A Continual Multiagent Planning Approach to Situated Dialogue

Michael Brenner
Albert-Ludwigs-Universität
Freiburg, Germany
brenner@informatik.uni-freiburg.de

Ivana Kruijff-Korbayová
DFKI GmbH
Saarbrücken, Germany
ivana.kruijff@dfki.de

Abstract

Situated dialogue is usually tightly integrated with behavior planning, physical action and perception. This paper presents an algorithmic framework, Continual Collaborative Planning (CCP), for modeling this kind of integrated behavior and shows how CCP agents naturally blend physical and communicative actions. For experiments with conversational CCP agents we have developed MAPSIM, a software environment that can generate multiagent simulations from formal multiagent planning problems automatically. MAPSIM permits comparison of CCP-based dialogue strategies on a wide range of domains and problems without domain-specific programming. Despite their linguistic capabilities being limited MAPSIM agents can already engage in fairly realistic situated dialogues. Our ongoing work is taking this approach from simulation to real human-robot interaction.

(1) Anne: ”Please give me the coffee, R2D2.”
(2) R2D2: ”Okay.”
(3) R2D2: ”Where is the coffee, Anne?”
(4) Anne: ”The coffee is in the kitchen.”
(5) R2D2: ”Thanks, Anne.”
(6) R2D2: ”Please open the kitchen door, Anne.”
(7) Anne: ”Okay.”
(8) Anne opens the kitchen door.
(9) R2D2: ”Thanks for opening the kitchen door, Anne.”
(10) R2D2 moves to the kitchen.
(11) R2D2 takes the coffee.
(12) R2D2 moves to the living room.
(13) R2D2 brings Anne the coffee.
(14) Anne: ”Thanks for bringing me the coffee, R2D2.”

Figure 1: Mixed-initiative dialogue between two artificial agents in MAPSIM (*Household* domain).

1 Introduction

When several agents are situated in a common environment they usually interact physically as well as verbally. Verbal interactions in such environments, i. e. situated dialogues, both *reflect* the past and *influence* the future physical behavior of the agents. As a result, situated dialogue is continually *interleaved* with behavior planning, physical action and sensing. But when and why do agents switch between these rather distinct activities? In particular, how is dialogue triggered by physical events and how, in turn, does it constrain them?

In this paper, we approach these questions from the perspective of *multiagent planning*. Specifically, we describe situated dialogue as Distributed Continual Planning (DCP) (DesJardins et al., 1999), i. e. as a process that integrates planning, acting and perception with communication. We present a novel DCP algorithm called Continual Collaborative Planning (CCP) and show how it can be used for situated dialogue modeling. Interestingly, the role of communication in CCP is twofold: A dialogue move can be part of the collaborative *planning* process; however, it is also the *execution* of a communicative action and, just like the execution of a physical action, it changes the “world” in ways that may lead to previously unforeseen changes in plans and, consequently, additional interactions. Since goals and plans of agents are continually revised, CCP models very dynamic interactions that naturally include mixed-initiative subdialogues and interleaved physical and communicative actions.

Approaches to situated dialogue can only be evaluated in environments where agents are actually situated, i. e. where they can not only communicate, but also perceive and act. Because we want to evaluate CCP (and related approaches) over a wide range of application domains we have developed MAPSIM, a simulation environment that turns formal multiagent planning problems into multiagent simulations. Crucially, MAPSIM creates the simulation as well as a domain-specific lexicon for natural-language dialogue *automatically* when analyzing the planning domain. Since no domain-specific *programming* is needed MAPSIM can be used to quickly evaluate dialogue strategies on a wide range of domains and problems.

The paper is structured as follows: We first introduce our multiagent planning formalism and discuss its suitability for dialogue planning. Then we present the CCP algorithm. In the subsequent sections we describe MAPSIM and analyze CCP dialogues in several domains. In the final sections we discuss related work and indicate our ongoing efforts.

2 Multiagent Planning Formalism

Planning in dynamic multiagent environments means reasoning about the environment, about (mutual) beliefs, perceptual capabilities and the possible physical and communicative actions of oneself and of others. All of these elements can be modeled in the multiagent planning language MAPL (Brenner, 2008). In this section we introduce MAPL informally and discuss its suitability for dialogue planning; formal definitions can be found in (Brenner, 2008).

MAPL is a multiagent variant of PDDL (Planning Domain Definition Language), the de facto standard language for classical planning (Fox and Long, 2003). One important extension in MAPL is the use of multi-valued state variables (MVSVs) instead of propositions. For example, a state variable *color(ball)* would have exactly one of its possible *domain* values *red, yellow,* or *blue* compared to the three semantically unrelated propositions *(color ball red), (color ball yellow), (color ball blue)*, all or none of which could be true in a given STRIPS state. MVSVs have successfully been used in classical planning in recent years (Helmert, 2006), but they also provide distinctive benefits when used for dialogue planning.

Firstly, we can use MVSVs to model *knowledge* and *ignorance* of agents: if no value is known for a state variable it is *unknown* (contrast this with the closed world assumption of classical planning: what is not known to be true is *false*). This concept can also be extended to beliefs about other agents' beliefs and mutual beliefs which are modeled by so-called **belief state variables**. Secondly, *wh-questions* can be modeled as queries about MVSVs in our model (see below). Thirdly, algorithms for generating referring expressions, such as the full brevity algorithm of (Dale, 1992), can be directly implemented using a MVSV representation.

(1) Bill goes home.
(2) Bill: ”Please bake the pizza, Oven.”
(3) Oven: ”Okay.”
(4) Oven bakes the pizza.
(5) Oven: ”I have finished baking the pizza, Bill.”
(6) Bill: ”Thanks for baking the pizza, Oven.”
(7) Bill: ”Please bring me the pizza, R2D2.”
(8) R2D2: ”Okay.”
(9) R2D2 brings Bill the pizza.
(10) Bill: ”Thanks for bringing me the pizza, R2D2.”
(11) Bill eats the pizza.

Figure 2: Dialogue between three artificial agents in MAPSIM (*Pizza* domain).

MAPL **actions** are similar to those of PDDL. In MAPL, every action has a **controlling agent** who executes the action and controls when it is done. Agents are fully autonomous when executing actions, i. e. there is no external synchronization or scheduling component. As a consequence an action will only be executed if, in addition to its preconditions being satisfied, the controlling agent *knows* that they hold. Implicitly, all MAPL actions are extended with such **knowledge preconditions** (cf. also (Lochbaum, 1998)). Similarly, there are implicit **commitment preconditions**, intuitively describing the fact that an agent will only execute actions if he has agreed to do so.

A MAPL domain can define three different ways to affect the beliefs of agents (necessary, e. g., in order to satisfy knowledge preconditions): sensing, copresence (joint sensing), and communication. All three are MAPL actions that have knowledge effects. **Sensor models** describe the circumstances in which the current value of a state variable can be perceived. **Copresence models** are multiagent sensor models that induce mutual belief about the perceived state variable (Clark and Marshall, 1981).

Informally, agents are copresent when they are in a common situation where they can not only perceive the same things but also each other. Individual and joint sensing are important for dialogue because they help *avoiding* it: an agent does not need to ask for what he senses himself, and he does not need to verbalize what he assumes to be perceived by the other agents as well. Communicative acts currently come in two forms: (i) **Declarative statements** are actions that, similarly to sensory actions, can change the belief state of another agent in specific circumstances. Line 5 of Fig. 2 shows an example of an agent explicitly providing another one with factual information. (ii) **Questions**, **commands** and **acknowledgments** are not explicitly modeled in a MAPL domain, but generated during CCP (as discussed in Sect. 3). These communicative acts potentially cover a broad range of speech acts, whose differentiation requires further refinement of the corresponding preconditions and effects.

MAPL **goals** correspond to PDDL goal formulae. However, MAPL has two additional goal-like constructs: **Temporary subgoals** (TSGs) are mandatory, but not necessarily permanent goals, i. e. they must be satisfied by the plan at some point, but may be violated in the final state. **Assertions**, on the other hand, describe *optional* "landmarks", i. e. TSGs that may helpful in achieving specific effects in later phases of the continual planning processes, which cannot be fully planned for yet because of missing information (Brenner and Nebel, 2006; Brenner, 2008). For example, the MAPL domain used to create the simulation in Fig. 1 contains an assertion stating that, informally speaking, to get something one must first know where it is.

MAPL plans differ from PDDL plans in being only *partially ordered*. This is inevitable since we assume that there is no central executive which could guarantee a totally ordered execution. We use the term **asynchronous plans** since MAPL plans also allow for *concurrent* occurrence of actions. Fig. 3 shows an example. An asynchronous plan that guarantees that the implied knowledge preconditions will be satisfied during execution (e. g. by explicitly naming the perceptions to be made and speech acts to be used) is called **self-synchronizing plan** because it "explains" how the agents can coordinate their behavior during execution.

It is often impossible for a group of situated agents to jointly commit to a self-synchronizing plan prior to beginning its execution. As an example, line 1 of Fig. 2 shows how an agent must start executing its individual multiagent plan (i. e. a plan for a group of agents but to which no other agent has committed yet) in order to even get the chance to negotiate the plan with the others: In this scenario, Bill must physically move first because he can only communicate with his household appliances "at home". This is modeled explicitly in the MAPL domain by means of a so-called **communication precondition** that the planner has to satisfy if agents should engage in dialogue. In future work, we will also use communication preconditions to model dialogue-specific requirements like *attention* (Grosz and Sidner, 1986) and *engagement* (Sidner et al., 2005).

3 Continual Collaborative Planning

Continual Collaborative Planning (CCP) agents switch between planning, partial plan execution, monitoring, plan adaptation and communication. Alg. 1 gives a high-level description of the CCP algorithm. Since the current state of the algorithm not only depends on what the agent has been doing, but also on the messages received from others, CCP is specified as a Distributed Algorithm (Lynch, 1996).

Algorithm 1 CCP AGENT(S, G)

```
$P = \emptyset$
**Received** no message:
  if $S$ satisfies $G$ do
    return "goal reached"
  else
    $P =$ MONITORINGANDREPLANNING$(S, G, P)$
  if $P = \emptyset$ then
    return "cannot achieve goal $G$"
  else
    $(S, P) =$ EXECUTIONANDSTATEESTIMATION$(S, P)$
**Received** (tell-val $vx$) **from** agent $a$:
  add $v \doteq x$ to $S$
**Received** request$(e)$ **from** agent $a$:
  $sg =$ TRANSLATEREQUESTTOGOAL$(e)$
  $P =$ MONITORINGANDREPLANNING$(S, G \cup sg, \emptyset)$
  if $P = \emptyset$ then
    **send** "cannot execute request $e$" **to** $a$
  else
    add $sg$ to $G$ as temporary subgoal
```

We will first discuss the base case when no communication has taken place yet, i. e. the CCP agent has neither sent nor received any messages yet.

Roughly speaking, the agent alternates between (re-)planning and acting in this case. The two phases are detailed in Algs. 2 and 3. Alg. 2 shows how a new planning phase is triggered: the agent *monitors* whether his current plan has become invalid due to unexpected (external) events or changes in his goals. If this is the case, the agent adapts its plan by replanning those parts that are no longer executable. In order to exploit the power of state-of-the-art planning systems, Alg. 2 uses an unspecified classical planner PLANNER to (re-)plan for the obsolete or missing parts of the old plan. The details of this process are irrelevant for the purpose of this paper; it results in an asynchronous plan that specifies actions for (possibly) several agents and the causal and temporal relation between them that is necessary to achieve the planning agent's goal.

Algorithm 2 MONITORINGANDREPLANNING(S, G, P)

```
if $res(S,P) \not\supseteq G$
   REMOVEOBSOLETESUFFIXGRAPH(P)
   $P' = \text{PLANNER}(A, res(S,P), G)$
   $P = \text{CONCAT}(P, P')$
return P
```

Fig. 3 shows such an asynchronous plan for the *pizza* scenario of Fig. 2, created with Alg. 2. Note that this plan contains special *negotiation* actions; they will be the triggers for task-orientated subdialogues in a later phase of CCP. The planning algorithm enforces such negotiation actions to be included in a plan whenever this plan includes actions or subplans to be executed not by the planning agent, but by another agent who is not yet committed to this plan. Thus CCP ensures that a (sub-)dialogue will take place that either secures the other agent's commitment or triggers replanning. Note how, in turn, the need for negotiation has forced the planner to include a physical action (Bill's moving home) into the plan in order to satisfy the abovementioned communication precondition.

As soon as a CCP agent has found (or repaired) a valid plan it enters the execution phase, described in Alg. 3. First, an action, e, on the first level of the plan, i. e. one whose preconditions are satisfied in the current state, is chosen non-deterministically. If the action is controlled by the CCP agent himself, it is executed. If not, the planning agent tries to determine whether the action was executed by its controlling agent. In both cases, the CCP agent will try to update its knowledge about the world state based on the expected effects and the actual perceptions made (FUSE function).

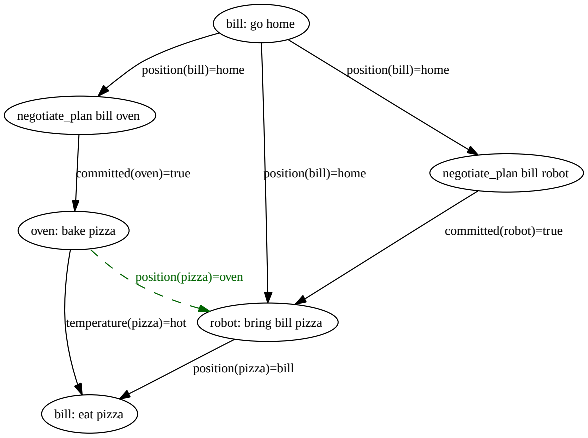

Figure 3: Bill's plan for getting pizza.

Algorithm 3 EXECUTIONANDSTATEESTIMATION(S, P)

```
$e$ = **choose** a first-level event from $P$
if $e$ ='negotiate_plan with agent $a$'
   $r$ = SELECTBESTREQUEST$(P, a)$
   **send** request($r$) **to** $a$
else if $agt(e) = \text{self}$ then
   EXECUTE$(e)$
$S' = app(S, e)$
$exp = \text{EXPECTEDPERCEPTIONS}(S', A^s)$
$perc = \text{GETSENSORDATA}()$
if $perc \supseteq exp$ or $exp = \emptyset$ then
   remove $e$ from $P$
$S = \text{FUSE}(S', perc)$
return (S,P)
```

The most interesting case for this paper is the one where the action chosen to be executed is *negotiate_plan*. This means that a CCP agent (A) is now in a situation where he is able communicate with another agent (B) that he intends to collaborate with, i. e. A's plan includes at least one action controlled by B, that B has not yet committed to. In this case, A will send a *request* to B. However, if a plan contains several actions by another agent, i. e. a whole subplan, it is often best not to request execution of the actions individually, but to ask for the end result or, respectively, the final action in the subplan. In other situations it may even be reasonable to request the achievement of subplans that include more than one agent. CCP does not stipulate a specific implementation of SELECTBESTREQUEST; we will describe one version in Sect. 5.

When an agent receives a request, Alg. 1 enters into a new phase. First the request is translated into a goal formula (Brenner, 2007) and tested for achievability. This is a simplification for the sake of processing efficiency, based on the assumption that what matters to the other agent is not the exact action, but its *result*, i. e. the achievement of a goal or precondition for a subsequent action by the requesting agent. Additionally, constraints on the arguments of requests (e. g. intended referents of natural language expressions) are easier to model as goal constraints than as actions (Brenner, 2007). Accepted requests are adopted as *temporary subgoals* (TSGs). This means that they must only be achieved temporarily and do not have to hold any more when the agent's main goal is achieved.

The adoption of requests as TSGs is a crucial element of CCP that, to the best of our knowledge, has not been described in other Continual Planning approaches: in addition to repeatedly revising their beliefs about the world, CCP agents also perform continual *goal revision*. In the simplest case, this leads to information-seeking *subdialogues*, as in lines 3–5 of Fig. 1. But newly adopted TSGs also explain why agents engage in subdialogues that mix communicative and physical actions (as in lines 6–9 of the same example).

4 MAPSIM

Continual Planning approaches can only be tested in environments where agents can actually execute, monitor and revise their plans. This is all the more true for our DCP approach to situated dialogue where agents need to interact collaboratively. To this end we have developed MAPSIM, a software environment that automatically generates multiagent simulations from MAPL domains. In other words, MAPSIM interprets the planning domain as an *executable model* of the environment. Thus, MAPSIM allows designers of DCP algorithms to evaluate their approaches on various domains with minimal effort. In this section, we give an overview of MAPSIM and describe how it is used for generating situated dialogues.

(1) Anne: request R2D2 'give R2D2 coffee Anne'.
(2) R2D2: accept_request 'give R2D2 coffee Anne'.
(3) R2D2: request Anne 'tell_val Anne R2D2 pos(coffee)'.
(4) Anne: execute 'tell_val Anne R2D2 pos(coffee)'.
(5) R2D2: ack_achieved 'tell_val Anne R2D2 pos(coffee)'.
(6) ...

Figure 4: The MAPSIM run of Fig. 1 without NL verbalization.

The MAPL domain description is parsed, analyzed and turned into perception, action, and communication models for CCP agents. During the simulation, MAPSIM maintains and updates the global world state and it uses the sensor models to compute individual and joint perceptions of agents. The agents interact with the simulation by sending *commands* in the form of plain MAPL actions. The simulator then executes the action, i. e. it checks the preconditions and applies effects as specified in the MAPL domain. If the controlling agent of a command is not identical to the agent who sent it to the simulator this is interpreted as a *request* which is not directly executed but passed on to the corresponding agent. MAPSIM also accepts specific commands for acknowledging subgoal acceptance and subgoal achievement.

Agents do not need to know anything about how their actions are executed. Thus, they can implement arbitrary deliberative or reactive methods to determine their behaviour and their reactions to requests. We believe that this can make MAPSIM a valuable evaluation tool even when the DCP and dialogue strategies investigated differ significantly from CCP. For example, the simulated dialogues produced by MAPSIM using different strategies could be evaluated using objective measures such as task success or dialogue costs from the PARADISE framework (Walker et al., 1997).

MAPSIM and the CCP agents described in this paper have been implemented in Python, using state-of-the-art planning technology as subsolvers.The generic planner currently used for CCP is a slightly modified version of Axioms-FF (Thiebaux et al., 2003). This enables MAPSIM to generate dialogues between artificial agents very fast.[1]

The main goal of this work is to show how a generic multiagent planning algorithm can be used

[1] For example, during the dialogue of Fig. 1 CCP called the PLANNER function 13 times with a total planning time less than half a second on a 1.6 GHz AMD Athlon.

for situated dialogue in natural language, e. g. in human-robot interaction (HRI). It is therefore important to investigate the efforts needed for mapping between the MAPL-based representation used by CCP agents and natural language. To that end MAPSIM includes a verbalization module, called the *reporter* agent. The reporter observes all physical and communicative events in the simulation and verbalizes them in English. All dialogues shown throughout the paper are unaltered outputs of the reporter. Fig. 4 shows the beginning of the MAPSIM run of Fig. 1 with reporting turned off.

The reporter is a simple template engine that first determines an appropriate pattern depending on the command type currently executed, then recursively replaces templates with concrete arguments until a template-free sentence is generated. Base values for arguments are generated directly from analyzing the MAPL domain. For example, operator names are assumed to directly correspond to verbs. Standard templates can be overridden by domain-specific patterns. However, the only general need for this we have experienced is the definition of verb complements. For example, the *Household* domain defines the complement of *"move"* as *"to the $arg0"* where *$arg0* is instantiated with the first argument of the respective command. Apart from verb complements, the *only* domain-specific template that was necessary to generate Fig. 1 states that the *interrogative* (wh-word) for state variables *position(x)* is *"where"*. While, compared to "real" natural-language processing systems, this is a simplistic approach with obvious limits, the minimal effort needed to achieve fairly realistic surface generation is noteworthy and will be exploited in future work.

5 Detailed Analysis of MAPSIM runs

This section provides a detailed analysis of several CCP runs in MAPSIM. It is important to realize that none of the sample runs in this paper is based on just one multiagent plan, but on a *series* of plans, devised, partly executed and revised several times according to Alg. 1.

All dialogue in CCP is driven by individual desires, i. e. agents engage in dialogue only if they need help in satisfying their individual goals. In the *household* scenario (Fig. 1) the necessity for collaboration stems from the fact that only R2D2 can move to the kitchen to get coffee, but only Anne can open the kitchen door. In the *pizza* scenario (Fig. 2) Bill needs the collaboration of his intelligent household appliances to be able to eat pizza.

As we have already seen, Bill's initial individual planning process resulted in the multiagent plan shown in Fig. 3. In this situation, Alg. 3 can only choose a *physical* action for Bill to execute, namely *go home*. Note that only the *execution* of this action enables Bill to subsequently communicate with Oven and R2D2 at all. Thus, Bill's problem can *only* be solved by a DCP approach that is able to interleave planning, physical execution and dialogue whenever necessary.

When at home, Bill can (and must) negotiate his plan with the two other agents he wants to involve. Alg. 3 uses the black-box function SELECTBESTREQUEST to determine an appropriate *temporary subgoal* whose achievement will be requested from another agent.

The currently used REQUESTSUBPLAN strategy works as follows: the agent first determines the longest possible subplan involving only one agent, then chooses an action on the *final* level of this plan as the best request. In other words, a CCP agent posing a request does not specify details about how he wants a temporary subgoal to be achieved. In the *household* example, Anne thus does not request R2D2 to go to the kitchen and get the coffee there, but just requests the last action in her multiagent plan, namely the robot giving her the coffee.

Admittedly, the straightforward verbalization of this action by the reporting agent using the verb "give" results in an unnatural dialogue contribution. Anne's request would be more appropriately formulated using "bring", "fetch" or "get", which unlike "give" do not presuppose that R2D2 already has the coffee. This reveals the need to take more of the subplan into account when verbalizing the request, a topic we are taking up in further work.

Anne thus leaves it to R2D2 to find its own solution to achieve the TSG. This "lazy" strategy mirrors on the dialogue level the idea of the Continual Planning approach, where an individual CCP agent postpones the solution of some subproblems to later phases in the planning-execution-monitoring cycle.

R2D2's previous plan was to do nothing (which satisfied his "empty" goal). After adopting the new

TSG, this plan is no longer valid and Alg. 2 triggers a new planning phase. Since R2D2 does not know where the coffee is this plan includes an appropriate information-gathering action and postpones detailed planning for getting the coffee until this information is known (by means of an *assertion* (Brenner and Nebel, 2006)). In our example, the information-gathering action is a request for information to Anne (cf. line 3 of Fig. 1). This request is generated as follows: R2D2's plan contains the action *(tell-val Anne R2D2 pos coffee)*, i. e. a speech act to be executed by Anne. According to Alg. 3, this action to be performed by another agent (from R2D2's point of view) must be requested first. Line 3 of Fig. 4 shows this request when executed without the *reporter* agent. Its verbalization results in R2D2 asking the question "Where is the coffee, Anne?".

MAPSIM provides several options for generating *acknowledgments*. In the dialogues presented here, agents provide acknowledgments when they accept a request (e. g. lines 2 and 7 of Fig. 1) and also when they realize that a request of theirs has been satisfied (e. g. lines 5, 9 and 14 of Fig. 1). Note that answers to questions are acknowledged only briefly, but satisfaction of physical subgoals is acknowledged more explicitly. While this is not necessarily the best acknowledgment strategy, it shows how the multiagent plan and the CCP history provide *context* as well as *focus* (Grosz and Sidner, 1986)) that can easily be exploited for surface generation and, in the future, also for interpretation (cf. Sect. 7).

For lack of space, we cannot discuss the rest of the dialogues in detail. Note, however, how the agents switch seamlessly between communicative and physical actions whenever necessary. Not shown by the reports are the *perceptions* made by the agents during the runs. Nevertheless, they are important for the dialogue, too, since agents also reason about their mutual perceptions and thus can avoid unnecessary verbalizations.

6 Related Work

This work shares many characteristics with previous approaches modeling dialogue as *collaborative planning*, most notably those based on the SharedPlans formalism (Grosz and Sidner, 1990; Grosz and Kraus, 1996; Lochbaum, 1998). SharedPlans use much more elaborate mental attitudes than MAPL and CCP, mainly because CCP agents rely on them only *implicitly* – until a violation of their assumptions prompts plan adaptation or new dialogue. In this respect, the commitments made by CCP agents more resemble the *joint persistent goals* of (Cohen and Levesque, 1991). Nevertheless, SharedPlans can be regarded as a "specification" of the kind of collaboration CCP intends to model *computationally*.

(Blaylock et al., 2003) note that SharedPlans do not model the cooperation that occurs during *execution*. They propose a high-level model of dialogue as *collaborative problem solving* (CPS); our approach can be regarded as an instantiation of that model. However, our work complements both SharedPlans and CPS by describing *how* knowledge preconditions prompt active sensing and information gathering during situated dialogue.

Distributed Continual Planning has been advocated as a new paradigm for planning in dynamic multiagent environment (DesJardins et al., 1999). To the best of our knowledge, ours is the first principled attempt to apply DCP to dialogue planning and also the first DCP approach describing deliberative *goal revision* as part of a DCP algorithm.

Collagen (Rich et al., 2001) is a system for building collaborative interface agents that is based on (Grosz and Sidner, 1986; Grosz and Sidner, 1990), which is domain-independent and has been used for various applications. Collagen's methods for representing the discourse state and doing plan recognition are much more sophisticated than CCP currently. However, Collagen does not (yet) include a first-principles planner, but relies on plan libraries and domain-specific code plug-ins (Rich and Sidner, 2007). It would be interesting to investigate whether CCP can be integrated with Collagen.

Similarly, the most prominent representative of the information-state-update approach to dialogue modeling, GoDiS (Traum and Larsson, 2003), has complementary rather than competing main strengths: GoDiS has a more elaborate repertoire of dialogue moves and can produce more sophisticated dialogue behavior than CCP and MAPSIM, but it uses static plans, and it is not clear how it would combine communication with physical action.

7 Conclusion and Outlook

We have presented a new algorithmic framework in which situated dialogue is modeled as Continual Collaborative Planning (CCP). We have shown how mixed-initiative dialogue that interleaves physical actions, sensing, and communication between agents occurs naturally during CCP. As a practical contribution, we have developed MAPSIM, a software tool that automatically generates multiagent simulations from formal planning domains, thus permitting the evaluation of CCP and other dialogue strategies on a wide range of applications.

The questions raised in the introduction about when and why agents switch between planning, acting, and execution have, intuitively, been answered as follows by CCP: Agents (re)start planning as soon as their plan becomes obsolete, possibly not because the world, but because their *goals* changed. They act whenever they have a valid plan containing executable physical actions. And they engage in dialogue whenever they want others to share subgoals or are requested to do this themselves. Since situated communication may have (physical) preconditions that must be satisfied first (e. g. being in the same room, having the other agent's attention/engagement, etc.) CCP explains how the need for dialogue may also trigger additional planning and acting.

From simulation to human-robot interaction The work presented in this paper provides a starting point for developing agents, e. g. robots, that can engage in situated dialogue with humans. Indeed, we are currently implementing CCP on a robotic system in the CoSy project. To that end, we are extending our approach in the following respects: (1) To allow for imperfect communication, we need to improve the handling of acknowledgments to include positive as well as negative feedback and clarifications. (2) To support the full range of plan-negotiation between dialogue participants, we need to allow agents to reject requests and accept rejections from others. This will enable us to handle situations with, e. g., conflicting goals, discrepancies in beliefs and execution failures.

Doing this amounts to refining and extending the repertoire of speech acts. Since the planning technology underlying CCP is known to scale very well (Thiebaux et al., 2003), we expect our dialogue approach to also scale up well to a larger repertoire of speech acts, more complex interactions and higher numbers of interacting parties.

We are also investigating how to better expose the purpose that an individual dialogue move serves in achieving an agent's overall goals, e. g. by deriving an explicit *dialogue plan* during CCP. Such a plan, in combination with the current state of the CCP process, will provide rich context information to the linguistic components of our robot, e. g. for the task of utterance interpretation and contextually appropriate surface generation.

Acknowledgments

This work has been supported by the EU in the Integrated Project "CoSy" (FP6-004250).

References

N. Blaylock, J. Allen, and G. Ferguson. 2003. Managing communicative intentions with collaborative problem solving. In *Current and New Directions in Dialogue*. Kluwer.

M. Brenner and B. Nebel. 2006. Continual planning and acting in dynamic multiagent environments. In *Proc. PCAR*, Perth, Australia.

M. Brenner. 2007. Situation-aware interpretation, planning and execution of user commands by autonomous robots. In *Proc. IEEE RO-MAN 2007*.

M. Brenner. 2008. The multiagent planning language MAPL. Technical report, Albert-Ludwigs-Universität, Institut für Informatik, Freiburg, Germany.

H. H. Clark and C. R. Marshall. 1981. Definite reference and mutual knowledge. In *Elements of discourse understanding*. Cambridge University Press.

P. Cohen and H. Levesque. 1991. Teamwork. *Noûs*, 25(4).

R. Dale. 1992. *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. MIT Press, Cambridge, MA.

M. DesJardins, E. Durfee, Jr. C. Ortiz, and M. Wolverton. 1999. A survey of research in distributed, continual planning. *The AI Magazine*.

M. Fox and D. Long. 2003. PDDL 2.1: an extension to PDDL for expressing temporal planning domains. *JAIR*.

B. J. Grosz and Sarit Kraus. 1996. Collaborative plans for complex group action. *Artificial Intelligence*, 86.

B. J. Grosz and C. L. Sidner. 1986. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3).

B. Grosz and C. Sidner. 1990. Plans for discourse. In *Intentions in Communication*. MIT Press.

M. Helmert. 2006. The Fast Downward planning system. *JAIR*, 26:191–246.

K. E. Lochbaum. 1998. A collaborative planning model of intentional structure. *Computational Linguistics*.

N. Lynch. 1996. *Distributed Algorithms*. Morgan Kaufmann, San Francisco, CA.

C. Rich and C. L. Sidner. 2007. Generating, recognizing and communicating intentions in human-computer collaboration. In *AAAI Spring Symposium on Intentions in Intelligent Systems, Stanford, CA*.

C. Rich, C. L. Sidner, and Neal Lesh. 2001. Collagen: applying collaborative discourse theory to human-computer interaction. *The AI Magazine*, 22(4).

C. L. Sidner, C. Lee, C. Kidd, N. Lesh, and C. Rich. 2005. Explorations in engagement for humans and robots. *AIJ*.

S. Thiebaux, J. Hoffmann, and B. Nebel. 2003. In defense of axioms in PDDL. In *Proc. IJCAI*.

D. Traum and S. Larsson. 2003. The information state approach to dialogue management. In *Current and New Directions in Discourse and Dialogue*. Kluwer.

M. Walker, D. Litman, C. Kamm, and A. Abella. 1997. Paradise: A framework for evaluating spoken dialogue agents. In *Proc. ACL-97*.