Intelligent Travel Planning: A MultiAgent Planning System to Solve Web Problems in the e-Tourism Domain

DAVID CAMACHO dcamacho@ia.uc3m.es
DANIEL BORRAJO dborrajo@ia.uc3m.es
JOSE M. MOLINA jmolina@ia.uc3m.es
Avda. de la Universidad, n.30, P.O. 28911, Leganes, Spain

Abstract. This paper presents Intelligent Travel Planning (ITP), a multiagent planning system to solve Web electronic problems in the Web, whose main goal is to search for useful solutions in the *electronic-Tourism* domain to system users. The system uses different types of intelligent autonomous agents whose main characteristics are cooperation, negotiation, learning, planning and knowledge sharing. Obviously the information used by the intelligent agents is heterogeneous and geographically distributed, since the main information source of the system is Internet. Other information sources are agent knowledge bases in the distributed system. The process to obtain, filter, and store the information is performed automatically by agents. This information is translated into a homogeneous format for high-level reasoning in order to obtain different partial solutions. Partial solutions are reconstructed into a general solution (or solutions) to be presented to the user. The system will show a set of solutions to the users that can be evaluated by them.

Keywords: information system, agent architectures, designing agent systems, multiagent systems

1. Introduction

There are a vast amount of information stored in the WEB available for any user connected to the network. This information is heterogeneous and distributed, so is impractical to build a single unified system that combines all of the possible information sources for any user. Currently, the only way to do this is to build specialized applications, which are difficult to maintain and to develop [11, 12]. This situation has originated several problems, one of the most important one is how an user could reuse all the available information to obtain satisfactory results when he is trying to obtain a solution to his problems. Currently, there are many systems that extract, filter and represent efficiently the information obtained from the Web. However, most of those systems are focused, mainly, on the amount of the information to be retrieved [6].

ITP is a distributed and cooperative Multiagent system to problem solving in the Web. ITP is based on the *agent* concept [10] in particular as any MAS [3, 15], ITP presents the following characteristics:

—Each agent has an incomplete amount of information or does not have the accurate abilities to solve the whole problem.

—It does not exist a global system control.
—Data is not centralized, so it must be shared by all agents.
—System execution is asynchronous; any agent can be working while it receives queries anytime.

Usually DAI [1, 8] has been studying and designing systems that are able to interact with other systems and with the users. MAS is a very active field in DAI, because it offers modularity and flexibility, these characteristics are essential in complex, large or unpredictable domains [13]. These systems are built using a set of modular components (agents) that are specialized at solving a particular problem aspect. This descomposition allows each software agent [2, 9], to use the most appropriate paradigm for solving its particular problem. When interdependent problems arise, the coordination among agents ensure that interdependencies are properly managed. This paper presents a distributed multiagent system that obtains data from the Web, reasons with it and obtain solutions that are finally suggested to a user.

ITP integrates traditional problem solving techniques, like planning and learning, with the advantages that provide distributed systems, like MAS. The introduction of planning in the MAS system through the concept of *planning agent* allows a high level of reasoning process. This high level reasoning could be shared among agents to obtain complex solutions. The introduction of learning in agents allow the system adaptation, to the user and to the dynamic environment. This paper analyzes two main characteristics of the system behaviour. First, how the information acquired from the Web is used to obtain a set of new solutions, and second how the system re-uses its own stored solutions (or solutions obtained from other agents) to gain efficiency in problem solving.

This paper is divided into six sections: Section 2 analyzes briefly the domain application (*e-Tourism* domain); Section 3 describes ITP architecture and its main goals; Section 4 presents an example over the Web using the designed system; Section 5 presents how the system will be evaluated, and finally, Section 6 shows the conclusions of the paper.

2. Electronic tourism domain

An electronic travel agent must have the ability to manage a travel planning. In this kind of domain (*e-tourism domain*) [5], the knowledge of the agent represents several towns, transports, lodging places, representing how the user could travel (through *planning operators*) between towns, airports, train stations, rent a car, or book a room in a hotel, etc....

The most important points in the management of a travel are:

1. Moving from the origin to the destination town.
2. Lodging at destination.
3. Local Transport possibility at target town,
4. Returning to initial (or other) town.

To handle item 1 and item 3, so far we have considered the airplane, train or bus as travel transport. To handle item 2, the system can use a set of information agents that can access to hotels and other lodging information available in the Web. Finally, the user may need to take a local train, bus or taxi to move to the airport, train station or bus station. The moving possibility around the target town refers to the possibility of renting a car, or to provide to the user information about public transport.

3. System architecture

ITP is a multiagent research tool whose main goals are:

—Solve travel problems given by a user. The user can propose to ITP his desired travel, and it will obtain a complete plan that include information about transport, lodging, etc.
—Extract, filter and store information automatically from the Web. The system uses the same information that the user could find if he wish planning the travel himself.
—Share different kinds of knowledge to gain efficiency in the problem solving task. The system could reuse old solutions or data stored by other agents in the system.
—Search for different solutions and customize the system behaviour to the users characteristics. The system learns from the user profile and try to adapt it possible request to the learned preferences.

ITP is a MAS approach that integrates a set of heterogeneous agents [4]. It could be summarized as:

UserAgent: This agent handles a user query and shows him the solution. To do so, it analyzes the problem and obtains an abstract representation. Subsequently it requests a *PlannerAgent* solutions to that problem. The userAgent has different skills like communication with PlannerAgents and users, or learning the user's profiles necessary to customize the system answer. The UserAgent has a set of interfaces to allow input and output information and the user evaluation of the solutions found.

PlannerAgent: The main PlannerAgent goal is reason about UserAgents and other PlannerAgents problems, and find out a set of possible solutions. PlannerAgents have different skills like communication (with different agents in the system), planning (its main reasoning module) and learning. The process of planning are made in two main steps, in first place, the system uses a classical[1] planner to obtain a set of abstract solutions that represent the set of necessary steps to solve the problem, in second place, case-base planning and automatically Web access are used to obtain specific information to complete these general steps that build the abstract solutions.

WebBot: These agents fill in the details (requested by PlannerAgents) obtaining the required information from Internet. Different partial solutions given by the

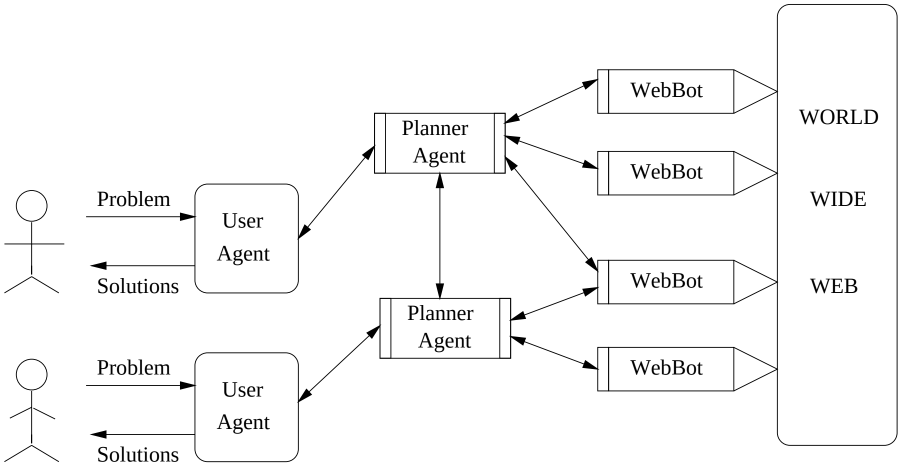


Figure 1. ITP general Architecture.

WebBot agents are combined by the PlannerAgents to obtain a detailed solution (or solutions) to the UserAgents queries.

CoachAgent: This agent controls a set of agents, manage tasks like register or unregister agents, search for new agents to help other agents group, suspend communication with any agent, order any task to a agent, etc. . ..

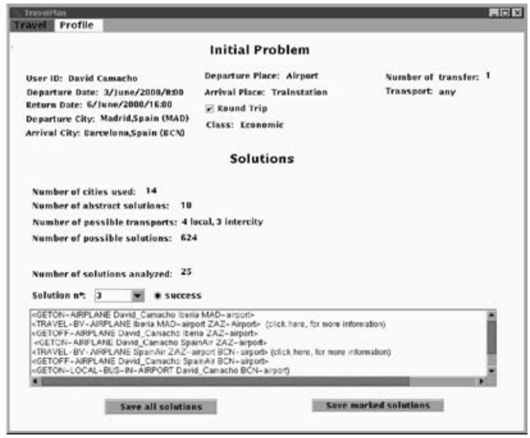
In Figure 1, a graphic representation of ITP is shown. The system is made by a set of agents that can communicate and cooperate among them to reach the problem solution.

All the agents in ITP use a common language representation to allow knowledge sharing, cooperation and other system characteristics, it has been implementated a set of Java performatives based on KQML [7].

4. Example of ITP application

Interaction among users and ITP is through the UserAgents. This kind of agent uses interfaces to communicate with the users. In this example an user wishes to travel to Barcelona (Spain) from Madrid (Spain) in June (see Figure 2(a)). The PlannerAgent returns 25 solutions to the user, from 624 possibles solutions. The system rejected a set of solutions in order to gain in efficiency, and only a subset of solutions are shown to the user. ITP suggests a particular solution if this solution matches with the learned preferences from the user (see Figure 2(b)). To do this, the UserAgent that pays attention to the users, extracts the main characteristics from the old stored solutions by the user and uses them to classifying all the possible solutions. Once a solution is selected, the user can consult especific information about the flight retrieved by WebBots from Internet.

(a) UserAgent input gui.



(b) UserAgent output gui.

Figure 2. ITP ⇒ UserAgent Interfaces.

In the example, we introduced a set of examples where their main characteristics were:

—The user always uses a particular airplane company (Iberia Airlines).
—All the solutions try to minimize *time*.
—Most of the examples use airplane or bus companies, but none use train companies.

In Figure 2(b) a possible solution founded by the system is showed to the user. The user should evaluate the solution given (in the example he will mark it with success), and this evaluation will be used by the system to learn the user preferences[2].

5. System evaluation

To allow the system evaluation it have being used two main approaches:

User evaluation: Any user that are using ITP could evaluate the system through the interfaces that the UserAgent showed to him (see Figure 2(b)). Any user can use the Profile interface to customize system behaviour to his preferences.

Automatic system evaluation: The CoachAgent has several statistical capabilities like: time to generate first solution, time to generate all the possible solutions, number of successful solutions found, etc. . . .

To evaluate the system we are developing two different types of experiments. In the first place a set of users that are evaluating the system solutions it have been employed, with these evaluations it is possible appreciate *solutions quality* from humans perspective. The second type of experiments are focused to measure performance system in the problem solving task, using only a PlannerAgent (molitical approach) or several PlannerAgents (multiagent approach) that can cooperate sharing planning skills and old store solutions.

6. Conclusions

We have presented a multiagent approach to problem solving in a dynamic environment. The motivation for this work is to develope a system that could reason with dynamic information and use planning processes to find the problem solution. We also studied how problem solving cooperative techniques can be used with data stored in the Web and in the system agents. We show a Web system application example, using data automatically obtained by the system to solve the user problem.

Notes

1. Actually it has being used PRODIGY4.0 planner [14] developed at Carnegie Mellon University.
2. Currently we are developing a set of new interfaces to allow a best hint evaluation.

References

1. A. H. Bond and L. Gasser, *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann: San Francisco, CA, 1988.
2. J. Bradshaw, *Software Agents*, AAAI: Press, Menlo Park, CA, 1997.
3. W. Brenner, R. Zarnekow, and H. Wittig, *Intelligent Software Agents. Foundations and Applications*, New York: Springer-Verlag. ISBN: 3-540-63411-8, 1998.
4. D. Camacho, D. Borrajo, and J. M. Molina, "TravelPlan: A multiagent system to solve Web electronic travel problems," in *Workshop on Agent-Based Recommender Systems, Fourth International Conference on Autonomous Agents*, Barcelona, Cataluña, Spain, 2000a.
5. D. Camacho, J. M. Molina, and D. Borrajo, "A multiagent approach for electronic travel planning," in *Second International Bi-Conference Workshop on Agent-Oriented Information Systems (AOIS-2000), Seventeenth National Conference on Artificial Intelligence (AAAI-2000)*, Austin, TX, 2000b.
6. O. Etzioni, "Moving up the infomation food chain," *AI Magazine*, vol. 18, no. 2, pp. 11–18, 1997.
7. T. Finin and J. Weber, *Draft Specification of the KQML Agent Communication Language*, June 15, 1993.
8. L. Gasser, 1992, "An overview of DAI," in *Distributed Artificial Intelligence: Theory and Praxis,* Kluwer Academic: Dordrecht, 1992, pp. 9–30.
9. M. R. Genessereth and S. P. Ketchpel, "Software agents," *Communications of the ACM*, vol. 37, no. 7, pp. 48–53, 1994.
10. M. N. Hunhs and M. P. Singh, *Readings in Agents*, Morgan Kaufmann: San Francisco, CA, 1997.
11. C. A. Knoblock and J. L. Ambite, "Agents for information gathering," in *Software Agents,* J. Bradshaw (ed.), AAAI/MIT Press: Menlo Park, CA, 1997.
12. C. A. Knoblock, S. Minton, J. L. Ambite, and N. Ashish, "Modeling Web sources for information integration," in *Proceedings of the Fifteenth National Conference on Artificial Intelligence,* 1998.
13. K. P. Sycara, "Multiagent systems," *AI Magazine*, vol. 18, no. 2, 1998.
14. M. Veloso, J. CarbonelI, A. Pérez, D. Borrajo, E. Fink, and J.Blythe, "Integrating planning and learning: The prodigy architecture," *Journal of Experimental and Theoretical AI*, vol. 7, pp. 81–120, 1995.
15. M. Wooldridge and N. R. Jennings, "Intelligence agents: Theory and practice," *Knowledge Engineering Review,* 1994.