On Partial Deduction and Conversational Agents

Mariela MORVELI-ESPINOZA and Josep PUYOL-GRUART [1],
Artificial Intelligence Research Institute (IIIA).
Spanish Scientific Research Council (CSIC).

Abstract.
Agents are situated autonomous entities that perceive and act in their environment, and communicate with other agents. An agent usually starts a conversation by querying another agent because it needs to satisfy a specific goal. This process allocates a new goal to the agent receiving the initial query, starting new dialogs with other agents, generating a recursive interaction. The generation of this kind of dialog is interesting when the system has the possibility of generating conditional answers with imprecise and uncertain values. We consider simple deliberative rule-based agents that proactively try to satisfy their goals. The mechanism to achieve this dialogs is based in the *specialization* of the mental state of agents, by means of the partial deduction of rule bases.

Keywords. Conversational agents, multi-agent systems, partial deduction, multiple-valued logic.

Introduction

Rule specialization has been used intensively in logic programming [14], mainly for efficiency purposes, but it has potential applications in other areas as multi-agent systems and particularly in communication among agents [11]. The proposal of this paper is not to explain the general advantages of an inference engine based on specialization [15, 16,17], but to show that this mechanism is useful to drive the communication among agents, generating *reasonable* dialogs. We propose the use of this technique to model the communication behaviour between agents, in an uncertain context, by allowing agents to use *conditional answers* [7,13].

In classical (boolean) rule bases, deduction is mainly based on the modus ponens inference rule: $a, a \rightarrow b \vdash b$. In the case that a denotes a conjunction of conditions $a_1 \wedge a_2$, the above inference rule is only applicable when every condition of the premise, i.e. a_1 and a_2, is satisfied, otherwise nothing can be inferred. However, if we only know that condition a_1 is satisfied, we can use *partial deduction* to extract the maximum information from incomplete knowledge in the sense of the following *specialisation* inference rule: $a_1, a_1 \wedge a_2 \rightarrow b \vdash a_2 \rightarrow b$. The rule $a_2 \rightarrow b$ is called the *specialisation* of $a_1 \wedge a_2 \rightarrow b$ with respect to the proposition a_1. The specialisation of a *rule base* consists

[1]Corresponding author: Josep Puyol-Gruart, Artificial Intelligence Research Institute (IIIA). Campus UAB. 08193 Bellaterra. Spain. Tel.: +34 935809570; Fax: +34 935809661; E-mail: puyol@iiia.csic.es.

on the exhaustive specialisation of its rules. Rules will be substituted by its specialized versions, and rules with no conditions will be eliminated and new propositions will be added. These new propositions will be used again to specialize the agent. The process will finish when the agent has no rule containing on its conditions a known proposition.

In an approximate reasoning context the specialization is much more interesting. The above boolean specialization inference rule can be transformed in the following way: $(a_1, \alpha), (a_1 \wedge a_2 \rightarrow b, \rho) \vdash (a_2 \rightarrow b, \rho')$, meaning that if the proposition a_1 is known to be true at least to the degree α and the rule $a_1 \wedge a_2 \rightarrow b$ is true at least to the degree ρ, then the specialised rule $a_2 \rightarrow b$ is true at least to a degree $\rho' = f(\alpha, \rho)$, where f a suitable combination function.

Using conditional answers and the specialization mechanism, agents are able to answer, when needed, with the information the questioner should know to come up with a value for the query, or they may also inform about other deductive paths that would be useful to improve the solution [15]. For instance the agent can answer: *with the current information x is quite true, but if y were true then x will be definitively true*.

We will use a very simplified vision of agents as message passing entities containing rules. When an agent receives a query it starts a process of finding new external information in order to obtain an answer for that query. The difference with other approaches is that the agent will use the external information to specialize the knowledge base of the agent, and incrementally build more precise answers. The answer can be conditional, that is, it can contain rules if it is not possible to obtain enough information.

In Section 2 we formally describe both the agents and the specialization of their mental state. Section 3 is devoted to the description of the protocols. We present an example of conversation in Section 4. Finally, some discussion and the conclusions are developed in Section 5.

1. Mental state and specialization

The state of our agents will be their mental state [20]. The main component of the mental state is the knowledge base containing beliefs (facts) and knowledge (rules) for deliberation. In this Section a simplified version of our propositional language[2] and the inference mechanism will be described.

Definition 1 *(Language and inference)* $\mathcal{L} = \langle T_n, \Sigma, \mathcal{C}, \mathcal{S} \rangle$ *is defined by:*

- $T_n = \{t_0, t_1, \ldots, t_n\}$ *is an ordered set of truth-values, where* t_0 *and* t_n *are the booleans* True *(1) and* False *(0) respectively.*
- Σ *is a set of propositional variables (atoms or facts).*
- $\mathcal{S}$ *are sentences composed by: atom pairs* (a, V)*, and rules of the form* $(p_1 \wedge p_2 \wedge \cdots \wedge p_n \rightarrow q, V)$*, where* $a, p_i, q \in \Sigma$*,* $V \in T_n$*, and* $\forall i, j (p_i \neq p_j, q \neq p_j)$

We will use the following inference rules:

- Parallel composition: from (φ, V_1) and (φ, V_2) infer $(\varphi, max(V_1, V_2))$

[2] In the complete version of the language we consider negation and the values of facts and rules are intervals of truth values. For the sake of simplicity here we use min and max operations instead of general triangular norms. For more information please see [16].

- Specialization: from (p_i, V) and $(p_1 \wedge \cdots \wedge p_n \rightarrow q, W)$ infer $(p_1 \wedge \cdots \wedge p_{i-1} \wedge p_{i+1} \wedge \cdots \wedge p_n \rightarrow q, min(V, W))$

The mental state of agents contains a set of facts and rules. In our model, both facts and rules are weighted with truth-values in T_n, meaning that the fact or the rule is true at least to some degree. Rules are tuples $r = (m_r, c_r, \rho_r)$ where m_r is the premise (a set of atoms), c_r is the conclusion (an atom) and $\rho_r \in T_n$ is the truth-value of the rule. The representation consists of mapping each atom in Σ to its truth-value and the (possibly empty) set of rules that conclude it.

Definition 2 *(Mental State) Let R be a set of rules, we define an agent mental state M of an agent A as a mapping: $M_A : \Sigma \rightarrow T_n \times 2^R$ where, for each $f \in \Sigma$, $M_A(f) = (\rho_f, R_f)$, being $R_f = \{(m_r, \rho_r) | (m_r, f, \rho_r) \in R\}$*

The representation of an agent's mental state will evolve as deduction proceeds. We represent the initial mental state of an agent as a mapping from any atom into *unknown* and the set of rules deducing it. It means that the atoms initially have their most imprecise value—that is 0.

We consider that a proposition has a *definitive* value when there are no rules that can contribute to its *provisional* value (initially *unknown* or 0), producing a more precise one by means of applications of the parallel composition inference rule. We will use a proposition to specialise rules only when that proposition has a definitive value. This permits rules to be substituted by its specialised versions being the condition eliminated from its premise. When there are no conditions left in the premise of a rule the conclusion of the rule is generated.

To describe the specialization algorithm we describe first the specialisation of a rule. Given an atom (p, ρ_p) and a rule (m_r, c_r, ρ_r) and considering that $p \in m_r$ then the specialization of the rule with respect to that atom will be a new specialized rule $(m_r - \{p\}, c_r, min(\rho_p, \rho_r))$, or a new atom if the rule had a single condition $(c_r, min(\rho_p, \rho_r))$.

We extend now the description of the specialisation of a rule to that of the specialisation of a set of rules concluding the same atom p, the mental state can be expressed as $M(p) = (\rho_p, R)$. In doing so, we select in turn a rule $r \in R$ to specialise. If its specialisation, with respect to a fact (f, ρ_f), returns a new rule r' then we substitute the rule by the specialised one in the agent's mental state representation, and the truth-value of p is not changed giving $M(p) = (V_p, R - \{r\} + \{r'\})$. If the rule is completely specialized and returns ρ_f, the rule is eliminated and a new truth-value for p is calculated by means of the parallel composition inference rule, and the new mental state would be $M(p) = (max(V_f, \rho_f), R - \{r\})$.

To specialise a complete agent's mental state we will use each fact with definitive value in the mental state in turn to make specialization steps that possibly will generate definitive values for other atoms to be later used to specialise more the state.

2. Agents

In the Section above we have explained what will be considered to be part of the mental state of agents and the basic mechanisms of specialization: given new external information, the mental state of an agent is completely specialized in a data driven style. In this

Section we present the concept of agent considering that it is a goal driven entity. Apart from the passively information acquired by perception, agents proactively find new information that will be useful to satisfy their goals. Consider a multi-agent system with n agents $\mathcal{A}_n = \{A_1, \ldots, A_n\}$. Each agent has the following structure:

Definition 3 *(Agents) A deliberative agent is a tuple $A_i = \langle M_i, G_i, I_i, O_i \rangle$ where:*

- *I_i is the input interface, the set of external facts that can be obtained querying other agents. They are tuples $\langle x, A_j \rangle$, where $x \in \Sigma$, $A_j \in \mathcal{A}$ and $A_j \neq A_i$.*
- *O_i is the output interface, this is, the set of facts that the agent can answer to other agents.*
- *G_i are the set of goals of A_i. They are tuples $\langle x, A_j \rangle$, where $x \in \Sigma$ and $A_j \in \mathcal{A}$.*
- *M_i is the mental state of agent A_i.*

We can see that an agent has two important elements: the mental state that is considered to be its building block, and a set of goals that guide its behavior. Goals are facts that the agent want to solve because it has commitments with other agents—generated from communication—or self commitments—internal facts not related with other agents. The input and output interface define the relation with the external world.

Definition 4 *(Fact privacy) The mental state of an agent A_i contains two kinds of facts:*

- *A fact $f \in O_i$ is called* public *then it can be answered to other agents.*
- *The facts $f \notin O_i$ are called* private*, then they can not be revealed to any other agent.*

Definition 5 *(Fact state) The mental state of an agent A_i contains three kinds of facts:*

- *The facts $f \in \{p \in \Sigma | M(p) = (V_p, \emptyset), V_p \neq 0\}$ are called* definitive *or* totally specialized *because there is no more knowledge that could increase their precision.*
- *The facts $f \in \{p \in \Sigma | M(p) = (V_p, R), V_p \neq 0, R \neq \emptyset\}$ are called* provisional *or* partially specialized *and can be improved if there is enough information.*
- *The facts $f \in \{p \in \Sigma | M(p) = (0, R)\}$ are called* pending *and they are (provisionally)* unknown.

2.1. Agents mental state cycle

When an *agent's life* begins and it receives a simple query, the agent can accept or reject it depending of multiple circumstances, for instance, privacy. In the case that the query is accepted, the agent begins a goal-driven—backward chaining style—work done over its mental state. This task will produce new goals (internal and external) that has to be solved. When new facts are known it is started a data-driven task of specialization—forward chaining style.

Agents can send and receive rules as conditional answers or knowledge communication. When the state of a query is *pending* or *provisional* we have to decide how to build a conditional answer. In the case of pending facts the conditional answer will be a set of rules useful to obtain a value for that fact; in the case of provisional facts the answer will be the provisional value and a set of rules useful to improve its value. When an agent receives a conditional answer it adds the new knowledge to its mental state.

Initially $G_i = \emptyset$ and all the facts have value *unknown* (0). We can summarize goal-driven work in the following steps:

1. When A_i receives a query q from an agent, and $q \in O_i$, then $G_i := G_i \cup \{\langle q, A_j \rangle\}$
2. For each goal $\langle g, A_k \rangle \in G_i$,
   (a) if $A_k \neq A_i$ we generate a query g to the agent A_k.
   (b) if $A_k = A_i$ it means that the goal is a self commitment and the agent starts a search process in order to find which is the information it needs.
3. Multiple specialization steps drives to reach goals. Given a goal $\langle g, A_i \rangle \in G_i$
   (a) If $M_i(g) = (V_g, \emptyset)$ and $V_g \neq 0$ then the agent generates a message for agent A_k with the contents $(g, V_g, \emptyset)$.
   (b) If $M_i(g) = (V_g, R)$ and $R \neq \emptyset$ and $\forall(m_r, c_r, \rho_r) \in R$, $m_r \subseteq O_i$ then the agent generates a message for agent A_k with (g, V_g, R).

   In both cases $G_i := G_i - \{\langle g, A_k \rangle\}$
4. When the agent receives answers from other agents, these are used to specialize the mental state. When the answer is (g, V'_g, R') and $M_i(g) = (V_g, R)$ then $M'_i(g) = (max(V_g, V'_g), R \cup R')$

The contents of answer messages are definitive facts or provisional facts with all the necessary rules to make it definitive. This does not mean that a fact with a provisional value will stop being a goal. This only means that a more precise value is reached. Stop criterion will be based on (i) goal value is found, (ii) goal is canceled or (iii) assigned time to find the goal is over (assigned time will depend on query priority and on priority agent AG wants to give it). Different criterions to choose a rule or an atom are out of the scope of this paper, in a backward chaining style we will choose the rule with best truth-value and the first premise in order of writing.

3. Communication

The communication is essential between agents because it is the base of important activities such us: cooperation, coordination and negotiation. It lets to send and receive knowledge, resolve conflicts in the tasks resolution or synchronize actions [19]. In our case, communication is the base in the conversational process between agents. Communication process is based on two important actions, these are: *querying* and *answering*.

After receiving a query, agents elaborate an answer with the information they have or get from other agents. Unquestionably the wished answer is the most precise fact value, nevertheless taking into account that there exist private facts or that their definitive values are not found yet, agents could answer with rules. Messages including rules could also be an option agents take when they have rules with facts that belong to other ones and do not want to obtain this information by themselves.

For querying or answering, agents use messages. To give a semantic to these messages, we use speech act theory [2,9] in form of performative verbs, which correspond to different types of speech acts. Based on FIPA standard [10], a message is a tuple $C_i = \langle P, S, H, B \rangle$, where P is the performative that indicates the message type (we use `QUERY`, `ACCEPT`, `INFORM`, `REJECT` and `CANCEL`), S (sender) is the agent that sends the message, H (hearer) is the agent that receives the message, and B (body) is the message content.

The body of performatives `QUERY`, `ACCEPT`, `REJECT` and `CANCEL` is the name of one fact. The performative `INFORM` has a more complex format because it may contain facts and rules. For this performative, the body is a set of tuples $\langle M_x, V_x \rangle$ where, x is a fact, M_x is the mental state of x and V_x indicates if the value of x is *provisional* or *definitive*. Taking the example above as reference, let's see two possibilities:

- A_j knows the definitive value of f:
  (INFORM, A_j, A_i, {(([1,1], $\emptyset$), definitive)})
- Otherwise it decides to send to A_i one or a set of rules (which must not have any private fact):
  (INFORM, A_j, A_i, {((ρ_1, {({a,b}, ρ_2)}), provisional)})

A dialog is a set of coherent messages: $D = \{C_1, \ldots, C_n\}$. We consider those which involve only two agents, which sequentially alternate dialogue moves. Protocols [12, 8] play a central role in agent communication to specify rules of interaction between communicating agents. In our model the following protocol will be used:

1. At the beginning $D = \emptyset$.
2. A dialog D is initiated by a *query*: (QUERY, A_i, A_j, f), where $A_i \neq A_j$. `QUERY` can appear, obviously, at any moment during a dialog.
3. Depending of the A_j output interface, it can accept or reject the query of A_i :
   - If $f \notin O_j$, then (REJECT, A_j, A_i, f)
   - If $f \in O_j$, then (ACCEPT, A_j, A_i, f)
4. If agent A_j has accepted, one of these five alternatives could happen:
   (a) A_j gives A_i the definitive value of proposition requested
   (INFORM, A_j, A_i, {((ρ_1, $\emptyset$), definitive)})
   (b) A_j gives A_i a provisional value of proposition requested
   (INFORM, A_j, A_i, {((ρ_1, $\emptyset$), provisional)})
   (c) A_j gives A_i one or a set of rules that help to deduce or improve the value of proposition requested
   (INFORM, A_j, A_i, {((ρ_1, R), provisional)})
   (d) A_i cancels the query made to A_j (CANCEL, A_i, A_j, f)
   (e) A_j could need more information to give an answer and instead of answer with a rule it decides to do all by itself.
   In this case, A_j will make all necessary queries to other agents, for example: (QUERY, A_j, A_k, f), where $A_k \neq A_i \neq A_j$, and when it have a value it will send to A_i. This makes process go to the beginning.

It is important to notice that performatives `ACCEPT` and `REJECT` allows agents to have social commitments [6]. A social commitment is defined as a structure indicating that there is a debtor committed to an action relative to a creditor [9]. In our case, when A_j accepts, it assumes a commitment with A_i, which is reflexed in its goals list.

Phil, Agent Leader $(A_l) \Rightarrow$	$\Leftarrow$ **Karl, Agent Programmer** $(A_p) \Rightarrow$	$\Leftarrow$ **Vicky, Agent Designer** (A_d)
adapt-game@A_p accept-adjustments project-begins	mobile-hw-supports adjust-graphics@A_d adapt-game guarantee-impact	screen-128x128 accept-adjustments@A_l guarantee-impact@A_p adjust-graphics
rule r1	rule r2	rules r3 and r4

$T_5 =$ (false (0), slightly-true (st), quite-true (qt), very-true (vt), true (1))
r1: (adapts-game@A_p $\rightarrow$ project-begins,0) r2: (mobile-hardware-supports $\wedge$ adjust-graphics@A_d $\rightarrow$ adapt-game,0) r3: (screen-128x128 $\rightarrow$ adjust-graphics,qt) r4: (accept-adjustments@A_l $\wedge$ guarantee-impact@A_p $\rightarrow$ adjust-graphics,0)

Figure 1. Mobile games company example.

4. Example

Consider a very simple scenario with three agents: Phil, Karl and Vicky; project leader, programmer and graphic designer respectively of a mobile games company. A new project has to be developed and Phil wants to know if Karl can do it.

–Phil (1): Hi Karl, there is a new project to adapt game *Drakon* for the mobile model *WX3*. Can you adapt it?
–Karl (2): Hi Phil, I will see the mobile and game information and I promise you to have an answer as soon as possible.
(To answer, Karl needs to analyze mobile hardware and to talk with Vicky. He sends her an e-mail with all information about the game and the mobile and call her later. Vicky analyzes the information. She knows that if minimum screen resolution is 128x128 pixels then it is possible to adjust graphics. But, for a definitive answer she would need to talk with Karl)
–Karl (3): Vicky, I sent you information about a new project, do you think you can adjust those graphics for model *WX3*?
–Vicky (4): Hi Karl, I think it is possible. However, I need to know if you guarantee me that the game will not lost its impact in users.
–Karl (5): Don't worry Vicky, I assure you the game won't lose its impact. Now, can I tell Phil that we will adapt the game?
–Vicky (6): One more thing Karl, I need Phil's agreement to make the adjusts you are suggesting.
(Karl decides to talk directly with Phil about it)
–Karl (7): Phil, I had to talk with Vicky because if she makes some graphic adjusts I will be able to adapt *Drakon*. She said that if you agree with those adjusts, she will make them.
(At this point, Phil has all the information to know if *Drakon* can be adapted or not)

In Figure1 we can see the set of fact and rules of the agents. Now, let's see their initial state:

$$A_l \begin{cases} I_l = \{(\text{adapt-game}, A_p)\} \\ O_l = \{\text{project-begins}\} \\ G_l = \{\text{project-begins}\} \\ M_l(\text{adapt-game}) = (0, \emptyset) \\ M_l(\text{accept-adjustments}) = (0, \emptyset) \\ M_l(\text{project-begins}) = (0, \{(\{\text{adapt-game}\}, 1)\}) \end{cases}$$

$$A_p \begin{cases} I_p = \{(\text{adjust-graphics}, A_d)\} \\ O_p = \{\text{adapt-game,guarantee-impact}\} \\ G_p = \emptyset \\ M_p(\text{mobile-hw-supports}) = (0, \emptyset) \\ M_p(\text{adjust-graphics}) = (0, \emptyset) \\ M_p(\text{guarantee-impact}) = (0, \emptyset) \\ M_p(\text{adapt-game}) = (0, \{(\{\text{mobile-hw-supports,adjust-graphics}\}, 1)\}) \end{cases}$$

$$A_d \begin{cases} I_d = \{(\text{accept-adjustments}, A_l), (\text{guarantee-impact}, A_p)\} \\ O_d = \{\text{accept-adjustments,guarantee-impact, adjust-graphics}\} \\ G_d = \emptyset \\ M_d(\text{screen-128x128}) = (0, \emptyset) \\ M_d(\text{accept-adjustments}) = (0, \emptyset) \\ M_d(\text{guarantee-impact}) = (0, \emptyset) \\ M_d(\text{adjust-graphics}) = (0, \{(\{\text{screen-128x128}\}, qt), (\{\text{accept-adjustments,guarantee-impact}\}, 1)\}) \end{cases}$$

(1) A_l has the objective to begin a new project. According to rule $r1$, A_l depends on A_p, therefore it sends a query: `(QUERY,`A_l`,`A_p`,adapt-game)`

(2) A_p can accept or reject it, let's suppose in this example that all agents will always accept queries, then it sends: `(ACCEPT,`A_p`,`A_l`,adapt-game)` and adds a new goal to its G_p list. To achieve this goal, A_p needs to know if the game can be programmed for that mobile model (this depends on mobile hardware and A_p gets this information by itself, reading the mobile guide, and assigns a value of vt). When A_p gets this value, it proceeds to specialize. Now the mental state of A_p is:

$$A'_p \begin{cases} I_p = \{(\text{adjust-graphics}, A_d)\} \\ O_p = \{\text{adapt-game,guarantee-impact}\} \\ G_p = \{\text{adapt-game}\} \\ M_p(\text{mobile-hw-supports}) = (vt, \emptyset) \\ M_p(\text{adjust-graphics}) = (0, \emptyset) \\ M_p(\text{guarantee-impact}) = (0, \emptyset) \\ M_p(\text{adapt-game}) = (0, \{(\{\text{adjust-graphics}\}, vt)\}) \end{cases}$$

(3) The value of the rule remains very high, then it is possible to adapt the game but A_p needs to know if A_d can adjust the graphics. A_d in turn will query A_p and A_l.

(4 & 5) A_d has two rules to get adjust-graphics value, one of them only needs own information and the other one needs information from other agents. Consider there is no problem with the screen resolution. According to the original conversation, Vicky talks with Karl about game impact: `(QUERY,`A_d`,`A_p`,guarantee-impact)`, and the answer is: `(INFORM,`A_p`,`A_d`,{((1,`$\emptyset$`),definitive)})`.

$$A'_d \begin{cases} I_d = \{(\text{accept-adjustments}, A_l), (\text{guarantee-impact}, A_p)\} \\ O_d = \{\text{accept-adjustments,guarantee-impact,adjust-graphics}\} \\ G_d = \{\text{adjust-graphics}\} \\ M_d(\text{screen-128x128}) = (1, \emptyset) \\ M_d(\text{accept-adjustments}) = (0, \emptyset) \\ M_d(\text{guarantee-impact}) = (vt, \emptyset) \\ M_d(\text{adjust-graphics}) = (qt, \{(\{\text{accept-adjustments}\}, vt)\}) \end{cases}$$

(6 & 7) It is interesting to consider the meaning of the current mental state of A_d: *with the current information* adjust-graphics *is quite true, but if Phil considers that* accept-adjustments *were true then* adjust-graphics *will be very true*. A_d needs one more value from A_l. It can ask A_l, but it decides to pass the job to A_p, and sends this new rule: `(INFORM,`A_d`,`A_p`,{((adjust-graphics;p;{(accept-adjustments,vt)}),provisional)})`. A_p can do nothing with this rule; it could ask to A_l about accept-adjustments but this is not an exportable

fact, then A_l can not give any answer. So that, A_p sends its own rule together with A_d rule.

$$A_p'' \begin{cases} I_p = \{(\text{adjust-graphics}, A_d), (\text{accept-adjustments}, A_l))\} \\ O_p = \{\text{adapt-game,guarantee-impact}\} \\ G_p = \{\text{adapt-game}\} \\ M_p(\text{mobile-hw-supports}) = (qt, \emptyset) \\ M_d(\text{adjust-graphics}) = (qt, \{(\{\text{accept-adjustments}\}, vt)\}) \\ M_p(\text{guarantee-impact}) = (0, \emptyset) \\ M_p(\text{adapt-game}) = (0, \{(\{\text{adjust-graphics}\}, vt)\}) \end{cases}$$

(8) A_l has now all the necessary information to say if project-begins is *quite true* or *true*. Depending on the value of finaldecision it will be qt—when finaldecision is false—or vt—when it is true.

$$A_l' \begin{cases} I_l = \{(\text{adapt-game}, A_p)\} \\ O_l = \{\text{project-begins}\} \\ G_l = \{\text{project-begins}\} \\ M_d(\text{adjust-graphics}) = (qt, \{(\{\text{accept-adjustments}\}, vt)\}) \\ M_p(\text{adapt-game}) = (0, \{(\{\text{adjust-graphics}\}, vt)\}) \\ M_l(\text{accept-adjustments}) = (\text{finaldecision}, \emptyset) \\ M_l(\text{project-begins}) = (0, \{(\{\text{adapt-game}\}, 1)\}) \end{cases}$$

5. Conclusions

In this paper we have presented how the specialization of rule-based knowledge bases can be the central mechanism to deliberate and also to produce *reasonable* dialogs among conversational agents [18,3]. Agents communicate exchanging data and knowledge in the form of conditional answers to solve their goals in a collaborative manner. The contents of the messages can be part of the mental state of agents, containing only public information. We believe that this model makes sense when we manage imperfect information: vague, imprecise and incomplete. In this case the specialization mechanism give new opportunities of richer conversations by using in each moment the more precise information to drive the questioning/answering protocols.

One important point not covered in this paper is related to the use of negation in the conclusions of rules. In our complete language a fact a has the value $[\alpha, \beta]$ because rules concluding a are responsible of α (the minimum of the interval) and rules concluding $\neg a$ of β (the maximum). More certain rules produces more precision for the conclusion. Provisional values for facts are those less precise that can be used also to produce provisional specialization and so provisional values for other facts.

Another important issue is time. It may be reasonable to think in different strategies of specialization using provisional values, i.e. when a concrete timeout has been reached or when we need a value, we can use a less precise but useful result, similar to *anytime* algorithms. The pass of time gives an opportunity to increase the accuracy, then the goals of agents can persist until it is no possible to obtain more precise values.

What we need to do now is to carry out experiments to see which are the emergent conversations among agents; to study different strategies for obtaining information: in parallel, using provisional values, etc.; to study different kind of collaborative effort and delegation [5] and coordination [4]; and to extend our model by adding concepts related to the Electronic Institution model [1].

Acknowledgements.

Authors acknowledge partial support by the Spanish projects IEA (TIN2006-15662-C02-01), MULOG2 (TIN2007-68005-C04-01) and Agreement Technologies (CONSOLIDER CSD2007-0022, INGENIO 2010). Mariela Morveli-Espinoza is supported by the Programme Alβan, the European Union Programme of High Level Scholarships for Latin America, scholarship No.(E06D101440PE). We would like to thank the referees for their valuable suggestions and comments.

References

[1] Josep Lluis Arcos, Marc Esteva, Pablo Noriega, Juan A. Rodríguez-Aguilar, and Carles Sierra. Engineering open evironments with electronic institutions. *Engineering applications of artificial intelligence*, 18(2):191–204, 2005.

[2] J.L. Austin. *How to Do Things with Words*. Harvard University Press, 1962.

[3] M Barbuceanu and WK Lo. Conversation oriented programming for agent interaction. In *Issues in Agent Communication*, volume 1916 of *Lecture Notes in Artificial Intelligence*, pages 220–234, 2000.

[4] J Bentahar, B Moulin, and B Chaib-draa. Commitment and argument network: A new formalism for agent communication. In *Advances in Agent Communication*, volume 2922 of *Lecture Notes in Artificial Intelligence*, pages 146–165, 2003.

[5] SE Brennan and Hulteen. Interaction and Feedback in a Spoken Language System - A Theoretical Framework. *Knowledge-Based Systems*, 8(2–3):143–151, APR-JUN 1995.

[6] M. A Colombetti. Commitment-based approach to agent speech acts and conversations. In *Fourth International Conference on Autonomous Agents, Workshop on Agent Languages and Conversation Policies*, pages 21–29, 2000.

[7] Robert Demolombe. A strategy for the computation of conditional answers. In *Proceedings ECAI'92*, pages 134–138, 1992.

[8] U. Endriss, N. Maudet, F. Sadri, and F. Toni. Protocol conformance for logic-based agents. In *Proceedings IJCAI-2003.*, 2003.

[9] Roberto A. Flores and Rob Kremer. Bringing coherence to agent conversations. In *AOSE*, pages 50–67, 2001.

[10] Foundation for Intelligent Physical Agents (FIPA). *Communicative Act Library Specification*, 2002.

[11] T Fruhwirth. Specialization of concurrent guarded multi-set transformation rules. In *Logic Based Program Synthesis and Transformation*, volume 3573 of *Lecture Notes in Artificial Intelligence*, pages 133–148, 2005.

[12] Mark Greaves, Heather Holmback, and Jeffrey Bradshaw. What is a conversation policy? In *Issues in Agent Communication*, pages 118–131. Springer-Verlag: Heidelberg, Germany, 2000.

[13] K Iwanuma and K Inoue. Minimal answer computation and SOL. In *Logics in Artificial Intelligence 8th*, volume 2424 of *Lecture Notes in Artificial Intelligence*, pages 245–258, 2002.

[14] J. W. Lloyd and J. C. Shepherson. Partial evaluation in logic programming. *The Journal of Logic Programming*, 11(3/4):217–242, October/November 1991.

[15] J. Puyol, L. Godo, and C. Sierra. A specialisation calculus to improve expert system communication. In Bern Neumann, editor, *Proceedings of the 10th European Conference on Artificial Intelligence, ECAI'92*, pages 144–148, Vienna, 1992. Jonh Wiley & Sons, New York.

[16] J. Puyol-Gruart, L. Godo, and C. Sierra. Specialisation calculus and communication. *International Journal of Approximate Reasoning (IJAR)*, 18(1/2):107–130, 1998.

[17] J. Puyol-Gruart and C. Sierra. Milord II: a language description. *Mathware and Soft Computing*, 4(3):299–338, 1997.

[18] F Rago. Conversational agent model in intelligent user interface. In *Fuzzy Logic and Applications*, volume 2955 of *Lecture Notes in Artificial Intelligence*, pages 46–54, 2006.

[19] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 1995.

[20] Yoav Shoham. Agent-oriented programming. *Artificial Intelligence*, 60:51–92, 1993.