IRES: On the Integration of Restaurant Services

Miquel Montaner, Beatriz López, Esteve del Acebo, Silvana Aciar, and Isabel Cuevas

Institut d'Informàtica i Aplicacions
Agents Research Laboratory
Universitat de Girona
Campus Montilivi (Ed. PIV)
17071 Girona, Spain
{mmontane,blopez}@eia.udg.es, acebo@ima.udg.es, {saciar,icuevas}@eia.udg.es

1   Entry and Entrant Details

ENTRY INFORMATION	
Entry Acronym / Short Name	IRES
Entry Full Name	On the Integration of Restaurant Services
Entry Reference Number	4
PARTICIPATION	
Category	Application
Consider for Special Prize	Yes
Keywords	Recommendation, Personalization, Trust, Opinion-Based Filtering, Case-Based Reasoning
TEAM	
Team Name	GenialChef
Number of Contributors	5
Affiliation	Universitat de Girona
CONTACT PERSON	
Full Name	Miquel Montaner
Affiliation	Universitat de Girona
Country	Spain
Postal Address*	Campus Montilivi (Ed. PIV), 17071 Girona
Email Address*	mmontane@eia.udg.es
Telephone Number*	+34 972 41 84 78
FAX Number*	+34 972 41 80 98
URL*	http://arlab.udg.es

2   Description

2.1   Abstract

Our contribution to AgentCities is a restaurant recommender service that consists of a multi-agent system of service agents and personal agents. Service agents offer information about restaurants and the personal agents. Personal agents are in charge of recommending restaurants to their users based on both, information on restaurants and interaction with other *friendly* personal agents. To achieve this purpose, personal agents interact with the restaurant server agent in order to know about the restaurants, the personal agent facilitator agent in order to know about other personal agents in the system, and the personal agents in order to find similar users and take advantage of their opinions and advices. To improve the performance of the personal agents, case-based reasoning and trust techniques have been applied.
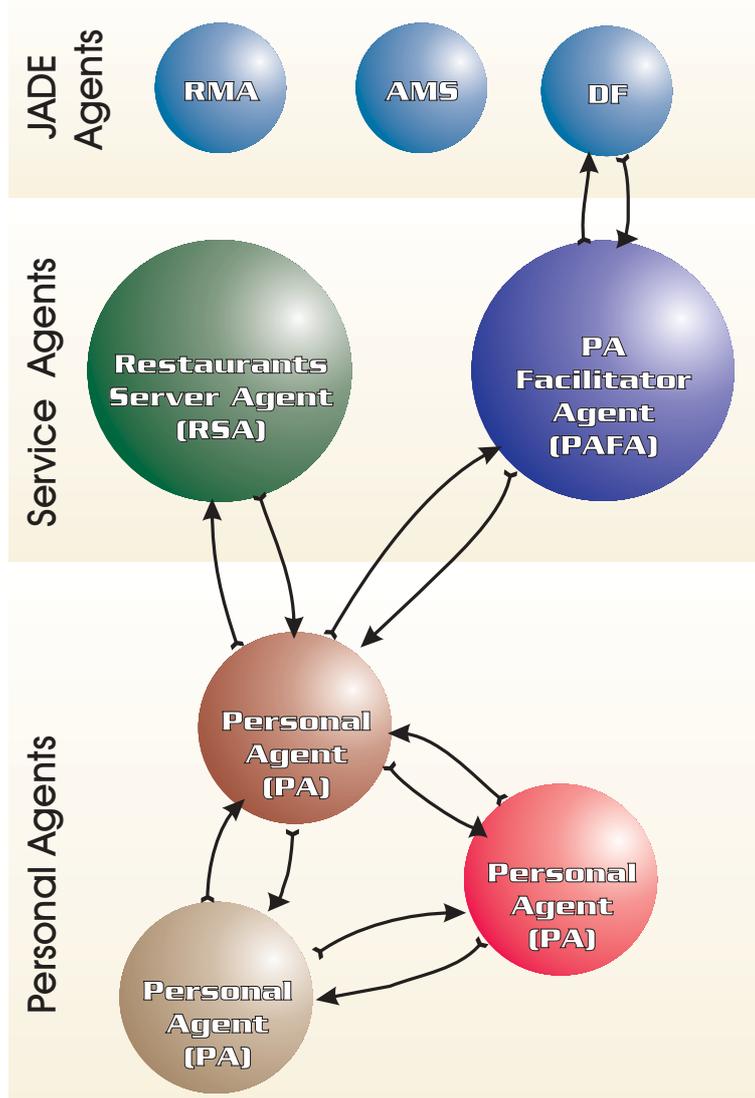
Fig. 1. System Architecture

2.2 Description

Recommender agents make recommendations to users according to the information available. Such information includes data on items (e.g., restaurants) as well as different profiles of other users on the system.

Many recommender agents have been developed on the Internet [4] where the typical architecture is a centralized server where users are connected to obtain their recommendations. However, when we apply the agents theory to recommender systems, a standard centralized recommender systems becomes a distributed world of recommender agents. Our contribution to Agentcities is in this latter XXX: a restaurant recommender service designed as a multi-agent system of recommender agents.

In our architecture we can group agents into service agents and personal agents (PA) (see Figure 1). Among service agents, we distinguish the restaurant server agent (RSA) and the personal agent facilitator agent (PAFA).

The RSA has all the information about restaurants. There could be many RSAs, each one having information about restaurants of a given region (e.g., a city, a county or a country). For example, a RSA could offer information about restaurants of Barcelona, the county of Barcelona or Spain. We dissuade RSAs that offer information about a country because the agent should have too much information. In addition, if we design RSAs that give information about a city, we are excluding some important restaurants near the city. Therefore, we advice RSAs covering a city and its outskirts. The RSA has several functions:

- give information about a certain restaurant, a restaurants situated in a certain address or a restaurants situated in a certain city,
- and functionalities regarding the case-based reasoning engine developed over the database. Due to the similarity functions among the restaurant attributes we can retrieve restaurants very similar to a given one, restaurants very different to a given one or the most similar restaurants to a certain idea that the user has in mind.

The PAFA acts as a broker agent: it is in charge of assisting PAs to find and contact other PAs. If the system have many PAs, should be interesting to have several PAFAs in the system in order to prevent overloads. Mainly, the PAFA informs the personal agents about:

- whether a given PA is online before contact him/her,
- the agent identification (AID) of a certain PA in order to contact him/her and
- the AID of other PAs in the system in order to make new friends.

PAs are in charge of recommending restaurants to the user. Each user have their own PA in their local machine and the service agents XXX in the server gives assistance to them. Each personal agent encapsulates the user profile and filter recommendations based on it. Therefore, the personal data is kept in private. The content-based filtering method is applied to make recommendations to the user based on the content of the items, in particular, a case-based reasoning approach is implemented [2]. However, collaborative filtering cannot be applied since user profiles are kept in private by each recommender agent and it is impossible a direct comparison among them. To take advantage of the collaborative world, we have implemented a new information filtering method based on a model of trust: the opinion-based filtering method [1, 3]. The main idea is to consider other agents as personal entities which you can rely on or not. Reliability is expressed through a trust value with which each agent labels its neighbors. The trust value is initially computed through interaction, following a proactive *playing agents* procedure [3]. Once the agent has a set of *friends*, it can use them to filter information. When the agent is not sure about a recommendation or discovers a new item, it asks the reliable agents for their opinion and uses their trust values to decide whether the item is interesting for the user or not. Once the agent has the opinion of other agents, a consensus is achieved through the use of an aggregation measure. The result of the consensus provides a confidence value upon which the agent can decide on the convenience of recommending an item to the user or not.

We suppose that similar agents will provide pertinent opinions, but they may also give inadequate ones. Trust, therefore, should be modified as goes by depending on the results of the recommendations, in order to improve acquaintance.

It is important to note that the opinion-based approach is combined with content-based and collaborative filtering methods. The performance of the content-based method is the same, but, because trust, we are able to apply a collaborative method and improve it in two ways: we are not looking for private data of other user but for an opinion and agents only believe in the recommendations of agents with a high trusting value.

In order to communicate our agents, an ontology has been developed. The ontology comprehends the communication between the PA and the RSA, between the PA and the PAFA and among PAs. This ontology is deeply explained in section 3.2.

2.3 Implementation / Use Status

The proposed system has been implemented as a prototype. To test its performance, we installed twenty two agents to people from our Department. We only implemented one RSA with the restaurants of Girona and its outskirts and one PAFA. The system was working during two months and, at present, we are in the beginning of the analysis of the results obtained.

2.4 Relevance to Open Agent Systems

Our contribution to open agent systems is twofold:

- Privacy of personal data: personal agents interact with the restaurant server agent to know information about new items and with the personal agent facilitator agent to know information about other agents in the system. Personal agents and service agents do not know specific information about other users (e.g., visited restaurants). In this way, the system guaranties the privacy of the user's personal information.

Moreover, personal agents are anonymous to others, each personal agent is identified by an unique id and other agents just know this id. Thus, the agents have no idea who is the user behind the personal agent.

– Quick adaptation of new agents: When a new user is incorporated to the system, a personal agent is created and it is registered to the DF. As from the interaction of this user with the service agents and other personal agents, the new agent is quickly adapted to the environment.

2.5 Relevance to AgentCities

We think that, in a first stage, we contribute to the Agentcities network in a way that we provide a service mainly to people of Girona, but also to people around the world. This service consists of a recommender system of restaurants of Girona. Users have to run a personal agent in their local machine to access the service. The user's personal agent interacts with a distributed world of agents and quickly recommend restaurants with a high probability of success.

In a second stage, when RSAs from cities all over the world and PAs of people elsewhere will be available, the benefits of our service will be more significative. PAs trained in one city will be able to recommend restaurants to other PAs of people visiting these cities (e.g., in business trips or holidays).

2.6 Support / Acknowledgments

3 Access Information

3.1 User Access

To run a personal agent (PA) in your machine follow the instructions:

1. Download the java classes from: ***http://arlab.udg.es/genialchef.zip***
2. Unzip the file ***genialchef.zip***
3. If you already have ***JDK1.2*** or later installed, jump to step 6. Otherwise, follow with step 4.
4. Download the Java Runtime Environment (JRE) from
   http://java.sun.com/j2se/1.4.1/download.html
5. Install the JRE
6. The PA is started running the file ***PA.bat***, which contains the following command[1]:

```
java -classpath .;Base64.jar;jade.jar;iiop.jar;jadeTools.jar;AL.jar;jdbc.jar jade.Boot
-host arlab.udg.es -mtp jade.mtp.iiop.MessageTransportProtocol -container PA1:PA
```

7. The instructions to use the PA can be found in the User Manual annexed with this documentation.

If you want to run more than one PA, you should unzip the file ***genialchef.zip*** in another folder and launch the command cited in the step number 6, taking into account that you must start your agent with a different name every time: instead of PA1:PA you must use PA2:PA, PA3:PA, etc.

3.2 Agent Access

HOSTING PLATFORM INFORMATION

Our project has a platform that is accessible via the *-host* parameter when the personal agents are launched. The deployed agent platform which participate in the application is the following:

---

[1] this command is used in Windows OS. If you are using Linux OS the classpath separator is ":" instead of ";". Thus, the command in Linux OS is:
```
java -classpath .:Base64.jar:jade.jar:iiop.jar:jadeTools.jar:AL.jar:jdbc.jar jade.Boot
-host arlab.udg.es -mtp jade.mtp.iiop.MessageTransportProtocol -container PA1:PA
```

Name: arlab.udg.es
Software Tools in Use: JDK 1.2 or later, JADE 2.6

AGENTS TO CONTACT

At present, four agents are accessible in our platform: a restaurant server agent, a personal agent facilitator agent and two personal agents:

Name: RSA
Agent Identifier: RSA@arlab.udg.es:1099/JADE
Purpose: It is a restaurant server agent that provides information about restaurants of Girona and its outskirts to PAs.

Name: PAFA
Agent Identifier: PAFA@arlab.udg.es:1099/JADE
Purpose: It is a personal agent facilitator that provides information about the personal agents in the system and how to contact them.

Name: PA9625781
Agent Identifier: PA9625781@arlab.udg.es:1099/JADE
Purpose: It is a personal agent that recommends restaurants to its user and, in addition, provides opinions and advices to other PAs.

Name: PA9625782
Agent Identifier: PA9625782@arlab.udg.es:1099/JADE
Purpose: It is another personal agent that recommends restaurants to its user and, in addition, provides opinions and advices to other PAs.

SERVICE INFORMATION

Each of the agents cited before offers a service: the RSA offers a restaurant service, the PAFA offers a personal agent finder service and the PAs offer a personal agent service.

Restaurant Service ("ServidorRestaurants")

Text Description: The Restaurant Server Agent (RSA) is in charge of providing information about the restaurants of a given zone. In particular, the RSA of our platform provide information about the restaurants of Girona and its outskirts. The RSA has several functions:

– give information about a certain restaurant, a restaurants situated in a certain address or a restaurants situated in a certain city,
– and functionalities regarding the case-based reasoning engine developed over the database. Due to the similarity functions among the restaurant attributes we can retrieve restaurants very similar to a given one, restaurants very different to a given one or the most similar restaurants to a certain idea that the user has in mind.

Protocols: FIPA Query Protocol
Languages: FIPA-ACL
Ontologies: Restaurants Ontology
DF Registration Message:

```
(df-agent-description
    :name (agent-identifier :name RSA@arlab.udg.es:1099/JADE
    :addresses(sequence IOR:000...100) :resolvers (sequence))
    :services (sequence (service-description
        :name RSA@arlab.udg.es:1099/JADE
        :type ServidorRestaurants
        :protocols (sequence)
```

```
            :languages (sequence)
            :ontologies (sequence)
            :properties (sequence)
      ))
      :protocols (sequence)
      :languages (sequence)
      :ontologies (sequence)
  )
```

Example Messages:

– Example1 - *Request Similars*, *Request No Similars* and *Request Restaurants*: these queries are used when
   the PA wants to know the most similar or the least similar restaurants to a given restaurant or to a
   certain set of attributes. The *Request Similars* query looks for the most similar restaurants to a given
   one, the *Request No Similars* query looks for the least similar restaurants to a given one and the *Request
   Restaurants* query looks for the most similar restaurants to a set of attributes. The inform message in
   response to the three kinds of message of the RSA is the same in each case.

   Request Similars Query:

```
      (QUERY-REF
          :sender  ( agent-identifier  :name PA9625782@arlab.udg.es:1099/JADE)
          :receiver(set ( agent-identifier  :name RSA@arlab.udg.es:1099/JADE) )
          :content (o_request_similars 5 "2002-07-02 10:24:36+02")
          :language  FIPA-ACL
          :protocol  fipa-query
      )
```

   Request No Similars Query:

```
      (QUERY-REF
          :sender  ( agent-identifier  :name PA9625782@arlab.udg.es:1099/JADE)
          :receiver(set ( agent-identifier  :name RSA@arlab.udg.es:1099/JADE) )
          :content (o_request_no_similars 5 "2002-07-02 10:24:36+02")
          :language  FIPA-ACL
          :protocol  fipa-query
      )
```

   Request Restaurants Query:

```
      (QUERY-REF
          :sender  ( agent-identifier  :name PA9625782@arlab.udg.es:1099/JADE)
          :receiver (set ( agent-identifier  :name RSA@arlab.udg.es:1099/JADE) )
          :content  (o_request_restaurant
              (o_men :pmenus "1" :menus " ")
              (o_cuin :tipuscuina Brasa :ptipuscuina "3")
              (o_ciu :pciutat "3" :ciutat Girona)
              (o_papr :paprox "De 18 a 24 Euros" :ppaprox "3")
              (o_booleans :fumadors false :minusvalids false
                  :animals false :parking false :refrigerats false
                  :espectacles false :reserves false :banquets false)
              (o_qual :pqualificacio "1" :qualificacio " ")
              5
          )
          :language  FIPA-ACL
          :protocol  fipa-query
      )
```

```
Inform:

    (INFORM
        :sender ( agent-identifier  :name RSA@arlab.udg.es:1099/JADE)
        :receiver (set ( agent-identifier  : PA9625782@arlab.udg.es:1099/JADE) )
        :content (o_restaurants 10 (sequence
            (o_rest_ret :similitud "1.0"
                :nom "VEGETARIA LA CATEDRAL"
                :id_restaurant "2002-07-02 10:24:36+02")
            (o_rest_ret :similitud "0.9795996"
                :nom "BAR VEGETARIA TILLA"
                :id_restaurant "2002-07-02 17:17:44+02")
            (o_rest_ret :similitud "0.86"
                :nom "EDELWEISS- AGUSTI ENSESA"
                :id_restaurant "2002-07-03 16:44:17+02")
            (o_rest_ret :similitud "0.85969967"
                :nom "L'SPAGHETTERIA DEL COCOLINO"
                :id_restaurant "2002-07-02 17:09:58+02")
            (o_rest_ret :similitud "0.84768766"
                :nom "CAL FRAY"
                :id_restaurant "2002-06-10 11:32:05+02")
            )
        )
        :language  FIPA-ACL
        :protocol  fipa-query
    )
```

– Example2 - *Request By Name*: this query is used when the user knows the name, the address or the city of the restaurant and ask for information on it.

```
(QUERY-REF
    :sender ( agent-identifier  :name PA9625782@arlab.udg.es:1099/JADE)
    :receiver (set ( agent-identifier  :name RSA@arlab.udg.es:1099/JADE) )
    :content  (o_request_by_name av Girona "la ")
    :language  FIPA-ACL
    :protocol  fipa-query
)

(INFORM
    :sender ( agent-identifier  :name RSA@arlab.udg.es:1099/JADE)
    :receiver (set ( agent-identifier  : PA9625782@arlab.udg.es:1099/JADE) )
    :content  (o_restaurants 1 (sequence
        (o_rest_ret :similitud "1.0"
            :nom "LA LLARGA"
            :id_restaurant "2002-07-03 19:08:48+02")
        )
    )
    :language  FIPA-ACL
    :protocol  fipa-query
)
```

– Example3 - *More Info*: this query is used when the user visits the information window of a restaurant. The PA queries the RSA for the full information of the restaurant.

```
(QUERY-REF
    :sender ( agent-identifier  :name PA9625782@arlab.udg.es:1099/JADE)
    :receiver (set ( agent-identifier  :name RSA@arlab.udg.es:1099/JADE) )
```

```
        :content  (o_more_inf "2002-07-02 10:24:36+02")
        :language  FIPA-ACL
        :protocol  fipa-query
    )


    (INFORM
        :sender  ( agent-identifier  :name RSA@arlab.udg.es:1099/JADE)
        :receiver(set ( agent-identifier  : PA9625782@arlab.udg.es:1099/JADE) )
        :content (o_info Vegetariana Migdia "  " "972218388" "Lluis Batlle i Prats, 4"
            "  " GIRONA "Sense Qualificacio" "VEGETARIA LA CATEDRAL" "Menys de 15 Euros"
            (o_booleans :fumadors false :minusvalids false
                :animals false :parking false :refrigerats false
                :espectacles false :reserves false :banquets false)
        )
        :language  FIPA-ACL
        :protocol  fipa-query
    )
```

Personal Agent Finder Service ("FacilitadorAgent")

Text Description: The personal agent facilitator agent (PAFA) acts as a broker agent: it is in charge of assisting PAs to find and contact other PAs. If the system have many PAs, should be interesting to have several PAFAs in the system in order to prevent overloads. Mainly, the PAFA informs the personal agents about:

– whether a given PA is online before contact him/her,
– the agent identifier (AID) of a certain PA in order to contact him/her and
– the AID of other PAs in the system in order to make new friends.

Protocols: FIPA Query Protocol
Languages: FIPA-ACL
Ontologies: Restaurants Ontology
DF Registration Message:

```
    (df-agent-description
        :name (agent-identifier :name PAFA@arlab.udg.es:1099/JADE
        :addresses(sequence IOR:000...100) :resolvers (sequence))
        :services (sequence (service-description
            :name PAFA@arlab.udg.es:1099/JADE
            :type FacilitadorAgent
            :protocols (sequence)
            :languages (sequence)
            :ontologies (sequence)
            :properties (sequence)
        ))
        :protocols (sequence)
        :languages (sequence)
        :ontologies (sequence)
    )
```

Example Messages:

– Example1 - *Request Agents*: this query is used when the PA wants to meet other agents. The PA send a query to the PAFA with his ID and the number of agents that it is looking for. The PAFA informs the PA with a list of other online PAs containing their ID and their AID.

```
    (QUERY-REF
        :sender  ( agent-identifier  :name PA9625782@arlab.udg.es:1099/JADE)
```

```
        :receiver (set ( agent-identifier  :name PAFA@arlab.udg.es:1099/JADE) )
        :content  (o_request_agents 2 "1037214032070")
        :language  FIPA-ACL
        :protocol  fipa-query
)

(INFORM
        :sender  ( agent-identifier  :name PAFA@arlab.udg.es:1099/JADE)
        :receiver(set ( agent-identifier  : PA9625782@arlab.udg.es:1099/JADE) )
        :content (o_agents (sequence
            (o_usu :usuari "1037213923654"
                :id_usuari2 (agent-identifier
                    :name PA9625781@arlab.udg.es:1099/JADE
                    :addresses (sequence  IOR:000...100)
                    :resolvers (sequence))
            )
            (o_usu :usuari "1037213928987"
                :id_usuari2 (agent-identifier
                    :name PA1@arlab.udg.es:1099/JADE
                    :addresses (sequence  IOR:000...100)
                    :resolvers (sequence))
            )
        ))
        :language  FIPA-ACL
        :protocol  fipa-query
)
```

– Example2 - *Get Info Agents*: this query is used when the PA wants to know, first, if their friends are
  online in a certain moment and, then, the AID of the agent to contact. Thus, the PAFA informs the PA
  with a list of the inquired agents that are online and their AID.

```
(QUERY-REF
        :sender  ( agent-identifier  :name PA9625782@arlab.udg.es:1099/JADE)
        :receiver (set ( agent-identifier  :name PAFA@arlab.udg.es:1099/JADE) )
        :content  (o_get_info_agents (sequence
            (o_idage :id_usuari "1037213923654")
            (o_idage :id_usuari "1037204976306")
            (o_idage :id_usuari "1037115756326")
        ))
        :language  FIPA-ACL
        :protocol  fipa-query
)

(INFORM
        :sender  ( agent-identifier  :name PAFA@arlab.udg.es:1099/JADE)
        :receiver(set ( agent-identifier  : PA9625782@arlab.udg.es:1099/JADE) )
        :content (o_set_info_agents (sequence
            (o_sia :id_agent2 (agent-identifier
                :name  PA9625781@arlab.udg.es:1099/JADE
                :addresses (sequence  IOR:000...100)
                :resolvers (sequence))

            )
        ))
        :language  FIPA-ACL
        :protocol  fipa-query
)
```

Personal Agent Service ("AgentPersonal")

Text Description: Personal agents (PA) also provide a service. Like in the real world, where people ask their friends for the opinion about a certain restaurant or ask their friends for advice when they are looking for a restaurant, the personal agents provide the service of providing opinions and advices to other PAs. When the PA is not sure about a recommendation, it can query its *friends* for their opinions and decide based on them. In addition, the PA can recommend to the user the restaurants most recommended by their friends.

Protocols: FIPA Query Protocol
Languages: FIPA-ACL
Ontologies: Restaurants Ontology
DF Registration Message:

```
(df-agent-description
    :name (agent-identifier :name PA9625782@arlab.udg.es:1099/JADE
    :addresses(sequence IOR:000...100) :resolvers (sequence))
    :services (sequence (service-description
        :name "1037115756326"
        :type AgentPersonal
        :protocols (sequence)
        :languages (sequence)
        :ontologies (sequence)
        :properties (sequence)
    ))
    :protocols (sequence)
    :languages (sequence)
    :ontologies (sequence)
)
```

Example Messages:

– Example1 - *Get Opinion*: this query is used when the PA wants to know the opinion of another PA in a given restaurant. The opinion is a value in [0-1].

```
(QUERY-REF
    :sender ( agent-identifier  :name PA9625782@arlab.udg.es:1099/JADE)
    :receiver(set ( agent-identifier  :name PA9625781@arlab.udg.es:1099/JADE) )
    :content (o_get_opinion "2002-06-14 15:14:45+02")
    :language  FIPA-ACL
    :protocol  fipa-query
)

(INFORM
    :sender ( agent-identifier  :name PA9625781@arlab.udg.es:1099/JADE)
    :receiver(set ( agent-identifier  : PA9625782@arlab.udg.es:1099/JADE) )
    :content (o_set_opinion "0.87")
    :language  FIPA-ACL
    :protocol  fipa-query
)
```

– Example2 - *Get Advice*: this query is used when the PA asks for advice to other PAs.

```
(QUERY-REF
    :sender ( agent-identifier  :name PA9625782@arlab.udg.es:1099/JADE)
    :receiver(set ( agent-identifier  :name PA9625781@arlab.udg.es:1099/JADE) )
    :content (o_get_advice 4 "1037214032070")
    :language  FIPA-ACL
```

```
        :protocol  fipa-query
    )

    (INFORM
        :sender  ( agent-identifier  :name PA9625781@arlab.udg.es:1099/JADE)
        :receiver  (set ( agent-identifier  : PA9625782@arlab.udg.es:1099/JADE) )
        :content   (o_set_advice (sequence
            (o_conjunt :nom "VEGETARIA LA CATEDRAL"
                :valoracio "0.7975"
                :id_restaurant "2002-07-02 10:24:36+02")
            (o_conjunt :nom LAGUNAK
                :valoracio "0.6925"
                :id_restaurant  "2002-07-02 10:20:33+02")
            (o_conjunt :nom "EL BULLI"
                :valoracio "0.675"
                :id_restaurant "2002-06-10 15:49:19+02")
            (o_conjunt :nom TELEPIZZA
                :valoracio "0.64"
                :id_restaurant "2002-06-12 11:06:31+02")
            ) 4
        )
        :language  FIPA-ACL
        :protocol  fipa-query
    )
```

References

1. M. Montaner, B. López, and J. L. de la Rosa. Developing trust in recommender agents. *In Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'02). Cristiano Castelfranchi and W. Lewis Johnson (Eds). ACM Press. vol. 1, pp. 304-305. Bologna (Italy)*, 2002.
2. M. Montaner, B. López, and J. L. de la Rosa. Improving Case Representation and Case Base Maintenance in Recommender Agents. *In Proceedings of the 6th European Conference on Case Based Reasoning (ECCBR'02). Susan Craw, Alun Preece (Eds.), Lecture Notes in AI N2416. Springer-Verlag. pp. 234-248. Aberdeen (Scotland)*, 2002.
3. M. Montaner, B. López, and J. L. de la Rosa. Opinion-Based Filtering Through Trust. *In Proceedings of the 6th International Workshop on Cooperative Information Agents (CIA'02). Matthias Klusch, Sascha Ossowski and Onn Shehory (Eds.), Lecture Notes in AI N2446. Springer-Verlag Berlin Heidelberg, pp. 164-178, Madrid (Spain)*, 2002.
4. M. Montaner, B. López, and J. L. de la Rosa. A Taxonomy of Recommender Agents on the Internet. *To be published in Artificial Intelligence Review*, 2002.

Appendix A. User's Manual

First of all, it is important to note that the language used to label the different parts of the interface is Catalan, since it has to be used by Girona citizens. In order to make it understandable for English speakers, this manual translates the most important labels from Catalan to English. In a future, a multilingual version will be deployed. The use of CBR makes the multilingual approach not trivial at all.

Configuration Steps (first execution)

The first time the personal agent (PA) is executed, two configuration steps have to be done:

1. Login and password generation: each PA assists only one user, therefore, in order to prevent the interaction between the PA a other users, a login and a password is generated with the window presented in Fig. 2.

Fig. 2. Login and Password Generation

2. Training Set: in order to start recommending to the user as soon as possible with a high probability of success, the agent needs to know the preferences of the user. Therefore, the agent fills in the user profile with initial experiences through the use of a training set. That is, the user is prompted to a set of restaurants (in particular 13 restaurants) and he/she must fill in the explicit information regarding his/her interests in restaurants. In order to enter the explicit interest of the user in a restaurant, the user has to evaluate the quality price relation, the ambient, the quantity of meal, the efficiency of the waiters and the originality of the cuisine (it is deeply explained in the section "Restaurant Information").
The inquired restaurants are representative of the different kinds of interests that users usually have. However, certainly these restaurants are not the ones that best represent the user's interests. For this reason, it is recommendable to look for the restaurants that best match what the user most likes and dislikes and evaluate them. The more accurate is the profile, higher is the probability of accurate recommendations of the PA.

Login

In order to identify the user every time the PA is started, a password is inquired with the window presented in Fig. 3.

The Personal Agent Interface

The most important information of the main window is at the bottom part under the title "Restaurants que Recomano" (recommended restaurants), where the PA recommend restaurants to the user (see Fig. 4). Clicking on the recommendation, the information about the restaurant is shown in a separated window (see section "Restaurant Information" and Fig. 10). It is very important for the PA that the user gives his/her opinion about the recommendations when the user has been in the given restaurants, since the PA needs feedback of the user to learn about the accuracy of the recommendations to improve them in the future.

At the top of the main window there is a user menu with several options. These options are deeply explained in the following sections.

Fig. 3. Login

Fig. 4. Main Window

Search

In the option menu "Cerca" there are three options of search: search by name ("Per Nom"), basic search ("Cerca Bàsica") and advanced search ("Cerca Avancada").

– Search by Name: this option is used when the user is looking for a given restaurant regarding which the name ("Nom"), the address ("Adreca") or the city ("Ciutat") is known (see Fig. 5).

Fig. 5. Search by Name

– Basic Search: this option is used when the user is looking for a restaurant with a certain basic features (see Fig. 6). These features are the city of the restaurant ("Ciutat"), the cuisine ("Tipus cuina") and the approximate price ("Preu aproximat"). The possible values for the cuisine and the approximate price are translated in tables 1 and 2. In addition, the user can point out different priorities in his/her searches in order to better define their requirements. The possible priorities are low ("Baixa"), normal and high ("Alta").

It is important to note that the kind of restaurant the user is looking for could not exist. However, the searcher gives to the user several results, which are the most similar restaurants to the user's requirements.

Fig. 6. Basic Search

– Advanced Search: it is the same search as the basic search but with some advanced features (see Fig. 7). These features are the class of the restaurant ("Qualificació"), the daily specials ("Menus"), whether the restaurant celebrates banquets ("Banquets"), whether it is necessary to make a reservation ("Reserves"), whether the restaurant has car park ("Parking"), whether the restaurant has conditioned air ("Aire Acondicionat"), whether there are performances in the restaurant ("Espectacles"), whether the restaurant have a pet care service ("Animals"), whether the restaurant have a non-smoking zone ("No fumadors") and whether the restaurant have special access for disabled people ("Acc. Discapacitats"). The possible values for the restaurant class and the daily specials are translated in tables 3 and 4.

Fig. 7. Advanced Search

The results of the search are presented in the lists at the bottom of the window. The results are grouped into four groups: evaluated restaurants that the PA recommends ("Valorats - Recomanats"), other evaluated restaurants ("Valorats - Altres"), non-evaluated restaurants that the PA recommends ("No Valorats - Recomanats") and other non-evaluated restaurants ("No Valorats - Altres"), which evaluated means that the user has provided feedback on the restaurant in the past.

Trustworthy Agents

In the option menu "Utilitats-Agents Afins" there are a list of the reliable *friends* that the PA has made (see Fig. 8). Initially the list is empty and the PA elaborate it using a procedure that we have called *playing agents*. The PA asks other agents for the opinion about the restaurants that the user most "loves" and "hates" and compare them in order to calculate the initial trust value. Then, taking into account the success of the opinions and advices of other agents, the PA updates their trust values.
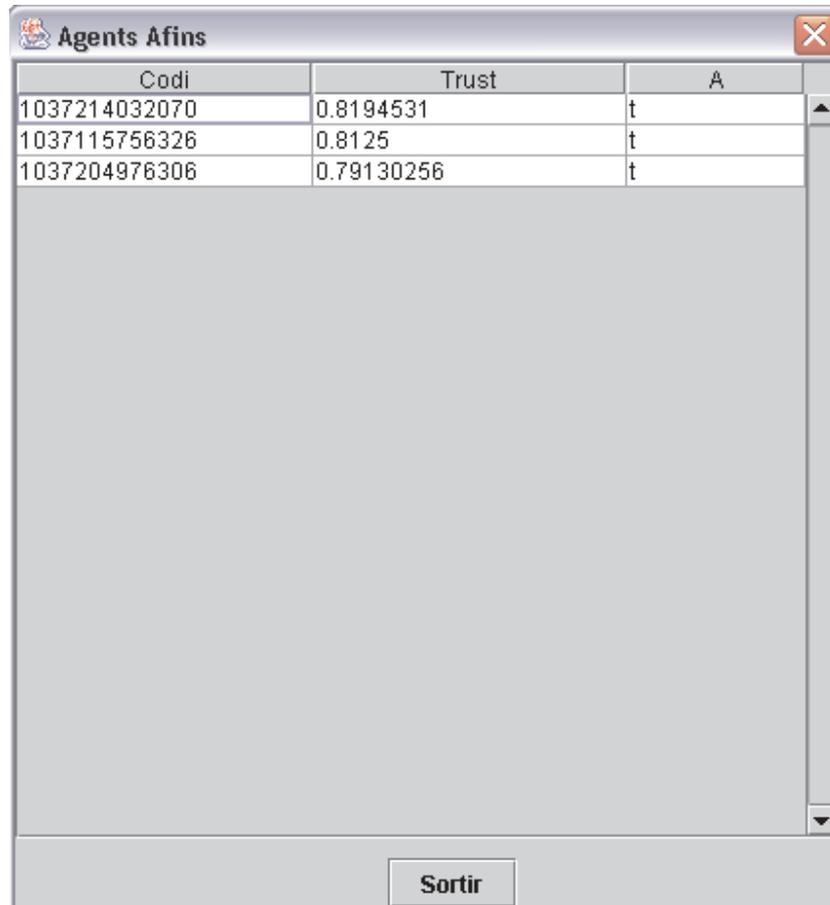
Codi	Trust	A
1037214032070	0.8194531	t
1037115756326	0.8125	t
1037204976306	0.79130256	t

Fig. 8. Trustworthy Agents

When the PA is not sure about a recommendation or discovers a new item, it asks the reliable *friends* for their opinions and uses the trust value to decide whether the item is interesting for the user or not.

User Profile

In the option menu "Utilitats-restaurants" there are a list of the restaurants contained into the user profile (see Fig. 9). The list is composed by:

- the name of the restaurant ("Nom"),
- the evaluation of the user after aggregating the different explicit interest attributes ("Valoració"). Restaurants with the value -1 means that the user has no given explicit feedback about this restaurant,
- and the drift attribute, a parameter that represents the age of the interest of the user in the given restaurant. Restaurants with a drift near to 1 represent that the user has an up-to-date interest in them and restaurants with a low drift represent that the user is forgetting them.

The user can click in the restaurants to see their information. The information showed in the pop up window is explained in the following section.

Nom	Valoració	Drift
CASA MARIETA	0.9	1.0
EL BULLI	0.9	0.9870779
MAS DE LA CREU	0.8	0.9870779
LA CUINA DE CAN PIPES	0.75	0.98806596
LAGUNAK	0.75	0.9870779
EL CELLER DE CAN ROCA	0.75	0.9870779
EL ROSER	0.75	0.98806596
EL MUSEU DEL VI	0.625	0.9870779
VEGETARIÀ LA CATEDRAL	0.6	0.9870779
LA CHURRASKITA	0.4	1.0
XINÈS GRAN MURALLA	0.35	0.9870779
PIZZERIA COCOLINO	0.325	0.9870779
DOWNTOWN	0.1	0.9870779
TELEPIZZA	0.1	1.0
MCDONALDS	0.05	0.9870779
LA PENYORA	-1.0	0.98806596
EL MOLÍ DE SANT VICENS	-1.0	0.98806596
CADILLAC CAFE	-1.0	0.98806596
PALAU DE LA MERCÈ	-1.0	1.0
TAHITÍ	-1.0	0.98806596
CAL BLAI	-1.0	0.989055
ACAPULCO	-1.0	0.989055
QUÉ TAL?	-1.0	0.98806596
PIZZERIA EL RACÓ	-1.0	0.98806596

Sortir

Fig. 9. Restaurants in the User Profile

Restaurant Information

When the user clicks in a restaurant, an information window is showed (see Fig. 10).

Fig. 10. Restaurant Information

At the top of the window there is information about the restaurant, in the middle there is information about the explicit interest that the user has shown about the restaurant and, at the bottom, information concerning to the user's implicit interest about the restaurant is presented.

Restaurant Information: it is the objective information about the restaurant.

- Restaurant: name of the restaurant.
- Telèfon: restaurant phone number.
- Adreca: restaurant address.
- Població: city of the restaurant.
- Horari: restaurant hours.
- Dia Tancat: day/s that the restaurant is closed.
- Cuina: cuisine of the restaurant. The different cuisines are shown in Table 1.
- Preu aprox.: approximate price of having a meal in the restaurant. The different possibilities are shown in Table 2.
- Menus: specifies when the restaurant has daily specials. The different possibilities are shown in Table 3.
- Capatitat: capacity of the restaurant.
- Qualificació: the restaurant class expressed as fork ratings. The different possibilities are shown in Table 4.
- Banquets: banquet service.
- Reserves: need to make a reservation.
- Parking: car park.
- Aire Acondicionat: conditioned air.

- Espectacles: performances in the restaurant (e.g., live music).
- Animals: pet care service.
- No fumadors: non-smoking zone.
- Acc. Discapacitats: special access for disabled people.

User's Explicit Interest: it is the opinion of the user about the restaurant. The possible values of the opinion are: "Molt Bona" (very good), "Bona" (good), "Regular", "Dolenta" (bad) and "Molt Dolenta" (very bad).

- Rel. qualitat-preu: the quality price relation.
- Ambient: the restaurant ambient.
- Quantitat menjar: the quantity of meal.
- Eficàcia servei: the efficiency of the waiters.
- Originalitat cuina: the originality of the cuisine.

User's Implicit Interest: it is the information that the agent captures about the implicit interest that the user has shown about a restaurant.

- Recomanacions: the number of times that the restaurant has been recommender to the user.
- Visites: the number of times that the user has visited the restaurant information window.
- Consultes: the number of times that the restaurant has been retrieved as a result of a user's search.

Table 1. Cuisines

Catalan	English
Totes	All
Alta Cuina	High Cuisine
Altres Cuines	Others
Argentina	Argentinean
Basca	Basque
Brasa	Grilled Meat
Casolana	Homelike
Castellana	Spanish
Catalana	Catalan
Creperie	Creperie
De Mercat	Market
De tapes	Appetizers
Empordanesa	Empurdanese
Francesa	French
Grega	Greek
Hamburgueseria	Burger
Imaginativa	Imaginativeness
Internacional	International
Italiana	Italian
Mediterranea	Mediterranean
Mexicana	Mexican
Peix i marisc	Fish and seafood
Pizzeria	Pizzeria
Rapida	Fast food
Tailandesa	Thai
Tradicional	Traditional
Vegeteriana	Vegetarian
Xinesa	Chinese

Table 2. Prices

Catalan	English
Menys de 6 euros	Less than 6 euros
De 6 a 12 euros	From 6 to 12 euros
Menys de 15 euros	Less than 15 euros
De 12 a 18 euros	From 12 to 18 euros
De 18 a 24 euros	From 18 to 24 euros
De 18 a 30 euros	From 18 to 30 euros
De 30 a 45 euros	From 30 to 45 euros
De 45 a 60 euros	From 45 to 60 euros
De 60 a 90 euros	From 60 to 90 euros
Mes de 90 euros	More than 90 euros

Table 3. Daily Specials

Catalan	English
Sense Menus	No daily specials
Migdia	Lunch
Nit	Dinner
Migdia i Nit	Lunch and dinner

Table 4. Restaurant qualifications

Catalan	English
	No rating
1 forquilla	1 fork
2 forquilla	2 forks
3 forquilla	3 forks
4 forquilla	4 forks