Advice-Giving Dialogue: An Integrated System

Didier Bronisz, Thomas Grossi, François Jean-Marie
Cap Gemini Innovation
Chemin du Vieux Chêne
38240 Meylan, France
email: {bronisz,grossi,francois}@capsogeti.fr

Abstract

In this paper we present the implementation of an *advice-giving system for financial investment* for the final phase of the project ESTEAM-316[1]. This system integrates multiple agents in a single architecture allowing cooperation between a natural language dialoguer, "intelligent" data base access modules, and a problem solver in the financial domain. Using a user model, this system adapts the mixed initiative dialogue during both the formulation of the problem and its resolution by the expert. A novice user thus has access to expert knowledge despite the weakness of his own knowledge.

1 The Demonstrator

In its final phase, the project was oriented towards the development of a demonstration prototype, incorporating various ESTEAM-316 research results and showing the feasibility of an AGES. The Cooperation Architecture is a conceptual framework for AGES design and a set of mechanisms to support implementation of that design. It is a cooperation architecture because it supports the active cooperation of independent components or modules of the AGES; it is thus the means for integration. Furthermore, it supports the integration of heterogeneous modules through encapsulation of modules as *agents*, and by providing module–module cooperation using any of three standard interaction models.

Our dialogue module integrates various subsystems, for example, a natural-language and intention recognition module, an expression generation module and a dialogue planning and management module. Great advances have been made in each of these domains but at present we are dealing only with the tip of the iceberg. In the project ESTEAM-316 we chose not to address specific issues such as explanation in great detail (there are many researchers already addressing these problems), but to build a general integrated system aimed at recognizing the user's intentions and answering him in an understandable way with expert knowledge. We will show how this system is able to incorporate modules which are more specialized in certain domains (problem solver or cooperative data base access).

2 The Agents

The cooperation architecture

To provide a design method and to support the efficient implementation of AGES, we have built a special architecture. The architecture aims to keep separate the design and implementation of internal Agent features from external features handling cooperative interaction. The architectural approach to communication allows Agents to share tasks by problem decomposition and to share knowledge by using special Agents for Data and Knowledge Base Management.

The main problems tackled in defining the architecture concern:

- support for the integration of heterogeneous Agents;
- control mechanisms for managing the interactions between Agents solving subproblems;
- efficient communication between Agents which share or exchange information.

The Problem Solver

The Problem Solver citehanet incorporates the results of two major research efforts: logic-based modeling and encoding, and explanation generation based on proof trees. The Problem Solver is the module which solves the user's problem(s) in a financial domain and provides proof trees that the Dialogue Manager can use to generate explanations. This Problem Solver module is based upon a knowledge representation formalism which integrates and object-oriented approach and logic, and is implemented in Prolog [1,2].

The Cooperative Answering Module

In the context of traditional applications devoted to company management, like payroll computation, people or programs who have to access data in a

[1] This work was supported in part by the Commission of the European Communities as ESPRIT project 316 (ESTEAM-316).

Database have a very precise definition of the data they want to access. There are many other applications where people want to access data in order to make a decision, or to solve a problem whose solution cannot be found applying a simple algorithm. An important feature of this context, from the point of view of data retrieval, is that users don't have a precise idea of the data which can help them to solve a problem, or to make a decision. The objective of the Cooperative Answering module is to simulate the behavior of a person who wants to help as much as possible an interlocutor who asks them a question. That is, to try to understand why this interlocutor asks this question, and to determine what additional interesting information, not explicitly requested, could be provided in addition to the answer[3].

the Dialogue Manager

In the final integrated AGES demonstrator, the main problems are controlling the cooperative functioning of several distinct sources of knowledge, finding ways of representing and reasoning about diverse matters, and integrating the different contributions to perform intelligently a single task, namely advice-giving. The role of the Dialogue Manager is to provide a "friendly", comprehensible user interface that makes available to the user all the expertise of the various modules of the system in a flexible, uniform fashion [4,6,7]. We present in the next section a simple scenario in order to provide its functionalities.

3 A sample dialogue

Let us consider the following session.

1 *System* : Hello, tell me about your investment plan.

2 *User* : I'd like to buy a car in two years.

3 *System* : Ok. An investment plan comprises three subparts: an *emergency fund*, a *fixed-term investment* and a *long-term investment*. How much do you want to invest?

4 *User* : $20,000.

5 *System* : Let's begin with the emergency fund. How much do you want to put into an emergency fund?

6 *User* : Let's talk about my car first.

7 *System* : Ok. How much money do you want to put into the fixed-term investment?

8 *User* : Oh, let's say $7,500.

9 *System* : I recommend that you put the money into a liquid security. Would you like me to propose one?

10 *User* : Yes.

11 *System* : Put your money into a 4.5 % savings account at Sofibanque. Let's return to the emergency fund, now. How much do you want to put in it?

12 *User* : $5,000.

13 *System* : Tell me also the delay.

14 *User* : 2 days.

15 *System* : Ok. I recommend that you put the money into a liquid security. Would you like me to propose one?

16 *User* : Are there any savings accounts with more than 4.5 % interest?

17 *System* : Barclay's has a savings account at 5 % interest, but it is in England

18 *User* : Ok, I'll take Barclay's.

19 *System* : So, that leaves you with $7,500 for your long-term investment, is this ok?

20 *User* : I wanted $10,000.

21 *System* : If you want an investment plan for $20,000 with an emergency fund of $5,000 and a fixed-term investment of $7,500, it only leaves $7,500 for a long term investment. If you want to put $10,000, then you should modify something else in your plan.

22 *User* : Could I put $2,500 into the emergency fund?

23 *System* : Ok. I recommend for the emergency fund a bank account at Sofibanque earning 4.5%, you can put $10,000 into the long-term investment. What risk-level would you like?

24 *User* : What is the risk-level of a long-term investment?

25 *System* : The risk-level is represented by an integer between 1 to 10. It shows the risk you want to have for your portfolio. So, what risk do you choose?

26 *User* : What if I put a risk-level of 5?

27 *System* : In this case, I recommend ...
Do you accept?

28 *User* : And if the risk-level is 2?

29 *System* : In this case, I recommend ...
Which one do you prefer?

30 *User* : The first one

31 *System* : Ok. So, to sum up, you want to invest $20,000, with $2,500 in an emergency fund with a delay of 2 days, $7,500 in a fixed-term investment to buy a car in two years, and $10,000 in an long-term investment. Is that right?

32 *User* : That's fine

33 *System* : Ok. Goodbye!!

Analysis

From this dialogue, we can easily demonstrate the different points we mentioned above.

- In sentence 6 the user prefers to begin the discussion with the fixed-term investment, while the system began the discussion with the emergency fund. The system changes the order of the different topics according to the will of the user and the feasibility of such an exchange.

- In sentence 12, the user does not say "The amount of the emergency fund is $5000" but only "$5000". The system must understand such an elliptical response. In the same way, the system does not ask "Tell me the delay of the emergency fund of your Investment plan?" but only "Tell me the delay?". Normally, the two participants hide a lot of information. They just give sufficient information to have an unambiguous dialogue. But having a partial knowledge necessitates being able to confirm sometimes what has been understood. In sentence 7, the system asks the user for a confirmation of the value of the fixed-term investment (2 years) which was previously given by him in the sentence 2. The system has understood that buying a car is equivalent to having a fixed-term investment, but it asks him to verify this supposition.

- In sentence 24 the user begins a digression in order to have an explanation about a new term introduced by the system in asking for a value. The system has to recognize this new user's intention, cope with (it may be longer than a two-turn dialogue (User, system) as in the discussion of a given solution) and come back to the previous dialogue.

- In sentence 10 the user accepts the system's offer to come up with a detailed investment plan, while in sentence 16 he decides to find out about other possibilities.

- In sentences 26 – 30 the system and the user explore the implications of a modification of one of the parameters.

All these considerations appear in any discussions, independently of the topics. There is implicit information (abbreviations, speaking manner) used in a discussion in a given domain. The dialoguer we are building does not handle such implicit information. We focus our attention on the domain-independent aspect of the dialogue organization. However, the advice-giving system must be able to explain what it does and how it does it, and also what the other components it is interfaced with do. We added, therefore, these domain-functionalities in order to implement a system able to help a *novice* user as well as an *experienced* one.

The system has to adapt its utterances and its explanations according to its perception of the user's knowledge. For instance, when the system presents a portfolio, it hides irrelevant information for a *novice* user while it shows it for the *experienced* one. In the same way, the system tries to use the user's vocabulary. In the example given above, the system uses the word *car* rather than *fixed-term investment* in order to help the user's understanding.

4 Conclusion

The integrated system elaborated in the project ESTEAM-316 is a demonstrator incorporating as much of the technology developed within Esteam as possible. This experience comes from research in the areas of Knowledge Representation, Cooperative Answering, Intentional Answering, Deductive Databases and Mixed-Initiative interfaces (also see the deliverables and the prototypes issuing from the project). The imitation of human behavior in the domain of the advice-giving remains very delicate and elaborate but we believe we have made a significant contribution.

Acknowledgements

We want to thank our partners within Esteam-316: CSELT in Torino, CERT in Toulouse and Philips in Brussels.

References

[1] A. Bruffaerts and E. Henin, Proof Trees for Negation as Failure: Yet Another Prolog Meta-Interpreter, in: *Logic Programming, Proc. of the Fifth International Conference and Symposium*, Seattle, August 15-19, 1988.

[2] A. Bruffaerts and E. Henin, Some Claims about Effective Explanation Generation in Expert Systems, in: *Proc. of the AAAI'88 Workshop on Explanation*, Saint Paul, August 22, 1988, 83-86.

[3] F.Cuppens, R.Demolombe. Cooperative Answering : a methodology to to provide intelligent access to a Database. Proceedings of 2d Int.Conf. on Expert Database Systems. The Benjamin/Cummings Publishing Company. 1988.

[4] P. Decitre, T. Grossi, C. Jullien, J.-P. Solvay, "Planning for Problem Formulation in Advice-Giving Dialogue." *Proceedings of the Association of Computational Linguistics*, 1987.

[5] M. Hanet, The Problem Solver, in: Deliverable no. 16, ESTEAM-316 Esprit Project, July 1989.

[6] C. Jullien, J.P. Solvay, "Person-Machine Dialogue for Expert Systems : The Advice-Giving Case." *Proc. 7th Int. Workshop Expert Systems & their applications*, Avignon, 1987.

[7] C. Jullien, J.C. Marty, "Plan Revision in Person-Machine Dialogue." *4th conf. of the European Chapter of the Association of Computational Linguistics*, Manchester, April 1989.