Anytime Reasoning Mechanism for Conversational Agents

Mariela MORVELI-ESPINOZA and Josep PUYOL-GRUART [1],

Artificial Intelligence Research Institute (IIIA).
Spanish Scientific Research Council (CSIC).

Abstract.
When an agent receives a query from another agent, it tries to satisfy it by building an answer based on its current knowledge. Depending on the available time or the urgency of the requirement the agent can produce answers with different levels of quality. Answers can contain the best one, a provisional one because it can be improved later, or a conditional answer because the agent ignores some information needed to build the answer. Agents always depend on the availability of information obtained from perception or from the communication with other agents. We assume that in the real world normally is better to receive an answer with poor quality than no answer. The answer can be good enough for the receiver or the receiver can spend more time to wait for a better answer. Autonomy implies taking the best decision with the available information, avoiding blocking situations and no action. In this paper, we propose an architecture for deliberative agents using anytime like reasoning to produce better answers as time increases.

Keywords. anytime algorithms, progressive reasoning, multi-agent systems, partial deduction, multiple-valued logic.

Introduction

Dean and Boddy first used the term *anytime algorithm* in the late 1980's [2]. The main characteristic of these algorithms is that the quality of its results can be measured and that it improves gradually as computation time increases. This kind of algorithms are normally related to real time, where the time granularity is thinner than the long time needed to calculate a complete solution. They are able to communicate the best result obtained when interrupted or they can establish a compromise to deliver it in a given time.

In the context of logics and knowledge-based systems some authors talks about *progressive (or anytime) reasoning* or *deduction* [3,6]. Anytime concepts are important for the techniques to build intelligent systems, for instance in probabilistic reasoning, ontologies or constrain propagation [4,12,14]. In multiagent systems, agents have particular goals. The conversations among deliberative agents aim to obtain information in order to produce solutions to those goals. In [5] we described how conversational agents could be

[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.

modeled. One important point not covered in that paper was related to the use of intervals of truth-values and negation in the facts and in the conditions and conclusions of rules. In our complete language [8] the value of a fact a is an interval of truth-values $[\alpha, \beta]$. Rules concluding a are responsible of α (the minimum of the interval) and rules concluding $\neg a$ of β (the maximum). Then we can introduce *quality* taking into account the precision of the intervals. We will consider values are *provisional* when it is possible to improve its precision using more information.

Another important issue not covered in that paper was 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. The pass of time gives an opportunity to increase the accuracy, and then the agent's goals can persist until it is no possible to obtain more precise values.

In this paper we will introduce how anytime or progressive reasoning based on specialization of rule-based systems can be the central mechanism to deliberate and also to produce *reasonable* dialogs among conversational agents [1,5,10]. Agents can produce answers with different levels of quality: containing the best, a provisional or a conditional answer. We assume that in the real world normally is better to receive an answer with poor quality than no answer. The answer can be good enough for the receiver or the receiver can spend more time to wait for a better answer.

In Section 1 we formally describe the specialization as an anytime mechanism of progressive reasoning. Section 2 is devoted to quality measures. We present the description of the agent and its pragmatics in Section 3. Comments on performance and validation of our approach are presented in Section 4. Finally, some conclusions and future work are developed in Section 5.

1. Specialization and progressive reasoning

Specialization [7,8,9] can be considered as an anytime algorithm because it allows to obtain information before the completion of the inference process. It can be considered also a mechanism for progressive reasoning because it is a technique that successively refines a solution while making available intermediate solutions. In the following we introduce briefly a simplified version[2] of the language and inference mechanism:

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

- *$T_n = \{t_1, t_2, \ldots, t_n\}$ is an ordered set of truth-values, where t_1 and t_n are the booleans* True *(1) and* False *(0) respectively. $Int(T_n) = \{[t_i, t_j] | i \leq j\}$ are intervals of T_n.*
- *Σ is a set of propositional variables (atoms or facts).*
- *Sentences S composed by: literals (a, V), $(\neg a, V)$, with $a \in \Sigma$ and $V \in Int(T_n)$ and rules of the form $(p_1 \wedge p_2 \wedge \cdots \wedge p_n \rightarrow q, [t_i, 1])$, where p_i and q are literals, and $\forall i, j (p_i \neq p_j, p_i \neq \neg p_j, q \neq p_j, q \neq \neg p_j)$*

[2] For the sake of simplicity here we use min operation instead of a general triangular norms. For more information please see [8].

We will use the following inference rules:

- Not-introduction: from $(a, [t_i, t_j])$ infer $(\neg a, [t_{n-j+1}, t_{n-i+1}])$
- Not-elimination: from $(\neg a, [t_i, t_j])$ infer $(a, [t_{n-j+1}, t_{n-i+1}])$
- Parallel composition[3]: from (a, V_1) and (a, V_2) infer $(a, V_1 \cap V_2)$
- Specialization: from $(p_i, [t_i, t_j])$ and $(p_1 \wedge \cdots \wedge p_n \rightarrow q, [t_k, 1])$ infer $(p_1 \wedge \cdots \wedge p_{i-1} \wedge p_{i+1} \wedge \cdots \wedge p_n \rightarrow q, [min(t_i, t_k), 1])$

The main component of the mental state of agents [13] is the knowledge base containing beliefs (facts) and knowledge (rules) for deliberation. In our model, both facts and rules are weighted with intervals of truth-values. Consider $R_q = R_q^+ \cup R_q^-$ the set of rules deducing the fact q. We can distinguish between the rules R_q^+ deducing (positive) q and those R_q^- deducing (negative) $\neg q$. Positive rules contribute to the minimum of the interval (positive evidences) and negative ones to the maximum (negative evidences).

The *specialization rule* above is the core of the progressive reasoning algorithm. When a rule is specialized it produces a new rule with less conditions and a new updated value. When a rule is totally specialized (there is no conditions) it produces a value for the literal of the conclusion. Given $(P \rightarrow q, [t_k, 1]) \in R_q^+$, the most precise value for that literal will be $[t_k, 1]$ because we use the min function in the specialization rule. Similarly, given $(P \rightarrow \neg q, [t_k, 1]) \in R_q^-$, the most precise value for that literal will be $[0, t_{n-k+1}]$.

Fact q is initially *unknown*, that is, its value is the most imprecise interval $[0, 1]$. Using the values obtained from totally—positive and negative—specialized rules we will obtain a more precise interval for q by means of the applications of *parallel composition rule*. Given a set of r rules R_q^+ with truth-values $\{[t_1, 1], \ldots, [t_r, 1]\}$ the most precise interval will be $[max_{i=1}^r(t_i), 1]$. Given a set of s rules R_q^- with truth-values $\{[t_1, 1], \ldots, [t_s, 1]\}$ the most precise interval will be $[0, min_{i=1}^s(t_{n-i+1})]$. Finally we can say that the most precise interval for q will be $[max_{i=1}^r(t_i), min_{i=1}^s(t_{n-i+1})]$. We have to take into account that each specialization step produces a new knowledge base and then the expected most precise interval will be changed.

The new rules are provisional if they are deduced with provisional information otherwise they are definitive. Facts are definitive if they are deduced with definitive information and there are no more rules that can improve its value. Depending on this, rules can be deleted or not, see Section 3.3.

2. Quality measures

Quality measures and their properties are important for anytime algorithms [15] . Quality has to be (i) Measurable and recognizable: the quality of an approximate result has to be determined precisely and easily at run time, and (ii) Monotonic: the quality of the result is a non-decreasing function of time and input quality.

Quality is evaluated based on a three-dimensional criterion that measures the level of certainty, precision and completeness of a given value, an interval of truth-values. The quality is determined based on the following characteristics:

[3]When the intersection of values is empty, then it is considered to be a contradiction in the knowledge base.

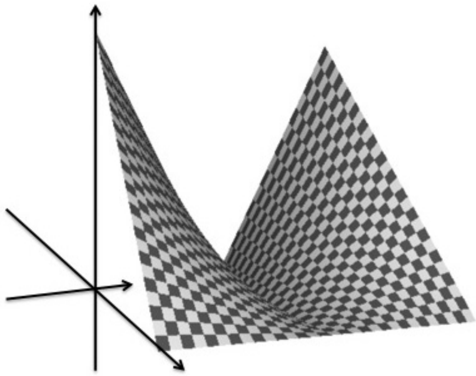

Figure 1. Representation of the quality function where $x, y, z \in [0, 1]$ are the middle point of the interval, the amplitude of the interval and the quality respectively. It is a symmetric function with respect to the plane $x = 0.5$

Certainty: In an approximate reasoning context we want to know the certainty and falsity of propositions. Then, given a set of knowledge deducing a fact we are interested in using those relations that provides values close to *true* or *false*.

Precision: Values of facts are intervals of linguistic terms. The most precise interval is when the difference between the maximum and the minimum is 0, and the least precise is when that difference is 1, that is, the only case $[0, 1]$, or *unknown*.

Completeness: To determine the value of a fact we need to know the values of other related facts. Given two answers, with the same level of certainty and precision, we will consider of more quality that with less number of definitive facts that could improve the result.

Precision and certainty are directly related because a good precision is interesting only when the value that represents is close to 0 or 1. Given a set of n truth-values, we can use the following expression to calculate a quality measure between 0 and 1:

$$q([t_i, t_j], n) = \left| \frac{(i + j - n - 1)(i - j + n - 1)}{(n - 1)^2} \right|$$

The first term of the numerator corresponds to the value represented by the middle point of the interval, better when more close to t_1 or t_n, that is, true or false. The second term corresponds to the precisions of the interval. It is a symmetric function with respect to the plane $i + j = n + 1$ and the divisor is a normalization constant. In Figure 1 we can see the shape of the function.

If all the facts used to deduce the goal would have a definitive value then the completenes will be of 100%. If all those facts would also have values—true of false with the maximum precision—such that the premises of rules was true then we will obtain the maximum quality degree. The current KB determines the maximum quality degree that can be obtained for a given fact. It is easy to see when the current quality degree of a given fact can be improved in the future.

3. Deliberative agents and anytime reasoning

The model of reasoning described above could take a long time to generate definitive results. This is not a consequence of the complexity of the deductive process. We consider that specialization time is irrelevant for our time restrictions, which are communication time, availability of agents, collaborative behavior, etc. The time granularity depends on the application but we have to take into account that the motivation is not classical real time.

We consider that agents have a deadline to answer a question. When an agent accepts a query, if necessary, it starts by asking other agents for information. But it cannot be waiting forever for the answers. When it is not possible to obtain a definitive value for a query and the deadline has been reached, it answers with less precision. Answers can contain the best one, a provisional one because it can be improved later, or a conditional answer because the agent ignores some information needed to build the answer.

3.1. Agents as anytime entities

Consider a multi-agent system with m agents $\mathcal{A}_m = \{A_1, \ldots, A_m\}$. Each agent has the following structure:

Definition 2 *(Agents) A deliberative agent is a tuple $A_i = \langle KB_i, G_i, I_i, O_i, t_i \rangle$ where:*

- *KB_i is the knowledge base of agent A_i.*
- *G_i is the set of goals of A_i. A goal g is a tuple $\langle x, A_j, t_b \rangle$, where $x \in \Sigma$, $A_j \in \mathcal{A}$ and t_b is the remaining time for deadline.*
- *I_i is the input interface of A_i, the set of external facts that can be obtained by querying other agents. These 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 of A_i; this is, the set of facts agent A_i can answer to other agents.*
- *t_i is the deadline for giving an answer.*

There are two type of anytime algorithms [11]: an *interruptible* one may be halted at any time and produces a result with a more or less good quality; a *contract* algorithm has a contract time—it must know the total allocation of time in advance—if interrupted at any point before the termination of the contract time, it might not produce results. We can consider that our agents has both anytime behaviors. It is a contract algorithm because the deadline is known in advance—autonomy gives agents freedom to define its own deadline independently of other agent's deadlines. It also could have an interruptible behavior because it can be asked at any time giving the current value, in the worst case *unknown*.

3.2. Agent architecture

An agent has a set of processes: (i) an *interface communication manager*, (ii) an *specialization engine*, (iii) an *answering machine*, (iv) an *evaluation machine* and (v) an *integration machine*; and a set of data repositories: the KB are the facts and rules of the problem domain, the current commitments (*goals*), and the data about other agents (*acquaintances*) and data about self, abilities and capacities (*competences*). In Figure 2 you can see an scheme of the relations among all these components.

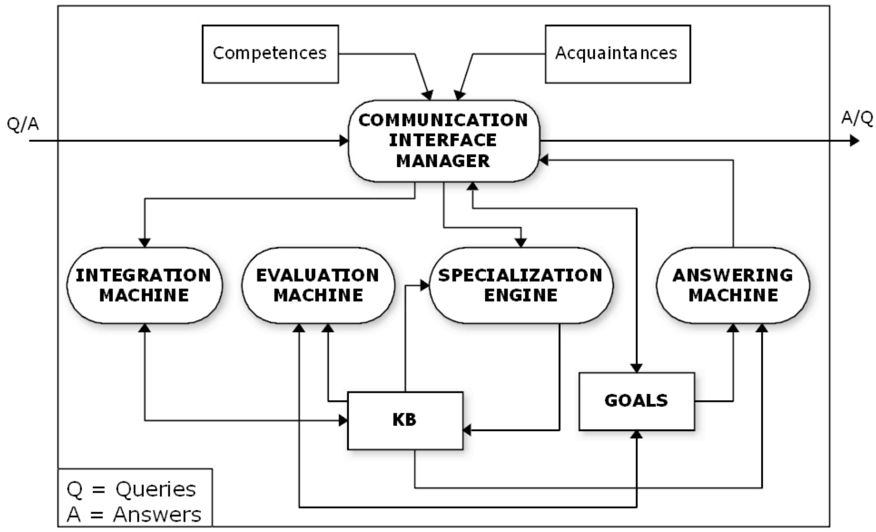

Figure 2. Agent architecture.

- *Communication Interface manager (CIM)* manages the input and output of queries and answers:

  * When it receives a query q and $q \in O_i$, a new goal is added to the goal list: $G_i := G_i \cup \{\langle q, A_j, t_f \rangle\}$.
  * When it receives an answer, it sends it to the *integration machine*.
  * It sends the answers and queries to the other agents, following the correct protocols and reporting all the activity.

- The *specialization engine* receives as inputs fact values and performs a *specialization cycle*: $S : KB \times f \rightarrow KB'$ is a data-driven process that begins when the input is a new fact value f. This triggers a complete specialization process over the KB and a new specialized KB' is generated.
- The *integration machine* receives as input a complete answer (facts and eventually a set of rules) and incorporates them into the KB.
- The *answering machine* receives as input a trigger signal indicating:

  i A goal deadline ends. If the goal doesn't have a definitive value, then the *answering machine* has to elaborate other kinds of answers (see Section 3.3)
  ii The definitive value for a goal is found, and then the obvious response is the definitive value.

- The *evaluator machine* is a goal-driven process $I : KB \times g \rightarrow g^*$ that begins when the agent process a goal g. It triggers a complete exploring process obtaining a set of new goals g^*, which are necessary to find values of g with better quality, as seen in Section 2.

3.3. Responses

One of the most important topics in our model is the different variety of answers agents can express:

Definition 3 *(Responses) A response is a tuple $R = \langle f, V, S, KB \rangle$ where:*

- *f is the fact which is been answered.*
- *V is the value of fact f (an interval of truth-values or linguistic terms).*
- *S is the state of the fact f value, i.e.* provisional*,* definitive *or* pending[4].
- *KB is a knowledge base useful to improve the value of f.*

Let's define now the kinds of responses the agent can give:

1. Definitive value $R = \langle f, V, def, \emptyset \rangle$: this is the most useful result because it means that there is no more information that can improve the result, this is the most precise. After the specialization we can substitute a rule using it by its specialized version.
2. Provisional value $R = \langle f, V, prov, \emptyset \rangle$: this is not a definitive value, it can be improved later. We can use it to produce only more provisional values. We can not delete rules that use it because they will be useful to produce more precise values.
3. Provisional value and a set of knowledge related to it, $R = \langle f, V, prov, KB_f \rangle$: this is similar to the case above but the answer includes all the information needed for improving the value. We can use this provisional value and start the mechanism to find more information.
4. A set of rules related to the question $R = \langle f, [0, 1], pending, KB_f \rangle$: the same that the case above but without a provisional value.

3.4. Evaluation cycle

When an *agent's life* begins and receives a simple query, the agent begins a goal-driven—backward chaining style—work. This task will produce new goals that have to be solved. The *evaluation machine* judges the impact of these new goals in the quality of the original one. Some of them can be internal and others have to be obtained from other agents. Internal goals are considered a self-commitment and the agent starts a search process in order to find which are the new goals it needs.

When new facts are known—maybe from other agents answers—it is started a data-driven task of specialization—forward chaining style. The transition from one solution to a more precise one happens in this specialization step.

An incomplete answer to a query is generated when there is no enough time to complete the query processing or there are agents that do not answer. Each agent goal could achieve a definitive or a provisional value. The *evaluation machine* decides if this value is enough. If further reasoning is required to improve the quality, new requirements are sent to the corresponding agents.

Agents can send and receive facts and rules as conditional answers or knowledge communication. When the deadline of a goal ends and it has a provisional value, the agent

[4] A pending fact is a fact that is provisionally unknown [5].

can send rules as part of the answer (see Section 3.3). There are sets of criteria that are out of the scope of this paper like privacy or protocol constraints that can limit the contents of rules in an answer. It is not necessary to send the provisional rules because with the provisional values of facts and the original rules we can easily deduce the provisional ones.

4. On performance and validation

Performance profiles [15] are used to measure how the quality of the output is improved over time. The simplest performance profile is a function $Q(t)$ representing the evolution of the quality with respect to time. Normally they have to be calculated using statistics over a set of inputs and they are normally monotonic functions. In our case it is easy to see that given the quality measures in Section 2 and the parallel composition rule (intersection) the performance profile of an agent is a monotonically non-decreasing function. New information improves the completeness and precision of the results.

There are other forms of performance profiles. For instance the conditional performance profile, where $P(q_{out}|q_{in}, t)$ is the probability of obtaining a result with quality q_{out}, given an input of quality q_{in} at time t. It is desirable that when the input quality improves the output quality also will do. From the specialization and parallel compositions rules we can see that it is true.

We have to study other type of performance profiles taking into account that our system is a multiagents system. We can think in social or join performance profiles and in the dependency of performance profile measures with respect to the deadline time of agents and other parameters.

It is obvious that an anytime system can produce answers before a standard one. The problem is to determine when this is or not an advantage. An agent can make a decision or execute an action depending on information of different quality. Depending on the available time or the urgency, the agent can accept a level of quality enough to exceed a particular threshold and then take action. Autonomy implies taking the best decision with the available information, avoiding blocking situations and no action.

Performance profiles can be objective measures of validation, but we are also interested in experimenting on subjective aspects of agent's behavior as the *emergence* of conversations among agents. We are interested in seeing if a very simple rule-based mechanism—as presented above—can produce conversations *similar to humans*.

5. Conclusions

In this paper we have presented an anytime mechanism for deliberative agents based on a monotonous reasoning over intervals of linguistic terms. There is a lot of things we plan to consider in the future. *Deadline* is considered fix for the sake of simplicity, but it could be variable and be calculated to improve agent's performance. Criteria like communication channels cost, confidence and agent's capacity can be considered for its estimation and effects on performance profiles. We have said that when an agent receives a provisional value it can be used to produce more provisional values, but we can think in a timeout or other rational subjective criteria to consider that a provisional value becomes definitive.

We are also designing a protocol to deal with provisional values and the knowledge received. It is reasonable to think that when a provisional value is received, agents can insist later in order to improve the value or use their own means to obtain that information.

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] 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.

[2] T. L. Dean and M. Boddy. An analysis of time-dependent planning. In *Proceedings of the Seventh National Conference on Artificial Intelligence, AAAI 88*, pages 49—54, 1988.

[3] Abdel illah Mouaddib and Shlomo Zilberstein. Knowledge-based anytime computation. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence, IJCAI 95*, pages 775–783, 1995.

[4] B. Jaumard and A. D. Parreira. An anytime deduction algorithm for the probabilistic logic and entailment problems. *International Journal of Approximate Reasoning (IJAR)*, 50(1):92–103, 2009.

[5] Mariela Morveli-Espinoza and Josep Puyol-Gruart. On partial deduction and conversational agents. In Teresa Alsinet, Josep Puyol-Gruart, and Carme Torras, editors, *Artificial Intelligence Research and Development*, volume 184 of *Frontiers in Artificial Intelligence and Applications*, pages 60–69. IOS Press, 2008.

[6] Abdel-Illah Mouaddib. A study of a dynamic progressive reasoning system. *Journal of Experimental and Theoretical Artificial Intelligence*, pages 101–122, 2000.

[7] 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, August 1992. Jonh Wiley & Sons, New York.

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

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

[10] 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.

[11] S. J. Russell and S. Zilberstein. Composing real-time systems. In *Proceedings of the Twelfth International Joint Conferences on Artificial Intelligence*, pages 212—217, 1991.

[12] S. Schlobach, E. Blaauw, M. El Kebir, A. ten Teije, F. van Harmelen, S. Bortoli, M. Hobbelman, K. Millian, S. Stam Y. Ren, P. Thomassen, R. van het Schip, and W. van Willigem. Anytime classification by ontology approximation. In Ruzica Piskac et al., editor, *Proceedings of the workshop on new forms of reasoning for the Semantic Web: scalable, tolerant and dynamic*, pages 60–74, 2007.

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

[14] Alan Verberne, Frank Van Harmelen, and Annette Ten Teije. Anytime diagnostic reasoning using approximate boolean constraint propagation. In *Constraint Propagation, Int. Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, pages 323–332, 2000.

[15] Shlomo Zilberstein. Using anytime algorithms in intelligent systems. *The AI magazine*, 17(3):73–83, 1996.