Send Fredo off to do this, send Fredo off to do that[1]

Luís Botelho, Hugo Mendes, Pedro Figueiredo, and Rui Marinheiro

ADETTI/ISCTE Av. das Forças Armadas, Edifício ISCTE, 1600-082 Lisboa, Portugal
{Luis.Botelho, Hugo.Mendes, Pedro.Figueiredo, Rui.Marinheiro}@iscte.pt; http://we-b-mind.org/

Abstract. Fredo is a generic domain-independent broker that creates value-added information taking into account the preferences specified by its clients. Fredo uses ontology services and yellow pages services to discover a set of agents that can provide information relevant to its clients' requests. Fredo uses an intelligent heuristic strategy based on a fuzzy evaluation mechanism to plan the queries it uses to gather relevant information for its clients' needs. In order to handle possible information overload, we have designed a special purpose interaction protocol, the paged information-request protocol, which is used to govern the interaction between Fredo and information providers. Fredo also uses a fuzzy inference engine to evaluate the gathered information with respect to the preferences specified by its clients. Fredo has been developed by and used in the Agentcities project. Fredo uses the FIPA ACL inter agent communication language with FIPA SL contents. It was implemented in JAVA and Prolog and runs on FIPA++, a FIPA compliant agent platform.

1. Introduction

In open agent systems such as that of the Agentcities project [17] it is important to have agents capable of dynamically creating value-added information for the user or for other agents. Further more, the process by which information is sought, integrated, and evaluated should be as independent as possible of the particular application domain so that it can be used in different domains, relying on different ontologies.

Fredo is a broker agent capable of searching information from various sources, pertaining diverse topics, integrating it in coherent ways and evaluating it according to specified preferences. Fredo uses absolutely general algorithms in the sense that they are totally independent of the application domain.

Fredo was built using some original proposals and ideas from several other authors and research groups. The main contributions to the state of the art of engineered software agents are the integration of several ideas put forth by other researchers in a single implemented agent working in an open environment; the original proposal

[1] The research described in this paper is partly supported by the EC project Agentcities.RTD, reference IST-2000-28385 and partly by UNIDE/ISCTE. The opinions expressed in this paper are those of the authors and are not necessarily those of the Agentcities.RTD partners. The authors are also indebted to all other members of the Agentcities ADETTI team.

regarding the representation of preferences; the paged information-request protocol, a new interaction protocol used to handle possible information overload; and deploying all the above using the FIPA standardization framework (FIPA++ [8], a FIPA compliant agent platform; FIPA ACL agent communication language [11]; and FIPA SL content language [12][1]). Fredo also relies on the existence of FIPA compliant directory and ontology agents (DF [10] and OA [9]).

Fredo may be used in several scenarios, such as the one represented in Figure 1. Fredo receives information requests from its clients. In general, it is impossible to satisfy the client's request consulting only a single information provider. The relevant agents are discovered by contacting Directory Facilitaror agents and Ontology agents, using the information contained in received requests.

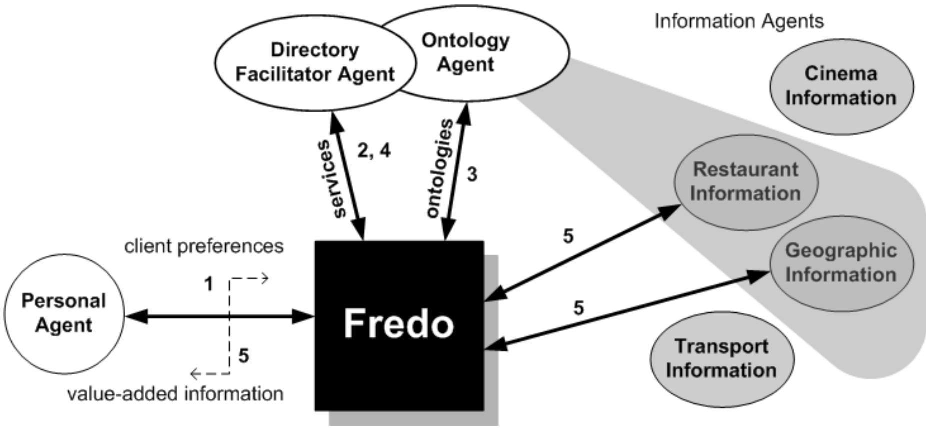

Figure 1. Scenarios where Fredo may be used.

Section 2 describes Fredo and analyses the obtained results. Section 3 compares our approach with related work. Finally, section 4 presents conclusions and future work.

2. Fredo: An Information Broker Agent

In order to satisfy clients' requests, Fredo discovers information providers relevant for the received requests (Task 1 in Figure 2); it plans queries to send to the discovered providers using a fuzzy heuristic rule (Task 2 in Figure 2); and it integrates the information received from the providers, evaluates it using a fuzzy inference mechanism and sends it to its clients (Task 3 in Figure 2). All these steps of Fredo's operation are absolutely general in the sense that Fredo does not hold any domain dependent information and it does not use any domain dependent algorithm.

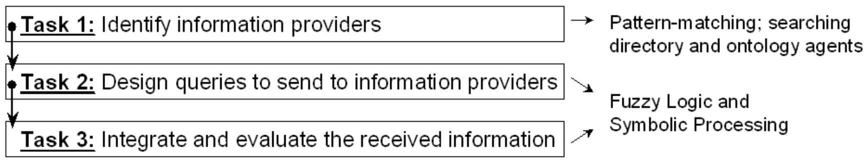

Figure 2. Fredo's major tasks and used technology

Although Fredo uses fuzzy reasoning, it does not assume its clients or its providers to use fuzzy logic.

2.1 Technical Overview

We defined the class Preference to represent information preferences. This class enables the representation of soft preferences (**values**) and hard preferences (**cut-off**), which constrain the attributes specified in the **attributes** property. Soft constraints do not need to be satisfied by the information to be sent back to the client; they are merely used to evaluate the returned information. Hard constraints, on the other hand, must fulfilled by the returned information.

The Preferences class also contains an attribute called **target** to hold the target object of the specified preferences. The object hold by this attribute is evaluated with respect to the specified preferences. **Table 1** shows a 0.7 weight preference for Italian cheap restaurants.

When Fredo receives a query with a set of preferences it must discover the information relevant for the preferences. For that purpose, Fredo consults agents providing yellow pages and ontology information services. In our application, these are the DF (Directory Facilitator) and the OA (Ontology Agent), which have been defined in the FIPA specifications.

Fredo asks the OA for ontologies containing the concepts mentioned in the received query. Then, Fredo asks the DF agent for information agents using the identified ontologies. Finally, Fredo sends queries to the identified information agents asking them for the relevant information for the received information request.

Table 1. Preference for cheap Italian restaurants.

```
(Preference
   :target <Restaurant> // specific object
   :targetClass Restaurant
   :attributes (sequence maxPrice foodType)
   :values (any   (sequence ?price ?type)
                  (and  (= ?type Italian)
                               (<= ?price 10)))
   :cut-off (sequence
             (AttributeCutOff :similarity 0.8)
             (AttributeCutOff))
   :weight 0.7)
```

In order to discover information providers for the preference in **Table 1**, Fredo asks the OA for an ontology containing the class *Restaurant* with attributes *maxPrice* and *foodType*, among others. Assuming the OA returns the ontology identifier "Agent-cities-Reastaurant-Ontology", Fredo asks the DF for information agents using that ontology.

After the DF sends Fredo the identifications of the necessary information providers, Fredo asks them for the relevant information. Since the soft constraints are not mandatory, the information to be returned by Fredo may not fulfill them. This means Fredo cannot use the softly constrained attributes to limit the amount of information that will be returned by each information provider, which may lead to the exchange to huge messages (e.g., messages containing 400 restaurants).

In order to avoid information overload, Fredo uses the hard constraints. Since hard constraints must be fulfilled by the returned information, Fredo uses them to limit the amount of information he asks the information providers. However, hard constraints are not mandatory therefore Fredo uses a fuzzy heuristic rule to create hard constraints whenever they are not specified in the received preferences.

The heuristic rule is based on the specified preference weight, which is mandatory. When the preference is very important (high weight), it means Fredo's client is willing to receive only information that is not too bad with respect to the preference. If the preference is not important (low weight), it means Fredo's client would not mind receiving information that is not highly rated with respect to the preference. Using this heuristic, Fredo computes a symbolic condition that constrains the asked information to those records that are as good as ¾ of the value of the preference weight. The quality of the record is evaluated using a fuzzy mechanism.

Even so, the returned information may be too much therefore Fredo uses a special purpose interaction protocol, called the paged-information-request protocol, that was created to ask information one page at a time (**Figure 3**). Using this protocol, the agent my find answer for its problem in the first page and decided to cancel the interaction.

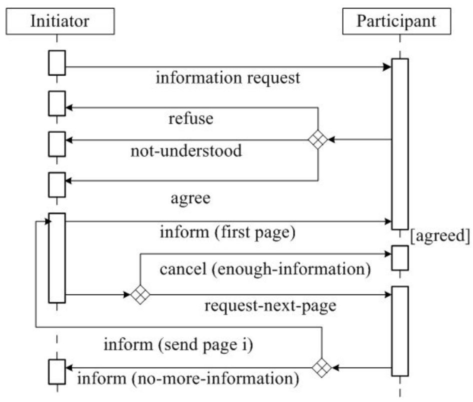

Figure 3. Paged Information-Request Protocol.

When the information asked from information providers is received, Fredo integrates and evaluates it and send it to its client. The evaluation of the received information with respect to the specified preferences is made by a fuzzy inference mechanism. Each preference is converted into a fuzzy proposition. The final evaluation is the average of the fuzzy values of each of the fuzzy propositions weighted by the preference weight.

2.2 Analysis of the Results

Fredo has been implemented and tested in several circumstances, including in the Agentcities audit in Barcelona in which it was used with agents (requesters, providers, directory and ontology agents) designed and developed by different teams. In one experiment Fredo integrates information from two main ontologies: restaurant ontology, and transportation ontology, which share the geographic information ontology. Different agents serve the information described in these ontologies, which means that Fredo is capable of interacting with different information providers.

With our approach, preferences representation is very expressive, allowing Fredo to integrate information from several ontologies provided by different agents, and supports a general domain-independent algorithm in all of its processing stages.

The same message may specify several preferences at the same time, about different classes and ontologies. Each preference may specify any logical combination of soft and hard constraints. The expressive power of the approach results not only of the proposed preferences representation but also of the flexibility of the used content language (FIPA SL [12]).

The process used by Fredo for discovering the relevant information providers relies upon the structure of the received information request whose content is completely expressed in FIPA SL. But it is possible only because Fredo can dynamically consult directory services and ontology services provided by FIPA agent platforms.

The use of cut-off values (whether explicitly specified within the received preferences or heuristically computed by the Fredo) greatly reduces the amount of information that is asked to information providers.

The use of the paged information-request protocol allows the agent to request the provider to stop sending information as soon as it becomes satisfied. This may happen because some *"satisficing"* criterion specified in the message is met or because the client does not want to receive more information. In some experiments, using our 600-restaurant database and a page length of 10 restaurants, the satisficing criterion is met by information contained in the first or in the second page sent by the provider. This means Fredo processes only 20 instead of 550 restaurants.

The choice of evaluating all received data with respect to the preferences, instead of considering only the data that totally satisfies all the preferences at once has shown appropriate in realistic applications. This is the case because, most of the times, it is impossible to find data that satisfies all preferences and also because the client prefers to receive imperfect solutions instead of no solutions at all.

3 Related Work

Brokers are intermediates who receive client requests, distribute the load to process those requests among several registered service providers and, in the end, return the obtained results. The design of these brokers may be tailored to a specific domain, like electronic commerce [2], be less domain dependent as in [3] and [14], where it is required a priory knowledge of the domain, or be domain independent as it is the case with [15] and Fredo.

The application scenario used in [14] is quite similar to Fredo's. However, their solution is quite rigid, since brokers have their own ontology with schema mappings to information providers' ontologies maintained locally. From those mappings it is then easy to distribute requests, but conversely we are constrained to the existence of those mappings. Fredo searching for providers is dynamic, since it does not hold domain ontology or mapping, it searches directory and ontology agents.

The search for providers may be supported by intermediates, like in [3], where a Jini-based directory and ontology service is used. Fredo works in a similar way, in the sense that it doesn't hold any registration regarding providers' domain information. Fredo has to use ontology agents and directory facilitator agents (yellow pages agents) to dynamically find providers, according to their ontologies and service types.

In order to distribute a request, one usually has to break it down and then search for the right information providers that satisfy the partial requests. This task can be done using previous acquired domain knowledge [3], or using existent mappings [14]. But Fredo's requests are decomposed using a domain independent process, relying on an analysis of specified preferences; domain information is just acquired from ontology agents, after receiving the request.

Domain knowledge may also be required to merge information from several providers [3]. Fredo does not have that limitation: value-added information is obtained using a generic process.

A more generic approach than [3] and [14] is used in [15] where domain knowledge about services is acquired only after receiving and analyzing requests. Those requests use hard-constraints, facilitating the finding and querying of providers. Fredo also uses preferences received in requests to better plan and distribute requests among providers, without previous domain knowledge. However, in [15], a simplistic approach is taken for preferences, by basically adding a new predicate. No weighted or fuzzy evaluation is used, as it is done in Fredo.

None of previously described approaches have used weights on preferences. However, soft-constraints considering weights in user queries, have been in use in several domains, such as information searching [5][7]. In [7] absolute weights, concerning individual terms, and relative weights, concerning a relation between terms, may be used. Fredo allows for both approaches, by permitting queries to cater for constraints on individual attributes, and constraints among attributes.

Soft-constraint fuzzy preferences are also used in several domains, like solving decision-making problems [4] [13], where it is necessary to satisfy multiple user preferences. Fuzzy searching and classification of information has also been in use in multimedia systems for some time [5] [6]. Fredo also uses a fuzzy approach to overcome the problem of possibly over restricted requests, poorly covered domains in providers or intrinsic vague definition of information in providers. Nevertheless, Fredo does not require that providers are aware of its fuzzy processing mechanism, where all queries sent by Fredo are tailored with hard constrains to retrieve the necessary results and to perform the fuzzy evaluation and merging the obtained results.

In [5], as in Fredo, fuzzy evaluation is associated with weighted preferences. These weights, assign different relevance to the required fuzzy conditions, in order to better adapt to user needs. But Fredo does more, by also considering hard-constraints (i.e., cut-offs) in client queries. This allows, at once, a better specification of user needs, and a better performance of the broker, since it is possible to considerably reduce exchanged information with providers, making the overall solution more scalable.

One can say that up to now no agent broker has considered all the techniques used by Fredo in one solution.

4 Conclusions and Future Work

We have proposed a flexible and expressive method for representing preferences that allows the specification of preferences about several attributes of an object. The representation provides for the specification of soft and hard constraints about individual attributes but also about relations among them.

The identification of the necessary information providers is an absolutely dynamic generic process. There is nothing that depends on a specific domain. Furthermore, the received set of preferences can refer to several ontologies possibly provided by several agents. Therefore, Fredo is a general agent (i.e., domain independent) that provides a value-added information service since it may integrate information services using several ontologies provided by different information providers.

We have also proposed a generic mechanism that is used to create the queries to send information providers in order to constrain the amount of returned information without discarding potentially interesting records. In addition, we have also designed an interaction protocol to be used when there is the risk that the answer to an information request may contain an exaggerated number of records.

Finally, we have designed a fuzzy mechanism that is used to evaluate gathered information from information providers with respect to the weighted preferences of the clients. This mechanism captures the fuzziness of human-like evaluation processes. Although Fredo uses a fuzzy evaluation mechanism, it does not assume other agents to also have fuzzy reasoning capabilities.

This set of proposals described in this paper makes up a general information brokering service that may be used in any domain without modification or recompilation.

References

1. Botelho, L.M.; Antunes, N.; Ebrahim, M.; and Ramos, P. 2002. “Greeks and Trojans Together”. In Proc. of the Workshop “Ontologies in Agent Systems” of the AAMAS2002 Conference

2. Caughey, S.; Ingham, D.; and Watson P. 1998 “Metabroker: A Generic Broker for Electronic Commerce”, Computer Networks and ISDN Systems, vol. 30, no. 1-7, pp. 619-620
3. Chakraborty, D.; Perich, F.; Joshi, A.; Finin, T.; and Yesha, Y. 2002. “Middleware for Mobile Information Access”, 5th Int. Workshop on "Mobility in Databases and Distributed Systems", in conjunction with DEXA
4. Chiclana, F.; Herrera, F.; and Herrera-Viedma, E. 1998. “Integrating three representation models in fuzzy multiple propose decision making based on fuzzy preference relations”, Fuzzy Sets and Systems, 97, 33-48
5. Ciaccia, P.; Montesi, D.; Penzo, W.; and Trombetta, A. 2000. “Imprecision and User Preferences in Multimedia Queries: A Generic Algebraic Approach”, In Proc. of the 1st Int. Symposium on Foundations of Information and Knowledge Systems (FoIKS 2000)
6. Fagin, R. 1998. “Fuzzy Queries in Multimedia Database Systems”, In Proc. of the 17th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems
7. Fagin, R.; and Maarek, Y.S. 2000. “Allowing users to weight search terms”, Proc. Recherche d'Informations Assistee par Ordinateur RIAO '2000, pp. 682-700
8. FIPA++ Webpage, http://fipapp.page.vu
9. Foundation for Intelligent Physical Agents. 2001. FIPA Ontology Service Specification. Document 00086. http://www.fipa.org/specs/fipa00086/XC00086D.html
10. Foundation for Intelligent Physical Agents. 2002. FIPA Agent Management Specification. http://www.fipa.org/specs/fipa00023/SC00023J.html
11. Foundation for Intelligent Physical Agents. 2002. FIPA Communicative Act Library Specification, Document SC00037J. http://www.fipa.org/specs/fipa00037/SC00037J.html
12. Foundation for Intelligent Physical Agents. 2002. FIPA SL Content Language Specification. Document SC00008I. http://www.fipa.org/specs/fipa00008/SC00008I.html
13. Fuller, R.; and Carlsson, C. 1996. “Fuzzy multiple criteria decision making: Recent developments”, Fuzzy Sets and Systems, 78, 139-15
14. Martin, D.; Oohama, H.; Moran, D.; and Cheyer, A. 1997. “Information brokering in an agent architecture”. In Proc. of the 2nd Int. Conf. on the Practical Application of Intelligent Agents and Multi-Agent Technology
15. Mcllraith, S.A.; Son, T.C.; and Zeng, H. 2001. “Semantic Web Services”, IEEE Intelligent Systems. Special Issue on the Semantic Web. 16(2): 46-53
16. Mota, L.; Botelho, L.M.; Mendes, H.; and Lopes, A. 2003. "O3F: An Object-Oriented Ontolgy Framework". Proc. 2nd Int. Joint Conf. on Autonomous Agents and MultiAgent Systems.
17. Willmott, S.; Dale, J.; Burg, B.; Charlton, P; and O'Brien, P. 2001. “Agentcities: a world-wide open agent network”. Agentlink News, 8:13-15