Simulating a Human Cooperative Problem Solving

Alexandre Pauchet[1], Amal El Fallah Seghrouchni[2], and Nathalie Chaignaud[3]

[1] LITIS - EA 4051
pauchet@insa-rouen.fr
[2] LIP6 - UMR CNRS 7606
Amal.ElFallah@lip6.fr
[3] LITIS - EA 4051
chaignaud@insa-rouen.fr

Abstract. We are interested in understanding and simulating how humans elaborate plans in situations where knowledge is incomplete and how they interact to obtain missing information. Our human interaction model is based on the speech act theory to model the utterances and it uses timed automata to describe the dynamics of the dialogs. Our human planning model is implemented as a hierarchical blackboard architecture which manages opportunistic planning. The system BDIggy we propose, is a concurrent implementation of the planning model and the interaction model through the BDI concept. This system is used to simulate the human processes during cooperative problem solving.

1 Introduction

Nowadays, heterogeneous multi-agent systems, compound of human agents and software agents, become commonplace. Therefore, humans frequently interact with intelligent systems deployed in open environments. When users encounter incompletely described situations, they have to acquire missing information with the system and reason about it. They often have to anticipate their problem-solving and build plans dynamically. We are convinced that studying human activities of problem-solving with incomplete information is essential to the design of agents which interact with humans. More precisely, it is necessary to integrate an explicit representation of the user's beliefs, abilities and intentions. This user's representation cannot be constructed without an accurate understanding of how subjects elaborate plans and how they interact to obtain missing information. The goal of this research is to develop a computational model of human interaction and of human planning, that may help to improve the design of software agents involved in heterogeneous multi-agent systems.

A psychological experiment was conducted, during which subjects have had to solve a problem with incomplete information. The problem submitted to subjects is related to a travel-agency application. Three salesmen are each in charge of a particular means of transport: the first one manages airlines, the second one manages railways and the last one manages taxis and coaches. Each of

H.-D. Burkhard et al. (Eds.): CEEMAS 2007, LNAI 4696, pp. 225–235, 2007.

them has to organize a journey for his own client. This problem consists in satisfying several constraints: a starting city and an arrival city in France, a time of departure, a time of arrival, a number of travelers and a budget. None of the journeys can be undertaken using a single means of transport. Thus, the subjects have to cooperate to solve the 3 problems. To solve their problem, the subjects use a software interface which records the performed actions and the emails while an experimenter notes the verbal utterances of the subjects. Each solving produces three experimental protocols (recorded actions, emails and concomitant verbalization), one for each subject.

The experiment was carried out with 14 groups of 3 students who solved the problems. We analyzed 9 groups in order to design the cognitive models and 5 groups were used for the validation of these models. The 9×3 experimental protocols have been analyzed from two points of view: human planning and human interaction. We modeled the human cooperative planning in an opportunistic way, while our human interaction model is based on the speech act theory to represent the utterances and uses timed automata to describe the dynamics of the dialogs. The system BDIggy we propose is a concurrent implementation of the planning and interaction models through the BDI concept. The validation of the models and the BDIggy architecture is done by comparing the output of this system with the human protocols collected during the psychological experiment.

2 The Human Planning Model

Each protocol has been analyzed and the resulting planning model uses the notions of phases, states of mind, strategies, tactics, observations and personality.

2.1 The Phases

Each protocol can be divided into several phases which differentiate the situations of the problem-solving.

- Subjects build plans during *planning*.
- Minor obstacles are removed during *correction*.
- *Deadlock-solving* corresponds to re-planning.
- When the solution is complete, the subject *tests* it.
- During *checking*, subjects verify their plans.
- *Cooperation* is used to help another subject.

2.2 The States of Mind

The decision to perform an action depends on the attention paid to the different constraints (or *criteria*) of the problem. The subjects take a subset of criteria into account, directly linked with the constraints. This subset, called state of mind, evolves during the problem-solving process according to the situation. The criteria are domain dependent. Five criteria are defined for the travel agency problem: *price*, *timetable*, *transfer*, *number of travelers* and *booking*.

2.3 The Strategies

Strategies model the way plans are built. The model differentiates the *approach strategies* (type of planning the subjects use), and the *planning strategies* (order in which subjects build plans). We observed:

- 2 approach strategies: *sequential planning* and *parallel planning*.
- 4 planning strategies: *prospective*, *retrospective*, *centrifugal* (from the middle of the travel to the edges) and *centripetal* (from the edges to the middle).

2.4 The Tactics

Subjects can instantiate one planning strategy by several actions and objects. Tactics are used to model the different choices of actions to be performed. The tactics are domain dependent. Tactics are used to choose the different stages of the journey (via the largest towns, the most direct, *etc.*) or the means of transport (the cheapest, the fastest, *etc.*).

2.5 The Observations

During the solving process, any change in one of the ingredients (phase, state of mind, strategy or tactic) corresponds to a new episode. These changes are triggered by the observations that subjects can make about the current situation of the problem. When the verbalization is sufficient, the observations correspond to the verbal utterances of the subjects. 18 observations appeared (Ex: the journey is too expensive, a particular town has a railway station, the departure or arrival of a plane is not compatible with the current solution, *etc.*).

2.6 The Personality

The personality models the individual differences between subjects. Each experimental protocol describes the behavior of a subject and this general behavior depends on values of the features of the personality. Eight features are used:

- *thrifty*: how the subject is concerned with the stage prices,
- *opportunistic*: how often the subject uses parallel planning,
- *good estimator*: the subject's capability to estimate correctly a solution,
- *careful*: how few mistakes are made,
- *optimizer*: how the subject tries to minimize the travel,
- *precise*: how exhaustive are the stage information searched by the subject,
- *patient*: how long the subject waits before sending re-queries,
- *cooperative*: how the subject participate easily to the other's problem,

When analyzed, a personality (8 features with a value in $\{1, 2, 3\}$) was attributed to each protocol.

3 The Human Interaction Model

Each of the 9 groups of 3 protocols have been merged into a single file, respecting the temporality. Both the utterance and the discourse levels are considered.

3.1 The Utterance Level

At the utterance level, each message of the experimental protocols was analyzed individually. To refer to the speech act theory [1], the observed speech acts are either *descriptives*, *directives* or *commissives*. Messages were first matched with a performative from FIPA-ACL [2] or from KQML [3]. When no matching performative exist, a new one is proposed. Table 1 presents the performatives observed in the protocols with their frequencies. This list is exhaustive only concerning the information-search dialogs.

Table 1. The observed performatives

Descriptives:		
inform	S sends a piece of information to R. From FIPA-ACL.	*2.7%*
notUnderstood	S does not understand one of R's previous message. From FIPA-ACL.	*0.8%*
reply	S answers R. Inform-if/inform-ref in FIPA-ACL.	*39.6%*
thank	S thanks R. New performative.	*2.7%*
Directives:		
acceptProposal	S accepts an information proposal from R. From FIPA-ACL.	*0.9%*
cancel	S tells R not to take into account a previous message. From FIPA-ACL.	*1.6%*
query	S asks R for a piece of information. Query-if/query-ref in FIPA-ACL.	*43.0%*
refine	S asks R to detail one of R's previous query. New performative.	*5.4%*
refuseProposal	S refuses an information proposal from R. Reject-proposal in FIPA-ACL.	*0.3%*
Commissives:		
propose	S proposes to send information to R. From FIPA-ACL.	*2.6%*

A message is represented by the predicate

$$pMessage(A_S\ A_R\ P\ C)$$

with A_S the sender, A_R the receiver, P the performative and $C \in BEL \cup DES$ the content on which the performative P is applied. BEL is the set of beliefs and DES is the set of desires. $C = pB(\alpha) | pD(A\ \delta)$ where $pB(\alpha) \in BEL$ with α a predicate, and $pD(A\ \delta) \in DES$ with $A \in \{air,\ railway,\ road\}$ an agent and δ a predicate. A desire is attached to an agent. Frequently, $\alpha \in STA$ and $\delta \in STA$. STA is the set of stages described by the predicate

$$pStage(C_D\ C_A\ T_D\ T_A\ M\ N\ P\ R)$$

where C_D and C_A are the starting and the arrival cities, T_D and T_A are the departure and the arrival times, M is the means of transport, N is the number of travelers, P is the price and R indicates if the stage is booked or not.

A performative is applied to a mental state (a belief or a desire), the scope of which is a predicate. A performative and its content are closely linked as the illocutionary force and its proposition (noted $F(P)$ in the speech act theory):

- A *descriptive* is applied to a *belief*. It describes how the sender perceives the world.

- A *directive* is applied to a *desire of the sender*. The sender wants to receive an information and he sends this desire to the subject who has the information.
- A *commissive* is applied to a *desire of the receiver*. The sender supposes that the receiver has a certain desire.

3.2 The Discourse Level

Our discourse analysis is based on the work of Vanderveken [4] which extends the speech act theory to discourse. He splits conversations into *illocutionary acts*, introduces *mental states* as basic reasoning units and calls *exchange* a set of bounded messages. The experimental protocols were divided into exchanges, classified into 4 categories: *information queries*, *information proposals*, *spontaneous sendings*, *error processings*. Each of these exchanges is guided by the discourse goal of the initiator, according to the first performative he sent. Error processing exchanges are the only ones whose discourse goal differs from the performative type of the initial message. Their discourse goal is directive because the receiver of the erroneous message wants the sender to cancel it. The way exchanges are terminated defines their satisfaction. An exchange can be either *satisfied* (the interlocutor's goal happens) or not. The starting performative is used to classify the exchange and the ending performative, if it exists, defines the exchange satisfaction. An exchange can be considered as terminated by the interlocutors even without explicit emission of an ending performative. Spontaneous sendings of information only need one main act: the starting performative.

As messages are emails, time is important regarding to re-queries and exchange terminations. When a subject needs an information from another subject, if he still has not received any answer after a certain time, he will re-ask for information. Similarly, exchange without any explicit ending speech act is considered as terminated after a certain time. Timed automata [5] are used to model these exchanges and the temporality. 4 pairs of automata are designed to represent the observed exchanges, a pair of automata (an automaton for each interlocutor) for each type of exchange. Fig.1 describes the behavior of the initiator of an information query (Q_{ini}). Each state represents a particular situation during the exchange. A transition can be crossed when a message is sent or received, after a delay or if a variable reaches a particular value. The 4 pairs of timed automata have been tested on the whole experimental protocols to ensure they are exhaustive. For each automaton, the frequency observed in the experimental protocols of each transition is specified. Timed automata are used

- *to generate a message*: messages are produced following an automaton. When there is no determinism, the decision is made according to the current situation and the simulated agent's personality. If several transitions can be crossed, the decision is made randomly respecting the frequency of the transitions.
- *to interpret a message*: an automaton describes the expected messages. 2 interlocutors can manage several exchanges simultaneously so these expected events help to know if a message belongs to an exchange.

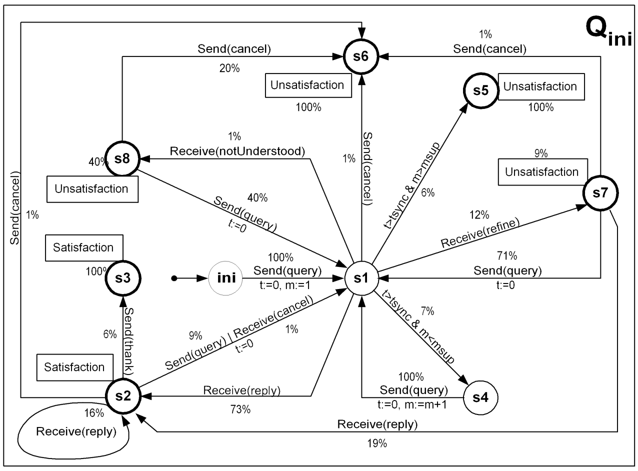

Fig. 1. Automaton of an information query

3.3 A Semantics of Performatives

This section describes the semantics associated with the sending/receipt of messages. The semantics of each performative is given by the generic reduction rule

$$[PreCond]\frac{A_X(si_1,...,si_n)\xrightarrow{send|receive(performative)}A_X(sf)}{a_1;...;a_n}$$

where *PreCond* are preconditions, A_X with $X \in \{ini,\ int\}$ is the automaton, $si_1...si_n$ are the states before processing the sending/receipt of a message, sf is the state after processing the sending/receipt of a message and $a_1,...,a_n$ are the internal "actions" to be performed in order to update the model.

We present here the semantics associated with the sending of a query in Q_{ini}.

Query

Syntax: *pMessage*(A_S A_R *query pD*(A_S S)) with A_S, $A_R \in \{air,\ railway,\ road\}$ and $S \in STA$.
Description: A_S can send a *query* if he wants the stage S, if S uses a different means of transport from A_S's one and if A_S has no belief about S.
Semantics:

$$\begin{bmatrix} pD(A_S,S) \\ \neg pMeans(S) \\ \neg pB(S) \\ \neg pB(\neg S) \end{bmatrix} \frac{Q_{ini}(ini,s2,s4,s7,s8)\xrightarrow{send(query)}Q_{ini}(s1)}{aUpdateTA(M)}$$

The postconditions are expressed with internal actions which can only affect the beliefs. Desires of the locutor and desires of the interlocutor are processed as knowledge (beliefs applied on a desire).

Moreover, when sending a message, the automaton states are updated with the action *aUpdateTA(M)*, *M*∈*MES*. This action checks if one of the opened automata matches the message, and modify it consequently. If there is no corresponding automaton, a new one is opened.

Using beliefs and desires in both the syntax and the semantics of a performative links the utterance and the discourse levels in our interaction model.

4 The Implementation

The human planning and interaction models are implemented into a new agent architecture, called BDIggy, which takes benefits from the BDI architectures. Further references to BDI model are based on dMARS [6]. A BDI agent includes a *queue of events* which stores internal and external events occurring in the system, some *beliefs*, a library of plans, a stack of *desires* and a stack of *intentions* (instantiated plans to reach the desires). The BDI interpreter runs as follows: first of all, events are updated generating new beliefs. Then, new desires are calculated matching plans of the library according to the beliefs. One of these plans is selected for execution and put into the intention stack. Finally, an intention is selected and its plan is performed, and so on.

A BDIggy instance is an agent written in Java which simulates a subject and generates artificial protocols. It is worth emphasizing that the only information shared by the agents is passed through the messages. In fact, three agents run concurrently, forming a multi-agents system.

The BDIggy architecture (see Fig. 2) includes the *perception module* which manages the agent's representation of the world. It analyzes the environment and generates beliefs. It contains a description of the problem (the agent's personality to simulate and the travel to be built), the domain knowledge and a representation of the current situation. All of them are beliefs and constitute the agent's *memory*.

The planning module is a hierarchical blackboard architecture, whose domain controllers manage the ingredients of the model. The module is initialized with a personality and a problem to be solved. From a description of the current situation, it generates the necessary observations and builds an episode (a phase, a state of mind, strategies and tactics), which can be considered as a short-term

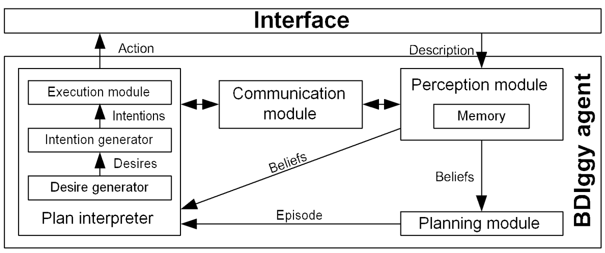

Fig. 2. The BDIggy architecture

abstract plan. An episode leads the system to perform actions. When a series of actions is performed, new observations can be made that create a new episode. This incremental process enables to build plans in an opportunistic way. The planning module replaces the plan library of the BDI architecture, but plans are dynamically constructed.

The plan interpreter works as the BDI interpreter. The *desire generator* translates the abstract plan into several desires. The *intention generator* refines a desire into intentions. The *execution module* performs the actions of an intention. During parallel planning, the plan interpreter manages as many desire and intention stacks as there are parallel plans. Parallel plans correspond to different solutions for the problem, *i.e.* travels crossing different cities but with the same starting city and the same ending city. In the desire generator, a desire can be *abstract* or *elementary*, *instantiated* (refined into intentions) or *non-instantiated*, *satisfied* or *unsatisfied*, *successful* or *unsuccessful*, *active* or *suspended*. In the intention generator, an intention can be *active* or *suspended*.

The *communication module* implements the human interaction models. The perception module delegates the communication module to understand the received messages and the execution module delegates the communication module to write the messages. The communication module can add beliefs in the memory when receiving or sending messages. When a re-query has to be sent, according to the timed automata, a belief is also added to the memory which can possibly be processed by the plan interpreter during the next cycle.

5 Simulation and Validation

To simulate the solving processes of the travel-agency problem, 3 BDIggy agents have been running simultaneously, generating new artificial protocols. The validation is done by hand analyzing a set of real and artificial protocols. The real protocols are drawn from the 5×3 protocols that had been put aside after the experiments. 10×3 protocols have been simulated, each having the personality of one of the real protocols analyzed. Our validation method is based on a Turing-like test. From the sets of protocols described above, we randomly drew one set with 2×3 real protocols and 4×3 artificial protocols and a second set with 3×3 real protocols and 3×3 artificial. Since our system does not support natural language processing, the messages from the real protocols were manually translated into our *pMessage* language.

"Experts" (fellow researchers who did not work on this study) were asked to hand analyze these sets of mixed protocols and to classify them according to their type (human or artificial). Table 2 presents the results of this validation. Expert 1 analyzed the two sets, Expert 2 analyzed only Set 1 and Experts 3 and 4 analyzed only Set 2. Each cell contains the proportion of well-classified protocols in each category. For example, the first cell indicates that Expert 1 classified correctly two protocols over the 3 human protocols of Set 1. The Total column specifies how many protocols each expert has classified correctly, among the number of protocols he analyzed (Expert 1 analyzed 12 protocols and classified correctly 6

Table 2. The validation results

	Set 1		Set 2		
	Human	**BDIggy**	**Human**	**BDIggy**	**Total**
Expert 1	2/3	1/3	1/2	2/4	6/12
Expert 2	1/3	1/3	-	-	2/6
Expert 3	-	-	0/2	2/4	2/6
Expert 4	-	-	2/2	2/4	4/6

of them and so on). Since the experts made around 50% of error, we can conclude that the experts were not able to reliably separate the two classes of protocols.

6 Related Work

This study deals with three issues: cognitive modeling, cooperative planning and interaction. To our knowledge, there are few works integrating all these aspects.

The TRAINS project [7] and TRIPS [8] are close to this work since they develop an intelligent planning assistant that interacts with humans. [7] proposes a model for collaborative agents which integrates an individual problem-solving model, a collaborative problem-solving model and an interaction model in natural language. The communication model works as in [9]: the goal is to recognize the interlocutor's plan to emit an appropriate answer. The notions used (*situations*, *objectives* and *recipes*) are similar to the BDI concept.

The COLLAGEN project [10] aims at developping an application-independent *collaboration manager* to facilitate collaborative interaction between software agents and users. COLLAGEN uses the SharedPlan theory of human collaborative dialogs [11], [12]. The decomposition of plans in recipes is similar to the one used in TRAINS. The utterances and actions to produce are selected according to how they contribute to the discourse purpose.

The TRAINS and COLLAGEN projects are concerned with collaborative interaction whereas we focus on cooperative interaction and planning. The main differences with our work is that we aim at simulating cooperative human planning and we proposed a cognitive model of opportunistic planning.

In comparison with KQML [3] and FIPA-ACL [2], our utterance model is better adapted to human dialogs. The list of performatives is improved and the link between the performative and its content is integrated. Many papers deal with performative semantics. [13] brings important results in the formalization of the Speech Act theory. [14] is oriented by the satisfaction of the speech acts. [15], used in FIPA-ACL, contains modal operators for describing the beliefs, desires, uncertain beliefs and persistent goals of the agents. Our semantics is based on the BDI concepts to use a single representation linking planning and interaction.

Concerning the communication protocols, FIPA-ACL cannot take into account the dynamics of human dialogs. The Dooley Graphs [16] contain information about the situation and the protagonists of a dialog, but do not consider the time dimension. We choose timed automata to represent duration of communicative actions and to synchronize interactions inside exchanges. To our knowledge, no other work uses this formalism to model interaction.

7 Conclusion

The strength of this work is to propose a complete study, from the collection of the experimental protocols to the implementation of the simulation system and its validation. The cognitive models are based on the analysis of the observed behaviors. From the planning point of view, the cooperative planning model allows to build opportunistic plans. From the interaction point of view, an utterance is represented by a performative applied to a mental state (a belief or a desire). Timed automata are a powerful formalism to introduce recursiveness and time management in the discourse representation. The performatives and the timed automata are linked thanks to a semantics using beliefs and desires. The planning and interaction models are integrated into the BDIggy architecture and are represented in a homogeneous way (through BDI) in the same system.

Because cooperation is enforced during the 3 simultaneous solving processes, planning and interaction are interleaved. Each subject depends on the others to acquire missing information and has to interact with them. Most of the time, planning is the prime activity and interaction is only a tool to obtain information: planning manages interaction. But when new information is acquired not corresponding to the subject's expectations, the plan has to be changed opportunistically: interaction manages planning.

The human model of cooperative planning can be re-used for other problem solving. Only the domain specialists, domain dependent, have to be reimplemented to support another problem. The interaction model is exhaustive concerning the information-search dialogs. It has to be extended to other kind of dialogs, such as collaborative ones. Concerning the BDIggy architecture, the problem description (predicates contained by the memory) and the interpretation of episodes into abstract plans are also dependent of the problem, whereas the process of the architecture is general.

The ultimate goal of this work was to propose a computational model which implements the cognitive model in order to simulate human processes during cooperative problem solving. In future work, these new knowledge would be used to design operational systems able to interact efficiently with human end-users.

References

1. Searle, J.: Speech Acts – An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969)
2. FIPA: FIPA communicative act library specification. Technical report, Foundation for Intelligent Physical Agents (2002)
3. Finin, T., Fritzson, R., McKay, D., McEntire, R.: KQML as an agent communication language. In: CIKM'94, pp. 456–463 (1994)
4. Vanderveken, D.: Illocutionary logic and discourse typology. Revue Internationale de philosophie 55(216), 243–255 (2001)
5. Alur, R., Dill, D.L.: A theory of timed automata. Theoretical computer science 126, 183–235 (1994)

6. d'Inverno, M., Kinny, D., Luck, M., Wooldridge, M.: A formal specification of dMARS. Technical note 72, Australian Artificial Intelligence Institute, Carlton, Victoria (1997)
7. Allen, J., Blaylock, N., Ferguson, G.: A problem solving model for collaborative agents. In: Falcone, R., Barber, S., Korba, L., Singh, M.P. (eds.) AAMAS 2002. LNCS (LNAI), vol. 2631, pp. 774–781. Springer, Heidelberg (2003)
8. Allen, J., Ferguson, G.: Human-machine collaborative planning. In: NASA Workshop on Planning and Scheduling for Space, Houston, USA (2002)
9. Traum, D.: A Computational Theory of Grounding in Natural Language Conversation. PhD thesis, University of Rochester (1994)
10. Rich, C., Sidner, C.: COLLAGEN: A collaboration manager for software interface agents. User Modelling and User-Adapted Interaction 8(3/4), 315–350 (1998)
11. Grosz, B., Kraus, S.: Collaborative plans for complex group action. Artificial Intelligence 86(2), 269–357 (1996)
12. Lochbaum, K.: A collaborative planning model of intentionnal structure. Computationnal linguistic 24(4), 525–572 (1998)
13. Cohen, P., Levesque, H.: Performatives in a rationally based speech acts theory. In: MACL'90, Pittsburgh, Pensylvania, pp. 79–88 (1990)
14. Singh, M.: Towards a formal theory of communication for multiagent systems. In: IJCAI'91, Sidney, Australia, pp. 69–74 (1991)
15. Sadek, M.: A study in the logic of intention. In: KR'92, Cambridge, pp. 462–473 (1992)
16. Parunak, H.V.D.: Visualizing agent conversations: using enhanced dooley graphs for agent design and analysis. In: ICMAS'96, Kyoto, Japan, pp. 275–282 (1996)