MARS: An Agent-based Recommender System for the Semantic Web *

Salvatore Garruzzo, Domenico Rosaci, and Giuseppe M.L. Sarné

DIMET, Università Mediterranea di Reggio Calabria
Via Graziella, Località Feo di Vito
89122 Reggio Calabria, Italy
{salvatore.garruzzo, domenico.rosaci, sarne}@unirc.it

Abstract. Agent-based Web recommender systems are applications capable to generate useful suggestions for visitors of Web sites. This task is generally carried out by exploiting the interaction between two agents, one that supports the human user and the other that manages the Web site. However, in the case of large agent communities and in presence of a high number of Web sites these tasks are often too heavy for the agents, even more if they run on devices having limited resources. In order to address this issue, we propose a new multi-agent architecture, called MARS, where each user's device is provided with a device agent, that autonomously collects information about the local user's behaviour. A single profile agent, associated with the user, periodically collects such information coming from the different user's devices to construct a global user profile. In order to generate recommendations, the recommender agent autonomously pre-computes data provided by the profile agents. This recommendation process is performed with the contribution of a site agent which indicates the recommendations to device agents that visit the Web site. This way, the site agent has the only task of suitably presenting the site content. We performed an experimental campaign on real data that shows the system works more effectively and more efficiently than other well-known agent-based recommenders.

1 Introduction

An overwhelming amount of different recommender systems [5, 9, 10] has been proposed in the last years to support users' Web navigation. They can provide users with useful suggestions, as the most promising pages to visit in a Web site, the items that could meet the user's interest in an E-commerce site, etc. Generally, recommender systems are partitioned in: (*i*) *Content-based*, that recommend to a user the objects which appear similar to those he already accessed in the past; (*ii*) *Collaborative Filtering*, that search similarities among users and consequently suggest to a user some objects also considered by similar users in the past; (*iii*) *Hybrid*, that use both content-based and collaborative filtering

* This work has been partially supported by the MIUR–"Italian Ministry of Education, University and Research", under the Research Project Quadrantis.

techniques to generate recommendations (e.g., a Web site can generate suggestions considering user's personal interests and user's commonalities among other known users). In these situations, hybrid recommender systems have been usually recognized as the most promising solution. Generally, these systems exploit in their recommendation algorithms an internal representation (profile) of the user. In order to construct such a user profile, many recommender systems proposed the use of *software agents*. Specifically, each user is associated to a software agent which monitors his Web activities. When the user accesses a Web site, his agent exploits the profile interacting with the site (e.g., through another software agent associated with the Web site). Finally, the site can use both content-based and collaborative filtering techniques to provide recommendations to the user agent by adapting the site presentation. In this scenario, an emerging issue is that nowadays users navigate on the Web using different devices as desktop PCs, cellular phones, palmtops, etc. Each of these devices presents: (*i*) its own interface characteristics (e.g., display capability), (*ii*) a different cost of Internet connection, (*iii*) different storage space and computational capability. These differences can influence the user's preferences; for example, when he accesses to a site with a cellular phone, he could desire a light site presentation. Consequently, we suppose that, for each user, there should be constructed a different profile for all the devices he uses. Furthermore, the issue (*iii*) leads us to argue that a user should be provided with a different and suitable agent for each device typology he exploits. Moreover, since the user's interests change with the exploited device also the recommender system should be adaptive with respect to the device [1, 7].

We have recently proposed [8] an agent-based recommender system, called Multi Agent System Handling Adaptivity (MASHA), that tackles this important issue. MASHA provides each device with an autonomous *client agent* to collect into a local profile the information about the user's behaviour associated to just that device. This local profile is continuously updated by a *server agent* that manages a global user profile that collects the information provided by the different devices exploited by the user. The third component of this architecture, called *adapter agent*, is capable to generate a personalized Web site representation. This representation contains some useful recommendations derived by (*i*) an analysis of the user profile and (*ii*) the suggestions coming from other users that exploit the same type of device. The main limitation of MASHA is the significant computational cost of the adapter agent activities, due to the execution of the recommendation algorithm. More in detail, let n be the number of site visitors and m be the number of objects present in the site. The computational complexity of the MASHA technique is $\mathcal{O}(m \cdot n^2)$ in the worst case, since it compares the profile of each visitor with those of the other visitors and considers up to m concepts for each visitor.

In this paper we present a *Multi-Agent Recommender System* (MARS) that is an evolution of the MASHA architecture. The main contribution of this work consists in proposing a new recommendation algorithm that takes into account both the visitor profile and the exploited device, and presents a smaller compu-

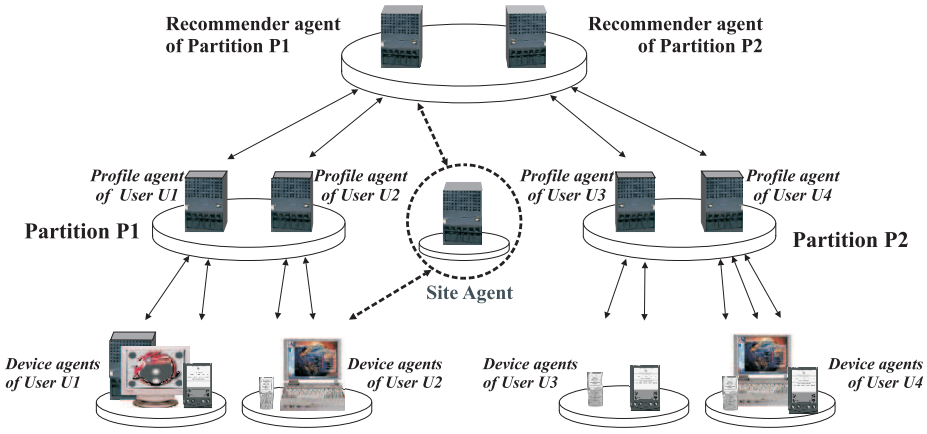

Fig. 1. The MARS Architecture

tational cost with respect to MASHA. The MARS architecture (see Figure 1) maintains the three MASHA agent typologies: (*i*) *device agent*, associated with each device, (*ii*) *profile agent*, associated with each user, and (*iii*) *site agent*, associated with each Web site. Differently from MASHA, the recommendations are not autonomously generated by the site agent, but they are the result of a collaboration between the site agent and a new agent type, called *recommender agent*. Indeed, the basic idea underlying MARS is that of partitioning the profile agents in clusters of users having similar global profiles, where each cluster is managed by a recommender agent. As a consequence, when a visitor accesses a Web site, the associated site agent does not have to make any onerous task, but it simply contacts the recommender agent that is associated with the cluster which the user belongs to. Each recommender agent r internally stores the following elements: (*i*) the profiles of the agents that belong to the partition of r; (*ii*) the catalogue of the objects of each site that has interacted in the past with r; (*iii*) the profiles of the past visitors of the site. Each of this profiles, associated with a past visitor v, collects the objects of the site that has been considered interesting by v. A significant advantage of this approach, is that a site agent that is contacted by n visitors, delegates the task of generating both content-based and collaborative filtering recommendations to the associated recommender agents. This way, the computational cost on the site agent side is $\mathcal{O}(m \cdot p)$ (where p is the number of different clusters) that results significantly lower than MASHA. Differently from other recommender systems [3, 6, 11], MARS presents three original characteristics, namely:

1. In order to construct the global user profile, it takes into account the different devices exploited by the user.
2. It generates the recommendations using the collaboration of the recommender agent, that runs on a server machine and pre-computes most of the necessary data. This way, the task of the site agent becomes very light.

3. It simplifies the task of the device agent, which does not perform neither the construction of the whole user's profile and the generation of the recommendations. Indeed, the user's profile is constructed by the more powerful profile agent running on a server machine, and the recommendations are generated by the site agent in conjunction with the recommender agent.

These characteristics make MARS more effective and more efficient when generating recommendations with respect to other systems, especially in presence of very large agent communities. We have experimentally evaluated MARS by comparing it with other recent profile-based recommender system approaches, and observing a significative improvements of the recommendation performances. The plan of the paper is as follows. In Section 2 we provide an overview of the MARS architecture; Related work is examined in Section 3; some experiments are presented in Section 4. Finally, in Section 5, some conclusions are drawn.

2 MARS Architecture

In this section we describe the MARS architecture that supports (i) the user in his Web navigation by generating personalized suggestions and (ii) the site manager to generate a site presentation in a format suitable for the device currently exploited by the user. To these purposes MARS exploits a suitable user profile. In order to define the notion of user profile, we have to preliminarily introduce the notion of agent *ontology*. We mean as agent ontology a dictionary of the terms used by the agents of the multi-agent system when interacting with each other. We call *concept* a term of an ontology. Here we assume that all the agents in MARS uses the same common ontology and consequently share the same concepts. Moreover, we also assume that all the objects present in the Web sites of MARS can be described by using the concepts of the common ontology. For instance, if an e-commerce site contains a given product (e.g. the book "Anna Karenina"), this product can be considered as an instance of the concept *book* (supposing that this concept is contained in the ontology). Therefore, when in some cases along this paper we say that a Web site contains concept instances, we mean that it contains actual objects. Differently, when we refer to concepts of a Web site, we deal with the concepts of the common ontology which the objects of the site belong to. For each concept of a given Web site visited by the user, the profile stores a value that represents the time spent on the instances of that concept. This time value is considered as a rough measure of the user's interest about the concept and it is strictly related to the characteristics of the exploited device. MARS uses four types of agents, described in detail below. As shown in Figure 1, each user's device is associated with a *device agent* that monitors the user and builds a local user profile. Then, each user is associated with a *profile agent*, running on a server machine, that constructs a complete profile of the user's interests. To this purpose, the profile agent collects the local profiles provided by the different device agents. Profile agents associated to different users, are grouped in partitions, each of them characterized by a specific domain of

interest (e.g. sport, travels, etc.). Each profile agent can belong to different partitions if its associated user is interested in different domains. In its turn, each partition is associated with a *recommender agent* that runs on a server machine and that is able of determining similarities between the agents of the partition. Furthermore, for each Web site of the MARS community, a recommender agent contains a complete list of the concepts of the site and, for each agent of the associated partition, a list of the concepts of the site accessed by that agent. These information are provided by a *site agent* associated to each Web site.

The recommender system works as follows. When the user U accesses a Web site W, the device agent of U interacts with the site agent of W and sends to it some information about the U's preferences. These preferences are relative to the presentation format desired by U when exploiting that device. Next, the site agent contacts the recommender agents of the partitions which U belongs to. These recommender agents pre-computed the concept instances of the site that best match with the device profile of U, to support content-based recommendations. Moreover, recommender agents also pre-computed the concept instances accessed by other users similar to U and that exploit the same device of U, to support information filtering recommendations. Then, these concept instances are transmitted to the site agent of W that generates recommendations for U with a suitable site presentation.

Note that the common ontology O exploited in MARS is realized as an XML-Schema document, where each element represents a *concept*. We suppose that all sites are XML sites that contains instances of concepts that belong to O. We also suppose that each Web page contains some *hyperlinks* represented by pairs (s, d), where s and d are instances of concepts. A hyperlink (s, d) in a page p can be clicked by a user that is visiting p, and the click leads to another page that visualizes d.

2.1 The Device Agent

We associate a device agent DA_i with each device D_i exploited by the user U. During a Web session, the device agent stores some device information and locally updates the user's profile based on the visited concepts. We describe below both the data structure and the behaviour of the device agent.

Device Data Structure The data structure of DA_i can be described by two data structures, namely the *Device Setting* (DS_i) and the *User Profile* (UP_i). DS_i contains the following parameters:

- $RIDSet$, that is the set of recommender agents associated to the partitions which U belongs to;
- $MSSet$, that is the *Maximum Size Set*, containing three parameters that represent the maximum sizes (in Kbyte) of text, audio and video contents that U desires to handle when using D_i;
- $\rho_1, \rho_2, \rho_3 \in [0, 1]$, associated to the actions performable by U (i.e., visiting, storing or printing a Web page);

- T, that is an integer coefficient used to evaluate the U's interest in a concept instance;
- P is the *attenuation period* expressed by the number of days between two consecutive U's actions after which the interest for an unvisited concept decreases;
- ψ, that is a function used to decrease each P days the U's interests relative to the associated concepts that are no longer accessed;
- k, z and r, that are parameters exploited by DA_i in its interaction with the site agent of each visited site (see Section 2.3). In particular, k, z and r respectively represent the number of: (i) interesting concepts belonging to the visited site that U desires to be considered in the site presentation; (ii) similar agents that U desires to be considered in collaborative filtering recommendations; (iii) recommendations to be considered for each similar agent.

UP_i stores the profile of U, based on the whole navigation history and updated on the basis of the hyperlinks that U clicked when exploiting D_i. More in detail, UP_i is a set of tuples $\langle c, IR_i, LU_i \rangle$, each one associated with a concept $c \in O$, where IR_i (*Interest Rate*) is a measure of the U's interest in c by using D_i and LU_i (*Last Update*) is the date of the last IR_i update. Analogously to the approaches [3, 6], in order to set a coefficient, belonging to the interval $[0, 1]$, that reaches the maximum value when $t > T$, we define the measure of interest in c by using the actual time t spent by U when visiting the page containing c. Moreover, U can store, print or simply read the Web page that contains c, and this is taken into account by weighting IR_i with a coefficient ρ_a for each action a (where $a = 1, 2, 3$). More formally, for each new update, IR_i is computed as follows:

$$IR_i = \begin{cases} (IR_i + \frac{t}{T} \times \rho_a)/2 \text{ , if } t \leq T \\ (IR_i + \rho_a)/2 \text{ , elsewhere} \end{cases}$$

In other words, IR_i is computed as the mean value between the previous value of IR_i and the current value $\frac{t}{T} \times \rho_a$, where the ratio $\frac{t}{T}$ is fixed to 1 if $t > T$. Besides, the function ψ is periodically used to decrease the interest rate of the unvisited concepts, based on the temporal distance from the last update.

Device Agent Behaviour DA_i supports U as follows: (i) in order to construct UP_i, DA_i monitors U's Web navigation sessions considering the concepts visited by U and his behaviour when accessing them (note that accessed concepts not yet occurring in UP_i require new elements to add into UP_i). DA_i periodically sends UP_i to its profile agent. (ii) When U visits a Web site, DA_i sends to the site agent the parameters relative to the exploited device to generate a personalized presentation of the Web site for U. (iii) In order to take in account the "age" of the interest rate, each P days DA_i updates the interest rate coefficient ($IR_i = \psi(IR_i, LU_i)$) associated to each concept c .

2.2 The Profile Agent

Each user U is associated with a profile agent (PA) that collects by each U's device agent the information about the concepts visited during U's Web activities. These information are sent to the recommender agents of the U's partitions. This is an important feature of MARS, since the device agents live on the associated devices and could have limited computation and storage capability. The contribution of PA, which runs on a more equipped machine, is fundamental to provide U with an off-line collector of all the information obtained by the different device agents that monitored the U's navigation. Below, both the data structure and the behaviour of PA are described.

Profile Data Structure The data structure of PA contains two elements, namely the *Profile Setting* (PS) and *Global User Profile* (GUP). In its turn, PS stores the following parameters:

- n is the number of device agents associated to PA;
- m is the number of parameters necessary to compute the global interest rates of the various concepts (see below);
- PM is a matrix having n rows and m columns, where each element PM_{ij} is the j-th parameter associated to the i-th device. This matrix is necessary to compute the contribution of DA_i to the computation of the global interest of a concept (see below). It is possible to use as PM_{ij} parameter several characteristics of the D_i connection, for instance the price per byte transmitted, the exploited bandwidth etc.;
- f is a function that accepts as input a PM row and computes as output the contribute of DA_i to the global interest of a concept.

The *Global User Profile* (GUP) stores a global representation of U's interests relative to the concepts visited in his whole navigation when exploiting his devices. It is represented by a pair (IR, GC), where IR is a list of pairs (c, IR_i) such that c is a concept and IR_i is its interest rate computed by DA_i. GC is described by a tuple of the form $\langle c, GIR\rangle$, where c identifies a concept visited by U and GIR is its *Global Interest Rate* shown by U. GIR is the weighted mean of all the interest rates for the concept c. Each weight of IR_i is evaluated by the *weighting function* f by using as input parameters the i-th row of the matrix PM. That is:

$$GIR = \frac{\sum_{i=1}^{n} f(PM_{i,1}, PM_{i,2}, .., PM_{i,m}) \times IR_i}{\sum_{i=1}^{n} f(PM_{i,1}, PM_{i,2}, .., PM_{i,m})}$$

Profile Agent Behaviour The behaviour of PA consists in updating GUP by exploiting the data that each U's device agent periodically sends to PA. These data consist, for each concept c visited by U with the device D_i, of a pair of the form $\langle c, IR_i\rangle$. If c also occurs in the GUP, IR_i is stored in IR and it is immediately exploited to update GIR; elsewhere, if c is a new concept, for the first time visited by the user, a new element is added both in IR and in the set of the global coefficients GC.

2.3 The Recommender Agent and the Site Agent

Two other types of MARS agents are the *recommender agent* (RA) and the *site agent* (SA). Each RA is associated with a set of users that are interested in the same domain. We denote by n the number of users associated with RA. Each SA is associated with a Web site in order to manage the site content.

Below, the data structure of RA and the behaviours of recommender and site agents, that interact together, will be briefly described. We omit to describe the structure of the site agent since it contains only the site catalogue.

Recommender Data Structure The data structure of RA is composed of three elements called *Site Catalogues* (SG), *Global Profile Set* (GPS) and *Profile Collector* (PC). SG contains, for each site W that interacted with RA in the past, a copy of the catalogue C_W that stores all the concept instances present in W. Each catalogue C_W is periodically updated by the corresponding site agent of W. GPS contains the global profiles of all the users associated to RA. The *Profile Collector* PC contains several data sections, each one relative to a site W of the MARS community and denoted by DS_W. In its turn, DS_W contains a list $PSet_W$ containing the profiles associated to the n_W past visitors of W. We denote by $P_{q,i}$ each of these profiles, associated to a given user q and his device i. In particular, $P_{q,i}$ is described by the pair (DP_i, L) that contains both the device profile DP_i and a list L. The elements of L are pairs (c, IR) where c is a concept instance, that q considers interesting in W, and IR is the interest rate of the associated concept. Note that $P_{q,i}$ denotes the profile of a visitor q using a specific device, and not his global profile. Both the information DP_i and L of each visitor profile $P_{q,i}$ are provided to RA by the site agent of W when q terminates its visit.

Recommender and Site Agent Behaviours Each Web site W is associated with a site agent SA. Suppose that the user U visits S by exploiting a device D_i; then the device agent of U sends to the site agent SA the device profile DP_i. Moreover, suppose that U's profile belongs to a partition associated to the recommender agent RA. In this case, SA contacts RA, that has pre-computed personalized recommendations for U, and sends to RA the device profile DP_i of the device D_i. In order to generate content-based recommendations, RA has built a list CB that contains those concept instances of W whose concepts belong to the global profile of U (this global profile is contained in the Global Profile Set of RA). Then, RA orders CB in a decreasing fashion based on the coefficient IR of each concept and maintains only the first k concepts deleting the remaining ones. Remember that k is a parameter contained in the Device Profile DP_i. Moreover, in order to generate collaborative filtering recommendations, RA_j compares the device profile $P_{U,i}$ contained in the data section DS_W, with each profile $P_{q,i} \in PSET_W$ of each other user q, that has visited W in the past and that has exploited the same device. As a result, a list CF of the concepts accessed by the z visitors mostly similar to U is obtained. Remember that also

z is a parameter contained in DP_i. The similarity between $P_{U,i}$ and that of another agent considered in PC is computed as the sum of all the contributions $(1 - d_j)$, with $j = 1, .., l$, where d_j is the difference, in absolute value, between the l instance rates of each concept common to both U and the other agent.

3 Related Work

Many recommender systems using software agents have been proposed in the last years. Below we present a qualitative comparison between some well-known agent-based recommender systems and MARS, pointing out differences and similarities. Other quantitative comparisons will be presented in the next section.

SUGGEST [11] supports user Web navigation dynamically generating links to pages (also belonging to dynamic Web sites) that are unvisited by a user and potentially interesting for him. In order to carry out its task, SUGGEST builds and maintains historical information about the user behaviour by means of an incremental graph partitioning algorithm. Navigational patterns information are extracted by SUGGEST modelling them as a complete graph $G = (V, E)$. The set V of vertices contains the identifiers of the different pages hosted on the Web server. The set of edges E is weighted by the relation: $W_{ij} = N_{ij}/max\{N_i, N_j\}$, where N_i, N_j and N_{ij} are the numbers of sessions (each one identified by the cookies stored on the client side) containing the page i, j or both, respectively. In order to find groups of strongly correlated pages, G is partitioned using a clustering algorithm, and a suggestion list is constructed in a straightforward manner, by finding the cluster which has the largest intersection with the page window correspondent to the current session.

C-Graph [2] proposes an agent model to support a Web user navigation, monitoring his behaviour and learning his preferences, to provide him with a set of recommendations. The user knowledge is modelled into an ontology as a rooted labelled direct graph $\langle N, A\rangle$, where: N is the set of nodes representing the set of concepts of interest for the user U; A, with $A \subseteq N \times N$, is the set of arcs encoding semantic relationships among concepts perceived by U, where the associated arc labels define a number of properties linked to the relationships containing also the model dependency by U. More precisely, an arc (s, t) is provided with a $label(s, t) = \langle d_{st}, r_{st}, h_{st}, \tau_{st}\rangle$, where $d_{st}, r_{st} \in [0, 1]$, h_{st} is a non negative integer and τ_{st} is a real number. The four *label coefficients* above are related to different properties computable by analyzing the visited documents and expressing some kind of relationships among concepts. The approach defines two functions ψ and ρ encoding the structural closeness and the user preferences, respectively. In order to summarize both structural and behavioural components, a function γ is defined to measure the "subjective" semantic closeness of two concepts. The user can set, in computing the semantic closeness, the degree of importance he gives to the structural preference with respect to the behavioural one, by setting an internal parameter k. More in detail, the semantic closeness between two concepts s and t is $\gamma(s, t) = k \times \psi(s, t) + (1 - k) \times \rho(s, t)$.

X-Compass [3] is an XML-based agent model that supports a user U in his Web activities by monitoring the behaviour in the Web pages access to automatically construct and manage an his profile. X-Compass exploits such profiles to provide content-based and collaborative filtering recommendations, as an example, the next page to visit. In particular, each user U is supported by an agent $Ag(U)$ having the following data structures: (i) a *user profile* $P(U)$ that stores U's interests and two relationships existing among them in a rooted graph, in which each node represents a concept of interest for U and has associated an attraction degree $DAttr$ and a key set $KSet$ of the semantics of the interest relative to that node; while each $P(U)$ arc represents both is-a relationships and associative rules, extracted by using data mining techniques (namely: *is-a relationships*, organize such interests in a generalization hierarchy; *associative relationships*, link U's interests appearing distant in the is-a hierarchy but closed from the analysis of the user behaviour); (ii) a list *history* $H(U)$ of elements, each one associated with a Web page access performed by U, ordered on the basis of the temporal access; (iii) an *aggregated history* $AH(U)$, that is a list of elements, each one representing the whole past U's history in visiting the associated Web page. During each Web session, $Ag(U)$ monitors each U access to a Web page to: extract the necessary information, and to update H, AH and $DAttr$ of the node representative of the currently visited page.

CBCF [4] (Content-Boosted Collaborative Filtering) uses a content-based predictor to enhance existing user data, to exploit collaborative filtering to generate personalized suggestions. The content-based approach views content information as text documents, and user ratings as one of six class labels. The collaborative filtering component uses a neighborhood-based algorithm, where a subset of users similar to the active users, and a weighted combination of their ratings is exploited to generate recommendations for the active user.

Similarities and differences with MARS Similarly to MARS, all the aforementioned systems exploits an internal profile to store information relative to the user. The main difference with MARS is that such a profile is stored in a unique agent that supports the user and manages his profile. Differently, in MARS the profile is managed by the profile agent and is built on the basis of the information provided by the device agents associated with the different user's devices. As a result, the device agents only collects information about the user, while the profile construction is performed by the profile agent. Moreover, in MARS recommendations are not generated by the only site agent, but they are the result of a collaboration between the site agent and the recommender agent that pre-computes most of the necessary data. This way, the task of the site agent becomes very light. Finally, none of the above described systems considers the effect of using different devices in the profile construction and, consequently, in the recommendation algorithm. Instead, MARS uses this information providing personalized content-based and collaborative filtering recommendations.

4 Experiments

In this section, we present some experiments devoted to evaluate the capability of MARS to perform both content-based and collaborative filtering activities by compare its performances in generating suggestions with all the systems previously described. We have chosen X-COMPASS, C-GRAPH and CBCF since they are, similarly to MARS, both content-based and collaborative filtering and exploit a user profile. They are, at the best of our knowledge, three of the most performative recommender systems. Moreover, to analyze separately the contribution of the content-based and the collaborative filtering algorithms, we have chosen also the content-based system SUGGEST, that is one of the most performative in this context. For our experiments, we have built 30 different XML Web sites by using a common ontology represented by a unique XML Schema; therefore each page has only instances of this XML schema. Furthermore, we have monitored 97 real users in their Web sites navigation, without using any recommendation support. For each user, a log file has recorded his choices in a list of 700 elements $\langle s, d, t \rangle$, relative to 700 different clicks performed by him during 15 days, where s (resp. d) is the identifier of the *source* (resp. *destination*) concept instance, and t is the time of his choice to go from s to d via a hyperlink.

We have also realized eight different types of device agents, developed by using JADE (Java Agent Development Framework) and JADE/LEAP (JADE Lightweight Extensible Agent Platform) for devices, as palmtop and cellular phones, with limited resources. Four of these agent types are MARS agents (namely device, profile, site and recommender agents) that implement our approach of generating user suggestions. The other four agent types are client agents built by following the recommendation-based approaches called SUGGEST, C-GRAPH, X-COMPASS and CBCF, that we have presented in Section 3 and that we compare in this section with our approach.

MARS Device Agents We have three device agents associated with three different devices, namely a desktop PC, a palmtop and a cellular phone. We have set their parameters (described in Section 2.1) as shown in Table 1. However, we remember that the interest for a concept has been assumed as "saturated" if the visit time of the concept is higher than T seconds. While, the coefficient ρ_1 (resp. ρ_2, ρ_3) weights the user's interest in a concept in the case the user simply visits (resp. stores, prints) a page containing an instance of that concept. Moreover, the attenuation period P is equal to 3 for each device agent; this means that the interest for a concept that has not been visited for three consecutive days is decreased by using the coefficient $\psi \in [0, 1]$. Finally, for each client agent the parameter k is equal to 4, thus showing to the user all the instances of the four most interesting concepts.

Other Client Agents The SUGGEST, C-GRAPH, CBCF and X-COMPASS device agents are built by following the descriptions of the relative data structures and recommendations algorithms proposed in [2–4, 11], respectively. In particular, relatively to the C-GRAPH agent, we have used a coefficient $k = 0.5$ to give the same importance to both the structural and semantics closeness.

Table 1. The setting of the MARS device agents

device agent	T (sec.)	ρ_1	ρ_2	ρ_3	P	ψ	k
desktop PC	200	0.6	0.8	0.9	3	0.90	4
palmtop	120	0.6	0.9	1.0	3	0.95	4
cellular phone	60	0.5	0.9	1.0	3	0.95	4

MARS Profile Agents Each user is associated with a profile agent. All the profile agents adopt the same parameters values: (*i*) $n = 3$, having only three types of device agents for each user. (*ii*) $m = 1$, since we have decided to use, as unique parameter to weight the contribution of the interest rate coming from each client agent, the *price per Mega Byte* (estimated for each adopted device typology). In this case, the matrix PM becomes a vector $[PM_1, PM_2, PM_3]$. The prices per Mbyte (in euro cents) that we have considered are: $PM_1 = 0.9, PM_2 = 1.4, PM_3 = 1.8$. (*iii*) We use the identity function $f(PM_i) = PM_i$ as weighting function. Thus the formula for computing the global interest rate GIR is:

$$GIR = \frac{\sum_{i=1}^{3} PM_i \times IR_i}{\sum_{i=1}^{3} PM_i}$$

4.1 Description of the experiments

In our experiments we monitored the users in their Web visits. We denote with a triplet (s, d, t) the transition of the user, that visits the instance s of the concept c_s, to the instance d of the concept c_d at time t. Initially, in order to allow the users' agents to build their user profiles, for each user we have collected the first 450 triplet (s, d, t) as training-set. Other 300 triplet have been used as test-set to evaluate the recommendation algorithm used by the site agent. That is, for each user, in correspondence of each triplet (s, d, t) belonging to the test-set, we have generated a recommendation $R(s)$, for each of the five algorithms MARS, X-COMPASS, C-GRAPH, CBCF and SUGGEST. Each recommendation $R(s)$ is a list of recommended concept instances. We have checked if d belongs to $R(s)$ in order to measure the effectiveness of the different approaches and we have stored the result in a value $c(s)$. Formally $c_s = \begin{cases} 1 \text{ , if } d \in R(s) \\ 0 \text{ , otherwise} \end{cases}$

The average precision Pre of each recommender method is defined as the average of the $c(s)$ values on all the triplets (s, d, t).
The first row of Table 2 presents the results obtained in this experiment comparing the five approaches. In terms of Average Precision MARS has resulted the best of the other approaches chosen for the comparison (the CBCF in all the occurrences), measuring about a 21, a 48 and a 35 percent better than CBCF, respectively. We argue that this very good performance that MARS obtains as recommender systems, than the other considered approaches, it is due to the fact that MARS considers, in determining its suggestions, also the devices exploited by the user. To confirm such an influence, we have repeated the above

Table 2. Performances of different recommendation algorithms

	MARS	*X-COMPASS*	*C-GRAPH*	*CBCF*	*SUGGEST*
Global	0.270 (0.189)	0.183	0.178	0.198	0.148
Content Based	0.196	0.139	0.140	0.156	0.148
Collaborative Filtering	0.139	0.086	0.081	0.100	-

comparisons, by using the only PC MARS device agent (already used in the previous experiment), instead of three different clients. In this way, the effect of the different devices, exploited by the user in the past, is not taken in account in the generation of the MARS recommendations. Result of this experiment is shown in round parenthesis in Table 2. In this conditions, MARS approach shows performances comparable with, but no higher than those of the other approaches. This confirms that the main advantage of MARS is in the introduction of different device agents associated to each devices exploited by the user. To understand more precisely how such a device consideration improves the recommendation performances, we have repeated the experiment considering separately the content-based and the collaborative filtering components of the experiments. That is, we have generated the recommendations of MARS, X-COMPASS, C-GRAPH, CBCF and SUGGEST only taking in account the concepts deriving from the similarity between the visitor profile and the site content, without considering the concepts suggested by the other users. Since SUGGEST is only a content-based recommender, the suggestions so generated are in this case the same than those of the previous experiments. The results of this experiment are reported in the second row of Table 2 and show that the performances of MARS is about a 26 percent higher than the best of the other systems. This confirms the supposition that taking into account the device exploited in accessing the concepts leads to model more precisely the user preferences, and this positively influences the suggestion performances. Furthermore, we have repeated the experiments with only the three approaches that act also as collaborative filtering methods, those concepts deriving from the suggestions of the other users, without taking into account the content-based component. The result of this latter experiment (reported in the third row of Table 2) shows that the performances of MARS are in this case significantly improved (about 39 percent higher than the other three approaches). We argue that this is the effect of having considered, in generating collaborative filtering recommendations for a user, only those users that exploited his same device.

5 Conclusions

In this paper we have presented a recommender system architecture, called *Multi-Agent Recommender System* (MARS), designed to generate recommendations on the basis of both user profile and exploited device. More specifically, our system is based on the following two ideas. The first is that a device agent monitors a

user that is exploiting a fixed device to build a light profile just for that device, while a profile agent constructs off-line a complete user profile. This leads to make more simple the task of the device agent, that often has limited resources and, on the other hand, to take into account the different exploited devices in constructing the user profile. The second is that each group of agents interested in the same domain is associated with a recommender agent. It computes off line the similarity between these agents and recording the behaviours of the agents in accessing the Web sites of the community, in order to support both content-based and collaborative filtering recommendations. This leads to generate very effective recommendations, taking into account also the exploited devices, and leaving to the site agent the only task of generating the graphical presentation. We have performed some experiments for evaluating the performances of our systems, in comparison with other four agent-based recommender systems, and the obtained results show a significative improvements of the suggestions. It is worth to point out that, besides these performance improvements, the main advantage of the system is, in our opinion, the particular lightness of both the device agent and the site agent that make very efficient the navigation of the agents through the Web sites.

References

1. C.R. Anderson, P. Domingos, and D.S. Weld. Adaptive web navigation for wireless devices. In *17th Int. Joint Conf. on Artificial Intelligence*, pages 879–884, 2001.
2. F. Buccafurri, G. Lax, D. Rosaci, and D. Ursino. A user behavior-based agent for improving web usage. In *CoopIS/DOA/ODBASE*, pages 1168–1185, USA, 2002.
3. S. Garruzzo, S. Modafferi, D. Rosaci, and D. Ursino. X-Compass: An XML Agent for Supporting User Navigation on the Web. In *5th Int. Conf. on Flexible Query Answering Systems, FQAS 2002*, volume 2522, pages 197–211, October 2002.
4. P. Melville, R.J. Mooney, and R. Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, pages 187–192, 2002.
5. M. Montaner, B. López, and J.L. de la Rosa. A taxonomy of recommender agents on the internet. *Artif. Intell. Rev*, 19(4), 2003.
6. J. Parsons, P. Ralph, and K. Gallagher. Using viewing time to infer user preference in recommender systems. In *AAAI Workshop on Semantic Web Personalization*, pages 52–64, San Jose, USA, July 2004.
7. F.J.G. Peñalvo, F. Paternò, and A.B. Gil. An adaptive e-commerce system definition. In *2nd Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Systems*.
8. D. Rosaci and G.M.L. Sarné. MASHA: A Multi Agent System Handling User and Device Adaptivity of Web Sites. *User Modeling and User-Adapted Interaction: The Journal of Personalization Research*, 16(5).
9. B.M. Sarwar, G. Karypis, J.A. Konstan, and J. Riedl. Analysis of recommendation algorithms for e-commerce. In *2nd ACM Conference on Electronic Commerce (EC-00)*, pages 158–167, Minneapolis, USA, October 2000. ACM.
10. J.B. Schafer, J.A. Konstan, and J. Riedl. E-commerce recommendation applications. *Data Mining and Knowledge Discovery*, 5(1/2):115–153, 2001.
11. F. Silvestri, R. Baraglia, P. Palmerini, and M. Serranò. On-line generation of suggestions for web users. *J. of Digital Information Management*, 2(2):104–108, 2004.