 summarizes the analysis of the annotated data.

Overall, the participants had no difficulties with the experiment. The mean error rates are low in both treatments: 1.78% (σ=3.36%) in SSS, and 1.80% (σ=2.98%) in LSS. 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 (p=0.985), supporting the claim that both treatments were of equal difficulty. Moreover, a MANOVA shows no significant effect of treatment-order for the verbal behavior under study, ruling out potential carry-over effects.

Production experiments always exhibit a considerable variation between participants. When modeling natural language processing systems, one needs to take this into account. A GRE component should produce REs that are easy to understand, i.e., ambiguities should be avoided and over-descriptions should occur sparingly. A GRE algorithm will always try to produce minimal descriptions. The generation of an under-description means a failure to construct an identifying RE, while over-descriptions are usually the result of a globally 'bad' incremental construction of the generated REs (as is the case, e.g., in the IA). An RRE component, on the other hand, should be able to identify as many referents as possible by treating as few as possible REs as under-descriptions.

The analysis of the SSS data with respect to G establishes the baseline for a comparison with other experiments and GRE approaches. 13.9% of the REs contain redundant information ($over_G$), compared to 21% in (Viethen and Dale, 2006). In contrast, however, our SSS scenes did not provide the possibility for producing more-than-minimal REs for every target object, which might account for the difference. $under_G$ REs occur with a frequency of 7.2% in the SSS data. Because under-descriptions result in the the hearer being unable to reliably resolve the reference, this means that the robot in our experiment cannot fulfill its task. This might explain the difference to the 16% observed in the task-independent study by Viethen and Dale (2006). The significantly (p<0.001) higher mean frequency of min_G than min_A underpins that G is an accurate model for the verbal behavior in SSS. However, G does not fit the LSS data well. An RRE algorithm with model G would fail to resolve the intended referent in 1 out of 4 cases (cf. $under_G$ in LSS). With only 0.7% $under_A$ REs on average, A models the LSS data significantly better (p<0.001). Still, there is is a high rate of $over_A$ REs. In comparison, R yields a significantly (p<0.001) lower amount of $over_R$. The mean frequency of $under_R$ is significantly (p=0.010) higher than for $under_A$, but still below $under_G$ in the SSS data. With a mean frequency of 76.7% min_R, R models the data better than both G and A. For the REs in LSS min_R is in the same range as min_G for the REs in SSS.

5 Conclusions

Overall, the data exhibit a high mean frequency of over-descriptions. However, since this means that the human-produced REs 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. In situated discourse about LSS, we thus suggest that A is suitable for the RRE task because it yields the least amount of unresolvable under-descriptions. For the GRE task R is more appropriate. It strikes a balance between producing short descriptions and supplementing navigational information.

Acknowledgments

This work was supported by the EU Project CogX (FP7-ICT-215181). Thanks to Mick O'Donnell for his support with the UAM CorpusTool.

References

John A. Bateman. 1999. 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.

Robert Dale and Ehud Reiter. 1995. Computational interpretations of the Gricean Maxims in the generation of referring expressions. *Cognitive Science*, 19(2):233–263.

Keith Devlin. 2006. 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.

Paul E. Engelhardt, Karl G.D. Bailey, and Fernanda Ferreira. 2006. Do speakers and listeners observe the Gricean Maxim of Quantity? *Journal of Memory and Language*, 54(4):554–573.

Kotaro Funakoshi, Satoru Watanabe, Naoko Kuriyama, and Takenobu Tokunaga. 2004. Generation of relative referring expressions based on perceptual grouping. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, Morristown, NJ, USA.

Simon Garrod and Martin J. Pickering. 2004. Why is conversation so easy? *Trends in Cognitive Sciences*, 8(1):8–11, January.

Stephen C. Hirtle and John Jonides. 1985. Evidence for hierarchies in cognitive maps. *Memory and Cognition*, 13:208–217.

Helmut Horacek. 1997. 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.

Emiel Krahmer and Mariët Theune. 2002. Efficient context-sensitive generation of referring expressions. In Kees van Deemter and R. Kibble, editors, *Information Sharing: Givenness and Newness in Language Processing*, pages 223–264. CSLI Publications, Stanford, CA, USA.

Emiel Krahmer, Sebastiaan van Erk, and André Verleg. 2003. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1):53–72.

Benjamin Kuipers. 1977. *Representing Knowledge of Large-scale Space*. PhD thesis, MIT-AI TR-418, Massachusetts Institute of Technology, Cambridge, MA, USA, May.

Timothy P. McNamara. 1986. Mental representations of spatial relations. *Cognitive Psychology*, 18:87–121.

Ivandré Paraboni, Kees van Deemter, and Judith Masthoff. 2007. Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, 33(2):229–254, June.

Massimo Poesio. 1993. 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.

Albert Stevens and Patty Coupe. 1978. Distortions in judged spatial relations. *Cognitive Psychology*, 10:422–437.

Kees van Deemter. 2002. Generating referring expressions: boolean extensions of the incremental algorithm. *Computational Linguistics*, 28(1):37–52.

Jette Viethen and Robert Dale. 2006. 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.

Hendrik Zender, Geert-Jan M. Kruijff, and Ivana 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.

Hendrik Zender, Geert-Jan M. Kruijff, and Ivana 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.

Anchor-progression in situated discourse about large-scale space

Hendrik Zender and Christopher Koppermann and Fai Greeve and Geert-Jan M. Kruijff
Language Technology Lab
German Research Center for Artificial Intelligence (DFKI)
Saarbrücken, Germany
zender@dfki.de

The use of natural language processing systems is no longer limited to small, fixed, fully known and fully observable domains. In interaction with mobile robots, with non-player characters in virtual worlds, or with mobile location-based applications alike references to entities outside the currently observable scene (i.e., in *large-scale space*) are becoming more and more important. *Referring expressions* (e.g., definite noun phrases, pronouns, and proper names) are used to convey which entities in the world are being talked about. Ideally, the natural language communication with such systems is not restricted to single one-way utterances. The way successful reference between such a system and its user is established must thus be viewed from a discourse-oriented perspective. Successful reference is established by the interplay of referring expressions and the way the discourse unfolds.

In this paper we address the challenge of producing and understanding referring expressions to entities in large-scale space during a discourse. To this end, we propose a general principle of *topological abstraction* (TA) for determining an appropriate spatial context. This principle is applied to the tasks of generating and resolving referring expressions. Further, we propose *anchor-progression* and *anchor-resetting* mechanisms to track the origin of the TA algorithms throughout the discourse. Finally, we present an empirical 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.

Introduction

For situated interaction with intelligent systems we need methods for establishing the relationship between entities in the world and the representations that the system has of its environment. Human natural language processing and algorithmic approaches alike have been extensively studied for application domains that are restricted to small visual scenes and other small-scale surroundings.

With autonomous mobile robots slowly finding their way into our everyday lives, non-player characters in 3D virtual worlds evolving from game opponents to social companions, and a rapidly growing market for mobile location-based and context-aware technology, we are faced with the challenge of situated communication about large-scale environments. That is, robotic assistants must understand instructions to fetch coffee "from the kitchen", and 3D avatars might want to emphasize the fact that a particular couch would fit perfectly well with "the armchair in your living room". Still, rather little research has addressed the specific difficulties involved in establishing reference to entities outside the currently visible scene. The challenge that we will 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.

In this paper, we identify *attention-direction* and *context determination* as crucial steps towards the generation and resolution of references to entities in *large-scale space*. We propose a general principle of *topological abstraction* for determining an appropriate spatial context. This principle is applied to the tasks of generating and resolving referring expressions. Then we move on from individual referring expressions to the mechanisms that determine the origin of the topolog-

This work has been supported by the European Community 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."

ical abstraction algorithms along the course of a discourse. We propose the principle of *anchor-progression*, which models the way attention-directing information unfolds during the course of a discourse. Finally, we present an empirical 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.

Background

Referring Expressions

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. Conversely, in natural language comprehension, resolving referring expressions (RRE) is concerned with identifying which domain entity is referred to by the speaker.

Usually, GRE has been viewed as an isolated problem, focussing on efficient algorithms for determining which information from the domain must be incorporated in a noun phrase such that this noun phrase 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. 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 referring expressions (Krahmer & Theune, 2002). Other, alternative GRE algorithms exist (Horacek, 1997; Bateman, 1999; Krahmer, van Erk, & Verleg, 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 the other members of the context set that act as *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 of 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.

Paraboni, van Deemter, and Masthoff (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 challenges of physically and perceptually situated dialogue.

All the different components of natural language dialogue systems contribute to the overall success of reference. Besides the already mentioned GRE algorithms this not only involves the other linguistic processes, such as discourse planning, sentence aggregation, lexical choice, and surface realization, but also knowledge base construction and maintenance, and the interface between the knowledge base and the NLP system. Moreover, dialogue systems need to perform bi-directional communication. In addition to natural language generation, a dialogue system must be capable of natural language understanding. Its counterpart for the GRE task is the task of resolving referring expressions to entities in the system's knowledge base (RRE).

In a nutshell, establishing reference is in general not solvable by an isolated GRE or RRE algorithm. Reference is established by conveying or processing the right information at the right point during the course of a discourse. For NLG it is not sufficient to determine which information needs to be included, but also when and where it should be realized. A natural language understanding system must be able to take into account the influence previous utterances have on the reference at hand.

With respect to this, we are addressing *attention-directing* factors that arise during a discourse, moving beyond the single, isolated referring expression. Together, our approach accounts for the progression of the focus of attention during a discourse about entities that are located in a domain that is larger than the immediate visual context.

Existing Corpora

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 & Dale, 2008b, 2008a), and the Drawer data set (Viethen & 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 & Fosler-Lussier, 2006) and the SCARE corpus (Stoia, Shockley, Byron, & Fosler-Lussier, 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 & Pickering, 2004; Pickering & Garrod, 2006) and *conceptual pacts* (Brennan & Clark, 1996), which among other things, include the use of "risky references" (Carletta & 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 & Wilkes-Gibbs, 1986). For several reasons these phenomena are very prominent in the aforementioned corpora. For one, the individual conversations recorded 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*.

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. Most referring expressions (that is, definite exophoric noun phrases) in such scenarios will be generated *in-situ*, treating the local visual scene as a small-scale spatial context. Embodied motion within the domain, visual salience, and short-term memory effects determine which objects qualify as referents and distractors.

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.

Cognitive Models of Large-Scale Space

The work presented here relies on the dichotomy between *small-scale space* and *large-scale space* for human spatial cognition (Herman & Siegel, 1978; Hazen, Lockman, & Pick, 1978). 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.

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 map*. 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.

Following the results of empirical studies, it is nowadays generally assumed that humans adopt a *partially hierarchical* representation of spatial organization (Stevens & Coupe, 1978; McNamara, 1986). The basic units of such a qualitative spatial representation are *topological* regions corresponding to more or less clearly bounded spatial areas (Hirtle & Jonides, 1985).

Context Determination in Hierarchically Ordered Domains

Large-scale space can be viewed as a hierarchically ordered domain. To keep track of the correct referential context in such a domain, we propose a general principle of *topological abstraction* (TA) for context extension, which is rooted in what we will call the *referential anchor*.

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. Another drawback of the approach is the omission of formalizing the notion of the referential anchor, and its progression during the course of a discourse.

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

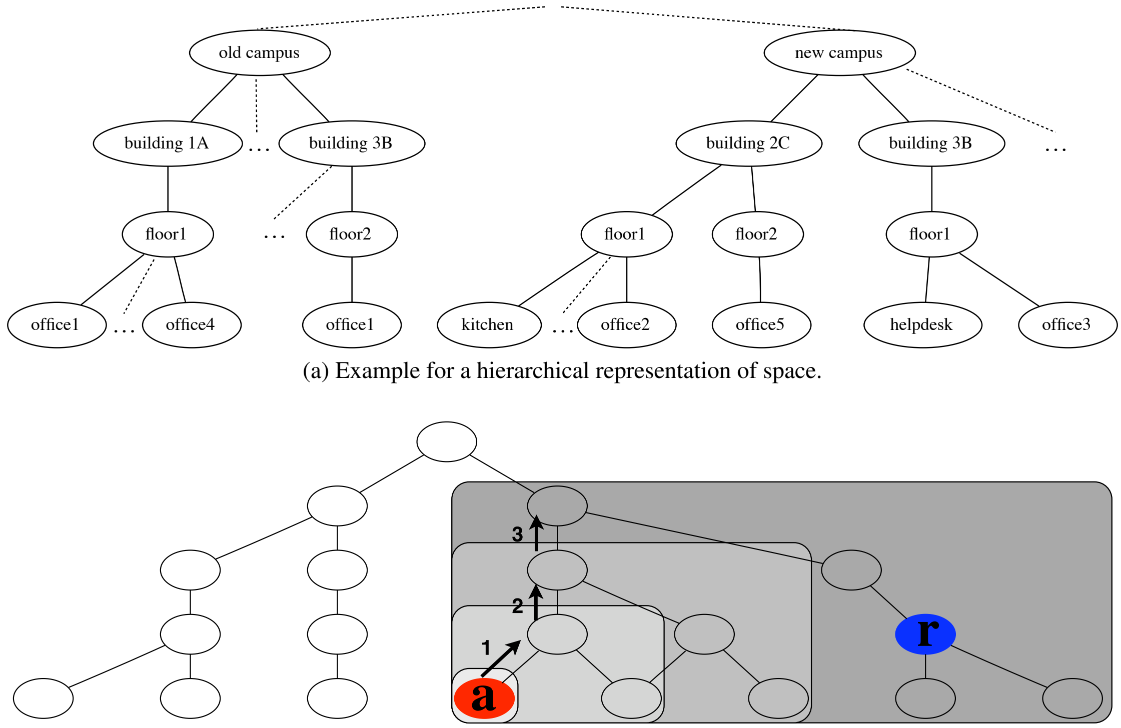
Topological Abstraction

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 1a). The referential 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. But during a longer discourse, a can move along the "spatial progression" of the most salient discourse entity. 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 subhierarchy (cf. Figure 1b).

Below we describe two instantiations of the TA principle, a TA algorithm for reference generation (TAA1) and TAA2 for reference resolution. They differ only minimally, namely in their use of an intended referent r or an RE $desc(x)$ to determine the conditions for entering and exiting the loop for topological abstraction. The way they determine a context through topological abstraction is identical.

Context Determination for GRE. TAA1 (cf. Algorithm 1 on page 6) constructs a set of entities dominated by the referential 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.

TAA1 is formulated to be neutral to the kind of GRE algorithm that it is used for. It can be used with the original Incremental Algorithm (Dale & 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 & Haddock, 1991), but we agree with Paraboni et al. (2007) that the mutually qualified



(a) Example for a hierarchical representation of space.

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

Figure 1. : Topological abstraction (TA) in a spatial hierarchy.

references that it can produce[1] are not easily resolvable if they pertain to circumstances where a confirmatory search is 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 & van Deemter, 2007). TAA1 is compatible with these approaches, as well as with the salience based approach of Krahmer and Theune (2002), which also provides a recursive variant that is able to handle relations.

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 overgenerating possible referents, we propose TAA2 (cf. Algorithm 2 on the next page) which tries to select an appropriate referent from a relevant subset of the full knowledge base.

It is initialized with a given semantic representation of the referential expression, $desc(x)$, in a format compatible with the knowledge base. 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.

A Model for Referential Anchor-progression in Discourse about Large-Scale Space

In order to account for the determination of the referential anchor, we propose a model that we call *anchor-progression*. The model assumes

[1] 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.

Algorithm 1 TAA1 (for reference generation)

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

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

Algorithm 2 TAA2 (for reference resolution)

```
Input: referential 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: $R := \emptyset$
  $C := C \cup \{a\} \cup topologicalDescendants(a)$
  $R := desc(x) \cap C$
  if $R \neq \emptyset$ then
    return $R$
  else
    Initialize abstraction queue: $Q := [a]$
    while $size(Q) > 0$ do
      $n := pop(Q)$
      for each $p \in topologicalParents(n)$ do
        $push(Q, p)$
        $C := C \cup \{p\} \cup topologicalDescendants(p)$
      end for
      $R := desc(x) \cap C$
      if $R \neq \emptyset$ then
        return $R$
      end if
    end while
    return failure
  end if
```

that each reference to an extra-linguistic entity in large-scale space serves as *referential anchor* for the subsequent reference. Formally speaking, each *exophoric* referring expression will set 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 referential anchor and the intended referent are then passed to the respective TA algorithms. Taking into account the verbal bevhavior observed in the experiment, cf. Section , we also propose a refined model of *anchor-resetting*. In this model, for each new turn (e.g., a new instruction), the referential anchor is re-set 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.

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 indented referent must be singled out against all entities in the domain.

Design Considerations

As discussed in the introductory sections, 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 that influence the participants' behavior are hard, if not nearly impossible, to control. That is why, usually, such experiments are conducted in specificly instrumented dedicated environments in order to be able to record the participants as unobtrusively as possible. Most participants will thus be unfamiliar with the experimental environment. Memory effects as well as different spatial reasoning capabilities in the participants will probably overshadow the observed verbal behavior. Embodiment in a virtual 3D world has a similar disadvantage, because the participants' mental map of the environment will be very brittle.

A common practice for the study of 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, Crocker, Pickering, & Scheepers, 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, Watanabe, Kuriyama, and Tokunaga (2004), Kelleher and van Genabith (2006), 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 hence 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.

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 2 on the following page. 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.

In order to preclude the aforementioned specific phenomena of collaborative, task-oriented dialogue, the participants had to utter instructions to an imaginary recipient of orders. The stimuli hence provide the participants 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.

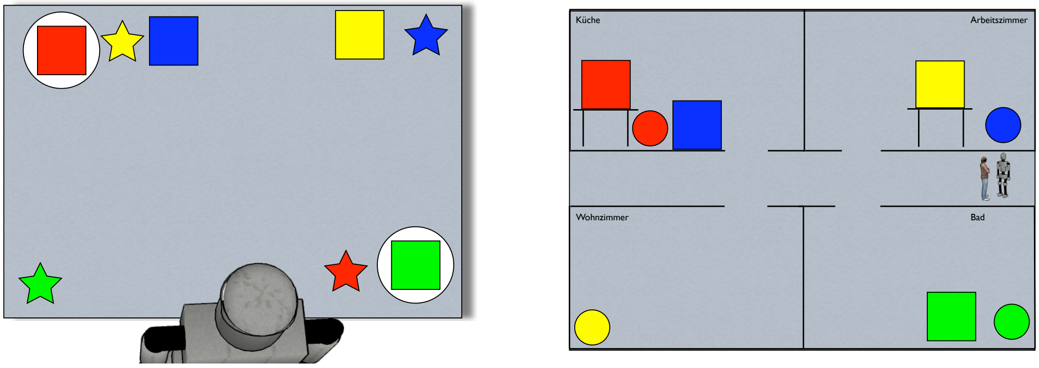
The small-scale setting shows a similar perspective on the scene as the experiment done by Funakoshi et al. (2004), i.e., a bird's-eye view of the table-top including an illustration of the robot's position with respect to the table. The target, landmark and distractor objects consist of cookies, small 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.

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 smoothen 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 equal 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



(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 labelled *Küche* 'kitchen', *Arbeitszimmer* 'study', *Bad* 'bathroom', *Wohnzimmer* 'living room'.

Figure 2. : Two of the scenes shown in the experiment.

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 (small-scale space) and an indoor environment (large-scale space), 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 the exclusion of visual salience as a factor.

The experiment was conducted in German.

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 for the second treatment and three corresponding practice scenes were shown, again allowing them to ask for clarification before starting with the eight 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). 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 differently colored 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, and the participants were paid for their efforts.

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]

The linguistic phenomenon we are interested in, i.e., referring expressions, was then manually annotated. Each session was annotated by one annotator. Samples of the annotations were cross-checked by a second or third annotator.

The annotation part consisted of several tasks. First of all, referring expressions were marked as 'refex' segments. Only definite noun phrases qualify as 'refex' segments. If a turn contained an indefinite noun phrase to introduce a new referent, the whole turn was discarded. 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 noun phrases 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 in 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 specifity with respect to each model according to the terms introduced by Engelhardt, Bailey, and Ferreira (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 identify the referent. If the participants made an error with respect to the instructions of the experiment, the respective 'refex' was coded as error.

Table 1 on the next page and Table 2 on page 11 show annotated examples taken from the data.

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. 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 × 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 Table 1 (3–4), this difference results from the participants' referring to intermediate waypoints that introduce new spatial contexts in addition to the target objects.

Overall, the participants had no difficulties completing the two treatments of the experiment. The error rates are low in both treatments: 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

[2] `http://www.wagsoft.com/CorpusTool/` Thanks to Mick O'Donnell for his support.

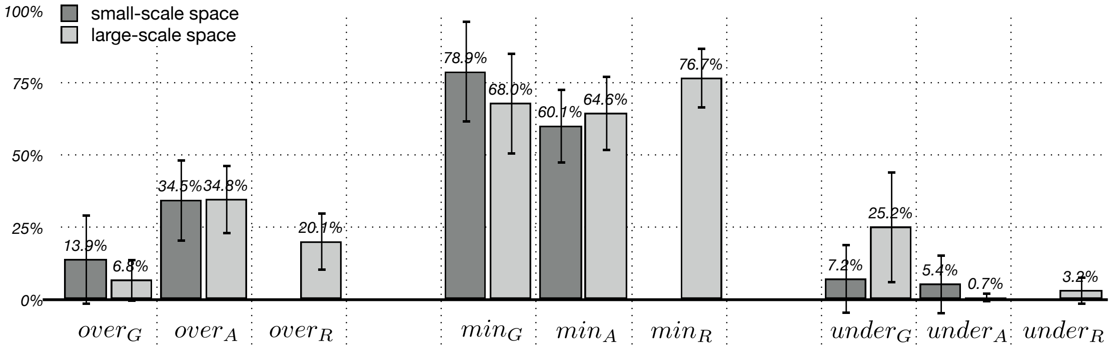


Figure 3. : Mean frequencies of overs-descriptions (*over*), minimal descriptions (*min*), and under-descriptions (*under*) with respect to the two models (anchor-progression, *A*, and global, *G*) in both treatments (large-scale space and small-scale space).

Table 1
: Example from the small-scale space scene in Figure 2a.

1. *nimm [das plätzchen unten links]$_{min_{G,A}}$, leg es [in die schachtel unten rechts auf dem teller]$_{over_{G,A}}$*
   take the cookie bottom left , put it into the box bottom right on the plate

   ‘take the cookie on the bottom left, put it into the box on the bottom right’

2. *nimm [das plätzchen unten rechts]$_{min_G,over_A}$, leg es [in die schachtel oben links auf dem teller]$_{min_{G,A}}$*
   take the cookie bottom right , put it into the box top 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 rechts]$_{min_{G,A}}$*
   take the cookie top left , put it into-the box top right

   ‘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 links]$_{under_{G,A}}$*
   take the cookie top right , put it into the box top left

   ‘take the top right cookie, put it into the top left box’

is no significant effect of treatment-order for the verbal behavior under study. This rules out potential carry-over effects.

Figure 3 shows the mean frequencies of over-descriptive, minimally descriptive, and under-descriptive referring expressions with respect to the models in both treatments.

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 on 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=15.8%) 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 am-

Table 2
: Example from the large-scale space scene in Figure 2b.

1. *geh [ins wohnzimmer]*$_{min_{G,A,R}}$ *und nimm [den ball]*$_{under_G,min_{A,R}}$ *und bring ihn [ins arbeitszimmer]*$_{min_{G,A,R}}$,

   go into-the living-room and take the ball and bring it into-the work-room ,

   leg ihn [in die kiste auf dem tisch]$_{under_G,over_{A,R}}$

   put it into the box on the 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 leg ihn [in die kiste auf dem boden]*$_{under_G,min_{A,R}}$

   and take the ball and bring it into the kitchen and 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 kiste auf dem tisch]*$_{under_G,min_{A,R}}$

   and then take you the ball in the kitchen and put it into the 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 liegt]*$_{min_G,over_{A,R}}$ *und legst ihn [in die kiste die dort steht]*$_{min_G,over_{A,R}}$

   and then go you into-the bathroom and take the ball that there 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’

biguous. 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=0.000) higher mean frequency of *min*$_G$ (80.8%, σ=18.6%) than *min*$_A$ (62.1%, σ=13.6%), 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=0.000) 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*$_G$ ‘refex’ 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 significantly better (t=6.776, df=29, p=0.000). 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=0.000) 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.

Discussion

In total, the data exhibit a high mean frequency of over-descriptions. 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.

However, since this means that the human-produced referring expressions 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, the re-mention of 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 underdescriptions, whereas for the GRE task, the *anchor-resetting* model is more appropriate. It strikes a balance between producing short descriptions and supplementing 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. Whereas in dialogue people sometimes make “risky” utterances, in the specific setting of one-way instruction-giving potential underdescriptions 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.

Conclusions

We presented an approach to the problem of generating and resolving referring expressions to entities in large-scale space. The challenges we 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 during the course of a discourse. We proposed the principle of topological abstraction with two specific instantiations in algorithms for 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 also reported on a production experiment for evaluating the proposed models. The evaluation shows that traditional 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.

References

Anderson, A. H., Bader, M., Bard, E. G., Boyle, E., Doherty, G. M., Garrod, S., et al. (1991). The HCRC Map Task corpus. *Language and Speech*, *34*, 351-366.

Bateman, J. A. (1999). 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)* (pp. 127–134). Morristown, NJ, USA: Association for Computational Linguistics.

Brennan, S. E., & Clark, H. H. (1996). Conceptual pacts and lexical choice in conversation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *22*(6), 1482–1493.

Byron, D., & Fosler-Lussier, E. (2006). 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).*

Byron, D., Koller, A., Striegnitz, K., Cassell, J., Dale, R., Moore, J., et al. (2009, March). 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: Association for Computational Linguistics.

Carletta, J., & Mellish, C. S. (1996). Risk-taking and recovery in task-oriented dialogue. *Journal of Pragmatics*, *26*(1), 71–107. Available from `http://www.sciencedirect.com/science/article/B6VCW-3VW8PNN-4/2/4478e0588dfac30c0bbfe4ff2a30733a`

Chomsky, N. (1957). *Syntactic structures.* The Hague / Paris: Mouton.

Clark, H. H., & Wilkes-Gibbs, D. (1986). Referring as a collaborative process. *Cognition*, *22*, 1–39.

Cooper, R. M. (1974). 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. Available from http://www.sciencedirect.com/science/article/B6WCR-4D6RJS3-5F/2/b0ed3950ccc4d75fe2af0ac66505ea7f

Croitoru, M., & van Deemter, K. (2007, January). 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.

Dale, R., & Haddock, N. (1991, April). Generating referring expressions involving relations. In *Proceedings of the fifth meeting of the european chapter of the association for computational linguistics.* Berlin, Germany.

Dale, R., & Reiter, E. (1995). Computational interpretations of the Gricean Maxims in the generation of referring expressions. *Cognitive Science*, *19*(2), 233-263. Available from citeseer.ist.psu.edu/dale94computational.html

Engelhardt, P. E., Bailey, K. G., & Ferreira, F. (2006). Do speakers and listeners observe the Gricean Maxim of Quantity? *Journal of Memory and Language*, *54*(4), 554–573. Available from http://www.sciencedirect.com/science/article/B6WK4-4J9X2WM-1/2/2031e4e6f7de36104e8678072246e13d

Funakoshi, K., Watanabe, S., Kuriyama, N., & Tokunaga, T. (2004). Generation of relative referring expressions based on perceptual grouping. In *COLING '04: Proceedings of the 20th international conference on computational linguistics.* Morristown, NJ, USA: Association for Computational Linguistics.

Garrod, S., & Pickering, M. J. (2004, January). Why is conversation so easy? *Trends in Cognitive Sciences*, *8*(1), 8–11. Available from http://www.sciencedirect.com/science/article/B6VH9-4B0PDXJ-1/2/94094ad5bea3448a9352fd05694d3b98

Hazen, N. L., Lockman, J. J., & Pick, H. L., Jr. (1978, September). The development of children's representations of large-scale environments. *Child Development*, *49*(3), 623–636.

Herman, J. F., & Siegel, A. W. (1978). The development of cognitive mapping of the large-scale environment. *Journal of Experimental Child Psychology*, *26*, 389–406.

Hirtle, S. C., & Jonides, J. (1985). Evidence for hierarchies in cognitive maps. *Memory and Cognition*, *13*, 208–217.

Hois, J., & Kutz, O. (2008). Natural language meets spatial calculi. In C. Freksa, N. S. Newcombe, P. Gärdenfors, & S. Wölfl (Eds.), *Learning, reasoning, and talking about space (SC'08)* (Vol. VI, p. 266-282). Berlin/Heidelberg, Germany: Springer Verlag.

Horacek, H. (1997). 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)* (pp. 206–213). Morristown, NJ, USA: Association for Computational Linguistics.

Kelleher, J. D. (2007). Attention driven reference resolution in multimodal contexts. *Artificial Intelligence Review*, *25*, 21–35.

Kelleher, J. D., & van Genabith, J. (2006). A computational model of the referential semantics of projective prepositions. In P. Saint-Dizier (Ed.), *Syntax and semantics of prepositions.* Kluwer Academic Publishers.

Knoeferle, P., Crocker, M., Pickering, M., & Scheepers, C. (2005). The influence of the immediate visual context on incremental thematic role-assignment: evidence from eye-movements in depicted events. *Cognition*, *95*(1), 95–127.

Koller, A., Moore, J., Di Eugenio, B., Lester, J., Stoia, L., Byron, D., et al. (2007). *Shared task proposal: Instruction giving in virtual worlds.* Working group report, Workshop on Shared Tasks and Comparative Evaluation in Natural Language Generation.

Krahmer, E., & Theune, M. (2002). Efficient context-sensitive generation of referring expressions. In K. van Deemter & R. Kibble (Eds.), *Information sharing: Givenness and newness in language processing* (pp. 223–264). Stanford, CA, USA: CSLI Publications.

Krahmer, E., van Erk, S., & Verleg, A. (2003). Graph-based generation of referring expressions. *Computational Linguistics*, *29*(1), 53–72.

Kuipers, B. (1977). *Representing knowledge of large-scale space.* PhD thesis, MIT-AI TR-418, Massachusetts Institute of Technology, Cambridge, MA, USA.

McNamara, T. P. (1986). Mental representations of spatial relations. *Cognitive Psychology*, *18*, 87–121.

Paraboni, I., van Deemter, K., & Masthoff, J. (2007, June). Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, *33*(2), 229–254.

Pickering, M. J., & Garrod, S. (2006, October). Alignment as the basis for successful communication. *Research on Language and Computation*, *4*(2–3), 203–228.

Stevens, A., & Coupe, P. (1978). Distortions in judged spatial relations. *Cognitive Psychology*, *10*, 422–437.

Stoia, L., Shockley, D. M., Byron, D., & Fosler-Lussier, E. (2008, May). SCARE: a situated corpus with annotated referring expressions. In *Proceedings of the sixth international language resources and evaluation (LREC'08).* Marrakech, Morocco: European Language Resources Association (ELRA). (http://www.lrec-conf.org/proceedings/lrec2008/)

van Deemter, K. (2002). Generating referring expressions: boolean extensions of the incremental algorithm. *Computational Linguistics*, *28*(1), 37–52.

Viethen, J., & Dale, R. (2006). Algorithms for generating referring expressions: Do they do what people do? In *Proceedings of the 4th international natural language generation conference (INLG 2006)* (pp. 63–70). Sydney, Australia.

Viethen, J., & Dale, R. (2008a, December). Generating relational references: What makes a difference? In *Proceedings of the australasian language technology association workshop 2008.* Hobart, Australia.

Viethen, J., & Dale, R. (2008b, June). The use of spatial relations in referring expressions. In *Proceedings of the 5th international natural language generation conference (INLG 08).* Salt Fork, OH, USA.

Contextually Appropriate Intonation of Clarification Requests in Human-Robot Interaction[*]

Ivana Kruijff-Korbayová
German Research Center for
Artificial Intelligence
DFKI GmbH
Saarbrücken, Germany
ivana.kruijff@dfki.de

Raveesh Meena
Department of Computational
Linguistics
Saarland University
Saarbrücken, Germany
rmeena@coli.uni-sb.de

Pirita Pyykkönen
Department of Computational
Linguistics
Saarland University
Saarbrücken, Germany
pirita@coli.uni-sb.de

ABSTRACT

It is established that assigning intonation to dialogue system output in a way that reflects contrast among entities available in the discourse context can enhance the acceptability of system utterances. Previous research has concentrated on the role of linguistic context in processing; dialogue *situatedness* and hence the role of visual context in determining the accent placement has not been studied. In this paper, we present an experimental study addressing the influence of visual context on the perception of nuclear accent placement in synthesized clarification requests. We predicted that variation in the placement of nuclear accent is perceivable and that visual context affects acceptability. We found that utterances with nuclear accent placement licenced by the visual scene are perceived as appropriate more often then utterances with nuclear accent placement not licenced by the visual scene.

Categories and Subject Descriptors

H.4 [**Human-Robot Interaction**]: we need to verify with the HRI conference webpage about the categories.

General Terms

Experimentation, Situatedness

Keywords

intonation, information structure, visual context, experimental methods, user study/verification

1. INTRODUCTION

Since the pioniering work of Pierrehumbert and Hirschberg it is generally accepted that speakers choose particular *intonation tunes* to convey relationships between their utterance, the currently perceived *beliefs* of a hearer(s), and anticipated contributions of subsequent utterances. These relationships are conveyed compositionally via selection of *pitch accents*, *phrase accents*, and *boundary tones* that make up tunes [5].

It is established that pitch accents mark the individual words with which they are associated as *salient* in the discourse context. The accented item is rendered salient not only phonologically but also from an informational standpoint. That is, the assignment of nuclear accent reflects contrast between the intended referent and contextually available alternative(s) [5, 8].

Although the discussion on contrast and placement of nuclear accent in the literature usually concerns discourse context established linguistically (i.e., by preceding utterances), it is generally assumed that the relevant observations also apply to a visual context. Thus, the presence of multiple objects in the visual scene, and hence the availability of competing visual properties should similarly affect the use of contrast and placement of nuclear accent in situated dialogue. Consequently, when generating system output in situated human-robot interaction we should therefore also account for the contextual appropriateness of its *intonation.*

Consider the following two possible realizations of the utterance "Is that a red box?" with different placement of the nuclear accent:[1] [2]

(1) R: Is that a RED box?
                    L*   HH%

(2) R: Is that a red BOX?
                          L* HH%

Based on the standard view in the literature, (1) but not (2) is appropriate in the visual context of Figure 1, where the presence of a 'red' and a 'blue' box licences the use of contrast on the *color* property for distinguishing the intended box from the other. The placement of accent in (1) is also appropriate when the robot is uncertain (and actually wrong)

[*] Supported by EU FP7 Project 'CogX'(FP7-ICT-215181).

[1] The words printed in SMALL CAPITALS indicate the alignment of the nuclear accent in the intonational contour. The intonational description shown beneath the utterances follows [4].

[2] The text labels on the objects were not present in the original pictures. We added them for presentation purposes in this paper, because the colors are not sufficiently distinguishable in black-and-white print.

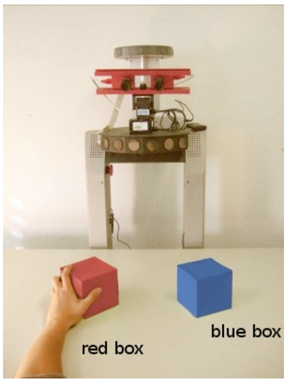


Figure 1: A visual context that is *congruent* for the *marked* accent placement in (1) but, *non-congruent* to the *unmarked* accent placement in (2).

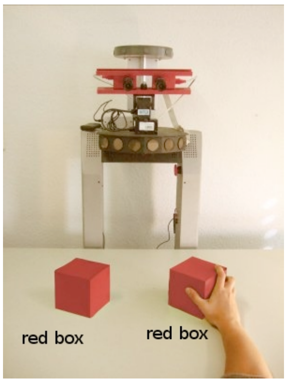


Figure 2: A visual context that is *non-congruent* for the *marked* accent placement in (1) but, *congruent* to the accent placement in (2).

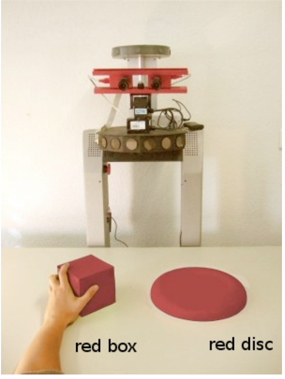


Figure 3: A visual context that is *congruent* for the accent placement in (2) but, *non-congruent* to the *marked* accent placement in (1).

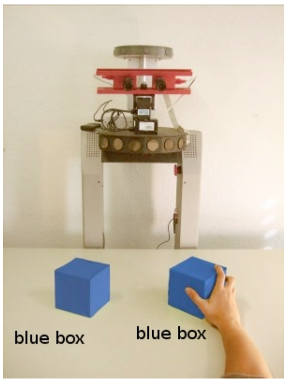


Figure 4: A visual context that is *congruent* for the accent placement in (1) but, *non-congruent* to the *marked* accent placement in (2).

about the color of the box it intends to refer to, as in the visual context of Figure 4. The nuclear accent placement in (2) is licensed in the visual contexts in Figure 2 and 3, since both objects have the color 'red', thereby offering no competing visual properties. The accent placement in (1) is not licenced in this case.

In order to verify the claim that visual context influences the perception of nuclear accent in an utterance, we have set up an experiment. In the experiment, a visual scene (such as those in Figure 1, 2, 3, and 4) is shown to a subject, and a robot's clarification request about the visual scene is played. The subject is asked to judge the appropriateness of the robot's utterance irrespective of its correctness. The underlying hypothesis of the experiment is:

If comprehension is sensitive to the relationship of the visual context and the nuclear accent placement then variations in the placement of nuclear accent in an utterance can be perceived. A preference of one pattern of accent placement over the other provides evidence in support of the role of visual context in determining the appropriate intonation of an utterance.

2. THE EXPERIMENT

2.1 Goal

In this experiment, we seek to verify whether the visual context influences the perception of nuclear accent in an utterance.

2.2 Methodology

2.2.1 Participants

Thirty-one subjects participated in the experiment. Twenty-one participants accessed an online version of the experiment. The remaining ten undertook the experiment in our lab. The experiment was targeted to native English speakers, but only six of the participants were confirmed to have English as their (only) mother tongue. Most non-native speakers claimed to speak US-English. Psycholinguistic findings reveal that L2 speakers of English are equally sensitive to intonational variations. However, their interpretation of tunes vary with the individual's experience with the L2 language [2]. Following this, we consider it appropriate to collapse data from both native and non-native English speakers for the experiment. All participants were offered a sum of 5 Euros or an Amazon Gift Card worth 5 Euros for their successful completion of the experiment, provided they register. Additionally, three participants were drawn for a prize gift voucher of worth 20 Euros each.

2.2.2 Material and Design

A stimulus in the experiment consists of a visual scene and an audio. The visual scene is a picture of a robot standing at a table with one object already present and another one being introduced by a human (therefore held by a hand, e.g. Figure 1). The audio consists of the robot's clarification request about the new object followed by the human's response 'Yes' or 'No', depending on the correctness of the robot's utterance.

The audio files were synthesized using the MARY[3] text-to-speech synthesizer (TTS) [7]. The MBROLA[4] 'mborla-us2' voice of a US-English male speaker was used for synthesizing the robot's clarification requests. The input to the TTS was provided in MaryXML format to indicate the type and location of nuclear accent and intonational boundary type. The human responses of 'Yes' and 'No' were also synthesized using MARY TTS, albeit with a US-English female speaker unit selection based voice.

Clarification requests of the form "Is that a `color type`" were chosen for the robot's utterances, e.g. "Is that a red ball". The color and type values were selected so that they were monosyllabic words, to maintain uniformity and avoid any other source of prosodic variation in the clarification request except for the contrastive accent placement. We

[3] `mary.dfki.de`

[4] `http://tcts.fpms.ac.be/synthesis/`

used the following eight object types: *ball*, *box*, *disc*, *heart*, *ring*, *sphere*, *star* and *wedge*. Each type appeared in six colors: *black*, *blue*, *brown*, *green*, *pink* and *red*. Using these eight object types and the six colors, we designed forty-eight (6x8) clarification sentences in the aforementioned form.

For the visual stimuli, two (not necessarily different) object types were paired in a picture (of 300x400 pixels), with a PeopleBot[5] standing at the table, see Figure 1. The pairing of object types was done such that each object occurs as an object that is already present on the table, and as an object that is being introduced (held by a hand). We used sixteen object-type pairs and twelve color pairs. The twelve color pairs for each of the sixteen object pairs result in a total of 12x16=192 unique pictures for the visual scenes. The object being introduced was randomly held in left-hand or right-hand to avoid visual saturation e.g. Figure 1 and Figure 2.

We used a 2x2x2 design with three factors of two levels each, i.e. visual context (congruent and non-congruent), intonation (marked and unmarked accent placement) and human response ('Yes' and 'No').

Visual Context. The first experimental condition captures the relationship between the visual context and the placement of nuclear accent in an utterance. Based on the presence or absence of competitive properties in a scene the nuclear accent placement in an utterance is *congruent* (C) i.e. licenced by the visual scene, or *non-congruent* (NC) i.e. not licenced by the visual context.

For example, the combination of accent placement in (1) and the visual scene in Figure 1 correspond to a congruent experimental condition. On the other hand, the combination of accent placement in (2) and the visual scene in Figure 1 correspond to a non-congruent condition.

Intonation. The second condition captures the placement of nuclear accent in an utterance. Two types of nuclear accent placement were chosen – *marked* and *unmarked*. In our stimuli an unmarked placement coincides with the assignment of nuclear accent to the last individual word in an utterance i.e. the noun. This is typically the default location of nuclear accent placement in a text-to-speech synthesizer. A marked nuclear accent placement does not correspond to this default position, instead the nuclear accent is assigned to the modifier. We label the intonation contour resulting from a marked nuclear accent placement as tune A (as in (1)) and the one resulting from an unmarked nuclear accent placement as tune B (as in (2)).

Response. The third condition in the experiment corresponds to whether the robot's hypothesis about the target object as expressed in the clarification request is correct or incorrect. The robot's hypothesis indicates the beliefs it currently holds about the visual scene. Since its perceptory senses are not perfect, its beliefs may or may not be the same as those of the human user. The human's response 'Yes' or 'No', indicates to the robot whether its perception about the target object scene is correct or not. Another reason for introducing this condition is to avoid bias in subject's judgement due to rightness or wrongness of the robot's clarifications.

We represent the eight combinations of these conditions as C-A-YES, C-A-NO, C-B-YES, C-B-NO, NC-A-YES, NC-A-NO, NC-B-YES and NC-B-NO. Each of the forty-eight stimuli sentences is then distributed over these eight conditions. This results into a stimuli set of 384 clarification requests.

In order to create the fillers, we introduce two additional nuclear accent placements. This is done to overcome auditory saturation due to tune A and tune B in the stimuli. The filler tunes exhibit accent placement on either the referential expression "that" or the verbal head "is". We label them as tune C and tune D, respectively. Table 1 summarizes the tunes and their corresponding intonation contours.

Table 1: Intonation Tunes.

Tune	Example
A	Is that a RED box? L* HH%
B	Is that a red BOX? L* HH%
C	Is THAT a red box? L* HH%
D	IS that a red box? L* HH%

The introduction of equally many filler tunes (i.e. 384 tune D and tune C in total) in the list results in a total of 768 items. The items are then divided into eight different lists of ninety-six items each, so that each list has all the forty-eight sentences, and evenly distributed over the eight conditions and two filler tunes. The rationale behind this is that by distributing the items across eight lists we ensure that a subject never sees an item more than once. For example, an utterance such as "Is that a red ball" is first distributed over the eight experimental conditions. Each of these eight items is then placed in one of the eight stimuli lists. Such a distribution allows us to ensure that a subject never sees a combination of a visual and linguistic stimuli twice. The items in each of these eight lists are randomized so that the subject cannot guess the next condition.

2.2.3 Predictions

For inferring the role of visual context in acceptability of the intonation tunes we predict that if comprehension is sensitive to the relationship of visual context and the nuclear accent placement then the utterances corresponding to the congruent condition will be judged more appropriate than utterances in a non-congruent condition.

For inferring the role of visual context in acceptability of a *marked* vs. *unmarked* accent placement we predict that if comprehension is sensitive to the relationship of visual context and the nuclear accent placement then the marked and

[5] One of the robots for the George scenario in the CogX project.

unmarked accent placement will be perceived more appropriate in congruent visual scenes than non-congruent scenes.

For inferring the role of visual context in acceptability of accent placement in a *correct* and *incorrect* robot hypothesis we predict that if comprehension is sensitive only to the relationship of visual context and the nuclear accent placement then a subject's perception of the appropriateness of an utterance will not be affected by the correctness of the robot's hypothesis. That is, congruent and non-congruent stimuli would have the same score distribution for both correct and incorrect robot hypothesis.

2.2.4 Procedure and Tasks

The experiment was implemented using the WebExp[6] system for conducting psychological experiments over the World Wide Web. The WebExp server has been hosted on a server running Linux version 2.6.26-2-amd64 with 1GB RAM. The Web-Experiment offered us a possibility to reach non-local native speakers of English for our experiment. We also ran the experiment in a on-site fashion. Interested participants were invited to our lab and were provided access to the Web-Experiment through a laptop.

On arrival at the Web-Experiment page the participants first read instructions about the task and the procedure. They were informed that in each robot scene there is one object already on the table that the robot knows about, and then another object is being presented by a human; The robot asks a question to verify whether it recognized correctly the type and the color of the object being shown; Since its recognition capacity is imperfect, it may make a mistake; The human responds to the robot with a 'Yes' or a 'No'; Their task is to evaluate whether the robot asked the question in a way appropriate to the current scene, irrespective of whether it recognized the object (its type and color) correctly or not.

Subsequently, the subjects filled in details regarding their age, gender, mother tongue, English they speak (US, UK, etc.), educational background, and their past experience with spoken language interfaces. After this subjects were automatically assigned one of the eight lists of stimuli. Next, through a set of six practice stimuli the subjects are introduced to the presentation style of the stimuli and their tasks.

In the practice session and the main experiment, the presentation of stimuli and the evaluation of the stimuli proceeds in three steps.

In the first step, the visual stimulus (a picture) is shown to the subject, and with a delay of 1500ms the corresponding audio stimuli for the robot's clarification request followed by the audio of the human user's response is played. This added delay is a standard procedure for visual preview as visual stimuli capture a subject's visual attention. In the absence of a visual preview, linking the attention captured by the visual scene with the audio stimulus from the clarification would have been a challenging task for the subject. The sentence would be over before the participants would have started to pay attention to the spoken stimuli. Once the audio stops playing, the visual scene disappears after a delay of 1s. This delay is added to give the subject some time for linking the dialogue with the visual scene.

In the second step, the subject is asked for their judgement of the robot's utterance: "Your evaluation of how appropriately the question was asked." The subject indicate their judgement by selecting a radio-button on a 5-point scale between good and bad.

In the third step, the subject is shown a simple math calculation task and asked to judge whether it is correct. An audio with the ticking of a clock is also played until the subject responds. The purpose of the calculation task and the clock audio is to interrupt the subject's visual and audio stimulation, due to the current presentation, before proceeding to the next presentation. Once the subject responds to the calculation task, the next stimulus is presented as just described.

The experiment was designed to take 20-25 minutes to finish.

2.3 Results

We analyzed data of thirty-one participants (i.e. 31x96= 2976 data points for analysis). We exclude the filler items from the analysis (this makes 2976/2=1488 data points under current investigations).

2.3.1 The effect of visual context on perception

We expected the stimuli to be more acceptable in the congruent then the non-congruent condition. That is, congruent stimuli should be judged more often good (score of 5) then non-congruent stimuli. From Table 2 (last column) we observe that only 50% of the congruent stimuli were judged good. However, 44.63% of the non-congruent were also judged good. Both these findings are unexpected. We expected the score for congruent stimuli to be much higher, and for the non-congruent to be very low.

Table 2: Distribution of the scores over 1488 data points.

Score	1-Bad	2	3	4	5-Good
C %	85 11.42%	83 11.15%	83 11.15%	121 16.26%	372 50%
NC %	77 10.34%	90 12.09%	139 18.68%	106 14.27%	332 44.62%
Count %	162 10.88%	173 11.62%	222 14.91%	227 15.25%	704 47.31%

The plots in Figure 5 and 6 suggest that the distribution for subjective judgement for congruent and non-congruent stimuli is rather similar. The high bars for the score of 5 and 4 in Figure 6 and the low score 1 and 2 in Figure 5 are contrary to our expectation. These are indicators that the congruent and non-congruent stimuli either failed to make an impression on the subjects or something hampered the perception of the visual context.

In order to further compare the subjective score of good and bad we collapsed the score 5 and 4 under the label 'GOOD', and, the score of 1 and 2 under label 'BAD'.

Table 3 shows the exact figures for the distribution of GOOD

[6] http://www.webexp.info/

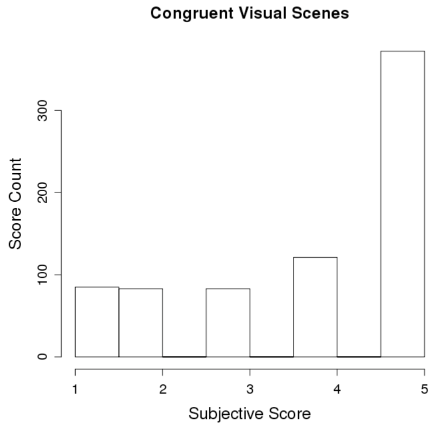


Figure 5: Subjective score distribution for congruent stimuli.

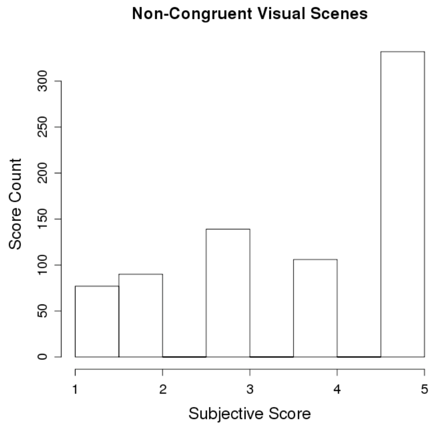


Figure 6: Subjective score distribution for non-congruent stimuli.

and BAD labels over the visual context. We observe that utterances in a congruent visual context were more often judged GOOD (66.26%) than BAD (22.53%). However, the distribution of judgement for the non-congruent visual context is not very different from the congruent context. About 58.87% of the stimuli in the non-congruent visual context were judged GOOD. That is, although the pitch accent placement was not licenced by the visual context of the scenes, the utterances were often judged GOOD. This is contrary to our prediction. We expected the subjective judgement of utterances in non-congruent visual contexts to be mostly BAD.

Table 3: Distribution of GOOD and BAD over Visual Context.

Visual Context	GOOD	BAD	NUTRL
C	493	168	83
%	66.26%	22.58%	11.15%
NC	438	167	139
%	58.87%	22.44%	18.68%

2.3.2 *The effect of visual context on perception of tunes*

We predicted that both tune A and tune B should be perceived more acceptable in congruent visual context than non-congruent context. Table 4 shows that a large portion (about 62%) of both tunes A and B were judged GOOD. Whether the visual context influenced these perceptions can be seen from the distribution of judgement of the tunes over the congruent and non-congruent visual context.

Table 4: Distribution of GOOD and BAD over Tunes.

Tune	GOOD	BAD	NUTRL
marked (A)	464	159	121
%	62.36%	21.37%	16.26%
unmarked (B)	467	176	101
%	62.76%	23.65%	13.57%

In Table 5 we see that tune A was judged GOOD more often in a congruent condition(C) than in a non-congruent condition(NC). Similar distribution is observed for tune B. Another way to verify our prediction is to observe the distribution of BAD label. Tune A has been judged more often BAD in non-congruent condition than congruent condition i.e. more often scored BAD in non-congruent. This provides us evidence in support of our prediction that marked and unmarked nuclear placement is perceived more acceptable when the visual context is congruent i.e. licences the accent placement.

Table 5: Distribution of GOOD and BAD over Tunes-Visual Scene.

Tune	GOOD	BAD	NUTRL
A-C	241	77	54
%	64.78%	20.69%	14.51%
A-NC	223	82	67
%	59.94%	22.04%	18.01%
B-C	252	91	29
%	67.74%	24.47%	7.79%
B-NC	215	85	72
%	57.79%	22.84%	19.35%

Both tunes A and B, however, have been judged more often BAD in congruent condition than in the non-congruent condition. This is contradictory to our expectations and we didn't find any explanation for this in the data.

2.3.3 *The effect of visual context on robot's hypothesis*

Observing the distribution in Table 6 we see that for a *correct* hypothesis i.e. when human response is 'Yes', tune A is judged more often GOOD in congruent condition than non-congruent condition. The same applies for tune B which is more often judged GOOD in the congruent condition than non-congruent condition. However, for an *incorrect* hypothesis i.e. for a human response of 'No', tune A is more often judged BAD in the congruent condition than non-congruent condition. The same also applies for hypothesis tune B, which is judged more often BAD in congruent condition than non-congruent. This is contradictory to our prediction that

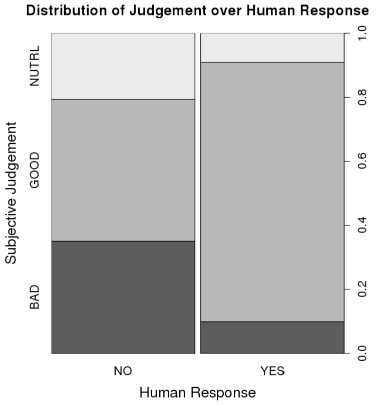


Figure 7: Subjective Judgement vs. Human Response.

the judgement of tunes in a visual context is not affected by the correctness or wrongness of the robot's hypothesis.

Table 6: Distribution of GOOD and BAD over Tunes–Visual-Scene–Hypothesis.

Tune	GOOD	BAD	NUTRL
A-C-YES %	156 83.87%	13 6.98%	17 9.13%
A-NC-YES %	140 75.26%	23 12.36%	23 12.36%
A-C-NO %	85 45.69%	64 34.40%	37 19.89%
A-NC-NO %	83 44.62%	59 31.72%	44 23.65%
B-C-YES %	167 89.78%	11 5.91%	8 4.3%
B-NC-YES %	139 74.73%	27 14.51%	20 10.75%
B-C-NO %	85 45.69%	80 43.01%	21 11.29%
B-NC-NO %	76 40.86%	58 31.18%	52 27.65%

The plot in Figure 7 provides the distribution of the subjective judgement over the human responses ('Yes' and 'No') respectively. It can be inferred from the plot that robot's clarification utterances with human response as 'Yes' were judged GOOD more often than those with human response 'No'. This indicates that the subjects were judging the *correctness* of the robot's hypothesis, rather than judging the *appropriateness* of the request in context of the visual scene.

The distribution of judgement over the human response clarifies to an extent why we do not see a significant difference between the subjective judgement for congruent and non-congruent visual contexts(cf. Table 3). As the subjects were judging the correctness of robot's hypothesis they perhaps paid attention to only the object being introduced. The presence of other object in the visual context and the nuclear accent placement in the intonation did not factor in their decisions.

Coming back to the issue of why the wrong hypothesis is judged more often BAD in a congruent case than non-congruent case, we observe the stimuli for these specific conditions (A-C-NO, A-NC-NO, B-C-NO and B-NC-NO). From the analysis we attribute these distribution to the visual context established by the pictures in these stimuli. In both congruent and non-congruent condition the stimuli was non-congruent from a subject's view point. In a congruent condition the visual scenes offered no ambiguity in the visual context and therefore a subject's visual attention is relatively relaxed, and hence the decision about the "correctness" of the robot's hypothesis is easier and harsher i.e. judged more often BAD. For non-congruent condition the visual scene offers some ambiguity for the subjects as well, and therefore presumably draws more of subject's visual attention, and perhaps this interferes with the subjective judgement as BAD, i.e. although they were judged BAD because of the "incorrect" robot hypothesis, the visual context compensated the harshness of BAD score for non-congruent cases.

3. DISCUSSION AND CONCLUSIONS

Existing attempts to model the intonation of dialogue system output in practical systems include [6, 1, 3, 9]. These systems illustrate various approaches to model the role of context in realizing intonation, but provide only limited experimental evaluation.

For example, in [3] intonation assignment in system turns that are direct answers to questions is done based on *information strcture partitioning* according to the preceding context, both in terms of what question is being answered and what alternative are salient. Accent placement is determined using semantic parallelism: two basic terms as parallel when they are either identical or alternative (i.e. belonging to same sort but non-identical). A perception experiment comparing system generated responses with controlled intonation against defaults indicated that contextual appropriateness of system output improves when intonation is assigned based on infromation structure.

A method of synthesizing contextually appropriate intonation with limited domain unit selection voices is presented in [9]. In a pilot study, they built an APML-aware limited domain voice for use in flight information dialogues, which involve comparing and contrasting the most compelling attributes of the most relevant flights available, rather than simply listing the query results [10]. In a perception experiment comparing the APML voice to a default version built using the same recordings without the additional structure, the intonation produced by the APML voice was judged significantly more contextually appropriate than that of the default voice.

Situated human-robot dialogue differs from the type of dialogue in these applications in that the dialogue context is not the only source of contextual information: te visual context is also part of the discourse context, and should be used for determining the placement of nuclear accent in system utterances. Moreover, whereas the abovementioned systems address intonation assignment in statements answering user's questions, we concentrate on clarification requests pertaining to changes in the visual context. Such clarification requests may not be related to prior mentions in the dialogue; they may concern objects or properties that exist in the visual scene but have not been spoken about.

The analysis of our experiment data reveals that the acceptability of a clarification request is influenced by the visual context. We observe that utterances in which the nuclear accent placement is licenced by the visual context are perceived more often as good than those where the visual context does not licence the accent placement. We do not know of any other similar study that investigates the role of visual context in establishing the appropriateness of intonation.

The findings further support the claim that intonational assignment (be it *marked* and *unmarked*) is governed by the visual context. Both marked and unmarked nuclear accent placements are preferred when the visual context licenses them.

The distinctive pattern for the condition (C-NO-A and C-NO-B) provides even stronger evidence on the role of perception of intonation in a visual context that is non-congruent from a subject's view point. The contextually appropriate usage of intonation in incorrect hypotheses leave no scope of ambiguity for the speaker in perception of the speaker's intentions. It can be claimed that an incorrect query in an unambiguous situation is least accepted.

In order to establish the role of the visual scene and intontation for comprehension, we are preparing an eye tracker experiment for verifying if the subjects pay attention to the already present object when making a judgement. We modify the design of this experiment with a change that instead of the human responding to the robot's query that subject would be required to answer the query. In this manner we will be able to involve the subjects in the interaction with the system. Moreover, since the subjects are required to respond to the robot's queries, the objective nature of the task enables us to measure the influence of visual scene and the intonation on their reaction. The hypothesis for this experiment is that with congruent intonation subject will be looking more at the right object, and that they will react faster. At least for the cases where the hypothesis is correct. It's an interesting question whether there will be any differences between the intonation patterns when the robot's hypothesis is wrong.

4. ACKNOWLEDGMENTS

This work was supported by the EU Project CogX (FP7-ICT-215181).

5. REFERENCES

[1] J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. Achorn, T. Becket, B. Douville, S. Prevost, and M. Stone. Animated conversation: Rule-based generation of facial expression, gesture spoken intonation for multiple conversational agents. pages 413–420, 1994.

[2] E. Garbe, B. S. Rosner, J. Garciá-Albea, and X. Zhou. Perception of english intonation by english, spanish, and chinese listeners. *Language and Speech*, 46(4):375–401, 2003.

[3] I. Kruijff-Korbayová, S. Ericsson, K. J. Rodríguez, and E. Karagjosova. Producing contextually appropriate intonation is an information-state based dialogue system. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 227–234. ACL, 2003.

[4] J. Pierrehumbert. *The Phonology and Phonetics of English Intonation.* PhD thesis, Massachusetts Institute of Technology, 1980.

[5] J. Pierrehumbert and J. Hirschberg. The meaning of intonation in the interpretation of discourse. In P. Cohen, J. Morgan, and M. Pollack, editors, *Intentions in Communication.* MIT Press, Cambridge MA, 1990.

[6] S. A. Prevost. *A Semantics of Contrast and Information Structure for Specifying Intonation in Spoken Language Generation.* Phd thesis, University of Pennsylvania, Institute for Research in Cognitive Science Technical Report, Pennsylvania, USA, 1996.

[7] M. Schröder and J. 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.

[8] M. Steedman. Information structure and the syntax-phonology interface. *Linguistic Inquiry*, 31:649–689, 2000.

[9] M. White, R. Baker, and R. A. J. Clark. Synthetizing contextually appropriate intonation in limited domains. In *Proceedings of the 5th ISCA Speech Synthesis Workshop*, 2004.

[10] M. White, J. D. Moore, M. E. Foster, and O. Lemon. Generating tailored, comparative descriptions in spoken dialogue. In *FLAIRS Conference*, 2004.