Integrating Joint Intention Theory, Belief Reasoning, and Communicative Action for Generating Team-Oriented Dialogue

Rajah Annamalai Subramanian^{1,2}, †Sanjeev Kumar^{2}, Philip Cohen^{1}

[1]Natural Interaction Systems, Seattle, USA
[2]Oregon Health and Sciences University, Portland, USA

rajah@ csee.ogi.edu, skumar@csee.ogi.edu, Phil.Cohen@naturalinteraction.com

Abstract

The goal of this research is to develop an architecture that can guide an agent during collaborative teamwork. The architecture should generate communication and dialogue during the performance of collaborative multi-agent tasks as a byproduct of the agent's rationally pursuing its intentions. This paper describes how a joint intention interpreter that is integrated with a reasoner over beliefs and communicative acts can form the core of a dialogue engine. As an interpreter of joint actions, the architecture enables agent programmers to describe a domain declaratively, specifying an agent's individual and joint intentions, plans and actions at a high level. The interpreter attempts to fulfill the agent's individual and joint intentions, subject to its beliefs and mutual beliefs. As a consequence, the system engages in dialogue through the planning and execution of communicative acts necessary to attain the collaborative task at hand. The dialogue engine is general enough to be applicable to both agent-agent and agent-human teams. The system has been implemented in a combination of Java and Prolog, and can be shown to obey the predictions of joint intention.

1. Introduction

Our motivation for building the STAPLE system is to build systems that exhibit collaborative teamwork. Instead of explicitly programming team behavior, collaboration arises in virtue of our embedding the foundational concepts of joint action/intention into a belief-desire-intention (BDI) architecture that is equipped with a repertoire of communicative actions. Thus an agent built around this architecture will engage in dialogues as a consequence of reasoning about joint action, intention, and belief.

One popular approach to the building of intelligent agents is via the Belief-Desire-Intention (BDI) architecture (Rao and Georgeff 1991). This architecture models an agent's acting based on its intentions and commitments, subject to its beliefs. Numerous implementations of this model have been developed, including SPARK (Morley and Myers 2004), the Procedural Reasoning System (PRS) (Georgeff and Lansky 1987), etc. Although these models point out the need for actions that model communication, little research into the classical BDI architecture has been directed at the integration of communication, collaboration, and action.

On the other hand previous research guided by the plan-based theory of communication (Cohen and Perrault 1979 Allen and Perrault 1980; Cohen and Levesque 1991) has shown that communication can be generated through the planning and executing of speech acts. This approach has motivated both research on dialogue systems (Bretier and Sadek 1996) and on inter-agent communication languages, such as FIPA (FIPA 2000) and KQML (Labrou and Finin 1997). In particular, ARCOL demonstrated that a rational agent implemented as a modal logic theorem-prover could in fact participate in dialogues, including spoken language dialogues conducted over the telephone network. However, a major drawback of the formal semantics specified for FIPA and KQML is the lack of motivation for when to communicate, what to communicate, how to handle failure, or more generally, how to conduct dialogue (beyond fixed protocols). For example, suppose that agent x request agent y to do an action to which agent y agrees. However, before y can do that task, agent x decides that it no longer wants it done. There is nothing in the plan-based theories of communication, or in the semantics underlying FIPA, that causes agent x to inform this fact to agent y.

The BDI approach has been extended to model agents to collaborate and communicate with others via concepts of shared plans (Grosz and Sidner 1990) and joint intentions (JI) (Levesque et al. 1990). JI theory specifies agents' mental states and has been integrated with models of communicative acts to build multi-agent systems (Jennings 1995; Tambe 1997a), that generate collaborative task-oriented behavior. In the example above, JI theory would require agent x to inform agent y that it has changed its mind. This paper shows that a JI interpreter should be able to serve as the basis for a collaborative dialog engine.

2. The STAPLE Joint Action Interpreter

The STAPLE engine has three main components, i) The JI interpreter, ii) the communicative acts generator, and iii) the belief reasoner. Here we briefly describe each of these components along with their integration.

† This author is currently at Cisco Systems Inc.

2.1 Joint Intention Theory

Similar to prior work by Grosz and Sidner (Grosz and Sidner 1990), Joint Intention theory (Levesque et al. 1990) stipulates what it means for agents to execute actions as a team. This same theory is used to specify formal semantics of communicative acts as an extension of standard speech act theory (Kumar et al. 2002b).

JI theory is expressed in a modal language with the usual connectives of a first order logic with equality, along with operators for propositional attitudes and event sequences. The primitive mental states include an agent's beliefs and goals. Temporal operators include $\Diamond p$ (eventually p) and $\Box$ p (always p). An action expression consists of action sequences, non-deterministic choices, concurrent actions, indefinite repetitions, and p? test actions. Constructs for conditional and iterative actions can easily be built. Details of this modal language and its model theory can be found in (Cohen and Levesque 1990).

The notion of an agent's commitment to achieving some state in the world is expressed as a persistent goal or PGOAL (Cohen and Levesque 1990). An agent x has a persistent goal (PGOAL x p q) if x wants p to become true and cannot give up the goal until it believes that p is accomplished, or is impossible, or is irrelevant (i.e. the relativizing condition q is untrue). An intention to do an action is defined as a persistent goal in which the agent is committed to performing the action believing throughout that it is doing that action. A persistent weak achievement goal or PWAG represents the one-way commitment of one agent directed towards another and it is used to define the semantics of various communicative acts (Section 2.2). The notion of teamwork is characterized by joint commitment (also known as joint persistent goal or JPG). The definition of JPG states that the agents mutually believe they have the appropriate goal, and that they mutually believe a PWAG to achieve it persists until the agents mutually believe that the goal has either been achieved, impossible, or irrelevant. In the logic, it can be shown that because of this joint commitment, agents have individual commitments that lead to individual intentions to do their parts relative to the overarching joint commitment. Joint commitment further stipulates that agents bound together by JPG are committed to establishing these mutual beliefs in the event of private belief about any of those three conditions (Section 2.3).

It has been shown in (Kumar et al. 2002b) that mutual belief in each other's PWAG towards the other to achieve a goal p is sufficient to establish a joint commitment to achieve p provided that there is mutual belief that p has not already been achieved. These mutual beliefs, and hence, the JPG can be established by performing appropriate communicative acts as discussed in the next section.

2.2 Interpreting Joint Intention Theory

STAPLE is a multi-stack BDI interpreter with built-in support for PGOAL (commitment), intention, and PWAG (See Figure 1). Using these foundational concepts, STAPLE can model joint commitment and joint intention. During the execution of an individual or joint action, an *Intention-Commitment (IC) stack* is built in a bottom-up manner, with the bottom element containing the original commitment and the other commitments or subgoal used to achieve the original commitment layered above it. When a PGOAL is being adopted or subgoaled, the interpreter first checks for its achievement, impossibility, and irrelevance by executing the belief reasoner in a Prolog engine over the agent's belief base. For instance, to check for impossibility, the belief reasoner is invoked to check if $\Box\neg p$ can be concluded from the belief base (this reasoning will make use of any domain dependent applicable rules of the form $\Box\neg p$:- q). Thereafter, a consistency checker that employs the belief reasoner is invoked to make sure that the content of the PGOAL is consistent with the existing commitments and intentions of this agent, and triggers on the belief base are created to monitor the escape conditions in the definition of PGOAL, and in the case of PWAGs, mutual belief of those conditions. The appropriate trigger is fired when one of the above three cases becomes true. The execution cycle is completed when one of the triggers for the original commitment is fired. During the interpretation of INTEND, primitive actions are executed directly, while complex actions are decomposed. It is possible to have multiple stacks because an agent can have more than one commitment at a given time, such as when it is performing concurrent tasks.

An agent specification in the STAPLE interpreter consists of an initial mental state (beliefs, goals, commitments, and intentions), capabilities (actions and plans), inference rules, initial state of the world, domain specific inference rules, etc.

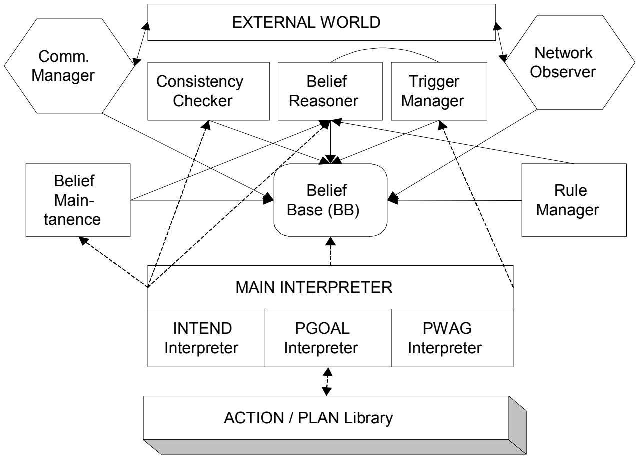

Figure 1: STAPLE Architecture

Agents in STAPLE are programmed using the usual Prolog syntax extended by operators for dynamic logic of actions (concurrent actions, test, repetition, etc.), temporal logic (eventually and always), and for negation, implication, conjunction, and some other miscellaneous constructs. Primitive actions may optionally specify a precondition that must be true before the action can be executed and a list of desired effects that may eventually become true as a result of the action being performed but there is no guarantee that they will ever become true. These conditions are used in a standard backward-chaining reasoner.

Plans in STAPLE are expressed as complex action expressions constructed using the action formation operators for sequence, non-deterministic OR, concurrent action, test action, and repetition. The interpreter recursively decomposes such expressions, executing the Self's actions, interpreting the action combinators, and waiting for the execution of other agents' actions to which it has committed. All primitive actions in a plan must have already been specified as mentioned above. Further, since plans are also actions (though not singleton actions), they are required to specify the effects that must be true after a successful execution of the action expression for that plan.

Joint intention theory leaves room for the agents to decide upon the best course of action consistent with the theory. For example, one of the declarative rules in STAPLE specifies that if an agent is committed to achieving a proposition p and it knows of actions that it can do to achieve that proposition, then the agent will intend to perform a non-deterministic OR expression of those actions with respect to the original commitment. We leave open precisely which of these actions to choose, for example, by maximizing utility. Similarly, there are several other rules defined in the rule base that characterizes rational behavior in different situations.

In the final design of the STAPLE interpreter, we clearly divided the execution process into two parts: i) Procedural Programming (Java) and ii) Logic Programming (Prolog). The Java part interprets the STAPLE program and asserts the statements into a Prolog knowledge base. It also handles the triggers, the intention-commitment stack, and the graphical user interface/console. The Prolog code is used for reasoning. Having this flexibility (integrating various languages to get the best of each of them) improved the performance of the STAPLE agents 70-fold when compared to a Java only implementation. This will allow the STAPLE interpreter to work in real-time application domains.

2.3 Integrating Communicative Actions

The semantics of communicative acts based on joint intention theory characterizes them as attempts having an associated goal and intention. The goal associated with performance of a communicative act (i.e. its desired effect) is what the agent would like to bring about by performing that act. The intention associated with attempting a communicative act (i.e. its intended effect) is what the agent is committed to bringing about via performance of that act.

The literature on the semantics of communicative acts based on JI theory defines two primitive communicative acts, REQUEST and INFORM. The goal of a request (REQUEST x y *e a q t*) is that the requestee y eventually does the action *a* and also come to have a PWAG with respect to the requester x to do *a*. The requester's and requestee's PWAG's are relative to some higher-level goal *q*. The goal of an inform (INFORM x y e *p* t) is that the listening agent y comes to believe that there is mutual belief between him and the informing agent x that the proposition *p* is true. The communicative act INFORM is represented in the STAPLE interpreter as follows:

```
action_definition(inform,3) :-
[args: [x,y,p],  %Informer, Informee, Proposition
 precondition: {bel(x, p) ∧ ~bel(x,bel(x, p))}
 code: {% code to compose and send message},
 effects: [(mb(x,y,bel(x, p)))]]
```

Table 1: Definition of INFORM in STAPLE

For the communicative act INFORM to be performed, the informing agent x has a precondition that it believes the proposition p and x does not believe that informee agent y believes p. A similar definition of REQUEST in STAPLE can be given in the language. These operator definitions are used in the usual backward-chaining fashion. Other communicative acts such as CONFIRM, INFORM-REF, INFORM-IF, ASK_REF, ASK-IF, PROPOSE, AGREE, REFUSE are composed using the basic communicative acts INFORM and REQUEST. For example, communicative act agree (AGREE x y *e a q t*) is defined as the agreeing agent x informs the listening agent y that he has a PWAG with y to perform action *a* with respect to y's PWAG that x do *a* relative to *q*. It has been shown in (Kumar et al. 2002b) that a REQUEST from x to y followed by an AGREE from y to x establishes a JPG between the requester and the requestee to do the requested action. Similarly, it is shown that it requires an INFORM from x to y followed by another INFORM from y to x that the informed proposition p was believed in order to establish mutual belief about p.

2.4 Integrating Belief Reasoner

The STAPLE interpreter includes a Horn-clause belief reasoner that implements weak S5 semantics and is capable of reasoning with quantified beliefs within the Horn subset of first-order logic. The beliefs of a STAPLE agent, including those beliefs common to all STAPLE agents, are stored in a knowledge base that supports concurrent access. The belief reasoner is used in the STAPLE interpreter for querying the belief base instead of deducing all possible consequences that can be inferred from the belief base. Some of the sample belief rules dealing with temporal formulas are as shown in Table 2.

(BEL self p) :- p	¬(BEL α p) :- \+ (BEL α p)
(BEL α p) :- (BEL α □p)	(BEL α ¬¬p):-(BEL α p)
(BEL α ◊p) :- (BEL α p)	(BEL α □□p):- (BEL α □p)
¬(BEL α p):- (BEL α ¬p)	(BEL α ◊◊p) :- (BEL α ◊p)
(BEL α (BEL α p)) :- (BEL α p)	(BEL α p∧q) :- (BEL α p) ∧ (BEL α q)

Table 2: Sample Deduction Rules for Belief Reasoner

A belief base maintenance system complements the belief reasoner and is primarily needed to help it avoid circular loops and infinite recursions For example, (BEL x (BEL x... (BEL x p)…))) is reduced to the equivalent fact (BEL x p). Beliefs about other agents are represented just like any other fact in the belief base. For instance, "I believe that x believes p" will be asserted into the agent's belief base as (BEL x p), which is a simplified form of (BEL self (BEL x p)). A few relations between beliefs and MB's are given in the table below (not a complete set).

(BEL x p) :- (MB x y p)
(BEL x (BEL y p)) :- (MB x y p)
(BEL x (MB x y p/\q)):- (BEL x (MB x y p))
/\ (BEL x (MB x y q))
(BEL x (MB x y □¬(p/\q))) :- (BEL x (MB x y □¬p))
∨ (BEL x (MB x y □¬q))

Table 3: Relations between beliefs and MB's

The consistency checker uses the belief reasoner to attempt to ensure that an agent does not adopt any commitment or intention that conflicts with any existing commitment or intention of that agent. For instance, the agent cannot adopt an intention to achieve ¬p if it already has an intention to achieve p (ideally speaking, it can have the intention to achieve ¬p and p at different times but the current version of STAPLE interpreter ignores this timing subtlety). An agent cannot adopt an intention or commitment to achieve p if it believes that the new commitment or intention makes an already existing commitment or intention impossible i.e. if (BEL x p) ⊃ (BEL x □¬q) where the agent has an existing commitment or intention to achieve q. There are other similar rules for maintaining consistency in the system.

The system properly reasons with disjunctive and "quantified-in" beliefs (Allen and Perrault 1980). The next section illustrates the use of KNOW-IF and KNOW-REF with an example.

KNOW-IF(x, *p*):- KNOW(x, *p*) ∨ KNOW(x,~ *p*)
KNOW-REF (x, i(v).p(v)):- ∃z (BEL x i(y) (p(y) ; y=z))
Note: functor 'i' is the Russellian quantifier 'iota'.

Table 4: Definitions of KNOW-IF and KNOW-REF

3. Example

The example presented in this paper is based on a "Lights World" domain similar to that used by Breazeal et al (Breazeal et al. 2004). In this domain, there are three lights that the human and a robot are to collaboratively turn on or off. The teammates engage in turn-taking, and engage in communicate acts to achieve their goals.

In accordance with joint intention theory, the robot maintains a shared mental state with the human, demonstrates a commitment to doing its own part of the joint action as well as the action to be done by its teammate, and communicates to establish the required mutual beliefs for the jointly committed goal.

Breazeal et al. need to distinguish goals that achieve some proposition from goals to perform actions without concern for outcome ("just do it" goals). This distinction, among many others, is easily expressed in STAPLE plans. STAPLE allows a user to declaratively specify and/or modify team behavior, thereby significantly short-cutting development time. This approach contrasts with Breazeal's framework where a new algorithm has to be programmed into the robot to get the new collaborative behavior.

The simulator has three lights (red, blue, and green) and supports two actions: switch_on, and switch_off each of which take the name of the light as an argument. The user and the agent (or two agents) communicate directly with each other using messages that encode the communicative actions that we define. A template-based language generator and Microsoft's text-to-speech subsystems produce human understandable output.

The examples that follow demonstrate the feasibility of obtaining team-oriented dialogue without having to program it explicitly. The most interesting part of these experiments is that we get a large range of dialogue behavior from the STAPLE interpreter under different initial conditions. For this particular experiment, we have the user, Bob, with an initial individual commitment (PGOAL) to establish a joint commitment (JPG) with the agent, Harry, to jointly turn lights on. It is to be noted that the user and the agent are considered to be cooperative.

3.1 Jointly executing an action expression

For our experiment, we set the initial conditions as follows: (i) Bob and Harry know the initial state of the world. (ii) Thereafter, Bob can observe the state of the world and the changes taking place in it whereas the Harry cannot. (iii) Only Harry can turn on the blue light (iv) The plan "jointly turn lights on" consists of an action sequence in which the first action specifies that "Some Agent" turns on the red light, after which the blue light is turned on only if it is not turned on already.

action(switch_on(redlight),SomeAgent),
((action(test(,~switched_on(bluelight)),Bob,E2),
action(switch_on(bluelight),Harry,E3)))

"SomeAgent" is a capitalized atom, which indicates a variable in STAPLE (as in Prolog.) Bob, who establishes the joint commitment to execute the action expression, knows the actors for each action in the sequence. Harry does not know the identity of "Some Agent" and treats it as an unbound variable whose value is to be found later. However, Harry believes that Bob knows who will switch on the red light, i.e., Harry believes there exists some K that Bob believes is the actor of the first action *switch_on(redlight). In other words,*

(KNOW-REF Bob,i(X). actor(switch_on(redlight),X))

We define two more communicative acts to use with this example. First we define an INFORM-REF to be an inform of the referent of an expression (Allen and Perrault 1980), and then we declaratively define ASK-REF as a REQUEST to do an INFORM-REF. In STAPLE, it is specified as a plan as shown below:

request(X,Y, action(informref(Y,X,C,i(Z,P)),Y),Q)

The above action specifies a request from agent X to agent Y that agent Y inform agent X the referent C of the expression i(Z,P). The precondition of the ASK-REF communicative act is that the agent performing ASK-REF knows that the other agent knows the answer to the question being raised.

The following actual dialogue between Bob and Harry takes place during execution.

Dialogue:

1. Bob: Let us jointly turn lights on.

Bob has an initial condition to form a team with Harry to jointly execute a plan. This automatically results in a dialog between Harry and Bob to communicate and form a team. Bob starts this process by sending a PROPOSE to Harry to do the action of jointly turning the lights on.

2**. Harry: I agree to jointly turn lights on.**

A PROPOSE or REQUEST is generally followed by an acceptance, refusal, or a clarification; Clarification is a sub-dialog that could result in the modification of the request or re-planning. We here concentrate on either AGREE or REFUSE to do the request. At this point there is a mutual belief between Harry and Bob that they have a PWAG to jointly turn lights on; this is due to the effect of AGREE after a REQUEST or PROPOSE.

3. Harry: Who will switch on Red light?

In continuing to interpret the joint action expression, Harry discovers that it does not know the actor of the action *switch_on(redlight).* Therefore, it adopts an individual commitment (PGOAL) for finding the agent who is the actor of action *switch_on(redlight).*

It should be noted that this PGOAL is created only if Harry believes that at least one other agent/human in the team knows the name of the agent to switch on the red light. Means-ends reasoning leads Harry to infer that he can achieve this committed goal by performing the ASK-REF communicative act[‡], provided he believes Bob knows who the actor is. Harry intends to perform ASK-REF, and then acts on this intention, resulting in utterance 3. This behavior contrasts with Breazeal's implementation (Breazeal et al. 2004) where the robot does the action if it can do it, and if not, looks towards the human for the human to do it.

4. Bob: You will switch on Red light[§].

Bob executes an INFORM to Harry that Harry will turn on the red light[**].

5. Harry: Ok, I will switch on Red light.

At this point, Harry realizes that he has to do an action. Harry does belief reasoning with his belief base and if he can do that particular action without leading to any contradictions in its existing belief, Harry will AGREE to do it. A mutual belief is now established there between Bob and Harry that Harry will turn on the red light.

6. <action: Harry pushes button; Red light turns on>

7. Harry: I have switched on Red light.

8. Bob: Ok.

Bob and Harry do not have a mutual belief that they can observe each other's actions, therefore Harry takes the responsibility to INFORM Bob about the effects of its action thereby establishing mutual belief that the action has been done. An INFORM followed by an acceptance establishes the mutual belief that Harry has turned on the red light. If the agents had mutual belief that Bob can observe the world, then there is no necessity for Harry to issue the INFORM.

The next step for the team is to check if the blue light is turned on or not. If it is not on, then they have to turn it on. Note that the agent that can test the state of the blue light cannot perform the action and vice-versa. They need to perform communicative acts to do this task together. There are two possible variations in the dialog at this point.

The first possible interaction is:

9a. Bob: The blue light is not switched on.

Bob can observe the state of blue light and establishes an individual intention (with respect to the joint commitment) to do the test condition, leading him to INFORM Harry of the result of the condition. Harry waits for Bob to do the test and inform him of the result.

10a. Harry: Ok.

11. <action: Harry pushes button; Red light turns on>

12. Harry: I have switched on Red light.

Harry takes an individual commitment (with respect to the joint commitment) to turn blue light on. Harry carries out the action and sends an INFORM to Bob as before,

The second possible interaction is as follows:

9b. Harry: Is the Blue light on?

Harry realizes that he cannot perform the test condition; Therefore Harry creates a PGOAL to KNOW-IF blue light is switched on or not. Harry issues the yes/no question (Allen and Perrault 1980) to BOB to come to know the status of the blue light

10b. Bob: The Blue light is not switched on.

Bob responds to the REQUEST by performing the INFORM-IF to let Harry know the status of the blue light. Harry then acknowledges the information and proceeds to turn on the blue light.

Dialogues 11 and 12 are exchanged as shown before.

4. Related Work

As far as teamwork is concerned, STAPLE is most related to STEAM (Tambe 1997) in its concern for teamwork capabilities. Regarding collaborative dialogue, STAPLE is most related to ARITMIS (Bretier and Sadek 1996) and Collagen (Rich et al. 2001). STAPLE is also related to Breazael's infrastructure for human-robot collaboration that builds upon both joint intention and shared plans theory similar to that done by STEAM.

STEAM has team operators based on SOAR rules (Laird et al. 1987), and is inspired by joint intention theory. It also has explicit representation of team goals, plans and joint commitments, and uses shared plan theories. One difference between our work and STEAM is the language for teamwork specification. STAPLE explicitly reasons about beliefs of other agents, and uses JI theory in generating communicative acts whereas STEAM uses a fixed sequence of messages to create and discharge joint commitments. STEAM has proven to be very useful for

[‡] There could be other acts as well that achieve this effect, such as perceptual ones.

[§] There is one more step in which Bob informs Harry that he will perform the INFORM_REF.

[**] In another scenario, Harry might know that it can switch on the red light and also knows that Bob cannot switch it on. Then, according to JI theory, Bob develops an intention to turn on red light relative to the joint intention of the team. Instead of a question being raised, Harry would have sent an INFORM to Bob that it will switch on red light.

building agent teams but the lack of belief reasoning and reasoning about semantics of communicative acts makes it difficult for STEAM to be used as a dialogue engine.

Collagen gets its dialogue behavior by implementing the algorithms and discourse structures of the Shared Plans theory (Grosz and Sidner 1990), which serves as a specification. Collagen does not do first principles reasoning about communication or teamwork. This makes it more difficult to generate dialog on a generic domain and substantial changes could be need from porting Collagen from one domain to another.

The ARTIMIS system by Bretier and Sadek (Bretier and Sadek 1996) was one of the early successes of integrating communication language with the intention model to achieve dialog and inspired the present research. The present system can replicate the basic speech act reasoning that ARTIMIS performs, but supplies additional goals for collaborative problem solving using JI theory.

5. Summary and Future Work

This paper demonstrates the feasibility of integrating the JI Theory, semantics of communicative acts and belief reasoning using a logic-based declarative language to obtain team and communicative behavior automatically without having to program this behavior explicitly. The example in this paper is created merely by encoding the initial conditions and stipulating the joint plan, from which team and communicative behavior followed automatically. There was no necessity to indicate when a question should be raised, when information should be shared etc. The plan and action library built into the STAPLE interpreter enables the agents to exhibit team-oriented dialogue by interpreting the constructs of joint intention theory along with first principles reasoning over a formal semantics of communicative acts based on that theory. This research shows that formal semantics of communicative acts can be fruitfully employed for inter-agent dialogue.

STAPLE supports several other interesting team and dialogue behavior not discussed here, including dialogue in teams of more than two agents. We have conducted various experiments in which agents can observe each others actions, can work with maintenance goals (of having all the lights always on, even if the user interferes and turns of some lights) and planning to achieve preconditions that need to be satisfied to execute a particular action.

Future work includes incorporating plan recognition for full-fledged dialogue, indirect speech-acts (Allen and Perrault 1980) and probabilistic reasoning for beliefs of the agents in uncertain domains such as spoken-dialog systems.

6. Acknowledgement

We gratefully acknowledge the support of the Defense Advanced Research Project Agency (DARPA) CoABS Program (contract F30602-98-2-0098, A0 G352) and through the Department of the Interior, NBC, Acquisition Services, under Contract No. NBCHD030010.

7. References

Allen, J. F. and Perrault, C. R. 1980. Analyzing Intention in Dialogues. *Artificial Intelligence.*, 15(3): 143-178.

Breazeal, C., Hoffman, G., and Lockerd, A. 2004. Teaching and Working with Robots as a Collaboration. In *Proceedings of AAMAS 2004*, New York, ACM Press.

Bretier, P. and Sadek, M. D. 1996. A rational agent as the kernel of a cooperative spoken dialogue system: Implementing a logical theory of interaction. In J. P. Muller, M. J. Wooldridge, and N. R. Jennings, Eds. *Intelligent agents III*, LNAI.

Cohen, P. R. and Perrault, C. R. 1979. Elements of a Plan-Based Theory of Speech Acts. *Cognitive Science*, 3(3): 177-212.

Cohen, P. R. and Levesque, H. J. 1990. Intention Is Choice with Commitment. *Artificial Intelligence*, 42: 213-261.

Cohen, P. R. and Levesque, H. J. 1991. Teamwork. *Nous*, 25(4): 487-512.

FIPA 2000. Agent Comm. Language Specifications.

Georgeff, M. P. and Lansky, A. L. 1987. Reactive reasoning and planning. In *Proc. of AAAI-87*, 677-682.

Grosz, B. J. and Sidner, C. L. 1990. Plans for discourse. In P. R. Cohen, J. Morgan, and M. E. Pollack, Eds. *Intentions in Communication*, MIT Press, Cambridge, 417-444.

Jennings, N. R. 1995. Controlling Cooperative Problem Solving in Industrial Multi-Agent Systems using Joint Intentions. Artificial Intelligence, 75(2): 195-240.

Kumar, S., 2006. A Formal Semantics of Teamwork and Multi- agent Conversations as the Basis of a Language for Programming teams of Autonomous Agents. Ph.D. diss., Dept. of Computer Science and Engineering, Oregon Health and Sciences University.

Kumar, S., Huber, M. J., Cohen, P. R., and McGee, D. R. 2002b. Toward A Formalism For Conversation Protocols Using Joint Intention Theory. Computational Intelligence, 18(2): 174-228.

Labrou, Y. and Finin, T. 1997. A Proposal for a New KQML Specification. *UMBC Tech. Report*, TR CS-97-03.

Laird, J. E., Newell, A., and Rosenbloom, P. S. 1987. SOAR: An Architecture for General Intelligence. *Artificial Intelligence*, 33(1): 1-64.

Levesque, H. J., Cohen, P. R., and Nunes, J. H. T. 1990. On Acting Together. In *Proceedings of AAAI-90*, 94-99.

Morley, D. and Myers, K. L. 2004. SRI Procedural Agent Realization Kit -- SPARK. In *Proc. of AAMAS-04, 2004.*

Rao, A. S. and Georgeff, M. P. 1991. Modeling Rational Agents within a BDI Architecture. In *Proc. of 2nd Int. Conference on Knowledge Representation and Reasoning*.

Rich, C., Sidner, C. L., and Lesh, N. B. 2001. COLLAGEN: Applying Collaborative Discourse Theory to Human-Computer Interaction. AI Magazine. 22: 15-25.

Tambe, M. 1997. Towards Flexible Teamwork. *Journal of Artificial Intelligence Research*, 7: 83-124.