International Journal of Artificial Intelligence,
Spring 2011, Volume 6, Number S11
ISSN 0974-0635;

A Multi Agent Architecture for Single User and Group Recommendation in the Tourism Domain

Sebastia L., Giret A. and Garcia I.

Department of Computer Science
Universidad Politecnica de Valencia
Valencia, Spain
{lstarin, agiret, ingarcia}@dsic.upv.es

ABSTRACT

In this paper, we present a Multi Agent System aimed to support a user or a group of users on the planning of different leisure and tourist activities in a city. The system integrates agents that cooperate to dynamically capture the users profiles and to obtain a list of suitable and satisfactory activities for the user or for the group, by using the experience acquired through the interaction of the users and similar users with the system. Moreover, the system is also able to generate a time schedule of the list of recommended activities thus forming a real activity plan. This paper focuses on the architecture and functional behaviour of our system.

Keywords: recommender systems, multi agent systems, group recommendation, planning.
1998 ACM Computing Classification: H.1.2 User/Machine Systems - Human Factors, H.3.5 Online Information Services - Web-based services, I.2.8 Problem Solving, Control Methods, and Search - Plan execution, formation, and generation, I.2.11 Distributed Artificial Intelligence - Multiagent systems.

1 Introduction

Nowadays, most people who plan a trip or a day-out will first initiate a search through the Internet. More and more people realize the advantages of the new technologies for planning leisure activities as an increasing number of companies and institutions offer tourist information which is easily accessible through web services. However, these sites provide the same piece of information to all users without considering the specific profile or needs of the user, which causes that the user has to process a large amount of information in order to pick up the preferable items.

e-Tourism (Sebastia L., Garcia I., Onaindia E., Guzman C., 2009) is an user-adapted tourism and leisure application for a single user or a group. It recommends the user/group a list of activities to perform in the city of Valencia (Spain) by analyzing the captured users profiles. It also computes a time schedule for the list of recommended activities taking into account the distance between places, the opening hours, etc. - that is an agenda of activities. We already have an implementation of this application, but we noticed some requirements that make it more appropriate to develop this application as a Multi-Agent System (MAS). These requirements include: new users should be able to enter the system at any time, existing users should be able to leave the system as well, users must be able to form a group and join and leave the group when required, recommendation technics must be easily added and changed as convenience, tourism activities and information need to be updated accordingly, and new recommendation and planning techniques should be easily integrated. Moreover, cooperation scenarios should be created on demand depending on the tourism preferences of the user and the recommendation provided. To this end, the *e-Tourism* application requires a flexible architecture to implement multiple users, multiple general and particular tourism preferences, a negotiated planned agenda, different planning and recommendation techniques, etc.

In this work we present a Multi-Agent architecture of *e-Tourism* focusing on the system components and its functional behaviour. This MAS architecture provides flexibility, openness, adaptability, and scalability to a tourism recommender and planning system.

This paper is organized as follows. Section 2 gives a brief state of the art of Multi-Agent RS and group RS. Section 3 introduces the MAS architecture of e-Tourism and describes the knowledge representation used. Sections 4 to 6 detail each agent and organization in this architecture. We finish with some conclusions and further work.

2 Background

A **Recommender System** (RS) (Resnick P., Varian H., 1997) is a personalization tool that attempts to provide people with lists of information items that best fit their individual tastes. A RS infers the user's preferences by analyzing the available user data, information about other users and information about the environment. Most of the implemented RS are either stand-alone or client-server systems (Niinivaara O., 2004). Stand-alone systems can only infer the recommendations by inspecting some existing information source. There is no possibility for explicitly stating opinions and no possibilities of exchanging information between agents (Foner L., 1997). A Multi-Agent RS uses agents to help in the solution process, trying to improve the recommendation quality. The agents cooperate and negotiate in order to satisfy the users' preferences, interacting among themselves to complement their partial solutions or even to solve conflicting preferences that may arise when planning a group tourism activity.

In the tourism domain, (Lorenzi F., 2006) proposes a MAS recommender approach where there is a community of agents in charge of searching information on the components of the travel package. The agents work in a distributed and cooperative way, sharing and negotiating knowledge with the global objective of recommending the best travel package to the user. *CASIS* (Lorenzi F., Santos D. S., Bazzan, A.L.C., 2005) uses a metaphor from swarm intelligence to define a case-based RS with the global objective of recommending the best travel package to the user. Nevertheless, unlike *e-Tourism*, in these two approaches there is no records about previous users so that a new recommendation process to the same user must be re-started from scratch. Besides these systems, mainly focused on recommending a travel package, in the tourism domain we can find other systems aimed at recommending a list of activities that a tourist can perform in a city. In addition, it is interesting to have the recommended visits organized as an agenda in order to visit as many places as possible and to minimize the trips between places. The definition of a tourist plan is a time consuming task that involves managing different kinds of information as opening hours, distances between places or the time spent on visiting each place. So, the task of the tourism recommender system is not only to help select the places to visit but also to organize a visit plan. In this paper we work on recommending tourist plans based on the user preferences and minimizing the trips between places. Our work is similar to *BerlinTainment* (Wohltorf J., Cissee R., Rieger A., Scheunemann H., 2004), which is a MAS that can be used to plan comprehensive day itineraries for entertainment on Berlin and determine current locations, points of interest, and routes. It provides the user with personalized recommendations (using a feature-based filtering) and location-based information. Our approach focuses on recommending a list of the activities that a tourist can perform in the city of Valencia. It also considers activities timetables and distances between the activities to compute the leisure and tourist agenda. But *BerlinTainment* is only able to give a recommendation for a single user.

Systems for recommending items to a group of two or more users in a tourist domain are particularly useful as people usually make group travels (family, friends, etc.). In this case travel recommendations should meet the preferences of the majority of the group members (Ardissono L., Goy A., Petrone G., Segnan M., Torasso P., 2003), taking into account the interests and tastes of the group as a whole and by identifying the individual preferences (Plua, C. and Jameson, A., 2002; Jameson A., 2004). The first task of this type of systems is to identify the individual preferences of the users in the group and then find a compromise that is accepted by all the members of the group.

We will illustrate some group recommender systems for tourism such as *Intrigue* (Ardissono L., Goy A., Petrone G., Segnan M., Torasso P., 2003), *Travel Decision Forum* (Jameson A., Baldes S., Kleinbauer T., 2003; Jameson A., 2004; Jameson A., Baldes S., Kleinbauer T., 2004) or *CATS* (McCarthy K., McGinty L., Smyth B., Salam M., 2006). *CATS* and *Travel Decision Forum* are, for instance, two group RS aimed at recommending, specifically, a vacation destination, whereas *Intrigue* assists a group of users in the organization of a tour and provides an interactive agenda for scheduling the tour.

Intrigue assists a group of users in the organization of a tour. Individual participants are not described one by one but the system models the group as a set partitioned into a number of homogeneous subgroups, and their possibly conflicting individual preferences are separately represented. *Intrigue* uses socio-demographic information about the participants and it elicits a set of preferences to define the subgroup requirements on the properties of tourist attractions, paying attention to those preferences possibly conflicting between subgroups. The group information stores a relevance value to estimate the weight that the preferences of a member should have on the recommendation. Each subgroup may have a different degree of influence on the estimation of the group preferences.

CATS is a conversational critique-based recommender that helps a group of members to plan a skiing vacation. The recommender manages personal as well as group profiles. The individual preferences are elicited by subsequently presenting a recommendation to the user. By critiquing a recommendation, the user can express a preference over a specific feature in line

with their own personal requirements. The group profile is maintained by combining the individual user models and associating critiques with the users who contributed them. The group recommendation displays information relating to group compatibility.
The *Travel Decision Forum* uses animated characters to help the members of a group to agree on the organization of a vacation. At any given moment, at most one member is interacting with the system, the other users in the group are represented as animated characters. The system uses a character that represents a mediator, who directs the interaction between the users. Users must reach an agreement on the set of preferences (group profile) that the recommendation must fulfil. Initially, each user fills out a preference form associating a degree of interest to each preference. The individual user profile as well as the group profile contain the same set of preferences. The degree of interest of a specific group profile is calculated out of the degree of interest of each user in the group. Once the group profile has been created, all group members must agree on the group preference model. The mediator asks each member of the group in turn whether the group model can be accepted or not. Using the users critiques, the mediator reconfigures the preferences ratios, and the recommendation is done using the group preference model.
e-Tourism is also a group recommender system for tourism. Similarly to *CATS* or *Travel Decision Forum*, *e-Tourism* elicits an individual profile for each member in the group, and the group profile is maintained by combining the individual user models. Whereas the systems described above only use aggregation methods, our system also employs other mechanisms as intersection or incremental intersection, a new functionality to elicit group recommendations such that no member in the group is specially promoted or harmed with the decisions; in other words, intersection mechanisms accounts for balanced decisions such that the preferences of all the group members are taken into account equally.
Another distinguishing characteristic of our system is that, instead of making recommendations that directly match the group preferences, *e-Tourism* applies a hybrid recommendation technique by combining demographic, collaborative, content-based recommendation and general preferences-based filtering. This way *e-Tourism* is always able to offer a recommendation, even when the user profile contains very little information. In comparison to other tourist group RS, *e-Tourism* provides a fully-automated mechanism for eliciting a group profile which does not require any interaction among the users, neither personally nor virtually. The resulting preferences of the *e-Tourism* preference elicitation could also be used as the starting point of a conversational mechanism in which users can further express and refine the group preference model. Besides, *e-Tourism* offers an agenda of activities for the group.

3 *e-Tourism* MAS Architecture Overview

In this section the MAS Architecture that implements the *e-Tourism* recommender system is outlined. To this end the main participating roles are described together with their internal functioning and the social relations that configures the whole system.
The Multi-Agent architecture designed and built for the *e-Tourism* application has six main **cooperation scenarios** in which different agent types (roles) interact to achieve some goals.

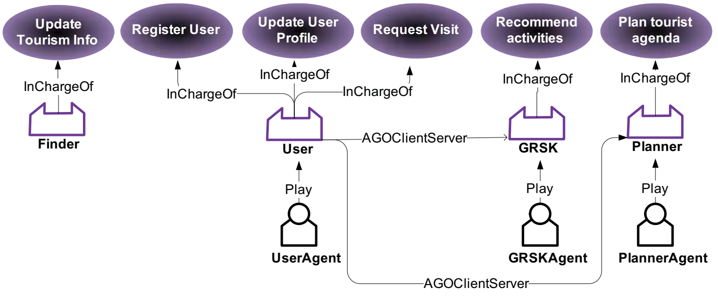

Figure 1: Organization diagram and Use Cases of the e-Tourism system.

Figure 1 shows the different scenarios and the roles in the MAS architecture.
We define four **roles**: the *User* role, to represent users; the *GRSK (Generalist Recommender System Kernel)* role, to represent the recommender system, the *Planner* role, to represent the planning system and the *Finder* role, to represent the information update mechanism.
These roles are in charge of six **use cases**:

1. **Register User**: when a user first enters the system, the first step is to register and enter his/her personal details and general preferences. A group is composed by users already registered in the system.

2. **Request Visit**: each time the user or the group enters the system for a new visit the user/group will be requested to introduce the specific preferences for the current visit: the dates of the visit, time schedule, etc.

3. **Recommend Activities**: when a user/group requests a visit, the GRSK is in charge of generating a list of activities that are likely of interest to the user or to the group of users.

4. **Plan Tourist Agenda**: from the list of recommended activities, the user selects those he/she is really interested in and discards those he/she does not want to be included in the final plan. At this stage the planning system schedules the activities according to the time restrictions of the user and the environment. If the recommendation is computed for a group, the group of users must agree on the activities selection.

5. **Update User Profile**: when the user logs again in the system, he/she is asked to rate the activities in the last recommended plan. These ratings are used to improve the user profile and to gain more suitable recommendations. If a group of users have request a recommendation all users in the group must individually rate the visit.

6. **Update Tourism Info**: the Finder role is in charge of keeping updated tourism information and activities in the system. In this way new activities can be added, existing ones can be deleted or modified, etc.

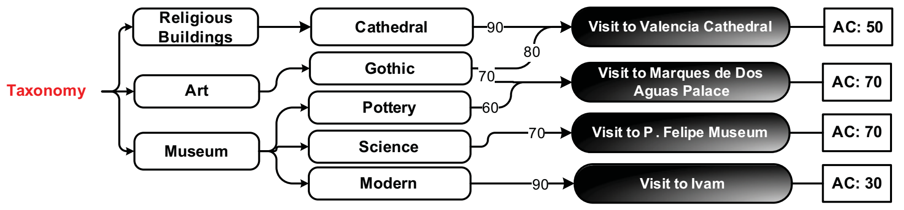

Figure 2: Part of the *e-Tourism* ontology.

3.1 The *e-Tourism* data

The GRSK relies on the use of a **taxonomy** (figure 2) to represent the user's preferences and the items to recommend. It comprises a set of features that are commonly managed in a tourism domain like architectonic styles or types of buildings, extracted from the "Art & Architecture Thesaurus"[1]. Unlike other ontologies (such as Harmonize[2]), which mainly cover accommodation facilities, events and activities, we need to represent feature of the locations and points of interest. Figure 2 shows part of the tourism ontology. An item i in *e-Tourism* (that is, an activity) is defined by a list of preferences.

Definition 1. Let T be the set of all features in the ontology. A *preference* is a pair on the form (f_n^i, r_n^i), where $f_n^i \in T$ is a feature defined in the ontology; $r_n^i \in [0, 100]$ is the degree of interest of f_n to the item i.

For example, according to the ontology in figure 2, the description of *"Visit to Valencia Cathedral"* is set to $\{(\text{Gothic}, 80), (\text{Cathedral}, 90), ...\}$. The item *"Visit to Marques de Dos Aguas Palace"* is described by the pair (Gothic, 70) because it is considered to be a less interesting gothic building than the Cathedral. Additionally, activities are associated a value AC^i to represent how many times it has been performed when recommended. The information about items is updated by means of agents that play the Finder role.
The **user profile** maintain the information required to the user for the recommendation process. It is maintained and stored by the UserAgent (section 4). The profile of a given user stores the following information:

- **Personal and demographic details**: like the age, the gender, the family or the country.
- The **General preferences model** of the user (GP^u) contains the description of the types of items the user u is interested in. This description is composed by a set of preferences (definition 1), that is, it comprises a list of the features in the taxonomy which the user u is interested in along with the user ratings for those features. This information is request to the user in the registration process. It contains a subset of features included in the ontology.
- Information about the **historical interaction** of the user with the recommender system,

[1]www.getty.edu/research/conducting-research/vocabularies/aat.
[2]www.harmonise.org.

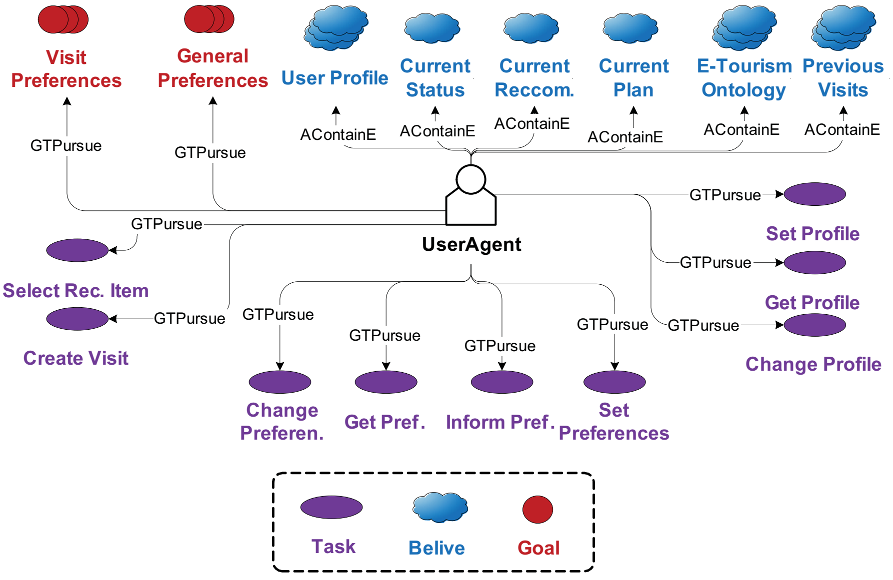

Figure 3: Agent diagram of the UserAgent.

namely the set of items the user has been recommended (and visited) and his degree of satisfaction with the recommended items (rating).

It is important to remark that the user may not indicate all the information required (demographic or interests information) and the system will still be able to provide a recommendation.

4 UserAgent

The User role of figure 1 is played and implemented by one or more UserAgents. This agent represents a user of *e-Tourism* that wants to plan a visit.
A UserAgent (figure 3) stores and handles the **user profile**. This profile can be initialized, modified or consulted by means of the *Set*, *Change* and *Get Profile* tasks, respectively. These tasks are executed whenever the *Register User* and *Update User Profile* Use Cases of figure 1 are executed. Second, the user general preferences model (GP^u) contains the description of the types of items the user u is interested in. We consider GP^u as a goal for our user agent (figure 3) because these preferences must be fulfilled by the recommendation provided to the user. These preferences are initialized when the *Register User* Use Case is executed and handled by means of the *Set*, *Change*, *Inform* and *Get Preferences* tasks.
Each time the user enters the system for a new visit (*CreateVisit* task), that is whenever the *Request Visit* Use Case is executed, he/she will be requested to introduce his current location, which is stored in the **Current Status** and the maximum number N of recommendations he/she desires. The UserAgent keeps data about the current interaction of the user with the system. The list of recommended items provided by the system to the UserAgent (as a result

of the *Recommend Activities* Use Case of figure 1) is stored as the *Current Recommendation*. From this list of recommendations, the user picks up the activities he/she is really interested in, and discards those ones he/she does not want to be included in the final agenda. The remaining items are marked as indifferent. This process is performed by means of the *SelectRecommendedItem* task. The list of selected and indifferent items is considered to build the agenda of activities, that is, the *Current Plan* in the *Plan Tourism Agenda* Use Case of figure 1 with the help of the PlannerAgent (see section 6). The result of the planning process is a list of activities joint with an specific start time and a duration.
When the user is recommended an activity, he/she can perform this activity or not. Moreover, if the user performs the activity, it can be satisfactory (at a certain degree) or not. This information is crucial for the system because it can be considered in order to improve future recommendations. Then, when the user logs again in the system for a new query, he/she is asked to rate the items recommended in the previous interaction. The user will specify which activities he/she has performed and the degree of satisfaction executing the *Update User Profile* Use Case of figure 1. This information is stored as **Previous Visits**.
A **group** of users is a set of users that plans to make a visit together and requests a recommendation for the group as a whole. The users in the group must be previously registered in the system, so the profiles of each user are already informed. The recommendation obtained and the planned agenda is the same for all users in the group and *e-Tourism* returns the same agenda to all the users in the group. The steps in group recommendation are:

- The users in the group make a request for recommendation.
- The users receive the recommended items and select the activities they are interested in.
- They receive the same planned agenda.
- Next time each user enters the system for a recommendation (single user recommendation or group recommendation), he/she is asked to rate the previous agenda. This process is done for each user in the group separately, so each user can give a different rating to the same activity.

5 GRSK Organization

The main goal of the *Generalist Recommender System Kernel* (GRSK) is to generate the list of activities to recommend to the user or the group of users. The GRSK is generalist in the sense that it can work with any application domain as long as the data of the new domain can be defined through an ontology representation (see section 3.1).
Before start explaining the recommendation process, it is important to note that, unlike most RS, GRSK is a semantic RS that does not initially work with the items that will be later recommended to the user. In contrast, GRSK makes use of the concept of feature to elicit the user preference model, which is a more general and flexible entity. This makes GRSK able to work with any application domain as long as the data can be represented through an ontology

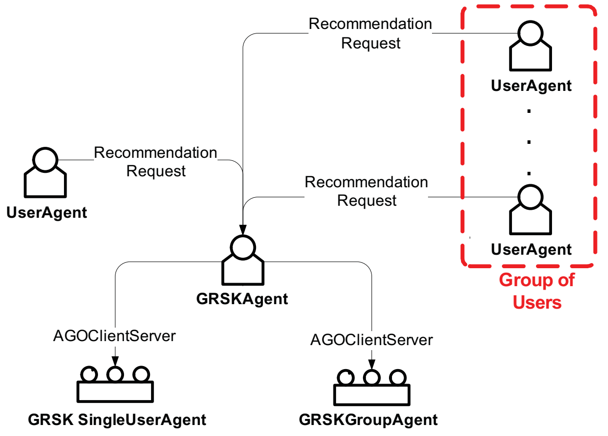

Figure 4: Agents of the GRSK organization.

(Garcia I., Sebastia L., Pajares S., Onaindia E., 2010). Each BRT generates a different set of preferences (see definition 1), an independent list of preferences and hence the lists may contain different features or the same feature with different degrees of interest.
GRSK is an **Hybrid RS** (Burke R., 2007) combines the four **Basic Recommendation Techniques** (BRT): demographic RS, content-based RS, collaborative RS and general preferences-based filtering[3], using a **Hybrid Technique** (HT) mixed or weighted. We have used an hybrid RS because each BRT used separately exhibits some advantages and disadvantages (Adomavicius G., Tuzhilin A., 2005). A common solution adopted by many RS is to combine these techniques into an hybrid RS thus improving recommendations by alleviating the limitations of one technique with the advantages of others (Pazzani M.J., 1999).

5.1 GRSKSAgent Organization

The core of the GRSK (figure 4) is called the **GRSKAgent**, which is the manager of the recommender organization. The GRSKAgent is a facilitator that, in order to respond to a request for a recommendation by an UserAgent or a group of UserAgents, invokes the agents in charge of obtaining the list of recommended items.
Two or more recommendation processes may differ because the process depends on the recommendation being computed for a single user or a group, and on the agents involved in the process. If this request comes from a single user, the GRSKAgent makes a request for recommendation to the GRSKSingleUserAgent. If the request comes from a group of UserAgents, the GRSKAgent makes a request to the GRSKGroupAgent.
The GRSKAgent returns the list of recommended items to the UserAgents involved in the recommendation process, a single UserAgent or the UserAgents in the group (broadcast).

[3]We opted for these techniques because we considered them more suitable for our current domain (tourism and leisure).

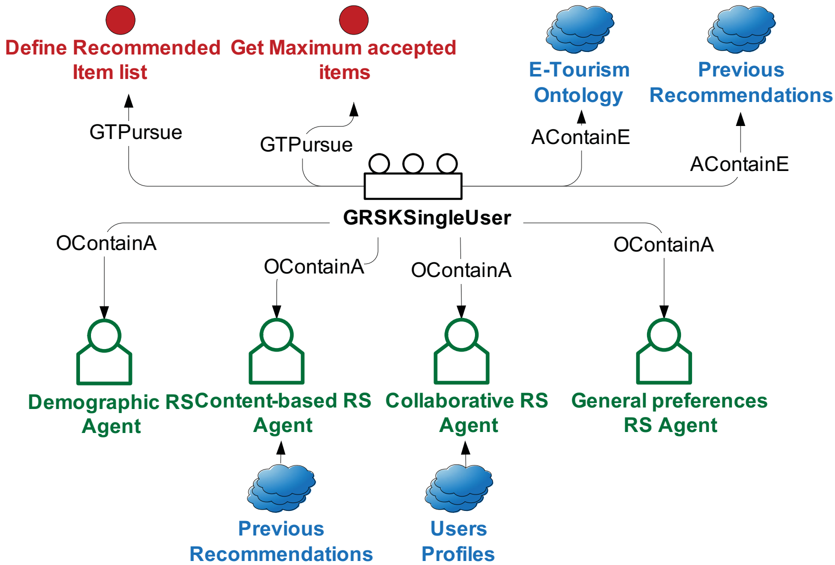

Figure 5: Agents of the GRSK organization (single user).

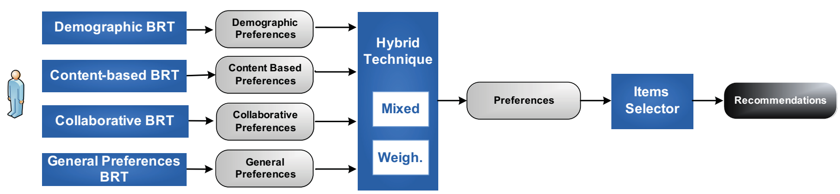

Figure 6: Single user recommendation process.

Both the GRSKSingleUserAgent and the GRSKGroupAgent invoke the different RS agents which derive a list of items to be recommended.

5.2 GRSKSingleUserAgent Organization: Single User Recommendation Process

The *GRSKSingleUserAgent* (figure 5) is in charge of controlling the single user recommendation process. This process (figure 6) consists of three steps:

1. **Apply the BRT to obtain the recommended preferences**: each BRT calculates the set of preferences that better match the user profile from the point of view of the BRT.

2. **Join the recommended preferences**: the HT joins the lists of preferences obtained by the BRTs into a single list.

3. **Obtain the list of recommended items**: the Items Selector selects the items that match the list of user preferences and obtain the final list of recommended items.

5.2.1 Obtaining the Recommended Preferences

In the first step, the BRTs are used to obtain the preferences (definition 1) that better match the user's interests and tastes from the point of view of each BRT by analyzing the user own profile, the profiles of other users and the items previously selected by the users. For a given user, each BRT creates a different list of preferences according to the parameters and data handled by the RS technique. The GRSK includes four agents, one for each RS technique used to compute the recommendation.

The **demographic BRT agent** (Pazzani M.J., 1999) classifies the user into a demographic category according to his profile details. Each demographic category is associated a list of preferences. The success of the demographic recommendation is strongly dependant of this user classification. We opted for a demographic BRT because it is a good alternative to solve the problem of the *new user* since it is able to always give a recommendation, but the recommendation obtained uses to be too generalist and not very accurate (although a good user classification could give good results).

The **content-based RS agent** (Pazzani M.J., Billsus D., 2007) computes a set of preferences by taking into account the items previously visited and rated by the user (historical interaction). This technique increases the user satisfaction by recommending items similar to those already accepted by the user, but, this RS does not introduce novelty in recommendation.

The **collaborative RS agent** (Schafer J.B., Konstan J.A, Riedl J., 1999) suggests those items preferred by people with a profile most similar to the given user profile (i.e. the user will be recommended items that people with similar tastes and preferences liked in the past). This technique is only useful when there is a great amount of data concerning items rated by other users. In order to obtain the corresponding list of preferences, this technique decides whether a user v is similar to the given user u by applying the Pearson Correlation with respect to the items that have been rated by both users v and u. This technique present some difficulties to be applied in this domain (Felfernig A., Gordea S., Jannach D., Teppan E., Zanker M., 2007), because it requires asking the users to rate a great variety of items. This represents a major shortcoming for this type of RS as visiting the same city is not a frequently done activity. But we have used this technique in GRSK because, if there is enough data, the results obtained are very satisfactory.

The **general-preferences-based filtering agent** (Hanani U., Shapira B., Shoval P., 2001) uses an information filtering technique that obtains the preferences that match with the main user interests specified by the user in his profile. The accuracy of this technique depends on the information provided by the user. However, GRSK is able to work with few information.

The BRT agents derive a set of preferences (P^u_{brt}, where $brt \in \{demographic, collaborative, content-based, general-preferences\}$) the recommended items must meet. These BRT agents can obtain a different set of preferences or a different degree of interest for the same feature. Moreover, they are autonomous to decide whether the preference they have obtained are accurate enough to be considered to obtain the final recommendation.

The individual preferences of a user u are denoted by the four lists of preferences: ($P^u_{demographic}$, $P^u_{collaborative}$, $P^u_{content-based}$, $P^u_{general-preferences}$).

5.2.2 Joining the Recommended Preferences

Once all the BRT have computed the corresponding P^u_{brt} the second step in the recommendation process is executed. The HT combines the lists of preferences ($P^u_{demographic}$, $P^u_{collaborative}$, $P^u_{content-based}$, $P^u_{general-preferences}$) in a single list P^u, that conform the final list of preferences for the user.

At this moment, GRSK decides what hybrid techniques to use: the *Mixed Hybrid Technique* or the *Weighted Hybrid Technique*. Only one of this techniques can be used for computing a recommendation for a given user or a group at a time.

The **Mixed Hybrid Technique** mixes the preferences in the lists of all the BRT. All preferences are handled in the same way without taking into account the BRT they belong to. The final user list of preferences is computed as follows. For each feature f_n included in a list of preferences P^u_{brt} returned by a BRT agent, a pair (f_n, r_n) is added to the final list of preferences P^u, where r_n is the average[4] of the values (degrees of interest) associated to f_n in all the preferences lists.

The **Weighted Hybrid Technique** mixes the preferences in the lists, but the value of the degree of interest r^i_n (see definition 1) is adjusted according to the weight of the BRT that have selected the preference. The weight of each BRT is defined in the configuration process. The process followed to obtain the list of preferences is similar. It only differs in the way the preference computed ratio is calculated. This ratio is adjusted using the weight of the BRT the preference belongs to.

5.2.3 Obtaining the List of Recommended Items

The **Items Selector** (IS) is in charge of obtaining the list of recommended items. It receives the list of preferences P^u and returns the set of items that better match the elements in the list. Then, the list of items that match P^u is computed ($GetItems(P^u)$). An item i matches P^u if it satisfies a preference and it has not already been visited by the user u.

The IS computes a priority for each item in this list according to the values of AC^i, the degree of interest associated in the ontology and the estimated degree of interest calculated by the BRT agents. The list of items is ordered according to the computed priority.

The IS selects the N best recommendations, which is the set of recommended items to the user u. This list is finally recommended to the UserAgent. At this moment, the UserAgent executes the *SelectRecommendedItems* task to obtain the filtered recommended items, which will be used by the PlannerAgent to build the agenda.

The GRSK has been designed to have the possibility of easily adding new basic and hybrid recommendation techniques. This is so thanks to the distributed GRSK architecture in which every recommendation technique is encapsulated into an agent. In this way a new technique

[4]For the sake of simplicity, we compute the average, but it could also be defined as the maximum, the median or the addition of the same set of values.

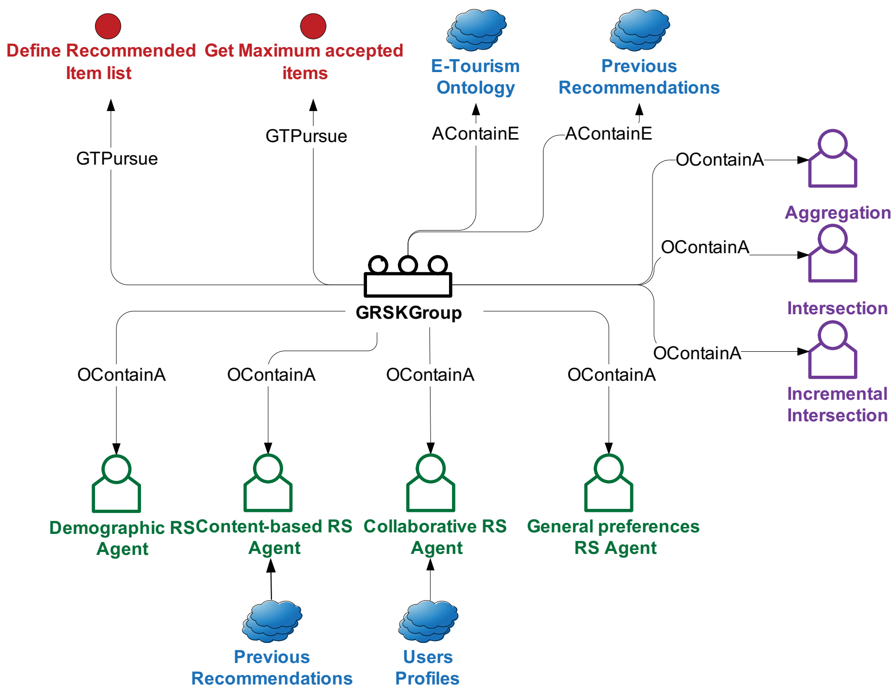

Figure 7: Agents of the GRSK organization (group).

can be easily added by means of a new agent that implements the given techniques and that understands the protocol detailed above. This dynamic structure makes the GRSK organization flexible and adaptable to new requirements, techniques and number of users in the system.

5.3 GRSKGroupAgent Organization: Group Recommendation Process

Since traveling is an activity that usually involves a group of users (family, friends, etc.), travel recommendations should meet the preferences of the majority of the group members (Ardissono L., Goy A., Petrone G., Segnan M., Torasso P., 2003). A group is a set of UserAgents that makes together a request to the recommender indicating they want a group recommendation. In this case, the preferences of all users in the group must be considered to compute the recommendation. The recommendation must equally satisfies all users in the group. The main issue in group recommender is to obtain the group profile (or group preferences model). It must represent the tastes and interest of the group as a whole. Once this group profile is obtained, the GRSK computes the list of recommended items considering the group as a single user.
When the GRSKAgent receives recommendation request from a group of UserAgents, it redi-

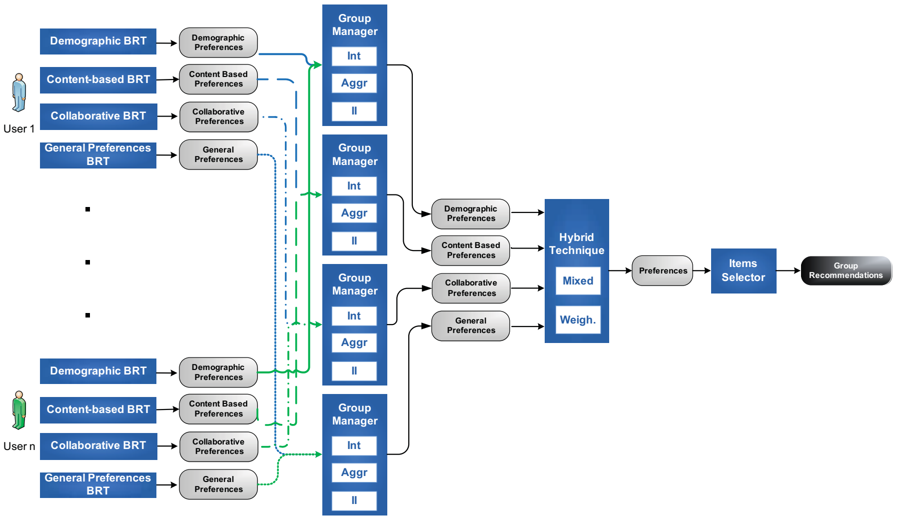

Figure 8: Group recommendation process.

rects the request to the GRSKGroupAgent. The GRSKGroupAgent (figure 7) is in charge of controlling the group recommendation process. This process is composed by the following steps (figure 8):

1. **Apply the BRT to obtain the recommended preferences for each user in the group**: the individual profiles of the users in the group are analyzed and the list of preferences for each user in the group are elicited using the BRT.

2. **Obtain the group preferences**: once the individual preferences of each user are modeled, they are combined to obtain the group preferences by means of a group preference modelling technique. This process is controlled by the *Group Preference Manager* (GPM), which, through methods like *Aggregation*, *Intersection* or *Incremental Intersection*, combines the individual preferences and reports the final group preferences model.

3. **Join the recommended group preferences**: the HT joins the lists of group preferences into a single list.

4. **Obtain the list of recommended items**: the Items Selector selects the items that match the list of group preferences and obtain the final list of recommended items.

This process is quite similar to the single user recommendation process, but adapted to group recommendation. The main difference is the second step, wich is the core of this process. It is in charge of obtaining the sets of preferences that match the group as a whole.

5.3.1 Obtaining the List of Recommended Items for each user in the group

The profiles of each user in the group are analyzed in order to elicit the lists of preferences that match the users in the group. This elicitation mechanism is performed as many times as users in the group using the BRTs (see first step in single user recommendation process, section 5.2.1).
The application of the BRT returns a different list of individual preferences for a user. Therefore, after this stage, we have obtained, for each user u in the group, four lists of individual preferences $(P^u_{demographic}, P^u_{collaborative}, P^u_{content-based}, P^u_{general-preferences})$ which describe the usual tastes of the user from the BRT's point of view. As we said above, each BRT generates an independent list of preferences for each user and hence the lists may contain different features or the same feature with different degrees of interest.

5.3.2 Obtaining the Group Preferences

The second step in group recommendation is to obtain a group preferences model or group profile. A key issue in group RS is how to combine the individual preferences of many different users and elicit a group profile that accurately reflects the tastes of all members in the group. How specification and elicitation of the individual preferences are managed to come up with the group recommendation will determine the success of the recommendation (Plua, C. and Jameson, A., 2002; Jameson A., 2004). The individual preferences of many different users must be combined and used to elicit a group profile, that accurately reflects the tastes of all members in the group.
The aim of this process is obtaining a group profile similar to a single user profile. That is, obtaining a list of preferences by BRT ($P^G_{demographic}, P^G_{collaborative}, P^G_{content-based}, P^G_{general-preferences}$), that satisfies all members in the group. So, the objective of this process is to transform the single user preferences models into a group preferences model. Once the group profile is obtained, the recommendation process follows regardless of the profile represents a user or a group.
To achieve this objective, the GRSK invokes the GPM as many times as number of BRTs are used to compute the recommendation. In this case, the GPM is invoked first to obtain a list of demographic preferences for the group using the list of demographic preferences of each user in the group, later the GPM is invoked to obtain the collaborative group list, and so on. Each time the GPM is invoked is fed with a list of preferences for each user in the group and a single joined list is obtained. The GPM returns $P^G_{demographic}$, the demographic group preference list, $P^G_{collaborative}$ the collaborative group preference list, $P^G_{content-based}$, the content-based group preference list, and $P^G_{general-preferences}$, the general preferences group preference list.
At this moment, the GPM makes use of three disjunctive methods to elicit the group preferences: *Aggregation*, *Intersection* and *Incremental Intersection*. These methods, detailed in (Garcia I., Sebastia L., Onaindia E., Guzman C., 2009), differ on the way the lists of individual preferences are combined. Only one of this methods can be applied in a given moment. The method is select in the GRSK configuration process. In (Garcia I., Sebastia L., Onaindia E., Guzman C., 2009) a comparison of the results when appling these techniques is presented.

The **Aggregation** mechanism is a common technique that has been used in various group RS (see section 2). Aggregation gathers the preferences, computed by the BRT modules, of all members in the group to make up a single set of preferences. A preference is included in P^G_{brt} $brt \in demographic, content-based, collaborative, general-preferences$ if it belongs at least to one of the P^u_{brt} lists. As (Garcia I., Sebastia L., Onaindia E., Guzman C., 2009) show, aggregating preferences does not necessarily account for the preferences of the group as a whole.

The **Intersection** mechanism is introduced as a counterpoint of the aggregation mechanism. This method finds the preferences that are shared by all the members in the group and make up the group preferences. As (Garcia I., Sebastia L., Onaindia E., Guzman C., 2009) show, the advantage of this mechanism is that all of the users in the group will be equally satisfied with the resulting group profile. However, the risk of using intersection is that we might end up with an empty list of preferences if the group is rather heterogeneous.

The **Incremental Intersection** is a more complex mechanism that uses a voting process to obtain the group recommendation. The preferences to recommend will be the ones that describe the most voted features. At the first iteration, we select the features with the highest number of votes (N_v). If the result of the recommendation process is not satisfactory, at the next iteration we select the features that received at least $N_v - 1$ votes, and so on. This way, we incrementally consider the features shared by the largest possible number of users in the group. A detailed explanation of this process is presented in (Garcia I., Sebastia L., Onaindia E., Guzman C., 2009). This mechanism obtains the best results, because it brings together the benefits of the Aggregation and the Intersection techniques.

At this moment GRSK includes these three group modelling techniques but new agents that cope with other functionalities could be added at any moment.

5.3.3 Joining the Recommended Group Preferences and Obtaining the List of Recommended Items

This step in group recommendation is similar to the second step of the single user recommendation (see section 5.2.2). The lists of preferences obtained in the previous step by the GPM ($P^G_{demographic}$, $P^G_{collaborative}$, $P^G_{content-based}$, $P^G_{general-preferences}$) are combined by the HT returning a single list of preferences P^G.

In the last step, the list of preferences P^G is then passed to the IS, which selects the items that best match these preferences. In group recommendation, the items in the final list of recommended items must not have visited by any user in the group. The IS selects the N best recommendations, which is the set of recommended items to the group of users. This list is finally recommended to the UserAgents in the group using a broadcast message. At this moment, the UserAgents in the group executes the *SelectRecommendedItems* task to obtain the filtered recommended items. The same list of recommended items is returned to all the users in the group. This list will be used by the PlannerAgent to build the agenda for the group. The behavior of the IS and the HT is independent from the fact that we are dealing with a single user or a group of users.

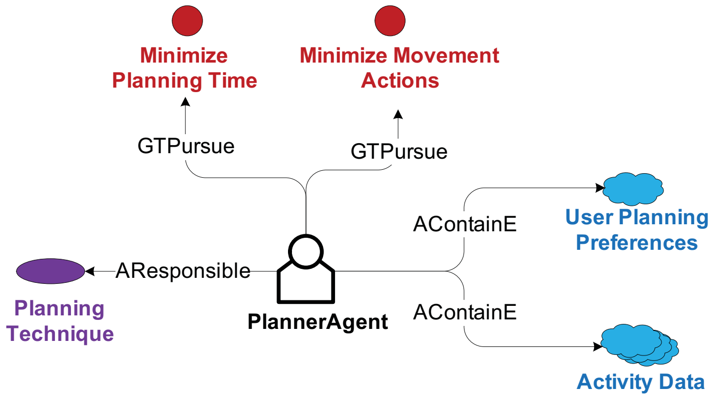

Figure 9: Agent diagram of the PlannerAgent.

6 PlannerAgent

The PlannerAgent (figure 9) is responsible of computing a feasible plan from the list of activities recommended by the GRSKAgent and then filtered by the user.
It manages three groups of different data:

- The **User Planning Preferences**, such as the visit date, the user available time, the current geographical location of the user, etc.
- The information about each activity (**Activity Data**) such as the opening hours of each activity, the address of the place where the activity takes place, the duration of activity, etc.
- Information about the city map (**City Data**), which comprises the city streets and the streets intersections.
- **Recommended items**. Once the GRSK has computed the set of recommended activities, the list of activities are shown to the user or the group and asked to select those activities he/she is interested in and to reject those activities he/she does not want to perform in this occasion. The remaining activities are considered as indifferent. Only the activities marked as selected or indifferent will be considered in the planning process. In group recommendation, the whole group must agree in the selection of these activities.

All these data are properly analyzed and combined by the PlannerAgent to build the user-adapted planning problem.
The PlannerAgent must select which activities among all the filtered activities to include in the plan, because not all these activities will be likely included in the plan since the scheduling will depend on the user available time, his temporal constraints and the time restrictions of the environment (i.e. opening hours of places). Therefore, the planning subsystem must select which activities to include in the plan as well as consider the initial user location and the distance between two consecutive actions to estimate the start time of the second activity.

The final solution plan will contain two types of actions: the performance of the *activities*, and the *movement* actions to go from one activity to the next one.
Each *activity* is described by a pair $(duration, utility)$. The duration of the activity depends on a user classification that distinguishes several types of user (i.e. the duration of the activity ”Visit IVAM Musseum” has a different duration if the user travel with children or not or if the user is very interested in modern art or not). Additionally, two more activities are added, to represent the actions ”*having lunch*” and ”*having dinner*”. If the user wants the plan to include the meals, the duration of both actions must be specified in planning preferences. If the user does not include this information in the planning preferences, the utility of these actions is set to 0. To plan this actions, the typical start/end hours of meals in the city to visit are considered, and, for the sake of simplicity, the locations of these activities coincide with the location of the last performed activity before lunch/dinner.
The set of *movement* actions, represents the movement from the place of activity to the place of following activity. Each movement action is described by its duration, which is computed by taking into account the distance between the places of activities.
(Sebastia L., Garcia I., Onaindia E., Guzman C., 2009) introduces two different ways to tackle this problem: as a *Constraint Satisfaction Problem* (CSP) (Ghallab M., Nau D., Traverso P., 2004), where each action is associated to a variable in the problem and the constraints that establish the relationships between these variables are also defined; and as a *Partial Satisfaction Planning* (PSP) problem (Smith. D.E., 2004). In this case, we specify the problem by means of the Planning Domain Definition Language (PDDL) version 3.0 (Gerevini, A., Long, D., 2005) and use an existing planner to solve it.
From our experience with both formulations, CSP and PSP, we can conclude PSP is better suited for this type of problems. CSP is a general framework for solving any type of constraint-based problem, by finding the variables values that satisfy the conditions imposed by the constraints. A more natural and human-oriented approach of solving a tourist agenda is to use a planning framework for the problem definition and the problem solving, like a PSP formulation, which provides a great flexibility and expressivity to tackle this type of problems.
We used the SAPA planner (Benton, J., Do, Minh B., Kambhampati, S., 2005) in our tourist agenda performance tests. For the particular problem instance we have presented in this section, both the CSP and PSP obtained the same solution plan but the PSP performance was much more efficient. The reason behind this efficiency is that, although SAPA is a domain-independent planner, it makes use of planning heuristics to guide search through the causal relationships between actions. In contrast, in a CSP formulation we cannot define such a causal structure with variables, and so only use generic heuristics for assigning values to variables can be used.
When the system solves the problem either using a CSP or a PSP formulation, we obtain a plan which contains a subset of the activities joint with the time when such activities should start and finish. This plan is shown as an agenda of activities (figure 10).

Figure 10: Tourist agenda.

7 Conclusions and further work

e-Tourism is a multi-agent system that generates personalized recommendations about tourist tours in the city of Valencia (Spain). The *e-Tourism* computes a recommendation for both a single user or a group of users. It is intended to be a service for foreigners and locals to become deeply familiar with the city and plan leisure activities. *e-Tourism* makes recommendations based on the user's tastes, his demographic classification, the places visited by the user in former trips and, finally, his current visit preferences. If the recommendation is requested for a group the *e-Tourism* tries to equally satisfy the preferences of all members of the group. The tool shows the user/group an agenda of recommended activities which reflects the user's tastes or the group's tastes and takes into account the geographical distance between places and the opening hours of these places.
A future research line is to extend the *e-Tourism* group recommendation adding new techniques to obtain the group profile. Our aim is to apply agreement technologies for group recommendation, in order to increase the reliability of electronic communities by introducing human social control mechanisms.

Acknowledgment

Partial support provided by Consolider Ingenio 2010 CSD2007-00022, Spanish Government Project MICINN TIN2008-6701-C03-01 and Valencian Government Project Prometeo 2008/051.

References

Adomavicius G., Tuzhilin A. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, *IEEE Transactions on Knowledge and Data Engineering* **17**(6): 734–749.

Ardissono L., Goy A., Petrone G., Segnan M., Torasso P. 2003. Intrigue: personalized recommendation of tourist attractions for desktop and handset devices, *Applied AI, Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries* **17**(8): 687–714.

Benton, J., Do, Minh B., Kambhampati, S. 2005. Over-subscription planning with numeric goals, *Proceedings of the International Joint Conference on Artificial Intelligence. Edinburgh, Scotland, UK*, Morgan Kaufmann Publishers Inc., pp. 1207–1213.

Burke R. 2007. *The Adaptive Web*, Springer Berlin / Heidelberg, chapter Hybrid web recommender systems, pp. 377–408.

Felfernig A., Gordea S., Jannach D., Teppan E., Zanker M. 2007. A short survey of recommendation technologies in travel and tourism, *OEGAI Journal* **25**(7): 17–22.

Foner L. 1997. Yenta: A multi-agent, referral-based matchmaking system, *Proceedings of the International Conference on Autonomous Agents. Marina del Rey, CA, USA*, ACM Press, pp. 301–307.

Garcia I., Sebastia L., Onaindia E., Guzman C. 2009. A group recommender system for tourist activities, *in* T. Di Noia and F. Buccafurri (eds), *Proceedings of the International Conference on Electronic Commerce and Web Technologies. Linz, Austria*, Vol. 5692 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, pp. 26–37.

Garcia I., Sebastia L., Pajares S., Onaindia E. 2010. Grsk: A generalist recommender system, *Proceedings of the International Conference on Web Information Systems and Techologies. Valencia, Spain*, Vol. 1, INSTICC, pp. 211–218.

Gerevini, A., Long, D. 2005. Plan constraints and preferences in pddl3: The language of the fifth international planning competition, *Technical report*, University of Brescia, Italy.

Ghallab M., Nau D., Traverso P. 2004. *Automated Planning. Theory and Practice*, Morgan Kaufmann.

Hanani U., Shapira B., Shoval P. 2001. Information filtering: Overview of issues, research and systems, *User Modeling and User-Adapted Interaction* **11**(3): 203–259.

Jameson A. 2004. More than the sum of its members: Challenges for group recommender systems, *Proceedings of the International Working Conference on Advanced Visual Interfaces. Gallipoli, Italy*, ACM, pp. 48–54.

Jameson A., Baldes S., Kleinbauer T. 2003. Enhancing mutual awareness in group recommender systems, *in* B. Mobasher and S. S. Anand (eds), *Proceedings of the Workshop on*

Intelligent Techniques for Web Personalization (International Joint Conference on Artificial Intelligence). Acapulco, Mexico, AAAI.

Jameson A., Baldes S., Kleinbauer T. 2004. Two methods for enhancing mutual awareness in a group recommender system, *Proceedings of the International Working Conference on Advanced Visual Interfaces. Gallipoli, Italy*, ACM, pp. 447–449.

Lorenzi F., Bazzan A., A. M. 2006. An architecture for a multiagent recommender system in travel recommendation scenarios, *in* A. Felfernig and M. Zanker (eds), *Proceedings of Workshop on Model-Based Systems (European Conference on Artificial Intelligence). Riva del Garda, Italy*, ECAI, pp. 88–91.

Lorenzi F., Santos D. S., Bazzan, A.L.C. 2005. Negotiation for task allocation among agents in case-base recommender systems: a swarm-intelligence approach., *in* E. Aimeur (ed.), *Proceedings of the Workshop Multi-Agent Information Retrieval and Recommender Systems (International Conference on Artificial Intelligence). Edimburgh, Scotland*, pp. 23–27.

McCarthy K., McGinty L., Smyth B., Salam M. 2006. Social interaction in the cats group recommender, *Proceedings of the Workshop on the Social Navigation and Community based Adaptation Technologies (International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems). Dublin, Ireland*.

Niinivaara O. 2004. Agent-based recommender systems, *Technical report*, Department of Computer Science, University of Helsinki, Finland.

Pazzani M.J. 1999. A framework for collaborative, content-based and demographic filtering, *Artificial Intelligence Review* **13**: 393–408.

Pazzani M.J., Billsus D. 2007. *The Adaptive Web*, Springer Berlin/Heidelberg, chapter Content-based recommendation systems, pp. 325–341.

Plua, C. and Jameson, A. 2002. Collaborative preference elicitation in a group travel recommender system, *Proceedings of the Workshop on Recommendation and Personalization in eCommerce (International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems). Malaga, Spain.*, pp. 148–154.

Resnick P., Varian H. 1997. Recommender systems, *Communications of the ACM* **40**(3): 56–58.

Schafer J.B., Konstan J.A, Riedl J. 1999. Recommender systems in e-commerce, *Proceedings of the ACM Conference on Electronic Commerce. Denver, Colorado, US*, ACM, New York, NY, USA, pp. 158–166.

Sebastia L., Garcia I., Onaindia E., Guzman C. 2009. e-Tourism: a tourist recommendation and planning application, *International Journal on Artificial Intelligence Tools (WSPC-IJAIT)* **18**(5): 717–738.

Smith. D.E. 2004. Choosing objectives in over-subscription planning, *Proceedings of the International Conference on Automated Planning and Scheduling. Whistler, British Columbia, Canada*, pp. 393–401.

Wohltorf J., Cissee R., Rieger A., Scheunemann H. 2004. Berlintainment: An agent-based serviceware framework for context-aware services, *Proceedings of the International Symposium on Wireless Communication Systems. Mauritius.*, pp. 245–249.