Accommodation through Tacit Sensing

Luciana Benotti

Accommodation through Tacit Sensing

Luciana Benotti
TALARIS Team - LORIA (Université Henri Poincaré, INRIA)
BP 239, 54506 Vandoeuvre-lès-Nancy, France
Luciana.Benotti@loria.fr

Abstract

The aim of this paper is to use insights from the theory of accommodation to study in a uniform way different kinds of acts involved in situated dialogue. When interlocutors are engaged in situated dialogue, their informational states evolve through dialogue acts, physical acts and sensing acts. We model this evolution in a non-traditional conversational system using tools from mature branches of artificial intelligence. In particular, we use a planner that is able to find plans in the presence of incomplete knowledge and sensing: PKS (Petrick and Bacchus, 2002). In the resulting model, we study the interactions among dialogue acts, physical acts and sensing acts, and their relationship with accommodation.

1 Introduction

The phenomena of accommodation has been widely studied from philosophical and linguistic perspectives, ranging from classical papers like (Lewis, 1979) to recent contributions like (Beaver and Zeevat, 2007). We view accommodation theory as a schema in which to study, in a uniform way, the different kinds of acts that occur in situated dialogue. We not only believe that such an approach can help us obtain better models of dialogue, but also that dialogue is an essential setting in which to test such theories, theories that are too frequently divorced from the commonest setting of language use: situated conversation.

When interlocutors are engaged in situated dialogue, it is evident that their informational states evolve as a result of the dialogue acts performed during the task, and through the physical acts that interlocutors perform on their environment. But their states are also updated with the information that the participants sense from their environment; embedded agents do not have complete information about the world but they can sense it.

The approach presented here uses insights (and tools) from mature branches of artificial intelligence, such as planning with incomplete knowledge and sensing, in order to build a model for non-traditional conversational systems. In general, traditional conversational systems assume that conversational partners share common goals and collaborate in order to perform the task at hand as efficiently as possible. Our setup explores a case where conversational partners do not share a common goal and they are not as cooperative as partners involved in task-oriented dialogue. Our setup is a text-adventure game, where one of the participants is the player and the other participant is the game. The game has all the information needed to solve the game task but this is not its goal; its goal is to make the interaction engaging and challenging, encouraging the player to explore and discover the game world.

This work is part of a larger project on reconciling linguistic reasoning and collaborative reasoning in conversation (Benotti, 2007; Benotti, 2008). We advance this program here by adding to our model the treatment of sensing actions. To this end, we have integrated in a conversational system a planner that is able to find plans in the presence of incomplete knowledge and sensing: PKS (Petrick and Bacchus, 2002). In the resulting model, we study the interactions among dialogue acts, physical acts and sensing acts. We believe that this issue relates in relevant ways with the theoretical question: “In which contexts can sentences that have particular implicatures felicitously oc-

cur?" Following (Beaver, 1994) we believe this to be a better formulation of the problem of accommodation than the traditional question: "What inferences do people draw from sentences?"

2 Accommodating when talking, acting and sensing: everyday examples

Accommodation and grounding of dialogue and physical acts are topics that have been widely studied. But the study of accommodation and grounding of sensing acts is also essential when agents are embedded. Moreover, even when interlocutors are co-situated, sensing acts are usually less evident than physical and dialogue acts. Hence, an important question to study is "When is the common ground of the dialogue updated with the sensed information?" Or in other words, "When is there in the state of the activity enough evidence that a piece of information has been sensed?"

Let us address these questions with an example:

In kindergarden, the teacher showed a green square to a boy and, offering a piece of paper, told him: "Paint a circle that has this same color".

This simple example illustrates the interaction of a *dialogue act* performed by the teacher (request) with a *sensing action* (sense color) and a *physical action* (paint) that the teacher expects from the boy. When giving this instruction the teacher relied on the ability of the boy to sense the colors, but the sensing action is left tacit in the teacher request. She could have make it explicit saying "Look at the color of the square and paint a circle that has the same color". However in conversation, sensing actions are more naturally left tacit than made explicit. Why? Because they are so natural for sensing agents (indeed, sometimes they are unavoidable) that it is extremely easy to take them for granted.

Now we are going to look at this example as an instance of the general *rule of accommodation* introduced by Lewis in the article in which he coins the word *accommodation*:

If at time t something is said that requires component s_n of conversational score to have a value in the range r if what is said is to be true, or otherwise acceptable; and if s_n does not have a value in the range r just before t; and if such and such further conditions hold; then at t the score-component s_n takes some value in the range r. (Lewis, 1979, p.347)

This rule will help us perform a detailed analysis of our example in order to address the questions raised in the beginning of this section. Bearing this schema in mind, let us analyze step by step the different values that the variables of the rule take for our simple example. First of all, what's t? This is what Stalnaker has to say here:

The prior context that is relevant to the interpretation of a speech act is the context as it is changed by the fact that the speech act was made, but prior to the acceptance or rejection of the speech act. (Stalnaker, 1998, p.8)

So in our example t is the time right after the teacher said "Paint a circle that has this same color" but before the acceptance or rejection of this request.

Now, let us determine what the relevant components s_n are. Suppose that the boy is color blind and the teacher knows it. Then her request does not make much sense and any side participant and the boy himself will start asking what the goal of the request is, because clearly it cannot be the literal one: to obtain a green circle. Therefore, the color referred to by the teacher is the s_1 of our example. And if what the teacher said is to be acceptable, s_1 is required to have a particular value r_1; the same color than the square has in the real world (or in fact, a representation of it). Furthermore, there is no evidence that s_1 already has the value r_1 before the teacher began to speak (that is, there is no evidence that the color has been under discussion before), so we can assume that it doesn't.

Now, what are the further conditions that need to hold so that, at t, the score-component s_1 takes some value r_1? The teacher and the boy both know (at least intuitively) that people can sense their environment, that members of the same culture usually assign the same name to the same parts of the spectrum of colors, that humans can remember facts that they sense, that the sensed object is accessible, that a person will actually sense the color of an object if he is required to know this fact; the teacher and the boy rely on these and many other things that are usually taken for granted. All this knowledge is necessary for the boy to come up with the right sequence of actions in order to respond to the teacher's request; that is, in order to sense the color of the square and paint the circle.

Following Lewis, we would finish our instantiation of the rule of accommodation with the fact

that at the time t the score-component s_1 takes value r_1. Two last comments are in order here. First, it is worth pointing out that at the moment t the request has not yet been accepted or rejected but the addressee has already taken it in and adjusted himself to the fact that the dialogue act has been performed. The acceptance or rejection can be seen as a second change to the conversational record that ocurrs after the rule of accommodation applies. It's very important to distinguish between these two changes. Why? Because even if the request is rejected, the update of the conversational record that resulted from the accommodation may remain. Even if the boy answers "I don't like green. I won't do it", we know that the boy sensed the color of the square.

Second, how does the score-component s_1 takes value r_1? This is a question that is not directly addressed by Lewis but he seems to suggest is that s_1 takes value r_1 and nothing else changes. However, we agree with (Thomason et al., 2006; Hobbs et al., 1993; Kreutel and Matheson, 2003) that what is accommodated in order for s_1 to take value r_1 could be much more than just this fact. If we claimed that only s_1 changes, how can we explain the fact that the boy may take off a blindfold (he was playing "Blind man's bluff") after hearing the teacher? The required updates can also have their requirements (or preconditions) and side-effects, and we think that a natural way to model the accommodation updates is through *tacit acts*.

We adhere then to the view that explains Lewis' broad notion of accommodation (not limited to classical cases of presupposition accommodation) as tacit acts. Physical acts can be left tacit (Benotti, 2007), dialogue acts can be left tacit (Kreutel and Matheson, 2003; Thomason et al., 2006), but also sensing acts can be left tacit (this paper, Section 4). This is not a new idea then, but it's a promising approach and needs to be further developed.

The analysis of our example so far has given us some insight on the questions that were raised in the beginning of this section. We have seen that tacit sensing can be grounded even if the dialogue act that required the sensing is directly rejected (the "boy doesn't like green" example). And it can also be the case that the tacit sensing is grounded even if it cannot be directly executed because, for instance, it requires the execution of some physical act first (the "Blind man's bluff" example). The interactions among sensing acts, dialogue acts and physical acts can be extremely subtle; modelling them (putting sensing, physical and dialogue acts in a common schema) and, in particular making explicit the information at play, is the topic of the rest of this paper.

But first, let us have a look at a few more everyday examples; the aim of these instances is to show how frequent and pervasive are the interactions among different kinds of acts.

2.1 Tacit sensing and referring

If referring is treated as a dialogue act on its own, as many current dialogue systems do (DeVault and Stone, 2006), then the interaction between tacit physical action, tacit sensing action and referring acts need to be controlled. Consider this example:

Suppose that you are told that the hidden treasure you are seeking is behind the blue door. Painting a door blue does not satisfy the goal of finding the blue door — it merely obscures the entity of the appropriate door. (Etzioni et al., 1992, p.116)

This is an example of the incorrect interpretation (painting a door blue) that a conversational system can assign to a command when the system does not have complete information about the environment and has no restrictions on the order in which it can execute actions.

A first conclusion given this observation would be that only sensing actions (and not physical actions) should be allowed before referring actions are resolved. However, it might be the case, for example, that the blue door is in a different room, so the physical action of moving should be allowed before resolving the reference. A more refined approach would be then to leave the relevant properties of the definite description unchanged until the referred object is found. Current off-the-shelf planners provide ways in which to represent properties that must not change (usually called hands-off properties). Using this it is possible to model the fact that searching for a blue door is legitimate, whereas painting some door blue is not.

2.2 Tacit grounding and sensing

During dialogue, grounding acts are frequently left tacit. Consider the following example:

A[1]: Helen did not come to the party.
B[2]: How do you know that?
A[3]: Her car wasn't there.

B[4]: She could have come by bicycle.
(Kreutel and Matheson, 2003, p.6)

In this example, [4] tacitly grounds the assertion [3] (Helen's car wasn't there) but [4] also rejects the fact that [3] is a reason for [1]. In other words, [4] performs two dialogue acts that can be made explicit with "[4']: Ok, her car wasn't there. She could have come by bicycle". Notice that [4] cannot tacitly reject the assertion [3], something is wrong with: "??I saw her car there. She could have come by bicycle".

Sensing actions offer a whole new world for tacit grounding in situated interaction. After giving an instruction, for example, just sensing the results of the required acts is often the best way to know whether the addressee understood (and hence, grounded) the instruction. If you tell your daughter "Turn off the light of your room" and, when you come back, the light is off then you are pretty sure that she heard you.

2.3 Sensing and tacit exogenous events

Unobserved exogenous events can change the value of properties that have already been sensed. So we may well be faced with the treatment of not only incomplete but also incorrect information. But if we assume that the state of the world evolves via the effects of actions and events, then there is a intuitive approach for updating sensed values. Whenever a sensed property needs to be updated in order to make sense of the evolution of the interaction, a tacit exogenous event that updates this property can be inferred. In the following example, Andrew might have sensed that the contents of the pot were raw, but after a while he observe Bess's actions and update his knowledge.

Perhaps only Bess will see when the contents of her boiling pot have cooked. Andrew might still infer that this event has taken place from observing Bess's actions — say, by watching Bess turn off the heat or empty the pot. (Thomason et al., 2006, p.16)

In this three subsections we have shown everyday examples of the interaction of sensing acts, physical acts and dialogue acts. But these are only the tip of the iceberg. We believe that research on such interactions will be fundamental to deepening our understanding of situated dialogue. But how can we model these interactions? This is the topic of the next two sections.

3 A technical framework for tacit sensing

In this section we will introduce the two systems used to implement the ideas discussed in the previous section. We first briefly present our conversational application (the text adventure game), and then describe the main features of the planner that we use for our formalization and case-studies.

3.1 Situated interaction in a text-adventure

We have implemented a text-adventure game which can interpret commands that require tacit sensing. In this game-engine, the player can be embedded in different simulated game environments. The player can issue natural language requests to the game in order to manipulate and change the game environment. She can request to *sense the game objects* through special actions such as read; or directly *perceive the environment* (for example, every time the player enters a new room the game describes it). The situated perspective, and the answers generated by the game as a result of the player requests, allow the player to discover the rules by which her environment is governed and to extend her knowledge accordingly.

A game scenario is represented by several informational components: a database that specifies STRIPS-like actions (Fikes et al., 1972), a grammar, a lexicon, and two description logic knowledge bases (Baader et al., 2003) that share a set of definitions (one knowledge base models the player knowledge and the other the game scenario). The natural language processing module receives the player command and outputs a flat semantic representation that is used by the action handling module to modify the game scenario. The natural language generation module verbalizes the results of the player actions. (Benotti, 2007) describes how classical planning capabilities can be integrated into the architecture of the game-engine. Such planning abilities allow the game to infer *physical actions* left tacit by the player using the off-the-shelf planner Blackbox (Kautz and Selman, 1999). Blackbox implements classical planning techniques and assumes complete knowledge about the planning domain. In this paper, the planner PKS (Planning with Knowledge and Sensing) is used in order to investigate the case in which *sensing actions* are tacit.

3.2 Planning with knowledge and sensing

PKS (Petrick and Bacchus, 2002) is a knowledge-based planner that is able to construct conditional plans in the presence of incomplete knowledge. PKS builds plans by reasoning about the effects of actions on an agent's knowledge state, as opposed to other approaches based on possible-world reasoning. By reasoning at the knowledge level, PKS can avoid some of the irrelevant distinctions that occur at the world level, improving efficiency and producing natural plans. The PKS specification language offers features such as functions and variables, allowing it to solve problems that can be difficult for traditional planners (and making it ideal for non-traditional dialogue systems).

PKS is based on a generalization of STRIPS. In STRIPS, the world state is modelled by a single database. In PKS, the planners knowledge state, rather than the world state, is represented by a tuple $\langle K_f, K_w, K_v, K_x \rangle$ of databases whose contents have a fixed, formal interpretation in epistemic logic. Actions are specified as updates to these databases using the knowledge primitives *know fact* which modifies the database K_f, *know value* which modifies the database K_v, *know whether* which modifies the database K_w, and *know which* which modifies the database K_x.

We briefly describe these four databases here. K_f is like a standard STRIPS database except that both positive and negative facts are stored and the closed world assumption does not apply. K_v stores information about function values that will become known at execution time, such as the plan-time effects of sensing actions that return numeric values. K_w models the plan-time effects of binary sensing actions that sense the truth value of a proposition. K_x models the agent's exclusive disjunctive knowledge of literals (that is, the agent knows that exactly one literal from a set is true).

PKS performance has been tested for the composition of web services with promising results (Martinez and Lesperance, 2004). Moreover, in the prototype we have implemented using PKS inside our text-adventure game, PKS response time was acceptable (less than 2 seconds) for the kind of planning problems that the text adventure typically gives rise to. We tested it using the breadth first search strategy, rather than depth first because we require optimal length plans.

4 Tacit sensing: 2 case-studies

In this section we are going to explain in detail how a command issued by the player that includes tacit sensing actions is interpreted using PKS, and then executed by the game. We first classify sensing actions as either disjunctive or existential. We then present a case-study of disjunctive knowledge that makes use of conditional plans. Finally, we describe a case-study of existential knowledge that makes use of parametric plans.

4.1 Incomplete knowledge and sensing

There are two sorts of sensing actions, corresponding to the two ways an agent can gather information about the world at run-time. On the one hand, a sensing action can observe the truth value of a proposition $P(c)$, resulting in a conditional plan. The kind of incomplete knowledge sensed by this kind of action can be described as *binary* because it represents the fact that the agent knows which of the two disjuncts in $P(c) \vee \neg P(c)$ is true. In PKS, binary sensing actions are those that modify the K_w knowledge base. On the other hand, a sensing action can identify an object that has a particular property, resulting in a plan that contains run-time variables. The kind of incomplete knowledge sensed by these kind of action can be described as *existential* because it represents the fact that the agent knows a witness for $\exists x.P(x)$. In PKS, existential sensing actions are those that modify the K_v database.

We will now explain in detail how these two kinds of sensing actions can be left tacit by the player in our text-adventure game.

We said that our model can handle incomplete knowledge about the interaction in which a dialogue is situated. But how incomplete is the knowledge the model can handle? There are several levels at which knowledge can be incomplete. The most studied scenario is one in which not all the properties and relations of the objects involved in the task are known, but the set of objects is finite and all objects are named (that is all objects are associated with a constant). If this simplifying assumption is made, existential and disjunctive incomplete knowledge collapse; one can be defined in terms of the other. If all objects are named, the fact that there exists an object that satisfies a particular property can be expressed as the disjunction of that property applied to all the objects in the domain. However, we cannot make this sim-

plifying assumption because we are dealing with an environment where not all objects are known at plan time. Thus we not only need to study the use of disjunctive plans, we also need plans with run-time variables.

4.2 Tacit actions in conditional plans

We are going to analyze commands issued by the player that involve the execution of binary sensing actions that result in conditional plans.

In order to motivate conditional plans, let us consider an example. Suppose that the player is in a room with a locked door. She is looking around searching for a way to open the door, when the game says that there are two keys (one silver and one golden) lying on a table in front of her. Then she inputs the command "Open the door". Correctly executing this command in the state of the game described amounts to executing the following conditional plan. The plan involves taking both keys, trying the silver one in the door, and (if it fits) unlocking and opening the door; otherwise the golden key is used.

```
<init>
  take(silver_key,table)
  take(golden_key,table)
  trykey(silver_key,door)
  <branch,fits_in(silver_key,door)>
  <k+>:
     unlock(door,silver_key)
     open(door)
  <k->:
     unlock(door,golden_key)
     open(door)
```

But should the game execute this plan for the player? There is no definitive answer to this question unless we refine it further; we need to consider what the goal of such a text-adventure game is. For a start, it certainly shares a number of similarities with task-oriented dialogue systems (such as (Ferguson et al., 1996)). In particular, like task-oriented dialogue systems, our text-adventure has knowledge of the task; it models the steps involved in the task and how to talk about them. But task-oriented dialogue systems typically strive to solve the task as efficiently as possible, even if this leads to unnatural dialogue. On the other hand, for games (and indeed for tutoring systems too) efficiency in task performance and brevity is not necessarily an advantage; the longer the interaction the greater the opportunity of having a useful interactive experience (and more opportunity for learning). If we take this perspective, then a natural answer to our question would be: the game should *not* open the door for the player; rather it must force the player to perform all the steps on her own so that she will learn the task.

However, in order for a game to be engaging it cannot force the player to do repetitive tasks over and over again. In games, rules are not stated in advance; games require the skills of rule discovery through observation, trial and error, and hypothesis testing. Figuring out the rules governing the behavior of a dynamic representation is basically the cognitive process of inductive discovery, and this is challenging and motivating. But once a rule is learned, the player will no longer find it motivating to automatically apply it again and again. How to best use facts and rules that the player learned is an issue that needs to be carefully considered when deciding how a system should behave.

So, what's the answer to our question? What should the game do? Or in more general terms, when can this command (which gives rise to particular implicatures) felicitously occur? This depends on what has already happened in the game. Has the player already been through enough experiences to have the knowledge that is necessary in order to "open the door"? If yes, don't force the player to repeat the boring steps.

But how can we represent the knowledge that is necessary in order to find the conditional plan involved by this command, in order to leave the necessary actions tacit? To illustrate our explanation, let us go back to the concrete input "Open the door" and its conditional plan and analyze how it is handled by the system. The sensing action involved in the conditional plan is `trykey` defined in PKS as follows:

```
<action name="trykey">
  <params>?x, ?y</params>
  <preconds>
    Kf(accessible(?x)) ^
    Kf(locked(?x)) ^
    Kf(key(?y)) ^
    Kf(inventory_object(?y))
  </preconds>
  <effects>
    add(Kw, fits_in(?y,?x));
  </effects>
  </action>
```

Intuitively, after executing the action `trykey(?x,?y)` the agent *knows whether* a particular key `?x` fits in a locked object `?y` or not. Is this knowledge enough to find the conditional plan above? No, because it could be the case that none of the two keys fit into the door. If this is a possibility, then the conditional plan may not

achieve the goal `Kf(open(door))`. In order to rule out this possibility the following facts have to the added to the initial state of the planning problem:

```
add(Kx, fits_in(k1,c1)|fits_in(k2,c1))
```

Given this information, PKS is able to come up with the conditional plan above.

In its current version, PKS only returns disjunctive plans that will always be successful given the specification of the planning problem. It doesn't matter what the actual configuration of the world is, PKS guaranties that there will be a branch in the plan that achieves the goal. If this cannot be achieved then PKS will say that there is no plan. However, it might be the case that there is some conditional plan that is successful for most but not all configurations of the world. It would be interesting to have a planner that could provide plans for these cases, even when some of the branches will not achieve the goal.

Implementation details

Conditional plans are executed by decomposing them in disjunctive plans. For example, the conditional plan shown above can be decomposed in two disjunctive plans, namely:

```
take(silver_key,table)
take(golden_key,table)
unlock(door,silver_key)
open(door)
```

and

```
take(silver_key,table)
take(golden_key,table)
unlock(door,golden_key)
open(door)
```

These two disjunctive plans can be directly inserted in the game flow. In the game, the semantic representation of a command is in disjunctive normal form (that is, it is a disjunction of conjunction of actions). Each disjunct corresponds to a different reading of the command, hence a command's semantic representation will contain more than one disjunct if the command is ambiguous. Here, each branch of the plan can be reinserted into the game flow as a disjunct in the semantic representation of the command. Only one of the branches will be successfully executed since the sensed information is known to be exclusive (only one of the keys fits).

4.3 Run-time variables in tacit actions

In this section we are going to analyze commands issued by the player that involve the execution of existential sensing actions. Existential sensing actions result in parametric plans, that is, plans that include actions with run-time variables, values that will only be known at run time.

In order to motivate parametric plans, let us consider an example in a multiplayer game scenario. There is a player called Beatrix who has found a room with a panel where the location of all other players can be checked. Beatrix knows that in this game scenario, a player can drive herself to any other location if she knows the destination, and that in order to kill someone you have to be in the same place. Beatrix wants Bill dead and so she utters the command "Kill Bill". How do we have to represent this information so that the planner will be able to come up with a successful plan? The goal of the command can be represented with `Kf(dead(bill))` and the information about how the game world works that is already available to Beatrix can be represented with the following action schemas:

```
<action name="checklocation">
  <params>?x</params>
  <preconds>
    Kf(player(?x))
  </preconds>
  <effects>
    add(Kv, haslocation(?x));
  </effects>
</action>
<action name="drive">
  <params>?x,?y</params>
  <preconds>
    Kf(player(?x)) ^
    Kv(?y) ^
    Kf(haslocation(?x)!=?y)
  </preconds>
  <effects>
    add(Kf, haslocation(?x)=?y);
  </effects>
</action>
<action name="kill">
  <params>?x, ?y</params>
  <preconds>
    Kf(player(?x)) ^
    Kf(player(?y)) ^
    Kf(haslocation(?x)=haslocation(?y))
  </preconds>
  <effects>
    add(Kf, dead(?y));
  </effects>
</action>
```

With this information and a factual representation of the initial state the planner should return the following parametric plan. The plan involves checking Bill's location in the panel, driving to that location and killing Bill. The plan is not fully instantiated, as the actual location of Bill will only become known when the command is executed.

```
checklocation(bill)
drive(beatrix,haslocation(bill))
kill(beatrix,bill)
```

When the action `drive` is actually executed in the game, Bill's location can be obtained from the player knowledge base because the action `checklocation` will already have been executed.

5 Conclusions

In this paper we studied Lewis's broad notion of accommodation as a natural schema for treating tacit dialogue acts, tacit physical acts and tacit sensing acts in a uniform way, and we analyzed examples of the interaction among these three types of acts. In particular, we looked at sensing acts and two widely studied dialogue acts: grounding and referring. We also investigated phenomena usually studied in collaborative models of reasoning, such as exogenous events, using this same schema. Following this agenda, our final aim is to reconcile linguistic reasoning and collaborative reasoning in situated conversation.

We then turn to the question of how to model these interactions. For this purpose we integrated the planner PKS in a text-adventure game. PKS is a knowledge-based planner that is able to construct conditional and parametric plans. Such planning abilities allow the game to infer physical and sensing actions left tacit by the player. We believe that the non-traditional setup offered by the game is particularly suited to the study of the differences between collaborative task solving and other (less collaborative) types of interaction.

The work presented in this paper is in its early stages, and it is crucial to carry out an empirical test of our claims. But we believe that we have started to define a path which is worth following for two main reasons. On the theoretical side, we believe that the use of different kinds of tacit actions is omnipresent in human interaction and will help generalize the theory of accommodation. On the practical side, sensing actions are an essential component if we want to build situated dialogue systems that are able to interact in a realistic way.

References

F. Baader, D. Calvanese, D. McGuinness, D. Nardi, and P. Patel. 2003. *The Description Logic Handbook: Theory, Implementation, and Applications*. Cambridge University Press.

D. Beaver and H. Zeevat. 2007. Accommodation. In *The Oxford Handbook of Linguistic Interfaces*, pages 503–539. Oxford University Press.

D. Beaver. 1994. Accommodating topics. In *The Proceedings of the IBM and Journal of Semantics Conference on Focus*.

L. Benotti. 2007. Incomplete knowledge and tacit action: Enlightened update in a dialogue game. In *Workshop on the Semantics and Pragmatics of Dialogue*, pages 17–24, Rovereto, Italy.

L. Benotti. 2008. Accommodation through tacit dialogue acts. In *Conference on Semantics and Modelisation*, Toulouse, France.

D. DeVault and M. Stone. 2006. Scorekeeping in an uncertain language game. In *The 10th Workshop on the Semantics and Pragmatics of Dialogue*, University of Potsdam, Germany.

O. Etzioni, S. Hanks, D. Weld, D. Draper, N. Lesh, and M. Williamson. 1992. An approach to planning with incomplete information. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 115–125.

G. Ferguson, J. Allen, and B. Miller. 1996. TRAINS-95: Towards a mixed-initiative planning assistant. In *Proceedings of the Conference on Artificial Intelligence Planning Systems*, pages 70–77, Edinburgh, Scotland.

R. Fikes, P. Hart, and N. Nilsson. 1972. Learning and executing generalized robot plans. *Artificial Intelligence*, 3:251–288.

J. Hobbs, M. Stickel, D. Appelt, and Paul Martin. 1993. Interpretation as abduction. *Artificial Intelligence*, 63(1–2):69–142.

H. Kautz and B. Selman. 1999. Unifying SAT-based and graph-based planning. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 318–325, Stockholm, Sweden.

J. Kreutel and C. Matheson. 2003. Context-dependent interpretation and implicit dialogue acts. In *Perspectives on Dialogue in the New Millenium*, pages 179–192.

D. Lewis. 1979. Scorekeeping in a language game. *Journal of Philosophical Logic*, 8:339–359.

E. Martinez and Y. Lesperance. 2004. Web service composition as a planning task: Experiments using knowledge-based planning. In *Proceedings of the Workshop on Planning and Scheduling for Web and Grid Services*, pages 62–69.

R. Petrick and F. Bacchus. 2002. A knowledge-based approach to planning with incomplete information and sensing. In *Proceedings of the Sixth International Conference on Artificial Intelligence Planning and Scheduling*, pages 212–221.

R. Stalnaker. 1998. On the representation of context. *Journal of Logic, Language and Information*, 7(1):3–19.

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