A scalable model of planning perlocutionary acts

Alexander Koller and **Andrew Gargett** and **Konstantina Garoufi**
Cluster of Excellence "Multimodal Computing and Interaction"
Saarland University, Saarbrücken, Germany
{koller|gargett|garoufi}@mmci.uni-saarland.de

Abstract

We propose a new model of perlocutionary acts, in which perlocutionary effects of communicative actions are simply effects of operators in a planning problem. A plan using such operators can be computed efficiently, under the assumption that all perlocutionary effects come true as intended; the speaker then monitors the plan execution to detect it when they don't. By scaling the complexity of the execution monitor up or down, we can reconstruct previous approaches to speech act planning and grounding, or build an instruction generation system with real-time performance.

1 Introduction

The reason why people say things is the same as why they perform physical actions: because they want to achieve some goal by doing it. This is most obvious when the communicative action is an instruction which asks an interlocutor to perform a certain physical action; but it is still true for utterances of declarative sentences, which are intended to change the hearer's mental state in some way. The goals which an utterance achieves, or is meant to achieve, are called *perlocutionary effects* by Austin (1962).

However, relatively little work has been done on precise formal and computational models of perlocutionary effects, and in particular on the goal-directed use of communicative actions for their perlocutionary impact. Mainstream approaches such as Perrault and Allen (1980), which rely on modeling complex inferences in the hearer's mind and use non-standard planning formalisms, have never been demonstrated to be computationally efficient enough for practical use. On the other hand, issues of grounding (Clark, 1996) are highly relevant for the problem of modeling perlocutionary effects: If an utterance has not been understood, it cannot be expected to have its intended effect.

In this paper, we propose a new, general model of perlocutionary effects based on AI planning. In this model, the speaker computes a plan of communicative actions, each of which may have perlocutionary effects, under the assumption that all intended perlocutionary effects come true. That is, we model the effect of uttering "please open the window" as changing the world state such that the window becomes open. Because communicative actions can fail to have the intended effects (perhaps the hearer misunderstood, or is uncooperative), the speaker then observes the hearer's behavior to monitor whether the communicative plan has the intended effects. If the speaker notices that something goes wrong, they can react by diagnosing and repairing the problem.

This model makes it possible to deliberately compute a sequence of communicative actions that is fit to achieve a certain perlocutionary effect. Because the assumption that perlocutionary effects come true makes the planning easier, we can compute communicative plans efficiently; furthermore, our model can subsume communicative and physical actions within the same framework quite naturally. By scaling the execution monitoring module, we can trade off the precision with which the hearer's state is modeled against the inefficiency and model complexity this involves, according to the needs of the application. We show how a number of existing approaches to speech act planning and grounding can be reconstructed in this way, and illustrate the use of our model for the situated real-time generation of instructions in a small but fully implemented example.

Plan of the paper. We introduce our model in Section 2, connect it to the earlier literature in Section 3, and show its application to instruction generation in Section 4. Section 5 concludes.

Aspects of Semantics and Pragmatics of Dialogue. SemDial 2010, 14th Workshop on the Semantics and Pragmatics of Dialogue

2 A new model of speech acts

We start by describing our model of speech acts, which combines communicative action planning with monitoring of the hearer's actions.

2.1 Communicative planning

The fundamental idea of our approach is to model a communicative act simply as an action in some planning problem. We take a communicative action to be some act of uttering a string of words. Ultimately, an agent performs such actions in order to achieve a (perlocutionary) goal which is external to language. This could be a physical goal (the light is now on), a goal regarding the mental state of the hearer (the hearer now believes that I have a cat), or something else. In this sense, communicative actions are exactly the same as physical actions: activities that are performed because they seem suitable for reaching a goal.

We model perlocutionary effects as effects of communicative actions in a planning problem. While we use classical planning throughout this paper, the basic idea applies to more expressive formalisms as well. A planning problem consists of a set of planning operators with preconditions and effects; an instance of an operator can be applied in a given planning state if its preconditions are satisfied, and then updates the state according to its effects. *Planning* (see e.g. Nau et al. (2004)) is the problem of finding a sequence of actions (i.e. operator instances) which transforms a given initial state into one that satisfies a given goal.

Consider the following example to illustrate this. An agent A is in a room with a light l_1 and two buttons b_1 and b_2; b_1 will turn on l_1, while b_2 is a dummy, and pressing it has no effect. We can encode this by taking an initial state which contains the atoms $\mathsf{agent}(A)$, $\mathsf{ltswitch}(b_1, l_1)$ (i.e. b_1 is the light switch for l_1), and $\mathsf{state}(l_1, \mathsf{off})$ (i.e. the light is off). Let's also include in the initial state that A is at location p_1 and b_1 is at a (different) location p_2, via atoms $\mathsf{at}(A, p_1)$, $\mathsf{at}(b_1, p_2)$ and $\mathsf{near}(p_1, p_2)$. Finally, let's assume that A wants l_1 turned on. We express this desire by means of a goal $\mathsf{state}(l_1, \mathsf{on})$. Then one valid plan will be to execute instances of the operators in Fig. 1, which encode physical actions performed by the agent. Specifically, **moveto**(A, p_1, p_2) and then **press**(b_1, A, p_2, l_1) will achieve the goal: The first action moves A to p_2, establishing the preconditions for the second action and turning on the light.

moveto(x, y_1, y_2):
  Precond: $\mathsf{agent}(x), \mathsf{at}(x, y_1), \mathsf{near}(y_1, y_2)$
  Effect: $\neg\mathsf{at}(x, y_1), \mathsf{at}(x, y_2)$

press(w, x, y, z):
  Precond: $\mathsf{agent}(x), \mathsf{ltswitch}(w, z), \mathsf{at}(x, y), \mathsf{at}(w, y), \mathsf{state}(z, \mathsf{off})$
  Effect: $\neg\mathsf{state}(z, \mathsf{off}), \mathsf{state}(z, \mathsf{on})$

Figure 1: Physical actions for turning on the light.

"press"(w, z):
  Precond: $\mathsf{ltswitch}(w, z), \mathsf{state}(z, \mathsf{off})$
  Effect: $\neg\mathsf{state}(z, \mathsf{off}), \mathsf{state}(z, \mathsf{on}), \forall w'.w' \neq w \rightarrow \mathsf{distractor}(w')$

"the light switch"(w):
  Precond: $\exists z.\mathsf{ltswitch}(w, z)$
  Effect: $\forall w'.(\neg\exists z.\mathsf{ltswitch}(w', z) \rightarrow \neg\mathsf{distractor}(w'))$

Figure 2: Communicative actions for turning on the light.

If there is a second agent B in the room, then A can alternatively achieve the goal of switching l_1 on by asking B to do it, using communicative actions along the lines of those in Fig. 2. Here we add a further formula $\forall x.\neg\mathsf{distractor}(x)$ to the goal in order to require that the hearer can resolve all referring expressions uniquely. A valid plan is **"press"**(b_1, l_1) and then **"the light switch"**(b_1); this corresponds to uttering the sentence "press the light switch". (We write the names of communicative actions in quotes in order to distinguish them from physical actions.) The first action already achieves A's goal, $\mathsf{state}(l_1, \mathsf{on})$, but also introduces the atom $\mathsf{distractor}(b_2)$ into the planning state, indicating that the hearer won't be able to tell which button to press after hearing only "press ...". Since b_1 is the only light switch, this atom is easily removed by the action **"the light switch"**, which brings us into a goal state.

These two plans involve completely different kinds of actions: One uses physical actions performed by A, the other communicative actions performed by A intended to make B perform appropriate physical actions. Nevertheless, both plans are equally capable of achieving A's goal. We claim that communication is generally a goal-directed activity of this kind, and can be usefully modeled in terms of planning.

2.2 Plan execution monitoring

One crucial feature of this model is that the **"press"** operator, which encodes the action of uttering the word "press", has the effect that the light is on. At first, this seems surprising, as if simply saying "press ..." could magically operate the light switch. This effect can be understood in the following way. Assume that the hearer of an utterance containing "press ..." which is complete in the sense that it is grammatically correct and all referring expressions can be resolved uniquely understands this utterance. Assume also that the hearer is cooperative and follows our request, and that they manage to achieve the goal we have set for them. Then the communicative plan underlying the utterance will indeed have the effect of switching on the light, through the physical actions of our cooperative hearer.

Our communicative planning operators directly contain the perlocutionary effects that the utterance will have if everything goes as the speaker intended. This makes it possible for a perlocutionary effect of one action in the plan to establish the precondition of another, and thus to form communicative plans that are longer than a single utterance; we will present an example where this is crucial in Section 4. But of course, we must account for the possibility that the hearer misunderstood the utterance, or is unwilling or unable to respond in the way the speaker intended; that is, that an action may not have the intended effect.

Here, too, communicative planning is no different from ordinary planning of physical actions. It is reasonable to assume for planning purposes that the operators in the physical plan of Subsection 2.1 have the intended effects, but the plan may fail if A is not able to reach the light switch, or if she made wrong assumptions about the world state, perhaps because the power was down. Inferring whether a plan is being carried out successfully is a common problem in planning for robots, and is called *plan execution monitoring* (Washington et al., 2000; Kvarnström et al., 2008) in that context. Although there is no commonly accepted domain-independent approach, domain-dependent methods typically involve observing the effects of an agent's actions as they are being carried out, and inferring the world state from these observations. Because there is usually some uncertainty about the true world state, which tends not to be directly observable, this can be a hard problem.

A speaker who detects a problem with the execution of their communicative plan has the opportunity to diagnose and repair it. Imagine that after hearing the utterance "press the light switch" in the earlier example, the hearer moves to a point where they can see both b_1 and b_2, and then hesitates. In this case, a hesitation of sufficient duration is evidence that the hearer may not execute the instruction, i.e. that the plan execution didn't have the intended perlocutionary effect. The speaker can now analyze what went wrong, and in the example might conclude that the hearer didn't know that b_2 isn't a light switch. This particular problem could be repaired by supplying more information to help the hearer remove distractors, e.g. by uttering "it's the left one". Deciding when and how to repair is an interesting avenue for future research.

2.3 A scalable model

Putting these modules together, we arrive at a novel model of perlocutionary acts: The speaker computes a plan of communicative actions that is designed to reach a certain goal; executes this plan by performing an utterance; and then observes the hearer's actions to monitor whether the intended perlocutionary effects of the plan are coming to pass. If not, the speaker repairs the plan.

By making optimistic assumptions about the success of perlocutionary effects, this model can get away with planning formalisms that are much simpler than one might expect; in the example, we use ordinary classical planning and move all reasoning about the hearer into the execution monitor. Among other things, this allows us to use fast off-the-shelf planners for the communicative planning itself. As we will see below, even relatively complex systems can be captured by making the execution monitor smart, and even shallow execution monitors can already support useful performances in implemented systems.

2.4 Limitations and extensions

The model proposed above is simplified in a number of ways. First, we have dramatically simplified the planning operators in Fig. 2 for easier presentation. At least, they should distinguish between the knowledge states of A and B and perhaps their common ground; for instance, in the effects of **"the light switch"**, only objects of which the *hearer* knows that they are not light switches should be excluded from the set of distractors. Koller and Stone (2007) show how to extend a planning-based model to make such a distinction.

Although we have only discussed instruction-giving dialogues above, we claim that the model is not limited to such dialogues. On the one hand, declarative utterances affect the hearer through their perlocutionary effects just like imperative utterances do: They alter the hearer's mental state, e.g. by making a certain referent salient, or introducing a new belief. The role of the truth conditions of a declarative sentence is then to specify what perlocutionary effect on a hearer's belief state an utterance of this sentence can bring about.

On the other hand, we believe that other types of dialogue are just as goal-directed as instruction-giving dialogues are. In an argumentative dialogue, for instance, each participant pursues a goal of convincing their partner of something, and chooses communicative actions that are designed to bring this goal about. The role of execution monitoring in this context is to keep track of the partner's mental state and revise the communicative plan as needed. Because both partners' goals may conflict, this is reminiscent of a game-theoretic view of dialogue. It is conceivable that certain types of dialogue are best modeled with more powerful planning formalisms (e.g., information-seeking dialogues by planning with sensing), but all of our points are applicable to such settings as well. In particular, even in more complex settings the planning problem might be simplified by moving some of the workload into the execution monitor.

Finally, we have focused on plans which only contain *either* physical (Fig. 1) *or* communicative (Fig. 2) actions. However, since we have blended the physical and communicative contributions of those acts together (as e.g. with the communicative act **"press"** of Fig. 2), we can also compute plans which combine both types of action. This would allow us, for instance, to interleave communicative actions with gestures. In this way, our proposal could pave the way for a future unified theory which integrates the various kinds of communicative and physical actions.

3 Speech act planning and grounding

We will now discuss how our model relates to earlier models of speech act planning and grounding.

The most obvious point of comparison for our model is the family of speech act planning approaches around Perrault and Allen (1980) (henceforth, P&A), which are characterized by modeling speech act planning as a complex planning problem involving reasoning about the beliefs, desires, and intentions (BDI) of the interlocutors. P&A model the perlocutionary effect of a speech act REQUEST(P) as causing the hearer to intend to do P. However, this effect has to be justified during the planning process by inferences about the hearer's mental state, in which the hearer first recognizes the speaker's intention to request P and then accepts P as their own intention. Although we agree with the fundamental perspective, we find this approach problematic in two respects. First, the perlocutionary effect of REQUEST is modeled as limited to the hearer: it is not that P happens, but only that the hearer wants P to happen. This makes it impossible to compute communicative plans in which a subsequent utterance relies on the intended perlocutionary effects of an earlier utterance, as e.g. in Section 4 below. Second, even if we limit ourselves to the effect on the hearer's mental state, the formal approach to planning that P&A take is so complex that computing plans of nontrivial length is infeasible.

Our model solves the first problem by modeling the intended physical and mental effects directly as effects of the operator, and it solves the second problem by using simple planning formalisms. Compared to P&A, it takes a more optimistic stance in that the default assumption is that perlocutionary effects happen as intended. Any reasoning about the hearer's BDI state can happen during the execution monitoring phase, in which we can compute the expected step-by-step effects of the utterance on the hearer's state (intention recognition, goal uptake, etc.) as P&A do, and then try to establish through observations whether one of them fails to come true. This allows us to compute very simple plans without sacrificing linguistic correctness. We believe that similar comments hold for other recent planning-based models, such as (Steedman and Petrick, 2007; Brenner and Kruijff-Korbayová, 2008; Benotti, 2009).

We share our focus on modeling uncertainty about the effects of communicative actions with recent approaches to modeling dialogue in terms of POMDPs (Frampton and Lemon, 2009; Thomson and Young, 2009). POMDPs are a type of probabilistic planning problem in which the effects of actions only come true with certain probabilities, and in which the true current world state is uncertain and only accessible indirectly through observations; the analogue of a plan is a *policy*, which specifies what action to take given certain observations. This makes POMDPs a very powerful and explicit tool for modeling uncertainty about effects, which is however limited to very simple reasoning about observations. Although our planning model is not probabilistic, we believe that the two approaches may be more compatible than they seem: Many recent approaches to probabilistic planning (including the RFF system, which won the most recent probabilistic planning competition for MDPs (Teichteil-Koenigsbuch et al., 2008)) transform the probabilistic planning problem into a deterministic planning problem in which probable effects are assumed to come true, monitor the execution of the plan, and replan if the original plan fails. This is a connection that we would like to explore further in future work.

Grounding – the process by which interlocutors arrive at the belief that they mutually understood each other – falls out naturally as a special case of our model. A speaker will continue to monitor the hearer's behavior until they are sufficiently convinced that their communicative action was successful. This typically presupposes that the speaker believes that the hearer understood them; traditional classes of devices for achieving grounding, such as backchannels and clarification requests, are among the observations considered in the monitoring. Conversely, the speaker can stop monitoring once they believe their perlocutionary goal has been achieved; that is, when their degree of belief in mutual understanding is "sufficient for current purposes" (Clark and Schaefer, 1989), i.e. the current perlocutionary goal. Our prediction and tracking of expected perlocutionary effects is reminiscent of the treatment of grounding in information state update models, in which utterances introduce *ungrounded discourse units* (Matheson et al., 2000) into the conversational record, which must be later grounded by the interlocutors. In our approach, the first step could be implemented by introducing the ungrounded unit as an effect and then verifying that grounding actually happened in the execution monitor.

In its reliance on planning, our approach is somewhat in contrast to Clark (1996), who fundamentally criticizes planning as an inappropriate model of communication because "people … don't know in advance what they will actually do [because] they cannot get anything done without the others joining them, and they cannot know in advance what the others will do". We claim that this ignorance of speakers about what is going to happen need not keep them from forming a communicative plan and attempting a promising speech act; after all, if the hearer does unexpected things, the speaker will be able to recognize this and react appropriately. In our perspective, communication is not primarily a collaborative activity, but is driven by each individual agent's goals, except insofar as collaboration is necessary to achieve these goals (which it often is). This seems in line with recent psycholinguistic findings indicating that a speaker's willingness to select an utterance that is optimal for the partner is limited (Shintel and Keysar, 2009; Wardlow Lane and Ferreira, in press).

We deliberately keep details about the execution monitoring process open, thereby subsuming approaches where the speaker explicitly models the hearer's mental state (Poesio and Rieser, 2010), or only does this if necessary (Purver, 2006), or which emphasize inferring success from directly accessible observations (Skantze, 2007; Frampton and Lemon, 2009). In this sense, the model we propose is scalable to different modeling needs.

4 Communicative planning in practice

At this point, we have argued that very expressive execution monitors can in principle be used to reconstruct a number of approaches from the literature. We will now demonstrate that even a very *in*expressive execution monitor can be useful in a concrete application. The example on which we illustrate this is the SCRISP system (Garoufi and Koller, 2010), which extends the CRISP NLG system (Koller and Stone, 2007) to situated communication. CRISP, in turn, is a planning-based reimplementation of the SPUD system (Stone et al., 2003) for integrated NLG with tree-adjoining grammars (TAG, (Joshi and Schabes, 1997)).

SCRISP generates real-time navigation and action instructions in a virtual 3D environment. The overall scenario is taken from the GIVE-1 Challenge (Byron et al., 2009): A human instruction follower (IF) must move around in a virtual world as in Fig. 3, which is presented to them in 3D as in Fig. 4. The NLG system receives as input a *domain plan*, which specifies the (simulated) physical actions in the world that the IF should execute, and must compute appropriate *communicative plans* to make the IF execute those physical actions. Thus the perlocutionary effects that the NLG system needs to achieve are individual ac-

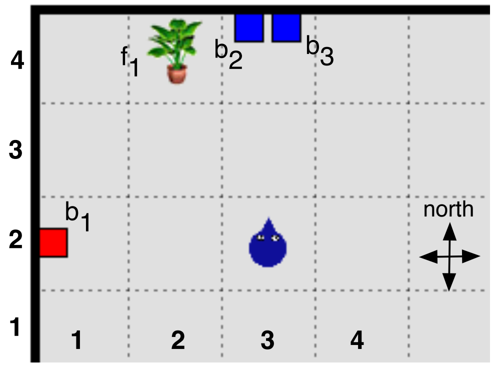

Figure 3: An example map for instruction giving.

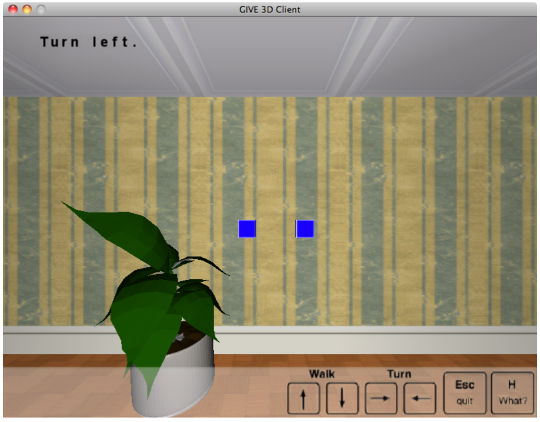

Figure 4: The IF's view of the scene in Fig. 3, as rendered by the GIVE client.

tions in the domain plan. In the example of Fig. 3, one action of the domain plan is $\mathsf{push}(b_1)$, i.e. the act of the IF pressing b_1. A sequence of communicative actions that has a good chance of achieving this is to utter "turn left and push the button".

SCRISP can compute such a communicative act sequence using planning, and can monitor the execution of this communicative plan. The average time it takes to compute and present a plan, on an original GIVE-1 evaluation world (as represented by a knowledge base of approx. 1500 facts and a grammar of approx. 30 lexicon entries), is about one second on a 3 GHz CPU. The plans are computed using the FF planner (Hoffmann and Nebel, 2001; Koller and Hoffmann, 2010). This shows that the approach to speech act planning we propose here can achieve real-time performance.

4.1 Situated CRISP

SCRISP assumes a TAG lexicon in which each elementary tree has been equipped with pragmatic preconditions and effects next to its syntactic and semantic ones (see Fig. 5). Each of these is a set of atoms over constants, free variables, and argument names such as obj, which encode the individuals in the domain to which the nodes of the elementary tree refer. These atoms determine the preconditions and effects of communicative actions.

For instance, the lexicon in Fig. 5 specifies that uttering "push X" has the perlocutionary effect that the IF presses X. It also says that we may only felicitously say "push X" if X is visible from the IF's current position and orientation. The position and orientation of the IF, and with them the currently visible objects, can be modified by first uttering "turn left". The two utterances can be chained together by sentence coordination ("and"). Finally, introducing the noun phrase "the button" as the object of "push" makes the sentence grammatically complete.

In order to generate such a sequence, SCRISP converts the lexicon and the perlocutionary goal that is to be achieved into a planning problem. It then runs an off-the-shelf planner to compute a plan, and decodes it into sentences that can be presented to the hearer. The operators of the planning problem for the example lexicon of Fig. 5 are shown in simplified form in Fig. 6, which can be seen as an extended and more explicit version of those in Fig. 2. We do not have the space here to explain the operators in full detail (see Garoufi and Koller (2010)). However, notice that they have both grammar-internal preconditions and effects (e.g., subst specifies open substitution nodes, ref connects syntax nodes to the semantic individuals to which they refer, and canadjoin indicates the possibility of an auxiliary tree adjoining the node) and perlocutionary ones. In particular, the **"push"** action has a perlocutionary effect $\mathsf{push}(x_1)$.

4.2 Planning and monitoring perlocutionary acts with SCRISP

Now let's see how SCRISP generates instructions that can achieve this perlocutionary effect.

First, we encode the state of the world as depicted in Fig. 3 in an initial state which contains, among others, the atoms $\mathsf{player\text{–}pos}(\mathsf{pos}_{3,2})$, $\mathsf{player\text{–}ori}(\mathsf{north})$, $\mathsf{next\text{–}ori\text{–}left}(\mathsf{north}, \mathsf{west})$, $\mathsf{visible}(\mathsf{pos}_{3,2}, \mathsf{west}, b_1)$, etc. As the goal for the planning problem, we take our perlocutionary goal, $\mathsf{push}(b_1)$, along with linguistic constraints including $\forall A \forall u.\neg\mathsf{subst}(A, u)$ (encoding syntactic completeness) and $\forall u \forall x.\neg\mathsf{distractor}(u, x)$

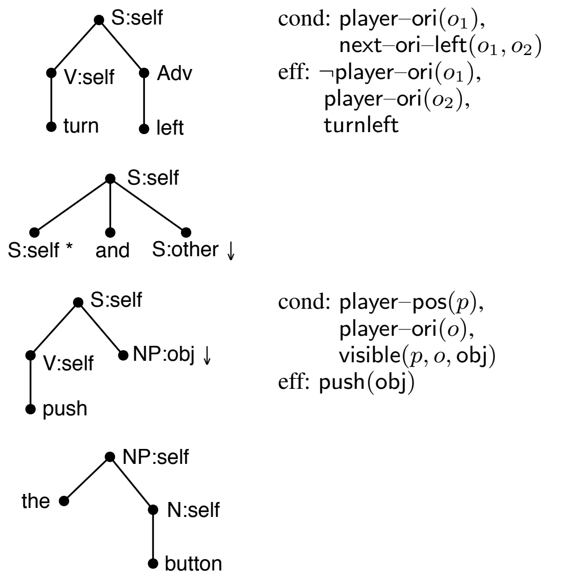

Figure 5: A simplified example of a SCRISP lexicon, focusing on pragmatic conditions and effects.

"turnleft"(u, x, o_1, o_2):
Precond: subst(S, u), ref(u, x), player–ori(o_1), next–ori–left(o_1, o_2), ...
Effect: ¬subst(S, u), ¬player–ori(o_1), player–ori(o_2), turnleft, ...

"and"(u, u_1, u_n, e_1, e_2):
Precond: canadjoin(S, u), ref(u, e_1), nextref(e_1, e_2), ...
Effect: subst(S, u_1), ref(u_1, e_2), ...

"push"$(u, u_1, u_n, x, x_1, p, o)$:
Precond: subst(S, u), ref(u, x), player–pos(p), player–ori(o), visible(p, o, x_1), ...
Effect: ¬subst(S, u), subst(NP, u_1), ref(u_1, x_1), $\forall y.(y \neq x_1 \wedge$ visible$(p, o, y) \rightarrow$ distractor$(u_1, y))$, push(x_1), canadjoin(S, u), ...

"the–button"(u, x):
Precond: subst(NP, u), ref(u, x), button(x)
Effect: $\forall y.(\neg$button$(y) \rightarrow \neg$distractor$(u, y))$, ¬subst(NP, u), ...

Figure 6: Simplified SCRISP planning operators for the lexicon in Fig. 5.

(encoding unique reference).

The planner can then apply the action **"turnleft"**(root, e, north, west) to the initial state. This action makes player–ori(west) true and subst(root, e) false. The new state does not contain any subst atoms, but we can continue the sentence by adjoining "and", i.e. by applying the action **"and"**(root, n_1, n_2, e, e_1). This produces a new atom subst(S, e_1), which satisfies one precondition of **"push"**(n_1, n_2, n_3, e_1, b_1, $pos_{3,2}$, west). Because **"turnleft"** changed the player orientation, the visible precondition of **"push"** is now satisfied too (unlike in the initial state). Applying the action **"push"** introduces the need to substitute a noun phrase for the object, which we can eliminate with an application of **"the button"**(n_2, b_1). We are thus brought into a goal state, in which the planner terminates.

The final state of this four-step plan contains the atom push(b_1), indicating that if everything goes as intended, the hearer will push b_1 upon hearing the instructions. Crucially, we were only able to compute the plan because we make the optimistic assumption that communicative acts have the intended perlocutionary effects: For instance, it is only because we assumed that uttering "turn left" would make the IF change their orientation in space that this action was able to establish the precondition of the "push" action. A REQUEST operator as in P&A, which only makes the IF *want* to turn left, would not have achieved the same.

Having uttered these instructions, SCRISP observes the IF's behavior in the virtual world to determine whether the intended effects actually come to pass. We achieve this through three simple submodules of the execution monitor. The "inactivity" submodule tracks whether the user has not moved or acted in a certain period of time, and resends the previous instruction when this happens. Fresh instructions from the user's current location are issued when the "distance" submodule observes that the user is moving away from the location at which they need to perform the next physical action. Finally, the "danger" submodule monitors whether the user comes close to a trap in the world, and warns them away from it. It is clear that these three modules only allow the system a very limited view into the hearer's mental state. Nevertheless, they already allow SCRISP to achieve competitive performance in the GIVE-1 task (Garoufi and Koller, 2010).

5 Conclusion

In this paper, we have proposed to model communicative actions as planning operators and their intended perlocutionary effects as effects of these operators; we further proposed that after uttering something, the speaker then observes the hearer's behavior to infer whether the utterance had the intended effect. This moves the main complexity of communication into the plan execution monitoring module, where it can be handled with as much effort as the application requires, while keeping the planning itself simple and fast.

We see the most interesting task for the future in working out some of the connections we sketched here in more detail, particularly a full model of the P&A approach and an extension to declarative utterances. In addition, it would be exciting to see how the notions of utility and uncertainty from POMDPs can be generalized in cases where a belief about the state is formed from observations using more than just a probability distribution, while retaining efficient planning.

Acknowledgments. We are indebted to Luciana Benotti, Oliver Lemon, Ron Petrick, Matt Purver, and David Schlangen for challenging and constructive discussions on the topic of this paper. We thank our reviewers for their insightful comments.

References

J. Austin. 1962. *How to do things with words*. Oxford University Press.

L. Benotti. 2009. Clarification potential of instructions. In *Proc. 10th SIGDIAL*.

M. Brenner and I. Kruijff-Korbayová. 2008. A continual multiagent planning approach to situated dialogue. In *Proc. LonDial*.

D. Byron, A. Koller, K. Striegnitz, J. Cassell, R. Dale, J. Moore, and J. Oberlander. 2009. Report on the First NLG Challenge on Generating Instructions in Virtual Environments. In *Proc. 12th ENLG*.

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

H. Clark. 1996. *Using Language*. Cambridge University Press, Cambridge, England.

M. Frampton and O. Lemon. 2009. Recent research advances in reinforcement learning in spoken dialogue systems. *Knowledge Engineering Review*, 24(4):375–408.

K. Garoufi and A. Koller. 2010. Automated planning for situated natural language generation. In *Proc. 48th ACL*.

J. Hoffmann and B. Nebel. 2001. The FF planning system: Fast plan generation through heuristic search. *J. Artificial Intelligence Research*, 14:253–302.

A. Joshi and Y. Schabes. 1997. Tree-Adjoining Grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69–123. Springer.

A. Koller and J. Hoffmann. 2010. Waking up a sleeping rabbit: On natural-language sentence generation with FF. In *Proc. 20th ICAPS*.

A. Koller and M. Stone. 2007. Sentence generation as planning. In *Proc. 45th ACL*.

J. Kvarnström, F. Heintz, and P. Doherty. 2008. A temporal logic-based planning and execution monitoring system. In *Proc. 18th ICAPS*.

C. Matheson, M. Poesio, and D. Traum. 2000. Modelling grounding and discourse obligations using update rules. In *Proc. 6th ANLP*.

D. Nau, M. Ghallab, and P. Traverso. 2004. *Automated Planning: Theory and Practice*. Morgan Kaufmann.

C. R. Perrault and J. Allen. 1980. A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6(3–4):167–182.

M. Poesio and H. Rieser. 2010. Completions, coordination, and alignment in dialogue. *Dialogue and Discourse*, 1(1).

M. Purver. 2006. CLARIE: Handling clarification requests in a dialogue system. *Research on Language and Computation*, 4(2-3):259–288.

H. Shintel and B. Keysar. 2009. Less is more: A minimalist account of joint action in communication. *Topics in Cognitive Science*, 1(2):260–273.

G. Skantze. 2007. *Error Handling in Spoken Dialogue Systems: Managing Uncertainty, Grounding and Miscommunication*. Ph.D. thesis, KTH, Stockholm, Sweden.

M. Steedman and R. Petrick. 2007. Planning dialog actions. In *Proc. 8th SIGdial*.

M. Stone, C. Doran, B. Webber, T. Bleam, and M. Palmer. 2003. Microplanning with communicative intentions: The SPUD system. *Computational Intelligence*, 19(4):311–381.

F. Teichteil-Koenigsbuch, G. Infantes, and U. Kuter. 2008. RFF: A robust, FF-based MDP planning algorithm for generating policies with low probability of failure. In *Proc. 6th IPC at ICAPS*.

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

L. Wardlow Lane and V. S. Ferreira. in press. Speaker-external versus speaker-internal forces on utterance form: Do cognitive demands override threats to referential success? *Journal of Experimental Psychology: Learning, Memory, and Cognition*.

R. Washington, K. Golden, and J. Bresina. 2000. Plan execution, monitoring, and adaptation for planetary rovers. *Electronic Transactions on Artificial Intelligence*, 4(A):3–21.