

# A multi-agent recommender system for supporting device adaptivity in e-Commerce

Domenico Rosaci · Giuseppe M. L. Sarné

Received: 18 October 2010 / Revised: 11 April 2011 / Accepted: 13 April 2011 /  
Published online: 7 May 2011  
© Springer Science+Business Media, LLC 2011

**Abstract** Traditional recommender systems for e-Commerce support the customers' activities providing them with useful suggestions about available products in Web stores. To this purpose, in an agent-based context, each customer is often associated with a *customer agent* that interacts with the *site agent* associated with the visited e-Commerce Web site. In presence of a high number of interactions between customers and Web sites, the generation of recommendations can be a heavy task for both these agents. Moreover, customers can navigate on the Web by using different devices having different characteristics that may influence customer's preferences. In this paper we propose a new multi-agent system, called ARSEC, where each device exploited by a customer is associated with a *device agent* that autonomously monitors his/her behaviour. Furthermore, each customer is associated with a *customer agent* that collects in a global profile the information provided by his/her device agents and each e-Commerce Web site is associated with a *seller agent*. Based on the similarity existing among the global profiles the customers are partitioned in clusters, each one managed by a *counsellor agent*. Recommendations are generated in ARSEC as result of the collaboration between the seller agent and some counsellor agents associated with the customer. The usage of the device agents leads to generating recommendations taking into account the device currently used, while the fully decentralized architecture introduces a strong reduction of the time costs. Some experimental results are presented to show the significant advantages obtained by ARSEC in terms of recommendation effectiveness with respect to other well-known agent-based recommenders.

**Keywords** Recommender systems · e-Commerce · Multi-agent systems

---

D. Rosaci (✉) · G. M. L. Sarné  
DIMET, Università Mediterranea di Reggio Calabria, Via Graziella, Località Feo di Vito,  
89122 Reggio Calabria, Italy  
e-mail: domenico.rosaci@unirc.it

G. M. L. Sarné  
e-mail: sarné@unirc.it

## 1 Introduction

The rapid growth of networking systems is stimulating an ever-increasing number of traders to participate in e-Commerce worldwide (Papazoglou 2001). Consequently, the Web is transforming from a simple information space to a more complete market space presenting a wide variety of commercial services, ranging from electronic Web-stores to auctions, on-line booking and other services (Adam and Yesha 2000; Levy and Weld 2000). In this context, customers might navigate among thousands Web sites to explore desired sources and make their purchases. However, in order to perform the various tasks typically involved in e-Commerce transactions, a customer has often to spend a large amount of time on the Web. On the other hand, electronic suppliers have similar problems for proposing their products to customers in the most suitable way, taking into account their preferences, habits, etc.

Moreover, nowadays it is usual for a customer to navigate on the Web by using, besides desktop PCs, many other different devices as notebooks, cell phones, palm-tops, etc. Each of these devices has its own physique and technological characteristics (as display or bandwidth capabilities) and usually is deeply different from its desktop counterpart; therefore, it is not suitable that providers deliver the same content to all device typologies. The necessity that recommender systems consider the exploited device in generating suggestions is becoming a key issue in the recommender systems field (Rosaci and Sarnè 2006, 2010; De Meo et al. 2003; Rosaci et al. 2009). Indeed, if a user accesses a site with a traditional desktop PC, the site manager can propose its recommendations by using a presentation with a high structural complexity, supposing the user can benefit from such a presentation by means of his/her browser equipped with full functionalities. Differently, when the device is a cellular phone, the site presentation has to be lighter than in the previous case. Furthermore, a user exploiting a cellular phone has to be provided with suggestions determined also considering the used device; as an example, in the collaborative filtering case the suggestions of the other users exploiting the same device should be preferred when generating recommendations. While in several computer science research fields, a large variety of approaches adapting their behaviour to the exploited device has been proposed (Anderson et al. 2001; Macskassy et al. 2000), none of these approaches has been specifically designed to support e-Commerce site visiting. It is necessary, in our opinion, to propose analogous techniques that address this important issue.

The importance of e-Commerce systems in the Information and Communication Technology is rapidly growing, due to the significant advantages introduced by these systems that improve adaptivity and personalization of the product presentation. In this context, a key challenge for the current generation of e-Commerce systems consists in the effort of increasing the personalization level of the offers. A possibility to face such a challenge is represented by the realization of an effective knowledge sharing among the customers of the system. In particular a customer, who needs to select the most suitable product to buy, should explore the opinions of other customers of the system. A number of new distributed and adaptive e-Commerce systems have been proposed in the last years (Badica et al. 2005; Di Stefano et al. 2002; De Meo et al. 2003) to support customers in their e-Commerce sessions. A strategy that has been widely exploited is the automated extraction of useful suggestions, as the most promising products to access an e-Commerce session, monitoring the customers' behaviour when they visit e-Commerce sites.

Often e-Commerce systems act also as recommender systems that generate some recommendations computed by using different possible approaches as: (1) *Content-based*, recommending to a customer those products that appear the most similar to those he/she already accessed in the past; (2) *Collaborative Filtering*, searching similarities among customers and consequently suggesting to a customer some products also considered by similar customers in the past; (3) *Hybrid*, using both content-based and collaborative filtering techniques to generate recommendations. Generally, these systems use a *profile* of the customer in order to represent his/her interests and preferences, and many of them propose the use of *software agents* to construct such a customer's profile. More in particular, each customer is associated with a software agent that monitors his/her Web activity. When the customer accesses to an e-Commerce site, his/her agent exploits the customer's profile for interacting with the site. In this interaction, the site might use both content-based and collaborative filtering techniques to provide recommendations to the customer's agent in order to adapt the site presentation.

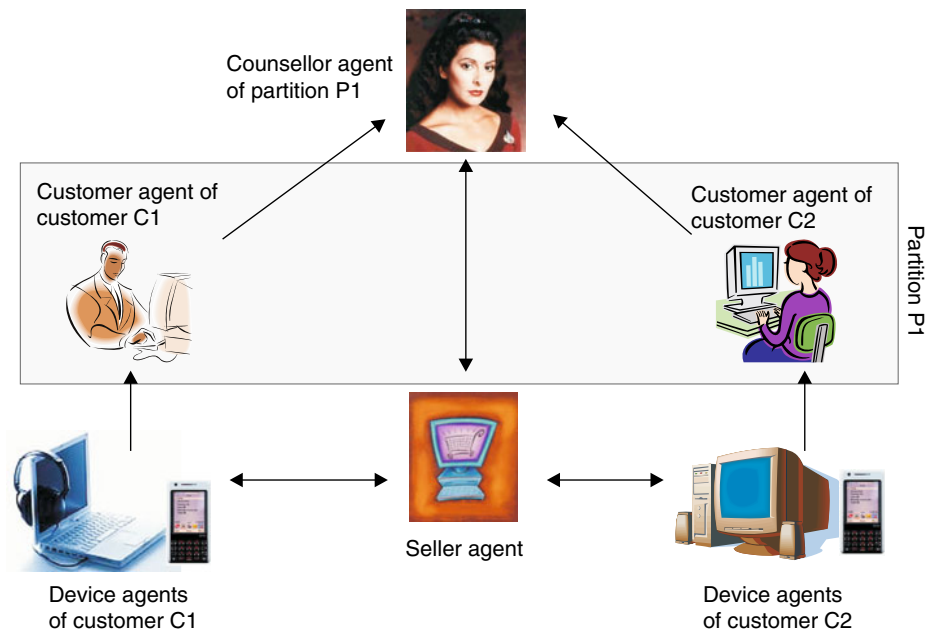
Traditional e-Commerce systems, that exploit content-based techniques for generating recommendations, are realized by means of a client-server architecture. Such an approach does not allow customers to share their experiences with others. A more effective approach is represented by the collaborative filtering, but this implies to implement e-Commerce systems using a decentralized architecture, as in intelligent distributed systems (Badica et al. 2010). However, as previously discussed, in such a scenario an emerging issue is that nowadays customers perform their e-Commerce activities by using different devices as desktop PCs, cellular phones, palmtops, etc. Each of these devices presents: (1) its own interface characteristics (e.g., display capability); (2) a different cost of Internet connection; (3) different storage space and computational capability. These differences can strongly influence the customer's preferences; for example, when he/she accesses to a site with a cellular phone, he/she could desire a light site presentation or, if this is not possible, he/she could avoid to perform that access. Consequently, we suppose that for each customer a different profile for each exploited device should be built. Furthermore, the issue (3) leads us to arguing that a customer should be provided with a different agent for each device typology he/she exploits and, since the customer's interests change with the exploited device, also the recommender system should be adaptive with respect to the device.

In order to face this important issue, some agent-based frameworks for developing recommender systems have been recently proposed. For example, the system MASHA (Rosaci and Sarnè 2006) provides each device with an autonomous *client agent* to collect into a local profile the information about the user's behaviour associated with just that device. A global user's profile is continuously updated by a *server agent* that collects the information provided by the different user's devices. The third component of this architecture, called *adapter agent*, is devoted to generate a personalized Web site representation. This representation contains some useful recommendations derived by both an analysis of the user's profile and the suggestions coming from other users that exploit the same type of device. However, although MASHA effectively handles the problem of taking into account the different devices in the recommendation generation, it presents a significant computational cost for the adapter agent activities, due to the execution of the recommendation algorithm. In fact, if we apply MASHA to an e-Commerce scenario, if  $c$  is the number of customers that visit a given e-Commerce site and  $p$  is the number of products present

in the site, then the computational complexity of the MASHA technique is  $\mathcal{O}(p \cdot c^2)$  in the worst case, since it compares the profile of each customer with those of the other customers, considering up to  $p$  products for each customer.

### 1.1 Contribution

In order to apply the MASHA framework to the e-Commerce context reducing the recommendation costs, we propose in this paper a new multi-agent architecture, called *Adaptive Recommender System for e-Commerce* (ARSEC), that represents an evolution of the MASHA architecture, appositely conceived to support e-Commerce activities. The ARSEC architecture (see Fig. 1) maintains the three MASHA agent typologies, namely: (1) a *device agent*, associated with each device, (2) a *customer agent*, associated with each customer (analogous to the *server agent* in the MASHA architecture), and (3) a *seller agent*, associated with each e-Commerce Web site (analogous to the *adapter agent* in MASHA). However, differently from MASHA, the recommendations provided by ARSEC are not autonomously generated by the seller agent, but they are the result of a collaboration between the seller agent and a new agent type, called *counsellor agent*. The basic idea underlying ARSEC is partitioning the customers in clusters of customers that have similar global profiles, where each cluster is managed by a counsellor agent. Consequently, when a customer visits an e-Commerce site, the seller agent of the site does not perform the onerous task of computing recommendations, but it exploits the help of the counsellor agent associated with the cluster that the customer belongs to.



**Fig. 1** The ARSEC architecture

More in particular, a seller agent that has to generate recommendations for  $c$  customers, delegates the task of computing both content-based and collaborative filtering suggestions to the counsellor agents of the customers. This way, the computational cost of the seller agent is  $\mathcal{O}(p \cdot \pi)$  (where  $\pi$  is the number of different clusters) that results significantly lower than MASHA. Furthermore ARSEC provides the customer with a list of similar customers that can be contacted in a P2P interaction. This feature is particularly useful in an e-Commerce scenario, since allows a customer to discuss the obtained suggestions with the other customers.

We have experimentally evaluated ARSEC by comparing it with MASHA and with three other agent-based recommender system, and we have observed a significant improvements in the quality of recommendations and an improvement of the efficiency.

It is important to remark that the new contribution provided by the approach we present in this paper is represented by the idea of pre-computing both content-based and collaborative filtering recommendations, using the counsellor agent. As described above, this agent is able to compute content-based recommendation for a customer belonging to the associated cluster, using information that are periodically transmitted by the customer agents. In MASHA, customer agents directly interact with the seller agent at the moment of providing recommendation for a customer, and the task of computing content-based recommendation is performed by the seller agent. Instead, in ARSEC the pre-computation of the recommendations is off-line performed by the counsellor agent, so allowing to significantly reduce the computation cost of the seller agent. This idea is also extended to the computation of collaborative filtering recommendations, based on a clustering of customer agents. The use of a clustering technique is not novel for grouping similar customers. However, in our approach, we propose the original idea of using clusters of similar customers to pre-compute collaborative filtering recommendations, off-line with respect to the activity of the seller agent that will use these recommendations when actually necessary. We have proposed to use clusters to extend the MASHA capabilities also in Rosaci et al. (2009), in the context of a multi-agent architecture called MUADDIB. However, MUADDIB exploits a centralized mechanism to manage clusters, that is not suitable to be applied in an e-Commerce scenario. In fact, customers' profiles rapidly change and this implies that clusters have to be periodically recomputed. In such a scenario, a central clustering manager as that of MUADDIB becomes a bottleneck that strongly reduces the efficiency and the robustness of the system. Differently to MUADDIB, ARSEC introduces a completely distributed clustering mechanism, where the task of determining the cluster to which a customer belongs is performed by the associated customer agent, without using any central clustering manager. Moreover, this clustering is dynamically recomputed when necessary, allowing the system to adapt itself to the evolution of customers' interests and preferences.

## 1.2 Plan of the paper

The paper is organized as follows. In Section 2 we provide an overview of the ARSEC architecture, while Section 3 describes the practical use of the system; related work is examined in Section 4; some experiments are presented in Section 5. Finally, in Section 6, some conclusions are drawn.

## 2 The ARSEC architecture

This section is devoted to give a general overview of the ARSEC platform that supports, on one hand, the customer by generating personalized suggestions and, on the other hand, the e-Commerce site by selecting those products potentially interesting for the visiting customer. To this purpose, ARSEC exploits a suitable customer's profile, which represents the categories of interest for the customer, giving to each category a measure of his/her interest.

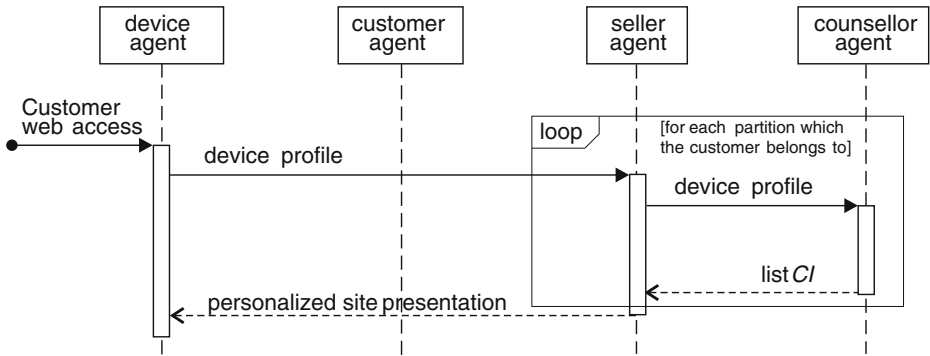
Indeed, in the ARSEC framework, each product of an e-Commerce site belongs to a given product *category*, e.g. *computer* or *books*. In order to make homogeneous the identification of the categories of interest for the customer and the description of the different contents of the e-Commerce sites, all the possible categories are included in a common *dictionary of the categories*, which is shared by all the users (i.e., the agents) of the system. Moreover, we assume that each e-Commerce site that adopts the ARSEC platform contains some products that can be described by using the categories of the common dictionary. For instance, if the e-Commerce site contains a given product, this product is considered as an *instance* of the category that the product belongs to. Therefore, when in some cases along this paper we say that an e-Commerce site contains category instances, we mean that it contains actual products.

For each category accessed by the customer, the profile stores a value that represents the time spent on the products of that category. This time value is considered as a rough measure of the customer's interest about the category and it is strictly related to the characteristics of the exploited device.

ARSEC uses four types of agents, described in detail below and depicted in Fig. 1. First of all, each customer's device is associated with a *device agent* that monitors the customer and builds a local profile of the customer related to just that device. Moreover, in order to collect all the information retrieved by the different customer's device agents, ARSEC associates with each customer a *customer agent*, running on a server machine, that constructs a complete profile of customer's interests and preferences. Customer agents, associated with different customers, are then grouped in partitions, and each customer agent can belong to different partitions.

The main component of the ARSEC architecture is represented by the set of the *counsellor agents*. A counsellor agent is associated with each partition and runs on a server machine. This agent determines similarities between the customers of the partition. Furthermore, for each e-Commerce site of the ARSEC community, the counsellor agent contains a complete list of the categories of the e-Commerce site and, for each customer of the associated partition, a list of the categories of the e-Commerce site accessed by that customer. These information are provided by the *seller agent* associated with each e-Commerce site.

In Fig. 2 is graphically shown how ARSEC works. More in detail, each time customer  $c$  accesses to an e-Commerce site  $E$ , the device agent of  $c$  interacts with the seller agent of  $E$  and sends to it some information about the preferences of  $c$ . These preferences, stored in the *device profile*  $DP$  (see below), are related to the format desired by customer  $c$  for accessing the products when exploiting that device. Furthermore, we suppose  $c$  belongs to  $\pi$  partitions. The  $c$ 's customer agent dynamically performs the task of determining the partitions which  $c$  belongs to. In order to propose useful suggestions for visiting customer  $c$  with a suitable site presentation, the seller agent contacts the counsellor agents  $dt_1, dt_2, \dots, dt_\pi$  of the  $\pi$  partitions



**Fig. 2** The behaviour of ARSEC

whose customer  $c$  belongs to, and transmits to it the device profile  $DP$  of  $c$ . For supporting content-based recommendations these counsellor agents pre-computed the products of the e-Commerce site that best match with the  $c$ 's device profile. Moreover, counsellor agents also pre-computed the products accessed by other customers that are similar to  $c$  and exploit the same device than  $c$ , in order to support collaborative filtering recommendations. Also this task is performed taking into account the customer's preferences contained in the device profile  $DP$ . Then, the so computed products are inserted into a list  $CI$  and transmitted to the seller agent of the site  $E$  that use them to produce on-fly a suitable site presentation for the customer  $c$  with the products offered by the e-Commerce site and considered the most interesting for him/her.

In this paper, we assume that the dictionary of the categories exploited by all the ARSEC agents is realized as an XML-Schema document, where each element represents a category. We suppose that all e-Commerce sites are XML sites containing products of categories that belong to the dictionary. We also suppose that a product of an e-Commerce site can be associated, by one or more *hyperlinks*, to other products contained in the same site. A hyperlink in ARSEC is represented by a pair  $(a, b)$ , where  $a$  and  $b$  are products (i.e. products) and a hyperlink  $(a, b)$  can be clicked by a customer for accessing  $b$  coming from  $a$ .

In the following subsections we describe in details the characteristics of the four types of agents introduced above.

### 2.1 An agent associated with both a customer and a device: the device agent

A device agent is associated with each device exploited by the customer. During an e-Commerce session, the device agent stores some device information and locally updates the customer's profile based on the visited categories. We describe below both the data structure and the behaviour of the device agent.

*Device data structure* The device agent contains two data structures, namely the *Device Profile (DP)* and the *Customer Profile (CP)*.

The Device Profile  $DP$  stores the set of parameters described in Table 1, that represent the customer's preferences when he/she uses that device.

**Table 1** Parameters contained in the device profile

Name	Description
<i>CounsellorSet</i>	Set of the counsellor agents associated with the partitions which the customer belongs to
<i>s1</i>	Maximum size (in Kbyte) of text content the customer desires to handle
<i>ss</i>	Maximum size (in Kbyte) of audio content the customer desires to handle
<i>s3</i>	Maximum size (in Kbyte) of video content the customer desires to handle
$\rho_1$	Parameter (ranging in [0..1]) that weights the importance of visiting a page describing a product
$\rho_2$	Parameter (ranging in [0..1]) that weights the importance of buying a product
$\rho_3$	Parameter (ranging in [0..1]) that weights the importance of monitoring a product
<i>T</i>	Integer coefficient used to evaluate the customer's interest in a product (see (1) below)
$\omega$	Number of days between two consecutive customer's actions after which the interest for a not accessed category decreases
$\psi$	Parameter ranging in [0..1], used to decrease each $\omega$ days the customer's interests related to the categories no longer accessed
<i>k</i>	Maximum number of interesting categories belonging to the e-Commerce site that the customer desires to be considered in an e-Commerce session
<i>z</i>	Number of similar agents that the customer desires to be considered in collaborative filtering recommendations
<i>r</i>	Number of recommendations to be considered for each similar agent

The customer's profile  $CP$  is based on the whole e-Commerce sessions history and updated based on the hyperlinks the customer has clicked exploiting that device. More in detail,  $CP$  is a set of tuples  $\langle \tau, IW, LU \rangle$ , each one associated with a category  $\tau$  that belongs to the common dictionary, where  $IW$  (*Interest Weight*) is a measure of the customer's interest in the category  $\tau$  by using the device and  $LU$  (*Last Update*) is the date of its last update. Analogously to the approaches (Garruzzo et al. 2002; Parsons et al. 2004), in order to have a measure, belonging to the interval [0, 1], that reaches the maximum value when  $t = T$ , we define  $IW$  by using the actual time  $t$  spent by the customer when visiting the page containing  $\tau$ . Moreover, the customer can buy  $\tau$  or add it to favorite or visit the Web page that contains  $\tau$ , and the action  $a$  (where  $a = 1, 2, 3$ ) performed is taken into account by weighting  $IW$  with a coefficient  $\rho_a$ . More formally, for each new update,  $IW$  is computed as follows:

$$IW = \begin{cases} \left( IW + \frac{t}{T} \times \rho_a \right) / 2, & \text{if } t \leq T \\ (IW + \rho_a) / 2, & \text{elsewhere} \end{cases} \quad (1)$$



In other words,  $IW$  is computed as the mean value between the previous value of  $IW$  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 parameter  $\psi$  is periodically used to decrease the  $IW$  value of the unvisited categories, based on the temporal distance from the last update  $LU$ . More in particular, when this temporal distance is a multiple of the parameter  $\omega$ , the current value of  $IW$  is multiplied by  $\psi$ .

*Device agent behaviour* The device agent constructs the customer’s profile  $CP$  by monitoring the customer’s e-Commerce sessions and considering the categories visited by the customer. As shown in Fig. 3, the device agent periodically sends  $CP$  to its customer agent. Moreover, when the customer visits an e-Commerce site, the device agent sends to the seller agent the parameters related to the exploited device to generate a personalized e-Commerce session for the customer. Finally, to take in account the “age” of the interest weight, periodically the device agent updates the interest weight coefficients.

2.2 An agent that build the global profile of the customer and determines the associated partitions: the customer agent

ARSEC associates with each customer a customer agent that collects from each customer’s device agent the information about the categories visited during the customer’s e-Commerce activities. These information are sent to the counsellor agents of the customer’s partitions (see Fig. 3). This is an important feature of ARSEC, since the device agents live on the associated devices and could have limited computation and storage capability. The contribution of the customer agent, which runs on a powerful equipped machine, is fundamental to provide the customer with an off-line collector of all the information obtained by the different device agents that monitored the customer’s sessions. Below, both the data structure and the behaviour of the customer agent are described.

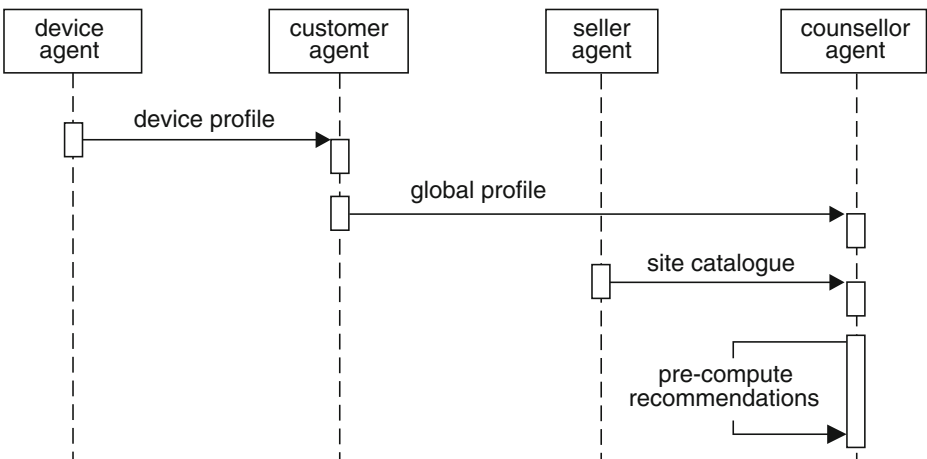


Fig. 3 Pre-computation of content-based and collaborative-filtering recommendations

*Profile data structure* The data structure of the customer agent contains two elements, namely the *Connection Setting (CS)* and the *Global Customer Profile (GCP)*. In its turn, *CS* stores the following parameters:

- *ND*: it is the number of device agents associated with the customer;
- *C*: it is a vector containing *ND* elements, where each element  $c_i$  is the cost of the Internet connection of the  $i$ -th device.
- $\Sigma$ : a threshold value, ranging in  $[0..1]$ , that is used when the customer agent computes the similarity with the available partitions, as we will describe below.
- $\chi$ : the number of hours between two consecutive computation of the similarities with the partition.

The *Global Customer Profile (GCP)* stores a global representation of the customer's interests related to the visited categories. In particular, *GCP* is a list of pairs  $(\tau, GIW)$ , where  $\tau$  identifies a category accessed by the customer and *GIW* is its *Global Interest Weight* shown by the customer, computed as the weighted mean of all the interest weights, related to the different devices. That is:

$$GIW = \frac{\sum_{i=1}^{ND} c_i \times IW_i}{\sum_{i=1}^{ND} IW_i} \quad (2)$$

where  $IW_i$  is the interest weight computed for the given category  $\tau$  by the  $i$ -th device,  $i = 1, \dots, ND$ , and  $c_i$  is the device cost of the  $i$ -th device.

*Customer agent's behaviour* The behaviour of the customer agent consists in (1) updating the global customer profile *GCP* by exploiting the data that each device agent of the customer periodically sends to the customer agent and (2) determining which partitions the customer is associated with. The task (1) does not need of further details, while the task (2) is performed as follows:

Let  $\pi_1, \pi_2, \dots, \pi_P$  the  $P$  available partitions, where each partition  $\pi_i$  managed by the counsellor agent  $\gamma_i$ . The customer agent periodically runs the following activities:

1. It sends a request message to each counsellor agent  $\gamma_i$ , and receives a response message by  $\gamma_i$ , having as content the list  $L_i = \langle GCP_{ik} \rangle$ , where each element  $GCP_{ik}$  is the global profile of a customer  $k$  belonging to the partition  $\pi_i$ .
2. For each list  $L_i$ , the customer agent computes a *similarity measure*  $s_i$  between the global profile *GCP* stored in the customer agent and the global profiles stored in the list  $L_i$ . The measure  $s_i$ , that is a real value ranging in  $[0..1]$  is computed as the mean of all the similarity values  $\sigma_{ck}$  between the global profile *GCP* of the customer  $c$  and each profile  $GCP_{ik}$  belonging to the list  $L_i$ . In order to determine the similarity between the global profiles of two customers  $a$  and  $b$ , we propose to exploit the Jaccard similarity measure (Greenstette 1994). Generally, the Jaccard similarity measure between two sets  $A$  and  $B$  is defined as the number of elements shared by  $A$  and  $B$ , divided by the total number of unique elements in both  $A$  and  $B$  (i.e., it is equal to  $\frac{|A \cap B|}{|A \cup B|}$ ). In our approach, we define the similarity measure between two global user's profiles as the Jaccard similarity measure between these global user's profiles.

**Definition 1** (Similarity value) Let  $GCP_a, GCP_b$  the global user's profiles of the user  $a$  and  $b$ , respectively. The *similarity value* between  $GCP_a$  and  $GCP_b$  is a real number ranging in  $[0..1]$  defined as:

$$\sigma_{ab} = \frac{|GCP_a \cap GCP_b|}{|GCP_a \cup GCP_b|}$$

The mean similarity  $s_i$  with respect to the partition  $i$  is thus computed as follows:

$$s_i = \frac{\sum_{k \in \pi_i} \sigma_{ck}}{|L_i|}$$

3. The customer agent determines those partitions that are associated with a means similarity greater than a fixed threshold  $\Sigma$ . This threshold is set by the customer and is stored in the customer profile. For each of this partition  $\pi_k$ , the customer agent sends a message to the associated counsellor agent  $\gamma_k$ , requesting the membership to the partition and having as content the customer agent's profile  $GCP$ .
4. The operation of computing the similarity with the partitions is periodically recomputed. The period  $\chi$  is fixed by the customer and stored in the customer profile.

In conclusion, we remark that the global activity of all the customer agents in determining the similarity values described above determines a clustering of the customers in the available partitions, that is dynamically recomputed and partially overlapped (since a customer can belong to different partitions. This means that the clustering activity is performed in off-line mode, completely independent of the users' activity. This is a crucial point of our method. Since the clustering activity and the customers' activities are performed independently, the cost of the clustering activity does not influence the efficiency of the customers' activities, in particular the generation of the recommendations. The price to pay for this efficiency is that the recommendations are generated on the current configuration of the clusters, that cannot be the best configuration since the clustering is executed periodically and not each time a modification is produced in the system. However, it appears as a reasonable approximation to use the results of the last clustering activity, performed in the last fixed period of re-clustering, instead to recompute the clustering each way the system is modified. Obviously, if the temporal period  $\chi$  between two consecutive re-clustering instances increases, the activity of the customer agent becomes less onerous. However, if  $\chi$  is too large, the approximation introduced in the recommendations can appear unacceptable. A suitable threshold for  $\chi$  has to be empirically determined by the system administrator, considering the dimension of the e-Commerce community and the personal preferences with respect to the opposite parameters effectiveness/efficiency.

### 2.3 Two agents for generating recommendations: the counsellor agent and the seller agent

A *counsellor agent* is associated with a set of customers interested in the same domain. A domain of interest is a set of categories related to the same subject (e.g. the

domain *hardware* contains the categories *PC*, *notebook*, *printer*, etc.). A *seller agent* is associated with an e-Commerce site in order to manage the products sold by the site. Below, the data structure and the behaviour of both counsellor and seller agents, that mutually interact, will be briefly described. We omit to describe the structure of the seller agent since it only consists in the catalogue of its products.

*Counsellor data structure* The data structure of the counsellor agent is composed of three elements called *Seller Catalogue (SC)*, *Global Profile Set (GPS)* and *Profile Collector (PC)*. The seller catalogue contains, for each e-Commerce site  $E$  that interacted with the counsellor agent in the past, all the products present in  $E$ . The global profile set  $GPS$  contains the global profiles of all the customers associated with the counsellor agent. The *Profile Collector (PC)* contains several data sections, each one related to a site  $E$  of the ARSEC community and denoted by  $DS_E$ . Each data section  $DS_E$  contains in its turn the list of the profiles associated with the past visitors of  $E$ . We denote by  $DS_E[c, d]$  each of these profiles, associated with a given customer  $c$  and his/her device  $d$ . The elements of  $DS_E[c, d]$  are obviously pairs  $(\tau, IW)$  where  $\tau$  is a category, that  $c$  considers interesting in the site  $E$ , and  $IW$  is the corresponding interest weight of  $\tau$ . The information related to each visitor profile  $DS_E[c, d]$  is provided to the counsellor agent by the site agent of  $E$  when the customer  $c$  terminates its session.

*Counsellor and seller agent behaviour* Suppose that a customer  $c$  visits the site  $E$  exploiting a given device  $d$ , as in Fig. 3; then, the device agent of  $c$  sends the device profile  $DP$  to the seller agent. The customer  $c$  belongs to some customer partitions, each one associated with a counsellor agent. In this case, the seller agent contacts each counsellor agent, that has pre-computed personalized recommendations for the customer  $c$ , and sends the device profile  $DP$  of the device  $d$  to the counsellor agent. In order to generate content-based recommendations, the counsellor agent has built a list  $CB$  containing those products of the site  $E$  whose categories belong to the global profile of the customer  $c$  (this global profile is contained in the Global Profile Set of the counsellor agent). Then, the counsellor agent orders the list  $CB$  in a decreasing fashion based on the coefficient  $IW$  of each category and maintains only the first  $k$  products deleting the remaining ones (remember that  $k$  is a parameter contained in the Device Profile  $DP$ ). Moreover, in order to generate collaborative filtering recommendations, the counsellor agent compares the profile  $DS_E[c, d]$  contained in the data section  $DS_E$  and related to the customer  $c$ , with each profile  $DS_E[q, d]$  of each other customer  $q$ , that has visited  $E$  in the past and that has exploited the same device  $d$  of the customer  $c$ . As a result, a list  $CF$  of the products belonging to those categories accessed by the  $z$  visitors less different to  $c$  is obtained (remember that also  $z$  is a parameter contained in  $DP$ ).

The difference between the customer  $c$  and that of another customer  $q$  considered in  $DS_E$  and that use the same device  $d$  is computed as follows. Let  $\tau$  be a category that belongs both to the data section  $DS_E[c, d]$  of  $c$  and the data section  $DS_E[q, d]$  of  $q$ , and let  $IW_c(\tau)$  be the interest rate assigned to the category in the profile of  $c$  and  $IW_q(\tau)$  be the corresponding interest weight in the profile of  $q$ . The value  $d(\tau) = |IW_c(\tau) - IW_q(\tau)|$  is assumed to be a reasonable measure of the difference between the two customer  $c$  and  $q$  in the evaluation of the category  $\tau$ . We measure

the global difference between the two customers  $c$  and  $q$ , denoted by  $D(c, q, d)$  by summing all the contributions  $d(\tau)$  related to all the categories  $\tau$  that the profile of  $c$  and  $q$  share. More formally:

$$D(c, q, d) = \sum_{\tau \in DS_E[c,d] \cap DS_E[q,d]} |IW_c(\tau) - IW_q(\tau)| \tag{3}$$

### 3 Presentation adaptivity

Each counsellor agent of a customer  $c$  that is visiting an e-Commerce site returns to the seller agent of the site the lists  $CB$  and  $CF$ , which contains products suitable to be recommended to the customer  $c$ . Besides these lists, the counsellor agent returns to the seller agent also the *similar customers list* that contains the  $z$  customers more similar to  $c$ . These lists are used by the seller agent to generate an adapted presentation for the visiting customer. In particular, the seller agent generates a Web page that contains the only elements compatible with the specification of the customer’s device, contained in the device profile  $DP$  (see the parameter  $s1, s2, s3$  described in Section 2.1). Moreover, the Web page contains two section of recommendations, namely *The seller recommends* and *The other customers recommend*, containing the products stored in the lists  $CB$  and  $CF$ , respectively. A third section, called *Contact other customers*, gives the possibility to send a message to the customers that have been considered when generating the  $CF$  list. Figure 4 shows an example of two different presentations of the same e-Commerce site for two different devices of the same customer, a desktop PC (Fig. 4a) and a palmtop (Fig. 4b), respectively.

We remark the differences in the graphical aspect of the presentation: that of the palmtop does not contain any figure since the customer has set a parameter  $s3$  (maximum size of graphic object) to a value smaller than the size of the available

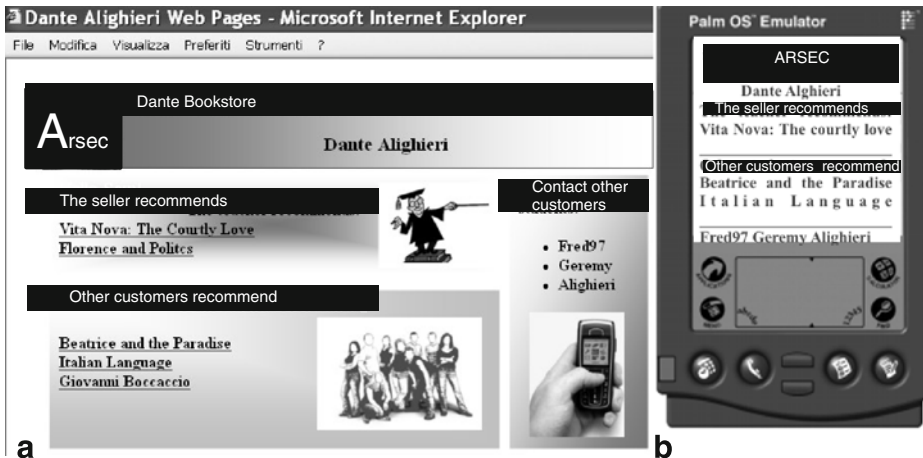
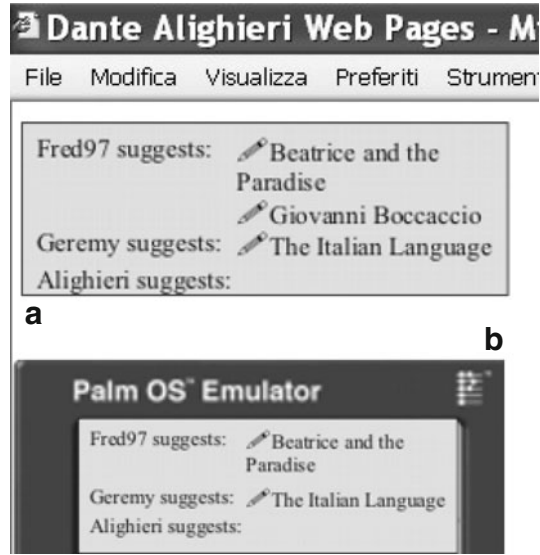


Fig. 4 The ARSEC presentation: (a) on a desktop PC; (b) on a palmtop

**Fig. 5** The other customers recommend: (a) on the desktop PC; (b) on the palmtop



figures, which are displayed on the desktop PC. Moreover, there are also differences in the generated suggestions. In particular, the parameter  $k$ , which represents the maximum number of products of the site that can be shown, is set to 2 (resp. 1) for the desktop PC (resp. palmtop). Consequently, the content-based recommendations in *The seller recommends* section consists of two items for the desktop PC, while for the palmtop only one item is shown. An analogous difference there is in collaborative filtering recommendations. Indeed, the parameter  $z$  is set to 3 for both the two device agents, therefore three similar customers are considered in the *Contact other customers* section: however  $r$  is equal to 2 (resp. 1) for the desktop PC (resp. palmtop) and in consequence the desktop PC shows more items than the palmtop in *The other customers recommend* section. Figure 5 shows how the system provides for each customer considered in *The other customers recommend* section the list of the suggested products. This way, the user can contact one of these customers for discussing, in a P2P interaction, of a given product.

#### 4 Related work

Nowadays, e-Commerce covers almost all the trading fields offering a wide number of goods and services in a real open-world market. As a consequence, users need to handle an increasing amount of information to carry out business processes. It is well known that a direct search usually returns a lot of irrelevant results mainly due both to the polysemy of many words used in the queries and to the difficulty to provide textual specification of product features. To help a user in this complex task, a large number of models and architectures of recommender systems (RSs) exploiting a representation of their past habits (i.e., users' profiles) (Adomaviciu and Tuzhilin 2001) have been proposed in a very large variety of works (also exploiting

agent technology) and described in a considerable number of surveys (e.g. Sarwar et al. 2000; Schafer et al. 2001; Wei et al. 2002; Burke 2002; Montaner et al. 2004; Lee 2004; Manouselis and Costopoulou 2007; Wei et al. 2007).

In this context, content-based and collaborative filtering techniques are commonly adopted to compute suggestions. Content based suggestions depend on the user's experience (i.e., a suitable set of key attributes and similar/relevant products data) and recommend products or services similar to the previous user's interests (Wang and Shao 2004). Content based systems are mainly affected from limitations related to the problems of attribute selection (Cheung et al. 2003) and over-specialization (due to the impossibility of exploring new items dissimilar to those known). In the collaborative filtering case, products or services are recommended to user based on preferences of other users having similar interests (Shardanand and Maes 1995) but in this case high computational costs are needed to compare many consumers' profiles and products (Weng and Liu 2004). Therefore, in presence of data having high dimensionality and sparsity, RSs can adopt clustering algorithm (Jain et al. 1999; Mobasher et al. 2002; Xu and Wunsch 2005; Berkhin 2006). Cluster models (and the associated algorithms) perform off-line some computations to group customers in many segments; a user will be assigned to the segment having the most number of similar customers. This way comparisons will be performed only in a subset of the overall data space.

However, content based and collaborative filtering methodologies are not alternative but complementary and to solve their weaknesses they can be combined, as in ARSEC, to generate most effective suggestions (Herlocker et al. 2000; Burke 2002). This type of RS is classified as hybrid, a category including all those RSs adopting two or more different techniques for generating suggestions.

To increase the personalization degree, RSs can exploit in building customers' profiles also Web usage data extracted by navigational and behavioral information related to e-Commerce activities (Zhang et al. 2000; Parsons et al. 2004; Wang and Shao 2004; Garruzzo et al. 2002; Kim et al. 2005; Liu and Keselj 2007). This additional information from one hand increases the opportunities for RSs to match users' needs but on the other hand increases the task required for its analysis. For such a reason, in ARSEC only a small number of Web usage data are exploited (see Section 2.1). Furthermore, users have currently the opportunity to access Web resources by using more device typologies, different for storage and physique characteristics. Since the exploited device can influence the user's behaviour, RSs should be adaptive also with respect to the device, taking into account its characteristics (considered as environmental data (Kobsa et al. 2001)) in generating recommendations, likely to ARSEC. Differently, many systems are device adaptive only with respect to the Web presentations (Anderson et al. 2001; De Bra et al. 2002; Ardissono et al. 2003).

A popular RSs classification is based on the adopted architecture that can be centralized or distributed. Centralized architectures (CRSs) are adopted by several RSs because are easy to implement. Indeed, each RS has to exploit a unique server and a unique database to perform all its tasks. On the contrary, CRSs are affected from several problems as efficiency, fault tolerance, scalability and customer's privacy. This architecture is implemented in many well known e-Commerce site (e.g., Amazon (<http://www.amazon.com>), CDNOW (<http://www.cdnw.com>), Dandang (<http://www.dandang.com>), eBay (<http://www.ebay.com>)) to support customers (and businesses) providing them with personalized content based and

collaborative filtering suggestions. In particular, the Amazon site generates recommendations based on individual behaviors, or on behaviors of other people. Content based recommendations are provided in its site section “News for You”, while collaborative filtering recommendations are shown in its site section “What Do Customers Buy After Viewing This Item?” suggesting some reasons (statistically computed) to be considered by the customer to buy some items. The system drives the customer to buy something because it is related to something that he/she purchased before or it is popular with other customers. Similar tools are also available in other Web stores such as CDNOW and Dandang. Differently, eBay provides recommendations based on its customers’ profiles built with the feedbacks provided from customers and sellers as measures of their satisfactions. Besides, eBay makes available the “Gift Finder” tool to help customers finding presents that match with the profile of their gift recipient.

WebSell (Cunningham et al. 2000) is an XML agent platform where customers are provided with a set of tools that would extend the range of products and services to trade on the Web. In this platform recommendations are generated exploiting the two considered approaches to bring customers together with products potentially of their interest. Focused on proposing new products to customers, two recent CRSs are presented in Stormer (2007) and Parikh and Sundaresan (2009); the first considers seasonal products (based on the quantities purchased, the turnover and the number of buyers); the second highlights novel products related to certain aspects of user’s behavior that are subject to huge surges and falls popular among other users. The approach called **CBCF** (Melville et al. 2002) (Content-Boosted Collaborative Filtering) adopts both (1) a content-based predictor exploiting a Bayesian classifier (Mitchell 1997), able to learn a user profile and predicting unrated products, and (2) a collaborative filtering that uses a neighborhood-based algorithm (Herlocker et al. 1999), in which similarity is computed by means of a Pearson correlation, for generating personalized suggestions. CDCF requires that a user has to rate, on one of six classes, each accessed item.

Many RSs adopt clustering algorithms as, for example, in Castro-Schez et al. (2011) that proposes a multi-agent system, FIPA compliant, that hierarchically arranges product categories in a dynamic catalogue based on sets of attributes, clusters personalized recommendations, computed also from imprecise and vague search preferences and finds possible product alternatives. To discover profiles that could be used for real-time Web personalization of suggestions, in Mobasher et al. (2002) it is proposed an approach for clustering both users’ transactions and visited Web pages in order to capture the most relevant users’ navigational patterns. Furthermore, some CRSs (EC-XAMAS (De Meo et al. 2007) and MASHA (Rosaci and Sarnè 2006)) take into account also the device in computing personalized suggestions. EC-XAMAS helps customers to find products and/or services of interest, according to their past interests and behaviors. It preserves customers’ privacy by entirely computing all the needed activities for generating suggestions on the client-side, while ARSEC uses a more efficient mechanism based on off-line pre-computation. To overcome limitations of device computational resources and to decrease the cost of the recommendation activities performed on the server side, as in EC-XAMAS, in MASHA is proposed an apposite architecture.

All the aforementioned RSs, as ARSEC, provide both content based and collaborative filtering recommendations, excepted Stormer (2007) in which suggestions



are content based and Cunningham et al. (2000) and Parikh and Sundaresan (2009) where is adopted a collaborative filtering approach. Differently from ARSEC, they are fully centralized, recording all the information about customers and sellers in a main database and processing them exploiting a single server. Similarly to ARSEC, also the approaches (De Meo et al. 2007; Rosaci and Sarnè 2006; Stormer 2007; Parikh and Sundaresan 2009) exploit Web usage data for calculating recommendations. Likely to ARSEC, only De Meo et al. (2007) and Rosaci and Sarnè (2006) are adaptive with respect to the exploited device. However, the cost for generating recommendations is high in presence of an environment having a great number of client agents. ARSEC introduces a mechanism of counsellors agents that pre-compute off-line the recommendations to decrease computational costs (as it is experimentally evaluated in the next section). Finally, only a restricted number of systems (Cunningham et al. 2000; De Meo et al. 2007; Rosaci and Sarnè 2006) are XML-based or adopt a clustering algorithm (Mobasher et al. 2002; Castro-Schez et al. 2011) in its computations, similarly to ARSEC.

Distributed recommender systems (DRSs) imply distribution of both computation and data storage activities among more autonomous entities that locally share such tasks. The presence of more databases and computational entities make DRSs able to overcome typical CRSs limitations by offering a significant scalability, fault tolerance, privacy preservation and security safeguard. On the other hand, DRSs are characterized by an intrinsic difficulty in design and performances optimization (Ackerman et al. 1999; Tanenbaum and Van Steen 2001; Canny 2002; Olson 2003; Zhong 2007) with a time and space complexity rapidly increasing with the number of involved entities to deal with (Jogalekar and Woodside 2000; Stormer 2007; Parikh and Sundaresan 2009; Rosaci et al. 2009). Often DRSs adopt peer-to-peer (P2P) networks to easily exchange in a decentralized domain data locally stored on each peer and provide them with efficient, scalable and robust routing algorithms to reduce the task of locate specific resources (e.g., CAN (Ratnasamy and McCanne 1999), Chord (Stoica et al. 2001), Pastry (Rowstron and Druschel 2001), and Tapestry (Zhao et al. 2002)).

To provide suggestions to a user about those items rated by the most similar users, an interesting proposal of a distributed personal collaborative filtering recommender is PocketLens (Miller et al. 2004). The aim of this system is to obtain good recommendations whenever and to run over whatever client (also disconnected from the Internet). In PocketLens is exploited a variant of the item-item algorithm proposed in Karypis (2001) and Sarwar et al. (2001) in which the matrix of similarity is constructed in an incremental manner in order to limit computational complexity. More in detail, for each user (1) the item rated are represented as vectors and the similarities is computed based on the *cosine similarity* (Herlocker et al. 1999; Karypis 2001) and (2) his/her matrix of similarities involves only, as rows, the items rated by him/her and, as columns, the items rated by the other users. Furthermore, in PocketLens users can preserve their privacy by deciding what information share with the others. Authors tested this recommender algorithm over more distributed architectures.

Many DRSs deal with adaptive distributed collaborative filtering applied to mobile commerce, to suggest multimedia contents. For example, the PEer-Oriented Recommender system (PEOR) (Kim et al. 2008) associates a personal agent with each peer and his computer. PEOR finds similar neighbors, considers changes

in user's interests, searches new contents recently rated by other users, suggests contents and minimizes time and computational costs. In Tveit (2001) products and services are suggested for marketplace users (associated with personal agent) provided with mobile devices by translating recommendation tasks in search tasks over a P2P topology like Gnutella. Push!Music (Jacobsson et al. 2006) supports interacting social network users, potentially provided with more devices, to recommend music files (considered as autonomous entities provided with individual information) with a collaborative filtering-like approach taking into account the exploited device. Developed to work over Ad-Hoc networks without access to remote online directory services, MobHinter (Schifanella et al. 2008) models similarities among users by means of a graph and uses the current neighbors (a narrow portion of the users' community) to locally refine suggestions computed with a collaborative technique.

Among the agent based DRSs that considers in their recommendations both the content based and collaborative filtering approaches likely to ARSEC, we cite the Competitive Attention-space System (CASy) (Bohte et al. 2004), that recommends shops and bids in competitive markets. Shop agents track market consumers, by means of transactions and visitors' interests (derived by profiles, keywords and product queries) also provided by the other shop agents, to propose market products potentially interesting for them. In this context, also in Lorenzi et al. (2008) cooperative agents exchange information extracted by their knowledge bases and recommend travel packages to the users. Each agent works as an expert in a particular kind of service on the basis of its past experiences (hotel, flights, interchanges, conferences, etc) and autonomously selects the most suitable sub-tasks of a recommendation task. Agent specialization provides to increase their confidence for improving the quality of suggestions obtained by composing all the agent contributions. As a third example of systems in this category, in Weng et al. (2006) recommender agents belonging to different organizations cooperate/compete to improve recommendations quality. For each user's request his/her agent provides recommendations from own resources but also consults and interacts with a set of other similar agents (selected by means of a specific algorithm) in reciprocal competition to provide suggestions.

Finally, similarities and differences of all the aforementioned DRSs with ARSEC are pointed out. All these systems take advantage of the distributed architecture in terms of scalability, risks failure, privacy and security. Only in Bohte et al. (2004), Weng et al. (2006) and Lorenzi et al. (2008) and in ARSEC, the recommendations are generated as content based and collaborative filtering, while the other presented systems are only collaborative filtering-based. In Tveit (2001), Bohte et al. (2004), Jacobsson et al. (2006) and Kim et al. (2008), recommendations are filtered based on the compatibility with the device characteristics.

Users provided with more devices are covered by ARSEC and Push!Music. However, among all the cited DRSs only ARSEC evaluates the weight of the exploited device on the user's behaviour and considers it in computing recommendations. Agent specialization is handled in Lorenzi et al. (2008) where, in a recommendation process, agents autonomously choose to carry out the subtask/s where they are most skilled, while in Weng et al. (2006) agent could have different point of views derived to belong to different organizations; differently, in ARSEC agent specialization derives to run on a specific device and from the assigned job (i.e. device, customer or counsellor agent).

## 5 Experiments

In this section we present some experiments devoted to evaluate the capability of ARSEC in supporting a customer with suggestions about those resources considered as the most useful for him/her. Being ARSEC an extension of the system MASHA, we have here compared the performances of ARSEC and MASHA. Moreover, among those recommenders described in Section 4, ARSEC has been compared also with the recommender systems EC-XAMAS (De Meo et al. 2003) (the most similar to ARSEC), CBCF (Melville et al. 2002) and pocketLens (Miller et al. 2004). In the experiments, 25 e-Commerce Web sites have been considered, where each site is provided with about 50 products. During the experiments 80 customers have been monitored in their e-Commerce sessions and, in particular, the experiments involved 10 of the 25 sites for building the customers’ profiles. The remaining 15 sites have been used to test the systems.

All the e-Commerce sites are realized in XML and a common dictionary is implemented by a unique XML Schema to represent the different categories. Therefore each site contains only instances of this XML schema. For each customer has been recorded his/her choices into a log file, that contains a list of 200 elements  $\langle a, b, t, r \rangle$ , related to 200 different products accessed by the customer, where  $a$  (resp.  $b$ ) is the identifier of the *source* (resp. *destination*) product,  $t$  is the timestamp associated with the choice to cross from  $a$  to  $b$  via a hyperlink and  $r$  is a rate provided by the customer (see below). The above systems have adopted JADE (Java Agent Development Framework) (<http://www.jade.tilab.org>) and JADE/LEAP (Caire 2003) for those devices, as palmtops and cellular phones, with limited resources. In particular ARSEC has been realized by using four agent types (namely device, customer, seller and counsellor agent) that implement our algorithm of generating user suggestions.

*ARSEC device and customer agents* As device agents in the experiments three device agents associated with three different devices have been considered, namely a desktop PC, a palmtop and a cellular phone. Their parameters (described in Section 2.1) are set as shown in Table 2. However, we remember that the interest for a category has been assumed as “saturated” if the product of that category is used for more than  $T$  seconds. The coefficient  $\rho_1$  (resp.  $\rho_2, \rho_3$ ) weights the customer’s interest in a category in the case the user simply visits (resp. buys, adds to favorites) a product that is an instance of that category. Moreover, the attenuation period  $\omega$  is equal to 3 for each device agent; this means that the interest in a category not accessed for three consecutive sessions is decreased by using the coefficient  $\psi \in [0, 1]$ . Finally, for each client agent the parameter  $k$  is equal to 3, to provide the user with all the products of the three most interesting categories. The device agents associated with a customer collaborate with the customer agent associates with him/her. All the customer agents adopt the same parameters values: (1)  $n = 3$ , having only three types

**Table 2** Setting parameters of the ARSEC device agents

<i>Device</i>	$T$	$\rho_1$	$\rho_2$	$\rho_3$	$\omega$	$\psi$	$k$
<i>PC</i>	200	0.65	0.85	0.90	3	0.80	3
<i>Palmtop</i>	100	0.60	0.90	1.00	3	0.90	3
<i>Cellular</i>	60	0.50	0.90	1.00	3	0.90	3

of device agents for each user. (2) The prices per Mbyte (in euro cents) considered are:  $PM_1 = 0.9$ ,  $PM_2 = 1.4$ ,  $PM_3 = 1.8$ .

*The other recommender agents* The MASHA, EC-XAMAS, CBCF and pocketLens agents implement the recommender algorithms proposed in Rosaci and Sarnè (2006); De Meo et al. (2003); Melville et al. (2002); Miller et al. (2004), respectively. In particular, note that: (1) in MASHA three device agents associated with the three different device typologies considered for ARSEC have been adopted; (2) CBCF and pocketLens are not naively developed as agent systems but they can be implemented under this paradigm without any problem; (3) the CBCF content-based predictor, developed as a Bayesian text classifier, has been adapted to deal with the adopted textual classification of products categories provided by the XML Schema.

*Comparison among ARSEC and the other approaches* From a methodological viewpoint, we point out the following main differences among ARSEC and the other approaches chosen for the comparison:

PocketLens (Miller et al. 2004) exploits a variant of the item-item algorithm proposed in Karypis (2001); Sarwar et al. (2001) in which the matrix of similarity is incrementally constructed in order to limit computational complexity. Technically, for each user (1) the rated item are represented as vectors and the similarities is computed based on the *cosine similarity* (Herlocker et al. 1999; Karypis 2001) and (2) his/her matrix of similarities involves only, as rows, the items rated by him/her and, as columns, the items rated by the other users. Differently, ARSEC uses the Jaccard similarity to generate collaborative-filtering recommendations, while the content-based recommendations are computed based on the interest coefficient. MASHA Computes the recommendations on the seller side, while ARSEC generates them via the counsellor agent.

EC-XAMAS (De Meo et al. 2007) and MASHA (Rosaci and Sarnè 2006) take into account also the device in computing personalized suggestions. EC-XAMAS preserves customers' privacy by entirely computing all the needed activities for generating suggestions on the client-side, while ARSEC uses a more efficient mechanism based on off-line pre-computation.

**CBCF** (Melville et al. 2002) also uses a client-side recommendation approach, by adopting a content-based predictor based on a Bayesian classifier, able to learn a user profile and predicting unrated products. Moreover, CBCF uses a neighborhood-based algorithm (Herlocker et al. 1999) to generate collaborative-filtering recommendations, in which similarity is computed by means of a Pearson correlation. ARSEC is different from CBCF both from the viewpoint of the exploited recommendation algorithms and from the viewpoint of the exploited architecture, where the recommendations are generated by the counsellor agent.

## 5.1 Description of the experiments

In our experiments, we have monitored the customers during their e-Commerce sessions. We denote with a tuple  $\langle a, b, t, r \rangle$  the choices of the customer, that selects a link from the product  $a$  of the category  $\tau_a$ , to the product  $b$  of the category  $\tau_b$  at time  $t$ , where  $r$  represents the rate, ranging in  $[0, \dots, 5]$ , provided by the customers about his/her choice (Note that a user's rate is required only by the CBCF and pocketLens

recommenders). Initially, as described above, in order to allow the customers’ agents to build their customer profiles, for each customer 10 e-Commerce sites have been used as training-set. For the other 15 sites, 200 triplets have been collected for each customer to be exploited as test-set in order to evaluate the products suggested by the e-Commerce systems. We considered four sets  $S_1, S_2, S_3,$  and  $S_4$  containing 20, 40, 60, and 80 customers, respectively. The following experiment was repeated for each set of customers  $S_1, S_2, S_3,$  and  $S_4$ .

For each customer, in correspondence of each tupla  $\langle a, b, t, r \rangle$  belonging to the test-set, a list of recommended products  $R(a)$  have been generated for all the evaluated systems. Then it has been checked if  $b$  belongs to  $R(a)$  in order to measure the effectiveness of the different approaches and stored the result in a value  $\delta_a$ . Formally:

$$\delta_a = \begin{cases} 1, & \text{if } b \in R(a) \\ 0, & \text{otherwise} \end{cases}$$

The Average Precision ( $\bar{P}$ ) of each e-Commerce system is defined as the average of the  $\delta_a$  values on all the tuples  $\langle a, b, t, r \rangle$ .

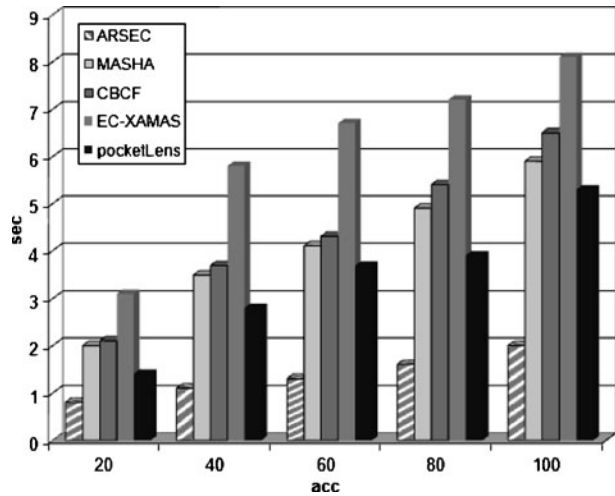
Table 3 presents the results obtained by the five approaches in this experiment in the generation of the recommendations considering, in terms of Average Precision, the global performance, the content-based and the collaborative filtering component, respectively. In this table we have denoted by A (resp., B and C) the global precision (i.e., content-based precision, collaborative filtering precision) and the respective values are shown for all the five cases and for different size of the customer agent community ( $S_1, S_2, S_3,$  and  $S_4$ ). The results obtained in this experiment clearly shown as the performances of ARSEC are better than those of the other tested recommenders (resp., MASHA, EC-XAMAS, CBCF and pocketLens). It is worth to point out that, for different sizes of the customer agent community, the collaborative filtering component improves for increasing number of customers in the community, while in the system provided with the content-based component it does not change significantly. The reason of this behaviour is that, while content-based recommendations are generated based on those products the customer already accessed in the past, apart from the other customers, the collaborative filtering recommendations are generated considering the products accessed by other customers.

Finally, we have compared the impact of the different recommendation algorithms on the performances of the e-Commerce sites. Figure 6 reports the average waiting

**Table 3** Global precision (A), Content Based precision (B) and Collaborative Filtering precision (C) performances of ARSEC, MASHA, EC-XAMAS, CBCF and pocketLens for different size communities

	$S_1$			$S_2$			$S_3$			$S_4$		
	A	B	C	A	B	C	A	B	C	A	B	C
<i>ARSEC</i>	0.90	0.68	0.47	0.95	0.70	0.54	0.96	0.71	0.67	0.97	0.72	0.71
<i>MASHA</i>	0.72	0.52	0.40	0.75	0.55	0.45	0.79	0.54	0.49	0.82	0.55	0.55
<i>CBCF</i>	0.70	0.51	0.41	0.72	0.53	0.44	0.76	0.54	0.51	0.77	0.54	0.52
<i>EC-XAMAS</i>	0.60	0.48	0.38	0.71	0.51	0.42	0.74	0.52	0.48	0.77	0.50	0.50
<i>pocketLens</i>	0.43	–	0.43	0.48	–	0.48	0.52	–	0.52	0.56	–	0.56

**Fig. 6** Waiting time of different recommender systems



time of the customers when accessing an e-Commerce site considered in the experiment above. The average value has been computed on all the e-Commerce sites and for different number of client accesses (the parameter *acc* of the Fig. 6). The experiment shows that ARSEC introduces a waiting time significantly smaller than the other test systems, and this positive gain in terms of time cost increases when the number of accesses increases too. For example, in presence of 60 client accesses, the waiting time of an ARSEC client is 1.3 s in average, while with MASHA the client waits 4.1 s, with EC-XAMAS the average waiting time is 6.7 s and with CBCF and pocketLens this time is respectively of 4.3 and 3.7 s. We argue that this good performance of ARSEC is due to the presence of the counsellor agents that pre-compute recommendations. In fact, the computational complexity of the seller agents in ARSEC is  $\mathcal{O}(p \cdot \pi)$  (where  $p$  is the number of the products and  $\pi$  is the number of agent partitions) while that of MASHA, PochetLens and CBCF quadratically depends on the number of customers present in the system. EC-XAMAS introduces even more high time costs, due to the fact that all the computation is performed on the client side. These results highlight that ARSEC allows the seller agent to be most effective than the other systems in proposing suitable suggestions for the device agent.

## 6 Conclusions

In the context of an e-Commerce support system, the importance of considering that the customer can access to the products by using different devices is assuming a key role. Recently, the MASHA framework has addressed this approach, but it presents a limitation when operating with large community of agents, as in the particular case of e-Commerce. In this paper we propose a contribution to this issue, by presenting a recommender system architecture for supporting e-Commerce, called ARSEC, designed to generate recommendations based on both customer profile and exploited device. In our framework the task of the device agent, that often has

limited resources, is not too heavy, being based on a fully decentralized approach. Furthermore, besides to compute both content-based and collaborative filtering recommendations, the system supports the customer in useful P2P interactions with other similar customers. This approach, as confirmed by experiments on a real community of customers, leads to generate very effective recommendations, taking into account also the exploited devices, and leaving to the site agent the only task of generating a graphical presentation. This also produces a significant reduction of the time cost of the customer when he/she waits for visualizing the Web pages of the e-Commerce sites. It is important to point out that the improvements introduced by ARSEC in the efficiency of the recommendation is a theoretical result, since the computational complexity of the ARSEC recommendation algorithm is  $\mathcal{O}(p \cdot \pi)$  while that of the MASHA system is  $\mathcal{O}(p \cdot c^2)$ . Our method allows the system to be scalable with respect to the number of customers present in the e-Commerce community. As highlighted above, the cost of generating recommendation for the seller agent only depends on the number of product contained in the seller catalogue and the number of available clusters. This is the most important aspect to be considered in an e-Commerce system from the viewpoint of the scalability. In fact, the customer in our system receives recommendations with the same quality of the service when the community size increases. Instead, our system does not improve the scalability with respect to the dimension of the seller catalogue. The computational cost for the seller is linear in the number of products, as in other previous system (e.g., MASHA). Instead, the better quality of the recommendations generated by ARSEC is derived by qualitative considerations and experimental evaluation that are promising results but that need to be confirmed by further analytical studies. Another important consideration is that the number of available clusters influences the performances of the system, since a too small number of clusters introduces a significant approximation in the recommendations, while a too high number reduces the efficiency of the seller agents when generating the recommendations. We do not address, in this current version of our proposal, the problem of determining a suitable trade-off between effectiveness and efficiency, computing an optimal number of clusters. This is an issue of our ongoing research.

## References

- Ackerman, M. S., Cranor, L. F., & Reagle, J. (1999). Privacy in e-Commerce: Examining user scenarios and privacy preferences. In *ACM conf. on electronic commerce* (pp. 1–8). New York: ACM.
- Adam, R. A., & Yesha, Y. (2000). *Electronic commerce: Current research issues and applications*. Berlin: Springer-Verlag.
- Adomaviciu, G., & Tuzhilin, A. (2001). Using data mining methods to build customer profiles. *Computer*, 34, 74–82.
- Anderson, C. R., Domingos, P., & Weld, D. S. (2001). Adaptive web navigation for wireless devices. In *Proc. of the 17th Int. Joint Con. on Artificial Intelligence (IJCAI 2001)* (pp. 879–884). San Francisco: Morgan Kaufmann.
- Ardissono, L., Goy, A., Petrone, G., Segnan, M., & Torasso, P. (2003). INTRIGUE: Personalized recommendation of tourist attractions for desktop and handset devices. *Applied Artificial Intelligence: Special Issue on Artificial Intelligence for Cultural Heritage and Digital Libraries*, 17(8–9), 687–714.
- Badica, C., Ganzha, M., & Paprzycki, M. (2005). Mobile agents in a multi-agent e-Commerce system. In *Proc. of the 17th int. symp. on symbolic and numeric algorithms for scientific computing* (pp. 207–215). Washington: IEEE Computer Society.

- Badica, C., Mangioni, G., & Rahimi, S. (2010). Intelligent distributed information systems. *Information Science*, 180(10), 1779–1780.
- Berkhin, P. (2006). A survey of clustering data mining techniques. In J. Kogan, C. Nicholas, & M. Teboulle (Eds.), *Grouping multidimensional data* (pp. 25–71). Berlin: Springer-Verlag.
- Bohte, S. M., Gerding, E., & La Poutré, J. A. (2004). Market-based recommendation: Agents that compete for consumer attention. *ACM Transaction on Internet Technology*, 4(4), 420–448.
- Burke, R. D. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adaptivity Interaction*, 12(4), 331–370.
- Caire, G. (2003). *LEAP 3.0: User guide, TLAB*.
- Canny, J. F. (2002). Collaborative filtering with privacy. In *Proc. of IEEE Symp. on research in security and Privacy* (pp. 45–57). Los Alamitos: IEEE Computer Society Press.
- Castro-Schez, J. J., Miguel, R., Vallejo, D., & López-López, L. M. (2011). A highly adaptive recommender system based on fuzzy logic for B2C e-Commerce portals. *Expert Systems with Applications*, 38(3), 2441–2454.
- Cheung, K. W., Kwok, J. T., Law, M. H., & Tsui, K. C. (2003). Mining customer product ratings for personalized marketing. *Decision Support Systems*, 35, 231–243.
- Cunningham, P., Bergmann, R., Schmitt, S., Traphöner, R., Breen, S., & Smyth, B. (2000). WEB-SELL: Intelligent sales assistants for the World Wide Web. In *Proc. of the Work. Programme at the 4th int. conf. on case-based reasoning* (pp. 104–109).
- De Bra, P., Aerts, A., Smits, D., & Stash, N. (2002). AHA! The next generation. In *Proc. of the 13th ACM conf. on hypertext and hypermedia, HYPERTEXT '02* (pp. 21–22). New York: ACM.
- De Meo, P., Rosaci, D., Sarnè, G. M. L., Ursino, D., & Terracina, G. (2007). EC-XAMAS: Supporting e-Commerce activities by an XML-based adaptive multi-agent system. *Applied Artificial Intelligence*, 21(6), 529–562.
- De Meo, P., Rosaci, D., Sarnè, G. M. L., Terracina, G., & Ursino, D. (2003). An XML-based adaptive multi-agent system for handling e-Commerce activities. In *Proc. of the 1st int. conf. ICWS-Europe 2003. LNCS* (Vol. 2853, pp. 152–166). Berlin: Springer-Verlag.
- Di Stefano, A., Pappalardo, G., Santoro, C., & Tramontana, E. (2002). A multi-agent reflective architecture for user assistance and its application to e-Commerce. In *Proc. of the 6th int. work. on cooperative information agents VI. LNCS* (Vol. 2446, pp. 90–103). Berlin: Springer-Verlag.
- Garruzzo, S., Modafferi, S., Rosaci, D., & Ursino, D. (2002). X-compass: An XML agent for supporting user navigation on the web. In *5th int. conf. on Flexible Query Answering Systems, FQAS '02. LNCS* (Vol. 2522, pp. 197–211). Berlin: Springer-Verlag.
- Greenstette, G. (1994). *Explorations in automatic thesaurus construction*. Hingham: Kluwer Academic Pub.
- Herlocker, J., Konstan, J., & Riedl, J. (2000). Explaining collaborative filtering recommendations. In *ACM 2000 conf. on computer supported cooperative work* (Vol. 12, pp. 241–250). New York: ACM.
- Herlocker, J. L., Konstan, J. A., Borchers, A., & Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *Proc. of the 22nd annual int. ACM SIGIR conf. on research and development in information retrieval, SIGIR '99* (pp. 230–237). New York: ACM.
- Jacobsson, M., Rost, M., & Holmquist, L. H. (2006). When media gets wise: Collaborative filtering with mobile media agents. In *Proc. of the 11th int. conf. on Intel. User Interfaces (IUI '06)* (pp. 291–293). New York: ACM.
- Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computer Survey*, 31, 264–323.
- Jogalekar, P., & Woodside, M. (2000). Evaluating the scalability of distributed systems. *IEEE Transaction on Parallel Distributed Systems*, 11(6), 589–603.
- Karypis, G. (2001). Evaluation of item-based top-N recommendation algorithms. In *Proc. of the 10th int. Conf. on Inf. and Knowledge Management, CIKM '01* (pp. 247–254). New York: ACM.
- Kim, J. K., Kim, H. K., & Cho, Y. H. (2008). A user-oriented contents recommendation system in peer-to-peer architecture. *Expert Systems with Applications*, 34(1), 300–312.
- Kim, Y. S., Yum, B. J. n., Song, J., & Kim, S. M. (2005). Development of a recommender system based on navigational and behavioral patterns of customers in e-Commerce sites. *Expert Systems with Applications*, 28, 381–393.
- Kobsa, A., Koenemann, J., & Pohl, W. (2001). Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, 16(2), 111–155.
- Lee, W. P. (2004). Towards agent-based decision making in the electronic marketplace: Interactive recommendation and automated negotiation. *Expert Systems with Applications*, 27(4), 665–679.



- Levy, A. Y., & Weld, D. S. (2000). Intelligent internet systems. *Artificial Intelligence*, 118(1–2), 1–14.
- Liu, H., & Keselj, V. (2007). Combined mining of web server logs and web contents for classifying user navigation patterns and predicting users' future requests. *Data Knowledge Engineering*, 61(2), 304–330.
- Lorenzi, F., Correal, F. A., Bazzan, A. L., Abel, M., & Ricci, F. (2008). A multiagent recommender system with task-based agent specialization. In *Proc. of int. work. on Agent Mediated Electronic Commerce (AMEC), in conjunction with AAMAS-2008*. AAMAS.
- Macskassy, S. A., Dayanik, A. A., & Hirsh, H. (2000). Information valets for intelligent information access. In *Proc. of AAAI spring symposia series on Adaptive User Interfaces, (AUI-2000)*. Menlo Park: AAAI Press.
- Manouselis, N., & Costopoulou, C. (2007). Analysis and classification of multi-criteria recommender systems. *World Wide Web*, 10(4), 415–441.
- Melville, P., Mooney, R. J., & Nagarajan, R. (2002). Content-boosted collaborative filtering for improved recommendations. In *Proc of the 18th national conf. on AI* (pp. 187–192). Menlo Park: AAAI Press.
- Miller, B. N., Konstan, J. A., & Riedl, J. (2004). PocketLens: Toward a personal recommender system. *ACM Transaction on Information Systems*, 22(3), 437–476.
- Mitchell, T. (ed.) (1997). *Machine learning*. McGraw Hill.
- Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2002). Discovery and evaluation of aggregate usage profiles for web personalization. *Data Mining Knowledge Discovery*, 6, 61–82.
- Montaner, M., Lopez, B., & de la Rosa, J. L. (2004). A taxonomy of recommender agents on the internet. *Journal on Web Semantics*, 19(4), 285–330.
- Olson, T. (2003). *Bootstrapping and decentralizing recommender systems*. Ph.D. Thesis, Dept. of Information Technology, Uppsala Univ.
- Papazoglou, M. P. (2001). Agent-oriented technology in support of e-business. *Communications of the ACM*, 44(4), 71–77.
- Parikh, N., & Sundaresan, N. (2009). Buzz-based recommender system. In *Proc. of 18th int. conf. on World Wide Web (WWW09)* (pp. 1231–1232). New York: ACM.
- Parsons, J., Ralph, P., & Gallagher, K. (2004). Using viewing time to infer user preference in recommender systems. In *AAAI workshop on semantic web personalization* (pp. 52–64). Menlo Park: AAAI Press.
- Ratnasamy, S., & McCanne, S. (1999). Scaling end-to-end multicast transports with a topologically-sensitive group formation protocol. In *Proc. of the 17th annual int. conf. on network protocols, ICNP '99* (pp. 79–88). Washington: IEEE Computer Society.
- Rosaci, D., & Sarnè, G. M. L. (2006). MASHA: A multi-agent system handling user and device adaptivity of web sites. *User Modeling User-Adaptivity Interaction*, 16(5), 435–462.
- Rosaci, D., & Sarnè, G. M. L. (2010). Efficient personalization of e-learning activities using a multi-device decentralized recommender system. *Computational Intelligence*, 26(2), 121–141.
- Rosaci, D., Sarnè, G. M. L., & Garruzzo, S. (2009). Muaddib: A distributed recommender system supporting device adaptivity. *ACM Transaccion on Information Systems*, 27(4). doi:10.1145/1629096.1629102.
- Rowstron, A. I., & Druschel, P. (2001). Pastry: Scalable, decentralized object location, and routing for large-scale Peer-to-Peer systems. In R. Guerraoui (Ed.), *Proc. of Middleware 2001, IFIP/ACM int. conf. on distributed systems platforms. LNCS* (Vol. 2218, pp. 329–350).
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2000). Analysis of recommendation algorithms for e-Commerce. In *Proc. of 2nd ACM conf. on Electronic Commerce (EC '00)* (pp. 158–167). New York: ACM.
- Sarwar, B., Karypis, G., Konstan, J. A., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proc. of the 10th int. conf. on World Wide Web, WWW '01* (pp. 285–295). New York: ACM.
- Schafer, J. B., Konstan, J. A., & Riedl, J. (2001). E-commerce recommendation applications. *Data Mining Knowledge Discovery*, 5(1–2), 115–153.
- Schifanella, R., Panisson, A., Gena, C., & Ruffo, G. (2008). MobHinter: Epidemic collaborative filtering and self-organization in mobile ad-hoc networks. In *Proc. of 2008 ACM conf. on recommender systems, RecSys 2008* (pp. 27–34). ACM.
- Shardanand, U., & Maes, P. (1995). Social information filtering: Algorithms for automating “word of mouth”. In *Proc. of the SIGCHI conf. on human factors in computing systems, CHI '95* (pp. 210–217). New York: ACM Press/Addison-Wesley Pub. Co.
- Stoica, I., Morris, R., Karger, D. R., Kaashoek, M. F., & Balakrishnan, H. (2001). Chord: A scalable Peer-to-Peer lookup service for internet applications. In *Proc. of SIGCOMM 2001* (pp. 149–160).

- Stormer, H. (2007). Improving e-Commerce recommender systems by the identification of seasonal products. In *Proc. of 22nd conf. on artificial intelligence (AAAI), work. on recommender systems* (pp. 92–99). Menlo Park: AAAI Press.
- Tanenbaum, A. S., & Van Steen, M. (2001). *Distributed systems: Principles and paradigms*. Upper Saddle River: Prentice Hall PTR.
- Tveit, A. (2001). Peer-to-Peer based recommendations for mobile commerce. In M. V. Devarakonda, A. Joshi, & M. S. Viveros (Eds.), *Proc. of 1st int.l work. on mobile commerce, 2001* (pp. 26–29). New York: ACM.
- Wang, F. H., & Shao, H. M. (2004). Effective personalized recommendation based on time-framed navigation clustering and association mining. *Expert Systems with Applications*, 27, 365–377.
- Wei, C. P., Shaw, M. J., & Easley, R. F. (2002). *e-service: New directions in theory and practice*, Chapter 9. A survey of recommendation systems in electronic commerce. Armonk: ME Sharpe.
- Wei, K., Huang, J., & Fu, S. (2007). A survey of e-Commerce recommender systems. In *Proc. of 13th int. conf. on service systems and service management* (pp. 1–5). Washington: IEEE Computer Society.
- Weng, L.-T., Xu, Y., Li, Y., & Nayak, R. (2006). A fair peer selection algorithm for an ecommerce-oriented distributed recommender system. In *Proc. of 4th conf. on advances in intelligent IT* (pp. 31–37). IOS Press.
- Weng, S. S., & Liu, M. J. (2004). Feature-based recommendations for one-to-one marketing. *Expert Systems with Applications*, 26, 493–508.
- Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3), 645–678.
- Zhang, W., Xu, B., Song, W., Yang, H., & Liu, K. (2000). Data mining algorithms for web pre-fetching. In *Proc. of the 1st int. conf. on Web Information Systems Engineering (WISE'00)*, WISE '00 (Vol. 2, pp. 2034–2044). Washington: IEEE Computer Society.
- Zhao, B. Y., Kubiawicz, J., & Joseph, A. D. (2002). Tapestry: A fault-tolerant wide-area application infrastructure. *Computer Communication Review*, 32(1), 81.
- Zhong, S. (2007). Privacy-preserving algorithms for distributed mining of frequent item sets. *Information Science*, 177(2), 490–503.