

Efficient Personalization of e-Learning Activities

Using a Multi-Device Decentralized Recommender System*

Domenico Rosaci Giuseppe M.L. Sarnè

University of Reggio Calabria

DIMET, Loc. Feo di Vito, 89122 Reggio Calabria

tel. +39 0965 875438, fax +39 0965 875238

domenico.rosaci@unirc.it, sarnè@unirc.it

Abstract

Personalization is becoming a key issue in designing effective e-learning systems and, in this context, a promising solution is represented by software agents. Usually, these systems provide the student with a *student agent* that interacts with a *site agent* associated with each e-learning site. However, in presence of a large number of students and of e-learning sites, the tasks of the agents are often onerous, even more if the student agents

*A short abridged version of this paper appeared in the Proceedings of the International Conference on Intelligent Agent Technology (IAT 2007), pp. 485–488, IEEE Press.

run on devices with limited resources. To face this problem, we propose a new multi-agent learning system, called ISABEL. Our system provides each student, that are using a specific device, with a *device agent* able to autonomously monitor the student's behaviour when accessing e-learning Web sites. Each site is associated, in its turn, with a *teacher agent*. When a student visits an e-learning site, the teacher agent collaborates with some *tutor agents* associated with the student, in order to provide him with useful recommendations. We present both theoretical and experimental results to show that this distributed approach introduces significant advantages in quality and efficiency of the recommendation activity with respect to the performances of other past recommenders.

Keywords: e-learning, Recommender Systems, Multi-Agent System, Devices

1 Introduction

Nowadays, with the progress of the Internet technologies, the number of e-learning platforms has drastically increased. In particular, an overwhelming amount of systems able to deliver educational resources to students, have been developed. However, this technological explosion has basically put effort into introducing new standards and learning mechanisms, while not as much headway has been made to tailor e-learning courses to the individual needs of learners. As widely recognized in the literature, a key challenge in designing e-learning systems (ELS) is improving adaptivity and personalization of the e-courses (Liu, Li, and Lau, 2006a; Sancho, Martínez-Ortiz, and Fernández-Manjón, 2005; Viet and

Si, 2006). This is a particular task, that does not fully cover the several issues involved in e-learning activities, since it only deals with the users' navigation between e-learning pages. However, such a task is central in the whole e-learning process, because supporting users in suitably accessing e-learning resources is the starting point to implement the other e-learning stages, as the evaluation of the resources, the interaction between students and teachers and so on. A possible solution to face such a challenge is realizing an effective knowledge sharing among the students of the system (Denman-Maier, 2004; Kurhila, Miettinen, Nokelainen, and Tirri, 2007). In particular a student that needs to select the more suitable educational resources to compose his learning course should be able to exploit the opinions of other students of the system. In the e-learning context, an educational resource (a lesson, a book, a tutorial, etc.) is called *learning object* (LO). More in detail, the IEEE Learning Technology Standards Committee (IEEE LTSC, 2005) states that a learning object is “any entity, digital or non-digital, which can be used, re-used or referenced during technology-supported learning”. A relevant amount of new distributed and adaptive ELSs have been proposed in the last years (Conlan, Wade, Gargan, Hockemeyer, and Albert, 2002; Karagiannidis, Sampson, and Cardinali, 2005; Weber, Kuhl, and Weibelzahl, 2002) to support students in their e-learning sessions. A strategy that has been strongly exploited is allowing the ELS to automatically extract useful suggestions, as the most promising LOs to access in a learning session, monitoring the students' behaviour when accessing e-learning sites. Often ELSs act as recommender systems (Soonthornphisaj, Rojsattarat, and Yim-Ngam,

2006; Tang and McCalla, 2004) that generate some recommendations which could be: (i) *Content-based*, recommending to a student the LOs which appear the most similar to those he already accessed in the past; (ii) *Collaborative Filtering*, searching similarities among students and consequently suggesting to a student some LOs also considered by similar students in the past; (iii) *Hybrid*, using both content-based and collaborative filtering techniques to generate recommendations. Generally, these systems use a *profile* of the student, which is a model representing his interests and preferences (Esposito, Licchelli, and Semeraro, 2004; Horváth and Rudas, 2006), and many recommender systems propose the use of *software agents* in order to construct such a student profile (Liu, Wang, and Fang, 2006b; Silveira and Vicari, 2002). More in particular, each student is associated with a software agent which monitors his Web activities. When the student accesses an e-learning site, his agent exploits the student's profile interacting with the site. In this interaction, the site can use both content-based and collaborative filtering techniques to provide recommendations to the student's agent by adapting the site presentation. Traditional e-learning systems generally act as content based recommender systems, and they are realized by means of a client-server architecture. Such an approach does not allow the students to share the study experience with others. A more effective approach should be also collaborative filtering, often implemented by adopting a decentralized architecture of the e-learning system and performing P2P interactions among the agents.

Furthermore, in such a scenario an emerging issue is that nowadays a stu-

dent can 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. As a consequence, student's preferences might be influenced by these differences; for example, when he accesses a site with a cellular phone, he could desire to exploit a light site presentation. Consequently, we argue that, for each student, a different profile for each used device should be built. Moreover, since the student's interests change with the exploited device also the recommender system should be adaptive with respect to the device (Dolog, Henze, Nejdl, and Sintek, 2004; Nejdl, Wolf, Qu, Decker, Sintek, Naeve, Nilsson, Palmer, and Risch, 2002).

To tackle this important issue, we have recently proposed (Rosaci and Sarné, 2006) a Multi Agent-based framework for developing recommender systems, called MASHA. MASHA 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. Moreover, each user is also associated with a *server agent* in order to build, manage and update his complete profile based on the local profiles periodically provided by the client agents associated with his exploited devices. 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 both an analysis of the user profile and the suggestions coming from other similar users that exploit the same type of device. However, although MASHA effectively handles the

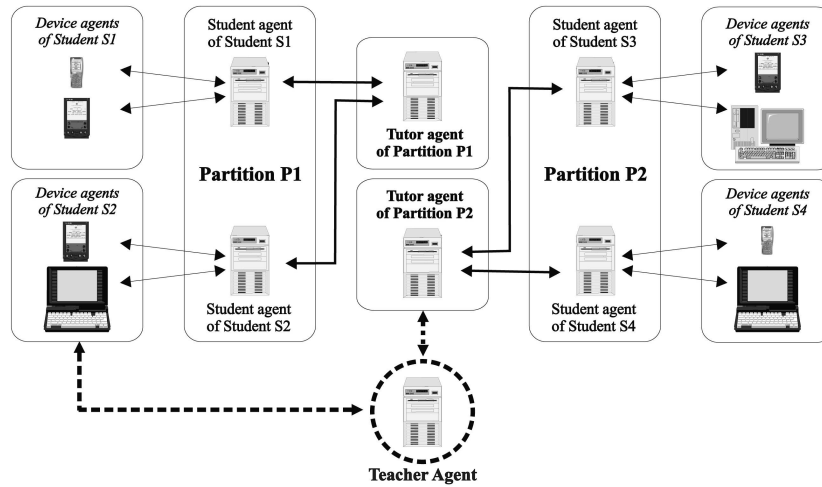


Figure 1: The ISABEL Architecture

problem of taking into account the different devices to generate effective recommendations, it presents a significant computational cost of the adapter agent activities, due to the execution of the recommendation algorithm. In fact, if we apply MASHA to an e-learning scenario, if s is the number of students that visit a given e-learning site and l is the number of learning objects which are present in the site, then the computational complexity of the MASHA technique is $\mathcal{O}(l \cdot s^2)$ in the worst case, since it compares the profile of each student with those of the others, considering up to l concepts for each student.

In order to apply the MASHA framework to the e-learning context reducing the recommendation costs, we propose in this paper a new multi-agent architecture, called *Information Software Agent-Based e-learning* (ISABEL), that is an evolution of the MASHA architecture, conceived to support e-learning activities. The ISABEL architecture (see Figure 1) maintains the three MASHA agent ty-

pologies, namely: (i) a *device agent*, associated with each device, (ii) a *student agent*, associated with each student (analogous to the *server agent* in MASHA architecture), and (iii) a *teacher agent*, associated with each e-learning Web site (analogous to the *adapter agent* in MASHA).

1.1 Differences with MASHA

Differently from MASHA, the recommendations provided by ISABEL are not autonomously generated by the teacher agent, but they are the result of a collaboration between the teacher agent and a new agent type, called *tutor agent*. The basic idea underlying ISABEL is that of determining groups of students that have similar profiles, where each group is managed by a tutor agent. Consequently, when a student accesses an e-learning site, the teacher agent of the site does not perform the onerous task of computing recommendations, but it exploits the help of the tutor agents that are associated with the groups which the student belongs to. A relevant advantage of this approach, is that a teacher agent that has to generate at the same time recommendations for s students, delegates the task of computing both content-based and collaborative filtering suggestions to the tutor agents of the students. This way, the computational cost of the teacher agent is $\mathcal{O}(l \cdot \pi)$ (where π is the number of different groups) that results significantly lower than MASHA. Some experiments we have performed show that even with a relatively small number of students and using very common computational resources for e-learning sites the advantage of ISABEL with respect of other systems is significant. Obviously, if the resources of the

servers are very high, the differences among the systems become more relevant, but the better scalability of ISABEL remains always a useful characteristic. This assumes a key role when the size of the student community increases (and in the future we expect to have e-learning communities with a very large number of students) and when the profiles of the students become heavy. In fact, it is important to remark that for each student it is necessary to compare his profile with those of the other students, and a single profile can be composed of a number (possibly large) of different features, making the comparison a not trivial task from a computational viewpoint. Besides this advantage in terms of efficiency, ISABEL also introduces an improvement of the performances in terms of effectiveness with respect to MASHA. Indeed, the collaborative filtering component of the recommendations for a student in MASHA is generated directly by the server agent of the site, taking into account the suggestions of only those students that visited just that site in the past. Instead, the collaborative filtering recommendations in ISABEL are generated by the tutor agent of the student, that is capable to consider the suggestions coming from all the students monitored by it, not only the students that visited a specific site. In other words, the introduction of the tutor agent introduces the possibility to generate collaborative filtering recommendations taking into account a large number of similar users (those that belong to the same partition), that have acquired in the past their experience over different sites (*inter-site* recommendations), while MASHA recommendations are generated only *intra-site*.

Furthermore ISABEL provides each student with a list of similar students

that can be contacted in a P2P interaction. Also this feature is particularly useful in an e-learning scenario, since allows a student to discuss the obtained suggestions with the other students. We have experimentally evaluated ISABEL by comparing it with other recent profile-based recommender system approaches, and we have observed a significative improvements of the recommendation performances and a low time cost for generating recommendations.

1.2 Practical significance and potential applicability of ISABEL

The discussion presented above leads us to conclude that the recommender system architecture proposed in this paper introduces a new mechanism to support e-learning activities, particularly useful (*i*) in those e-learning contexts where the number of students is large, (*ii*) when the information to manage for each student has a complex structure and (*iii*) when multiple devices are used by the students to access to the educational resources. These conditions are produced in a number of realistic scenarios, such as in e-learning platforms for universities, e-learning portals for Web communities, digital libraries that need to delivery a large number of e-books to many users and also those platforms that integrate multiple sub-platforms. In these contexts, our proposal of using a pre-computation of the recommendations and maintaining a separate profile of the user for each exploited device can generate more effective and efficient results in personalizing the presentation of e-learning resources. On the other hand, it is necessary to point out that our system is not particularly suitable in case

of small e-learning communities, or in presence of a unique modality of access, since in these situations the complexity of our architecture is not justified.

The plan of the paper is as follows. In Section 2 we provide an overview of the ISABEL 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 ISABEL Architecture

This section is devoted to a general overview of the ISABEL platform which supports, on one hand, the student in his learning tasks by generating personalized suggestions and, on the other hand, the e-learning site by selecting those documents potentially interesting for the visiting student. To this purpose, ISABEL exploits a student profile, which represents the categories of interest of the student, giving to each category a measure of the interest. Indeed, in the ISABEL framework, each learning object of an e-learning site belongs to a given object category, that we call *topic*. A topic is a string identifier that represents a category of interest, e.g. *computer* or *sports*. In order to make homogeneous the identification of the topics of the student with the description of the content of the e-learning sites, all the possible topics are contained in a common *dictionary of the topics*, which is shared by all the users of the system. Moreover, we assume that each e-learning site of the ISABEL platform contains some learning objects (LOs) as documents, videos and so on, that can be described by using

the topics of the common dictionary. For instance, if the e-learning site contains a given LO (e.g. a document related to the topic “Computational Complexity”), it can be considered as an *instance* of the topic which the LO belongs to. For each topic accessed by the student, the profile stores a value that represents the time spent on the instances of that topic. This time value is considered as a rough measure of the student’s interest about the topic and it is strictly related to the characteristics of the exploited device.

In order to choose a suitable indicator to represent the user’s interest in a category, we have considered several approaches proposed in the literature (Badi R., Bae S., Moore J.M. Meintanis K., Zacchi A., Hsieh H., Shipman F. and Marshall C.C., 2006; Al Halabi W.S., Kubat M. and Tapia M., 2007; Kim H.-R. and Chan P.K., 2008; Kelly D. and Belkin N.J., 2004). In (Badi R., Bae S., Moore J.M. Meintanis K., Zacchi A., Hsieh H., Shipman F. and Marshall C.C., 2006), the reading time of a document is recognized as a good indicator of the user’s interest, together with other secondary information depending on the interaction with the user, as the number of performed clicks and the consideration of only the “no-idle” component of the time, i.e. of that part of the time spent on the page during which the user moves the mouse. Also in (Al Halabi W.S., Kubat M. and Tapia M., 2007), authors argue that the time spent on a Web page is sufficient to infer the user’s interest. The mouse activity is instead considered in (Goecks J. and Shavlik J., 2000) and the display time is used in (Kelly D. and Belkin N.J., 2004) as implicit feedback to evaluate the user’s interest. All these approaches agree with the particular importance of considering the read-

ing time, intended as “no-idle” time, for computing the user’s interest, therefore we have decided to adopt this solution also in our approach, being conscious of the underlying limitations and considering it as a “sufficient” measure of the user’s interest. Furthermore, our approach deals with the particular case of exploiting several devices for navigating, and therefore the “no-idle” time spent of a page is relevant for comparing the accesses to a same page with different devices.

ISABEL uses four types of agents, described in detail below and depicted in Figure 1. First of all, each student’s device is associated with a *device agent* which monitors the student and builds his profile related to just that device. Moreover, in order to collect all the information retrieved by the different device agents of a student, ISABEL associates with each student a *student agent*, running on a server machine, that constructs a complete profile of student’s interests and preferences. Student agents associated with different students are then grouped in partitions, each of them characterized by a specific *domain of interest* (e.g. physic, chemistry, computer science, etc.). Each student agent can belong to different partitions if its associated student is interested in different domains.

The main component of the ISABEL architecture is represented by the set of the *tutor agents*. A tutor agent is associated with each partition and runs on a server machine and, for each e-learning site of the ISABEL community, contains a complete list of the topics of the e-learning site and, for each student of the associated partition, his complete profile and a list of the topics of the e-learning

site accessed by that student. These information are provided by a *teacher agent* associated with each e-learning site and by a *student agent* associated with each student. Figure 2 graphically shows how ISABEL works. When the student s accesses to an e-learning site E , the device agent of s interacts with the teacher agent of E and sends to it some information about the preferences of s . These preferences, contained in the *device profile* DP , are related to the format desired by the student s for accessing the LOs when he exploits that device. We suppose that the student s belongs to p partitions, being interested in p different domains. Then, the teacher agent contacts the tutor agents t_1, t_2, \dots, t_p of the p partitions which the student s belongs to, and transmits to it the device profile DP . To support content-based recommendations, each tutor agent pre-computed the LOs of the e-learning site that best match with the *student profile* of s , and that are compatible with the DP of the device exploited by s . Moreover, the tutor agent, that is able of determining similarities between the students of its partition, also pre-computed the topic instances accessed by other students which are similar to s and which exploit the same device of s . Then, in order to support collaborative filtering recommendations, each tutor agent selects those LOs that match the student's preferences contained in the device profile DP . Finally, the so computed topic instances are inserted into a list L and transmitted to the teacher agent of the site E that generates recommendations for the student s with a suitable site presentation.

In this paper, we assume that the dictionary of the topics exploited by the agents on the platform is realized by an XML-Schema document, where each

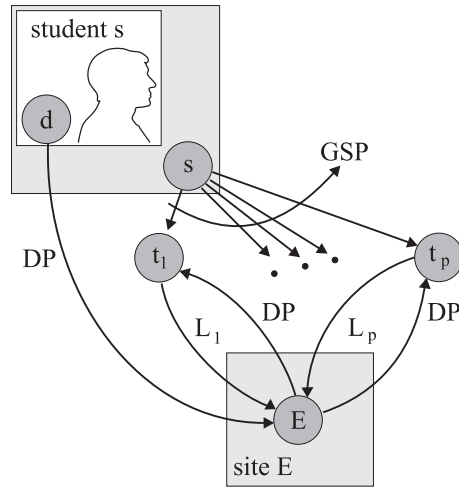


Figure 2: The behaviour of ISABEL

element represents a *topic*. We suppose that all e-learning sites are XML sites that contains LOs of topics that belong to the dictionary. We also suppose that an LO of an e-learning site can be associated with one or more *hyperlinks* to other LOs, contained in the same site. A hyperlink in ISABEL is represented by a pair (a, b) , where a and b are instances of topics (i.e. LOs) and a hyperlink (a, b) can be clicked by a student for accessing b coming from a .

The assumption to work only with XML files, embedding in the structure of the XML file the annotation related to a topic, appears reasonable in the limited context of an e-learning community, where we suppose that each e-learning site can be realized in XML, possibly using some ad-hoc design tool.

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

2.1 An agent associated with both a student and a device:

The Device Agent

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

2.1.1 Device Data Structure

The device agent contains two data structures, namely the *Device Profile (DP)* and the *Student Profile (SP)*. In its turn, *DP* contains the following parameters, that describes the preferences of the student when he uses that device:

- The set of the tutor agents associated with the partitions which the student belongs to;
- three parameters s_1 , s_2 and s_3 that represent the maximum sizes (in Kbyte) of text, audio and video contents, respectively, of a LO that the student desires to handle when using the device;
- three parameters $\rho_1, \rho_2, \rho_3 \in [0, 1]$, associated with the actions performable by the student (i.e., reading, storing or printing a LO content);
- a parameter T , an integer coefficient used to evaluate the student's interest in a topic instance, that represents a time-threshold;
- a parameter ω that represents the number of days between two consecutive student's actions after which the interest for a topic not accessed decreases;

- a parameter ψ , which belongs to the interval $[0, 1]$, used to decrease each ω days the student's interests related to the associated topics that are no longer accessed;
- three parameters k , z and r , that are exploited by the device agent in its interaction with the teacher agent of each visited e-learning site (see Section 2.3). In particular, k , z and r respectively represent the maximum number of: (i) interesting topics belonging to the e-learning site that the student desires to be considered in the e-learning session; (ii) similar agents that the student desires to be considered in collaborative filtering recommendations; (iii) recommendations to be considered for each similar agent.

Note that all the above parameters can be changed in any moment by the user, giving the possibility to dynamically modify his preferences.

The student profile SP stores the profile of the student, based on the hyperlinks clicked by the student exploiting the device. More in detail, SP is a set of tuples $\langle \tau, IW, LU \rangle$, each one associated with a topic τ which belongs to the common dictionary, where IW (*Interest Weight*) is a measure of the student's interest in the concept τ by using the device and LU (*Last Update*) is the date of the last IW update.

Analogously to the approaches (Garruzzo, Modafferi, Rosaci, and Ursino, 2002; Parsons, Ralph, and Gallagher, 2004), in order to obtain a measure, that belongs to the interval $[0, 1]$ and that reaches the maximum value when $t = T$, we define IW by using the actual time t spent by the student when visiting the

page containing τ . Moreover, the student can store, print or simply read the Web page that contains τ , and this is taken into account by weighting IW with a coefficient ρ_a for each action a (where $a = 1, 2, 3$). More formally, for each new update, IW is computed as follows:

$$IW = \psi \cdot \frac{IW + \frac{t}{T} \cdot \rho_a}{2} \quad (1)$$

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 $t \geq T$. Besides, the parameter ψ is periodically used to decrease the IW value of the unvisited topics, based on the temporal distance from the last update LU . More in particular, when this temporal distance is a multiple of the parameter ω , the current value of IW is multiplied by ψ .

2.1.2 Device Agent Behaviour

The device agent constructs the student's profile SP by monitoring the student's e-learning sessions and considering the topics visited by the student. The device agent (together with the other student's device agents) periodically sends SP to its student agent in order to build a complete student profile. Moreover, when the student visits an e-learning site, the device agent sends to the teacher agent the parameters related to the exploited device to generate a personalized e-learning session for the student. 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 student:

The Student Agent

ISABEL associates with each student a student agent that collects by each device agent of the student the information about the topics visited during the student's e-learning activities. These information are sent to the tutor agents of the student's partitions. This is an important feature of ISABEL, since the device agents live on the associated devices and could have limited computation and storage capability. The contribution of the student agent, which runs on a powerful equipped machine, is fundamental to provide the student with an off-line collector of all the information obtained by the different device agents that monitored the student's sessions. Below, both the data structure and the behaviour of the student agent are described.

2.2.1 Profile Data Structure

The data structure of the student agent contains two elements, namely the *Learning Setting (LS)* and the *Global Student Profile (GSP)*. In its turn, *LS* stores the following parameters:

- *ND*: it is the number of device agents associated with the student;
- *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.

The *Global Student Profile (GSP)* stores a global representation of the student's interests related to the visited topics. In particular, *GSP* is a list of

tuples $\langle \tau, GIW \rangle$, where τ identifies a topic accessed by the student and GIW is its *Global Interest Weight* shown by the student, 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 concept τ by the i -th device, $i = 1, \dots, ND$, and c_i is the connection cost of the i -th device.

2.2.2 Student Agent's Behaviour

The behaviour of the student agent simply consists in updating the global student profile GSP by exploiting the data that each device agent of the student periodically sends to the student agent.

2.3 Two agents for generating recommendations: The Tutor Agent and the Teacher Agent

In ISABEL the students are partitioned in clusters of students that are interested in the same topic. A *tutor agent* is associated with each cluster in order to manage it, while a *teacher agent* is associated with each e-learning site in order to manage the LOs contained in the site. Below, the data structure and the behaviour of both tutor and teacher agents, that interact each other, will be briefly described. We omit to describe the structure of the teacher agent since it contains only the catalogue of its learning objects.

2.3.1 Tutor Data Structure

The data structure of the tutor agent is composed of three elements called *Teacher Catalogue (TC)*, *Global Profile Set (GPS)* and *Profile Collector (PC)*. The teacher catalogue contains, for each e-learning site E that interacted with the tutor agent in the past, all the learning objects present in E . The global profile set GPS contains the global profiles of all the students associated with the tutor agent. The *Profile Collector (PC)* contains several data sections, each one related to a site E of the ISABEL 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 $DS_E[s, d]$ each of these profiles, associated with a given student s and his device d . The elements of $DS_E[s, d]$ are obviously pairs (τ, IW) where τ is a topic, that s considers interesting in the site E and IW is the interest weight of τ . The information related to each visitor profile $DS_E[s, d]$ is provided to the tutor agent by the site agent of E when the student s terminates its session.

2.3.2 Tutor and Teacher Agent Behaviour

Suppose that a student s visits the site E exploiting a given device d ; then, the device agent of s sends to the teacher agent the device profile DP . The student s belongs to some student partitions, each of which is associated with a tutor agent. In this case, the teacher agent contacts each tutor agent, that has pre-computed personalized recommendations for the student s , and sends to the tutor agent the device profile DP of the device d . In order to generate

content-based recommendations, the tutor agent has built a list CB that contains those topic instances of the site E whose topics belong to the global profile of the student s (this global profile is contained in the Global Profile Set of the tutor agent). Then, the tutor agent orders the list CB in a decreasing fashion based on the coefficient IW of each topic and maintains only the first k topic instances, 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 tutor agent compares the profile $DS_E[s, d]$ contained in the data section DS_E and related to the student s , with each profile $DS_E[q, d]$ of each other student q , that has visited E in the past and that has exploited the same device d of the student s . As a result, a list CF of the topics accessed by the z visitors less different from s is obtained (remember that also z is a parameter contained in DP).

The difference between the user s and another user q considered in DS_E and that uses the same device d is computed as follows. Let τ be a topic that belongs both to the data section $DS_E[s, d]$ of s and the data section $DS_E[q, d]$ of q , and let $IW_s(\tau)$ be the interest rate assigned to the topic in the profile of s and $IW_q(\tau)$ be the corresponding interest weight in the profile of q .

The value $d(\tau) = |IW_s(\tau) - IW_q(\tau)|$ is assumed to be a reasonable measure of the difference between the two student s and q in the evaluation of the topic τ .

We measure the global difference between the two students s and q , denoted by $D(s, q, d)$ by summing all the contributions $d(\tau)$ related to all the topics τ

that the profile of s and q share. More formally:

$$D(s, q, d) = \sum_{\substack{\tau \in \\ DS_E[s, d] \cap DS_E[q, d]}} |IW_s(\tau) - IW_q(\tau)| \quad (3)$$

We remark that our measure of difference aims at representing only the interest-based component of the difference between two students. In other words, since we want to provide a student s with suggestions coming from other students, we try to identify those students showing similar interests to s , i.e. those students that rate in a similar way the learning objects accessed by s . It is worth to point out that, in the general case of determining the similarity between two students, our similarity measure would not be sufficient. As an example, if the similarity were computed for grouping students in coalitions, for collaborative purposes, the importance of considering similar users' goals instead of similar interests becomes fundamental.

3 Presentation Adaptivity

Each tutor agent of the student s that is visiting an e-learning site returns to the teacher agent of the site the lists CB and CF , which contains learning objects suitable to be recommended to the student s . Besides these lists, the tutor agent returns to the teacher agent also the *similar student list* which contains the z students most similar to the student s . These lists are used by the teacher agent for generating an adapted presentation for the visiting student. In particular,

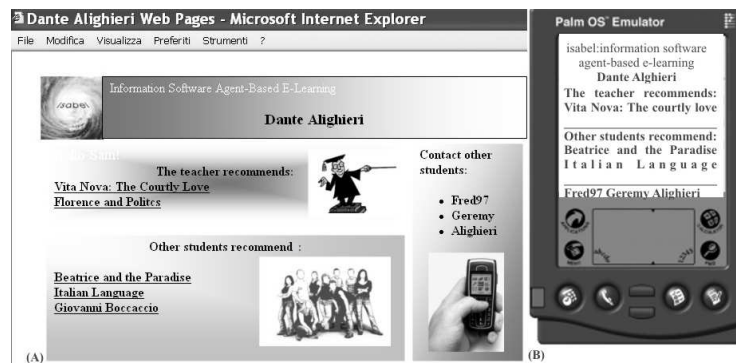


Figure 3: The ISABEL presentation (A) on a desktop PC; (B) on a palmtop the teacher agent generates a Web page that contains only elements that are compatible with the specification of the student’s device, contained in the device profile DP (see the parameter $s1$, $s2$, $s3$ described in Section 2.1.1).

The GUI used for ISABEL is totally similar to that of MASHA, described in Rosaci and Sarné (2006), being the difference between the two systems only in the underlying architectures. The aspect of the GUI is shown in Figure 3. As we can see, the GUI contains two sections of recommendations, namely *The teacher recommends* and *The other students recommend*, that have the function of showing the learning objects contained in the lists CB and CF , respectively. A third Section, called *Contact other students*, gives the possibility to send a message to the students that have been considered when generating the CF lists. Figure 3 shows an example of two different presentations of the same e-learning site for two different devices of the same student, a desktop PC (Figure 3-A) and a palmtop (Figure 3-B), respectively.

We remark the differences in the graphical aspect of the presentation: that

of the palmtop does not contain any figure since the student has set a parameter $s3$ (maximum size of graphic object) to a value smaller than the size of the available figures, which are displayed on the desktop PC. Moreover, there are also differences in the generated suggestions. In details, k is set to 2 (resp. 1) for the desktop PC (resp. palmtop). Consequently, the content based recommendations in *The teacher 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 students are considered in the *Contact other students* 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 students recommend* section. Figure 4 shows how the system provides for each student considered in *The other students recommend* section the list of the suggested LOs. The user can contact one of these students for discussing, in a P2P interaction, of a given LO.

4 Related Work

The various aspects related to e-learning systems have been dealt within a very large variety of scientific works. Therefore an overall contextualization of this paper within these backgrounds would require too much space and could go beyond our scopes. For such a reason, in this section, we prefer to mention only

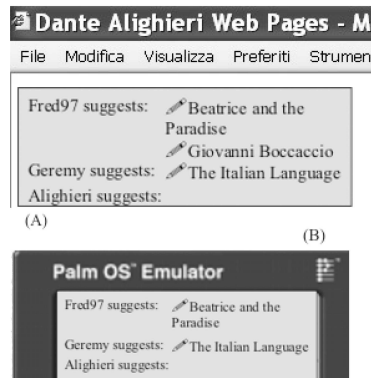


Figure 4: The other students recommend: (A) on the desktop PC; (B) on the palmtop

those works that, to the best of our knowledge, contribute to better define the context of interest of ISABEL. However, the interested reader could refer to a considerable number of surveys that in these later years have investigated the state of the art of the e-learning world (Anderson and Whitelock, 2004; Burgos, 2007; Friesen, 2005; Hauger and Köck, 2007; Uskov and Uskov, 2008).

Most of the recommender systems for e-learning proposed in the past (Brusilovsky, 2004; Dolog, Henze, Nejd, and Sintek, 2004; Rafaeli, Barak, Dan-Gur, and Toch, 2004) try to support the student by suggesting him the most useful educational resources. All the systems store in an internal profile a description of the student's interests and preferences, likewise ISABEL.

For example, ELENA (Dolog, Henze, Nejd, and Sintek, 2004) is an ELS able to provide a personalized support for learners based on distributed services, without the need of centralized control, and considering also the different device preferences or characteristics. The central component in this ELS is the Personal

Learning Assistant which uses and integrates the other framework services to find LOs that are suitable for its learner, exploiting a learner's profile. As result, a learner receives recommendation accordingly with both his profile and the LOs targeted by other learners that belong to the same group and share the same interests. The learner could exploit such recommendation by means of a traffic light metaphor that he will provide to highlight the recommended LOs.

As another example, QSIA (Rafaeli, Barak, Dan-Gur, and Toch, 2004) provides recommendations founded on a statistical matching approach adding a social dimension. It is realized by means of the ranks about a LO provided by the other users of the e-learning communities for sharing their acquired knowledge. The system matches, based on statistical similarities or user choices, these feedbacks with the required recommendations to search for those LOs that are relevant, reliable and authoritative.

Another recommender system for e-learning is KnowledgeTree (Brusilovsky, 2004), which is a distributed architecture for adaptive ELSs based on the re-use of LOs. It assumes the presence of four kinds of servers, where the most important is doubtless the *portal server*, that is able to manage a complete course providing learning content and student support. In particular, the support provided by the portal server is based on information retrieval technologies and self-organized neural networks. The LOs available in the systems are organized in knowledge maps constituted by matrix of 8x8 cells that links LOs semantically similar. On the basis of the student characteristics, the system adapts the provided support examining resources classified in the same cell or in the

cells around. Furthermore, the system attracts student's attention to cells and resources that result the most often visited by the student or by a group of students with similar goals and knowledge.

However, among these systems only ISABEL and ELENA consider the different devices exploited by the student. In particular, ISABEL considers and integrates "dynamically" in the global student's profile the information provided by all the different device agents. Differently, ELENA considers the various preferences of the student related to a specific device separately for each device without integrating such information, in a manner that can be considered "static" with respect to ISABEL. Since student profiles are then exploited in the recommendation algorithm, we easily argue that the recommendations provided by ISABEL, based on a global user profile dynamically updated, will probably result more precise than those of the other systems. Furthermore ISABEL is also the unique ELS which builds and updates the student profile in a fully automated manner, while the other systems explicitly require the human help. Finally, ISABEL pre-computes the recommendations, thus reducing the time cost of the student that is waiting for the recommendations. These considerations are simply qualitative comparisons: A quantitative evaluation of the advantages introduced by ISABEL with respect to the other aforementioned systems is described in the next section.

5 Experiments

In this section, we present some experiments devoted to evaluate the capability of ISABEL to adequately support a student suggesting those resources considered the most useful for him.

Firstly, we remark that our approach does not deal with the task of evaluating the student's knowledge. In other words, the student is personally responsible to evaluate the content of a learning object, and to decide if this content is relevant or not. Obviously, it is possible that the student uses an evaluation tool, or that he can interact with some teacher able to provide him with an evaluation. However, our system does not cover this (very important) issue, but only provides recommendations about those learning objects that appear the most attractive for the students based on their interests and preferences, leaving to other engines the task of evaluating the relevance of the acquired resources.

In order to perform these experiments, we have used a preliminary implementation of ISABEL, that is a research prototype whose architecture we have preliminarily presented in (Garruzzo, Rosaci, and Sarné, 2007), and that it is currently under development.

We have compared the performances of ISABEL with both MASHA (the system from which ISABEL is derived) and three traditional approaches described in Section 4. We have used in our experiment 21 e-learning Web sites related to object oriented programming and we have provided each site with about 200 LOs. In the experiments we have monitored 78 students in their e-

learning sessions during 45 days and in particular we have used 9 of the 21 sites for building the students' profiles in all the considered systems. The remaining 12 sites have been used to test the systems.

All the e-learning sites have been realized in XML, and we have used a common dictionary implemented by a unique XML Schema to represent the different topics. Therefore each site contains only instances of this XML schema. We have recorded, for each student, the student's choices into a log file, that contains a list of 300 elements $\langle a, b, t \rangle$, related to 300 different LOs accessed by the student, where a (resp. b) is the identifier of the *source* (resp. *destination*) LO, and t is the timestamp associated with the choice to cross from a to b via a hyperlink. We have realized all the proposed systems by using JADE (Java Agent Development Framework) ¹ and we have used JADE/LEAP for those devices, as palmtops and cellular phones, with limited resources (Caire, 2008). In particular ISABEL has been realized using four agent types (namely device, student, teacher and tutor agent) that implement our algorithm for generating user suggestions. The other systems are built by following the approach MASHA, and three other traditional approaches called $S1$, $S2$ and $S3$, based on the descriptions provided in (Dolog, Henze, Nejdil, and Sintek, 2004; Rafaeli, Barak, Dan-Gur, and Toch, 2004; Brusilovsky, 2004), respectively.

ISABEL Device Agents We have considered three device agents associated with three different devices, namely a desktop PC, a palmtop and a cellular phone. We have carried out some tests in order to suitably set their parameters

¹<http://jade.tilab.com>

Table 1: Setting parameters of the ISABEL device agents

<i>device</i>	T	ρ_1	ρ_2	ρ_3	ω	ψ	k
<i>PC</i>	300	0.6	0.8	0.9	3	0.90	3
<i>palmtop</i>	150	0.6	0.9	1.0	3	0.95	3
<i>cellular</i>	60	0.5	0.9	1.0	3	0.95	3

(described in Section 2.1) as shown in Table 1. However, we remember that the interest for a topic has been assumed as “saturated” if the LO of that topic is used for more than T seconds. The coefficient ρ_1 (resp. ρ_2, ρ_3) weights the user’s interest in a topic in the case the user simply visits (resp. stores, prints) a LO that is an instance of that topic. Moreover, the attenuation period ω is equal to 3 for each device agent; this means that the interest in a topic that has not been 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 instances of the three most interesting topics.

The other approaches The MASHA recommendation system has been built following the description presented in (Rosaci and Sarné, 2006). The approach $S1$ is based on the recommendation service described in (Dolog, Henze, Nejdl, and Sintek, 2004). For both the system MASHA and $S1$, we have adopted three device agents associated with the same three device typologies considered for ISABEL. The approach $S2$ and $S3$ are based on the recommendation techniques presented in (Rafaeli, Barak, Dan-Gur, and Toch, 2004) (a statistical

matching approach) and (Brusilovsky, 2004) (knowledge maps constitutes by matrix of cells), respectively. We note that the approaches $S1$, $S2$ and $S3$ do not implement the overall architectures proposed in (Dolog, Henze, Nejd, and Sintek, 2004; Rafaeli, Barak, Dan-Gur, and Toch, 2004; Brusilovsky, 2004), but only the recommendation techniques, opportunely adapted to our comparison purposes.

ISABEL Student Agents Each student is associated with a student agent. All the student agents adopt the same parameters values: (*i*) $n = 3$, having only three types of device agents for each user; (*ii*) the prices per Mbyte (in euro cents) that we have considered are: $c_1 = 0.9, c_2 = 1.4, c_3 = 1.8$.

5.1 Description of the experiments

We have performed three different experiments. The first experiment, described in subsection 5.2, aims at evaluating the effectiveness of ISABEL in comparison with the other e-learning systems. The effectiveness is the capability of the system of giving recommendations considered attractive for the student. The second experiment, described in subsection 5.3, measures the capability of each e-learning system of predicting with accuracy the choices of the student. Finally, the third experiment described in subsection 5.4 evaluates the efficiency of the different e-learning systems.

5.2 Effectiveness

In our experiments we have monitored the students during their e-learning sessions. We denote with a triplet (a, b, t) the choices of the student, that selects a link from the instance a of the topic τ_a , to the instance b of the topic τ_b at time t . Initially, as described above, in order to allow the students' agents to build their student profiles, for each student we have used 9 e-learning sites as training-set. For the other 12 sites we have collected 300 triplets for each student to be exploited as test-set in order to evaluate the LOs suggested by the e-learning systems. More in particular, for each student, in correspondence of each triplet (a, b, t) belonging to the test-set, we have generated a list of recommended topic instances $R(a)$, for each of the five systems ISABEL, MASHA, $S1$, $S2$ and $S3$. We have checked if b belongs to $R(a)$ in order to measure the effectiveness of the different approaches and we have stored the result in a value $\delta(a)$. Formally:

$$\delta_a = \begin{cases} 1 & , \text{ if } b \in R(a) \\ 0 & , \text{ otherwise} \end{cases} \quad (4)$$

The Average Effectiveness (\bar{E}) of each e-learning system is defined as the average of the $\delta(a)$ values on all the triplets (a, b, t) .

The first three rows of Table 2 presents the results obtained by the five approaches in this experiment in the generation of the recommendations considering, in terms of Average Effectiveness. In particular, the first row of the table reports the global value of the average effectiveness (i.e., considering both

Table 2: Performances of different e-learning systems.

	<i>ISABEL</i>	<i>MASHA</i>	<i>S1</i>	<i>S2</i>	<i>S3</i>
<i>Global \bar{E}</i>	0.89(0.66)	0.76(0.75)	0.74(0.66)	0.50	0.58
<i>CB \bar{E}</i>	0.71(0.57)	0.68(0.64)	0.62(0.54)	0.42	0.49
<i>CF \bar{E}</i>	0.69(0.59)	0.58(0.52)	0.54(0.48)	0.58	0.39
<i>Feedback</i>	3.93	2.99	2.88	2.71	2.43

the content-based and the collaborative filtering recommendations), while the second and the third rows reports separately the content-based and the collaborative filtering component of the average effectiveness, respectively. We can note that, in all the three cases (global precision, content-based precision, and collaborative filtering precision), ISABEL performs better than the other approaches, showing a precision that is significantly better than the best competitor systems. We argue that this very good performance that ISABEL obtains in recommending LOs is due to the fact that our system considers, when determining its suggestions, also the devices exploited by the student. To confirm such a supposition, we have repeated the above comparisons, by using the only desktop PC device agents (already used in the previous experiment), instead of three different clients. This way, the effects of the different devices, exploited by the student in the past, does not influence the recommendations of the systems. Results of this experiment are shown in round parenthesis in Table 2. In this condition, ISABEL shows performances comparable with, but no higher than

those of the other approaches. This confirms that the advantage shown in the previous experiment (represented by the main numbers in Table 2) is due to the capability of ISABEL to manage a situation with different devices, that is not adequately faced by the other systems. In e-learning also subjective measures are significative. In this sense we have investigated how the student perceives the support provided by the ELSs. To evaluate this perception, we have asked to our students to rank the suggestions provided by the five systems in a blind manner. Each student has ranked each suggestion with a rate from 1 to 5. The average rate, computed on all the suggestions and reported in the last row of table 2, qualitatively confirms the objective measures previously discussed.

5.3 Accuracy

The experiment described in this subsection aims at evaluating the degree of accuracy of the recommendations generated by ISABEL in predicting the choices of the students. To this purpose, we have observed the navigation of the students without the support of the e-learning system, leaving them free to choice learning objects of their interests among those contained in the catalog of each visited e-learning site. We have performed this experiment on the same set of 78 students involved in the experiment described in subsection 5.2, considering the profiles yet constructed in that experiment. We have monitored these students in their navigation among the 12 e-learning sites exploited in subsection 5.2 and we have compared the choices performed by the students with those computed by ISABEL and the other e-learning systems considered in the previous exper-

iment. Obviously, the recommendations computed by the e-learning systems are not shown to the students. In our tests we have computed three widely accepted accuracy measures, namely *Precision*, *Recall*, and *F-Measure*. Precision is defined as the share of the learning objects selected by the student among those recommended by the system; vice versa, Recall is the share of the pages suggested by the system among those chosen by the student. F-Measure represents the harmonic mean between Precision and Recall (see (Van Rijsbergen, 1979) for details about these measures). These parameters have been computed as follows:

- Given a page, say p_k , accessed by a student s_i , the links l_1, \dots, l_n (referring learning objects p_1, \dots, p_n) present in p_k have been considered. We call $PTempSet_i^k$ the set of these learning objects.
- s_i was asked to identify the subset $UserPSet_i^k \subseteq PTempSet_i^k$ of pages that he considered interesting.
- Our prototype was run for obtaining the set $SystemPSet_i^k \subseteq PTempSet_i^k$ of pages to be recommended to s_i .
- The Precision Pre_i^k , the Recall Rec_i^k and the F-Measure F_i^k , related to the visit of s_i to p_k , have been obtained by applying the formulas (see (Van Rijsbergen, 1979)):

$$Pre_i^k = \frac{|SystemPSet_i^k \cap UserPSet_i^k|}{|SystemPSet_i^k|} \quad (5)$$

$$Rec_i^k = \frac{|SystemPSet_i^k \cap UserPSet_i^k|}{|UserPSet_i^k|} \quad (6)$$

$$F_i^k = 2 \cdot \frac{Pre_i^k \cdot Rec_i^k}{Pre_i^k + Rec_i^k} \quad (7)$$

For each student s_i , we have computed the values Pre_i^k , Rec_i^k and F_i^k in correspondence of $J = 10$ different pages p_k , in order to determine the average values Pre_i , Rec_i and F_i as follows:

$$Pre_i = \frac{\sum_{k=1}^J Pre_i^k}{J} \quad Rec_i = \frac{\sum_{k=1}^J Rec_i^k}{J} \quad F_i = \frac{\sum_{k=1}^J F_i^k}{J} \quad (8)$$

- Finally, the *Average Precision* $AvgPre$, the *Average Recall* $AvgRec$ and the *Average F-Measure* $AvgF$ have been obtained as follows:

$$AvgPre = \frac{\sum_{i=1}^n Pre_i}{n} \quad AvgRec = \frac{\sum_{i=1}^n Rec_i}{n} \quad AvgF = \frac{\sum_{i=1}^n F_i}{n} \quad (9)$$

where n is the number of the students present in the systems.

Observe that Precision, Recall, F-Measure, Average Precision, Average Recall and Average F-Measure belong to the real interval $[0, 1]$; specifically, the higher these coefficients are the better the system works.

The accuracy measure computation was performed for various values of $TOP N$; this parameter denotes the number of learning objects that each approach must recommend. $TOP N$ is a parameter that varies on the basis of the desires of each student, that prefers to have given number of recommendations.

Table 3: Average Precision against $TOP N$ for different e-learning systems

	$TOP 2$	$TOP 4$	$TOP 8$	$TOP 16$	$TOP 20$
<i>ISABEL</i>	0.804	0.747	0.705	0.679	0.670
<i>MASHA</i>	0.777	0.737	0.699	0.679	0.648
<i>S1</i>	0.755	0.721	0.687	0.654	0.633
<i>S2</i>	0.726	0.700	0.669	0.635	0.617
<i>S3</i>	0.728	0.709	0.676	0.641	0.624

Tables 3, 4 and 5 present the values of the Average Precision, the Average Recall and the Average F-Measure obtained, in this experiment, by ISABEL, MASHA, S1, S2 and S3, respectively.

From the analysis of these tables we can see that as far as Recall and F-Measures are concerned, the highest values can be found for ISABEL. In particular the advantage of using ISABEL, as well as of the other systems into consideration, increases when $TOP N$ increases.

Table 7 shows, for various values of $TOP N$, the share of sessions in which ISABEL operates better than the other approaches, in terms of Average Precision, Average Recall and Average F-Measures, respectively. The analysis of this table further confirms the high accuracy of our system.

We observe that the effectiveness of ISABEL is better than that of MASHA, that is the only system among those considered that takes into account the effect of the different devices in the computation of recommendations. As pointed

Table 4: Average Recall against *TOP N* for different e-learning systems

	<i>TOP 2</i>	<i>TOP 4</i>	<i>TOP 8</i>	<i>TOP 16</i>	<i>TOP 20</i>
<i>ISABEL</i>	0.671	0.693	0.699	0.724	0.731
<i>MASHA</i>	0.568	0.597	0.599	0.626	0.629
<i>S1</i>	0.557	0.581	0.589	0.613	0.620
<i>S2</i>	0.546	0.570	0.576	0.601	0.610
<i>S3</i>	0.548	0.572	0.580	0.603	0.611

Table 5: Average F-Measure against *TOP N* for different e-learning systems

	<i>TOP 2</i>	<i>TOP 4</i>	<i>TOP 8</i>	<i>TOP 16</i>	<i>TOP 20</i>
<i>ISABEL</i>	0.73	0.739	0.744	0.751	0.762
<i>MASHA</i>	0.617	0.621	0.630	0.641	0.651
<i>S1</i>	0.600	0.608	0.614	0.620	0.631
<i>S2</i>	0.578	0.587	0.594	0.599	0.608
<i>S3</i>	0.58	0.591	0.600	0.605	0.612

Table 6: Average F-Measure against $TOP N$ for ISABEL and MASHA, computed for both Content-based (CB) and Collaborative filtering (CF) recommendations

	$TOP 2$	$TOP 4$	$TOP 8$	$TOP 16$	$TOP 20$
$ISABEL - CB$	0.698	0.707	0.718	0.729	0.744
$MASHA - CB$	0.698	0.707	0.718	0.729	0.744
$ISABEL - CF$	0.714	0.722	0.736	0.743	0.750
$MASHA - CF$	0.580	0.588	0.594	0.603	0.618

out in Section 1, this better result is mainly due to the introduction of the tutor agent, that is able to compute a more precise collaborative-filtering component of the recommendation. To better understand this fact, we have reported in Table 6 the values of F-Measure of both ISABEL and MASHA for the distinct components of the recommendations. In this table, we have indicated by $ISABEL - CB$ (resp. $ISABEL - CF$) and $MASHA - CB$ (resp. $MASHA - CF$) the content-based component (resp. the collaborative filtering component) of the recommendations generated by ISABEL and MASHA, respectively. We remark that ISABEL and MASHA identically perform for the content-based component (since they compute the recommendations on the same set of items) while ISABEL performs significantly better than MASHA for the collaborative-filtering component (that ISABEL computes considering all the students in the same partition while MASHA computes only for a given

Table 7: Share of sessions in which ISABEL operates better than the other approaches

	<i>TOP 2</i>	<i>TOP 4</i>	<i>TOP 8</i>	<i>TOP 16</i>	<i>TOP 20</i>
<i>AvgPrecision</i>	62%	63%	68%	74%	56%
<i>AvgRecall</i>	75%	79%	82%	83%	87%
<i>AvgF</i>	80%	86%	89%	91%	92%

site).

We remark that the reduction of the hyperlink alternatives in presence of devices with limited resources doubtless advantages the system’s prediction. However, Tables 3, 4 and 5 show that, also in the favorable case of a small value of $TOPN$, the results produced by our approach appear better than the other ones, although the differences of performances with the other systems are smaller than for higher values of $TOPN$, as we expected. We remark that also in this case ISABEL is advantaged by the possibility of taking into account, for the same learning object, different values of interest for different devices, and this assumes a significant role when we are in presence of devices with limited resources and consequently we generate a small number of recommendations. While a traditional system evaluates in the same way the interest for an object if accessed by a desktop PC or a cellular phone, ISABEL introduces a difference in the evaluation, and this leads to generate better results also in presence of a limited number of alternatives.

5.4 Efficiency

Finally, we have compared the impact of the different recommendation algorithms on the performances of the e-learning sites. Figure 5 reports the average waiting time of the students when accessing an e-learning site considered in the experiment above. The average value has been computed on all the e-learning sites and for different number of client accesses (the parameter *acc* of Figure 5). The experiment shows that ISABEL introduces a waiting time significantly smaller than the other systems, and this positive gain in terms of time cost increases when the number of accesses increases too. This advantage can be explained by theoretical considerations, since the computational complexity of the ISABEL recommendation algorithm is $\mathcal{O}(l \cdot \pi)$ (where l is the number of the learning objects and π is the number of student partitions) while that of the other systems is $\mathcal{O}(l \cdot s^2)$, linearly depending on the number s of students in the system. The number of students in the system strongly influences the number of client accesses to each e-learning site, since in average when the number of students increases the number of accesses increases too. Accordingly with this consideration, Figure 5 shows a slight increasing of the time cost of ISABEL with respect to the number of client accesses, while the other systems present a significant monotonic increasing.

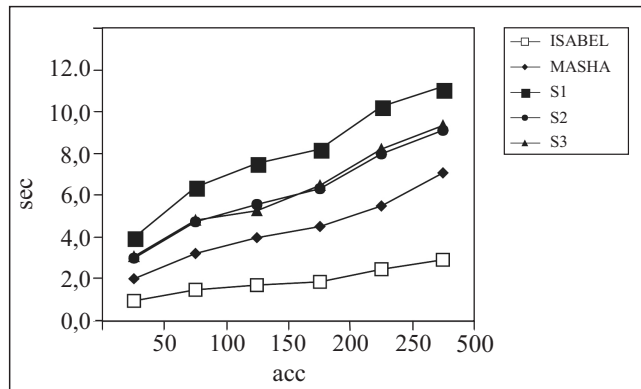


Figure 5: Waiting time of different recommender systems

6 Conclusions

In this paper, we have dealt with the issue of e-learning systems that allow students to access educational resources by using different devices. Such an issue has been addressed in the past by other approaches, which however present a significant limitation in terms of computational time costs when operating with large community of students. The system which we here propose, called ISABEL, gives a contribution to this issue. ISABEL is a recommender system architecture for supporting e-learning, designed for generating recommendations based on both student profile and exploited device. Our approach, as confirmed by experiments on a real community of students, leads to generate very effective recommendations, taking into account also the exploited devices, and leaving to the site agent only the task of generating a graphical presentation. This also produces a significant reduction of the time cost of the student when he waits for visualizing the Web pages of the e-learning sites. It is important to

point out that the improvements introduced by ISABEL in the efficiency of the recommendation is a theoretical result, while the better quality of the recommendations generated by ISABEL is derived by quantitative considerations derived by experiment. These experimental results seem promising, but they need to be confirmed by further analytical studies. This is a subject of our ongoing research. As another issue for further development of our research, in order to further reduce the ambiguity of topics, we plan to consider the use of ontologies for representing the domains of interests, instead of exploiting a simple dictionary. We argue that ontologies would even open new possibilities for recommendations since there would be semantic relations between the topics that can be leveraged for recommending related/similar content in the absence of the most suitable content.

References

- Al Halabi W.S., Kubat M. and Tapia M. (2007). Time Spent on a Web Page is Sufficient to Infer a User's Interest. In *Proc. of the IASTED Europ. Conf. (IMSA '07)*, pages 41–46, Anaheim, CA, USA. ACTA Press.
- Anderson, T. and Whitelock, D. (2004). The Educational Semantic Web: Visioning and Practicing the Future of Education, Special Issue on the Educational Semantic Web. *J. of Interactive Media in Education*.
- Badi R., Bae S., Moore J.M. Meintanis K., Zacchi A., Hsieh H., Shipman F. and Marshall C.C. (2006). Recognizing User Interest and Document Value

- from Reading and Organizing Activities in Document Triage. In *Proc. of the 11th Int. Conf. on Intel. User Interfaces (IUI '06)*, pages 218–225, New York, NY, USA. ACM.
- Brusilovsky, P. (2004). KnowledgeTree: a Distributed Architecture for Adaptive E-Learning. In *Proc. of the 13th Int. WWW Conf. on Alternate Track Papers & Posters (WWW Alt. '04), 2004*, pages 104–113. ACM Press.
- Burgos, D. (2007). Adaptation and IMS Learning Design, Special Issue on the Adaptation and IMS Learning Design. <http://www-jime.open.ac.uk/2007/>.
- Caire, G. (2008). *LEAP 3.0: User Guide*. <http://mia.ece.uic.edu/papers/MediaBot/jade/doc/LEAPUserGuide.pdf>.
- Conlan, O., Wade, V., Gargan, M., Hockemeyer, C., and Albert, D. (2002). Integrating Adaptive Hypermedia Services with Open Learning Environments. In *Proc. of the World Conf. on Educational Multimedia, Hypermedia and Telecom. (ED-MEDIA), 2002*, pages 344–350. AACE.
- Denman-Maier, E. (2004). Intercultural Factors in Web-based Training Systems. *J. UCS*, **10**(1), 80–104.
- Dolog, P., Henze, N., Nejdl, W., and Sintek, M. (2004). Personalization in Distributed e-Learning Environments. In *Proc. of the 13th Int. WWW Conf. on Alternate Track Papers & Posters (WWW Alt. '04), 2004*, pages 170–179. ACM Press.

- Esposito, F., Licchelli, O., and Semeraro, G. (2004). Discovering Student Models in e-Learning Systems. *J. UCS*, **10**(1), 37–47.
- Friesen, N. (2005). Interoperability and Learning Objects: an Overview of E-Learning Standardization. *Interdisciplinary J. of Knowledge and Learning Objects*, **1**, 23–31.
- Garruzzo, S., Modafferi, S., Rosaci, D., and Ursino, D. (2002). X-Compass: An XML Agent for Supporting User Navigation on the Web. In *Proc. of the 5th Int. Conf. on Flexible Query Answering Systems, FQAS 2002*, volume 2522 of *LNCS*, pages 197–211. Springer-Verlag.
- Garruzzo, S., Rosaci, D., and Sarné, G. (2007). ISABEL: A Multi Agent e-Learning System That Supports Multiple Devices. In *Proc. of the 2007 Int. Conf. on Intel. Agent Technology (IAT 2007), Malaga, Spain, 2002*, pages 485–488. IEEE.
- Goecks J. and Shavlik J. (2000). Learning Users' Interests by Unobtrusively Observing Their Normal Behavior. In *Proc. of the 5th Int. Conf. on Intel. User Interfaces (IUI '00)*, pages 129–132, New York, NY, USA. ACM.
- Hauger, D. and Köck, M. (2007). State of the Art of Adaptivity in E-Learning Platforms. In *Proc. of the LWA 2007: Lernen - Wissen - Adaption, Halle, September 2007*, pages 355–360. Martin-Luther-Univ. Halle-Wittenberg.
- Horváth, L. and Rudas, I. (2006). Course Modeling for Student Profile Based Flexible Higher Education on the Internet. *J. UCS*, **12**(9), 1254–1266.

- IEEE LTSC (2005). IEEE P1484.12 Learning Object Metadata Working Group.
<http://ltsc.ieee.org/wg12/>.
- Karagiannidis, C., Sampson, D., and Cardinali, F. (2005). Adaptive Learning Resources Sequencing in Educational Hypermedia Systems. *Educational Technology & Soc.*, **8**(4), 128–147.
- Kelly D. and Belkin N.J. (2004). Display Time as Implicit Feedback: Understanding Task Effects. In *Proc. of the 27th Int. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR '04)*, pages 377–384, New York, NY, USA. ACM.
- Kim H.-R. and Chan P.K. (2008). Learning Implicit User Interest Hierarchy for Context in Personalization. *Applied Intell.*, **28**(2), 153–166.
- Kurhila, J., Miettinen, M., Nokelainen, P., and Tirri, H. (2007). EDUCO: Social Navigation and Group Formation in Student-Centred E-Learning. *J. of Interactive Learning Research*, **18**(1), 65–83.
- Liu, W., Li, Q., and Lau, R., editors (2006a). *Advances in Web Based Learning - Proc. of the 5th Int. Conf., Penang, Malaysia (ICWL 2006), 2006. Rev. Papers.*, volume 4181 of *LNCIS*. Springer.
- Liu, Z., Wang, Z., and Fang, Z. (2006b). An Agent-Based E-Learning Assessing and Instructing System. In *Proc. of the 10th Int. Conf. on CSCW in Design, CSCWD 2006, Southeast Univ., Nanjing, China*, pages 1414–1419. IEEE.
- Neijdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., Nilsson, M.,

- Palmer, M., and Risch, T. (2002). EDUTELLA: A P2P Networking Infrastructure Based on RDF. In *Proc. of the 11th Int. World Wide Web Conf. (WWW 2002)*, 2002, pages 604–615. ACM Press.
- Parsons, J., Ralph, P., and Gallagher, K. (2004). Using Viewing Time to Infer User Preference in Recommender Systems. In *Proc. of the AAAI Work. on Semantic Web Personalization*, pages 52–64. AAAI Press.
- Rafaeli, S., Barak, M., Dan-Gur, Y., and Toch, E. (2004). QSIA - a Web-based Environment for Learning, Assessing and Knowledge Sharing in Communities. *Computer & Education*, **43**(3), 273–289.
- Rosaci, D. and Sarné, G. (2006). MASHA: A Multi Agent System Handling User and Device Adaptivity of Web Sites. *User Modeling and User-Adapted Interaction*, **16**(5), 52–64.
- Sancho, P., Martínez-Ortiz, I., and Fernández-Manjón, B. (2005). Semantic Web Technologies Applied to e-learning Personalization in <e-aula>. *J. UCS*, **11**(9), 1470–1481.
- Silveira, R. A. and Vicari, R. M. (2002). Improving Interactivity in e-Learning Systems with Multi-agent Architecture. In *Proc. of the 2nd Int. Conf. on Adaptive Hypermedia and Adaptive Web-Based Sys. (AH 2002)*, Malaga, Spain, volume 2347 of *LNCS*, pages 466–471. Springer.
- Soonthornphisaj, N., Rojsattarat, E., and Yim-Ngam, S. (2006). Smart E-Learning Using Recommender System. In *Proc. of the Int. Conf. on Intel.*

Computing, ICIC 2006, Kunming, China, volume 4114 of *LNCS*, pages 518–523. Springer.

Tang, T. and McCalla, G. (2004). Evaluating a Smart Recommender for an Evolving E-learning System: A Simulation-Based Study. In *Proc. of the 17th Conf. of the Canadian Soc. for Computational Studies of Intel., Canadian AI 2004, London, Ontario, Canada*, volume 3060 of *LNCS*, pages 439–443. Springer.

Uskov, V. and Uskov, A. (2008). 2008 World Survey on Strategic Issues of Web-based Education. In *Proc. of the Web-Based Education (WBE 2008)*, pages 2008–2015. ACTA Press.

Van Rijsbergen, C. (1979). *Information Retrieval*. Butterworth.

Viet, A. N. and Si, D. H. (2006). ACGs: Adaptive Course Generation System - An Efficient Approach to Build E-Learning Course. In *Proc. of the 6th Int. Conf. on Computer and Information Technology (CIT 2006), Seoul, Korea*, page 259. IEEE.

Weber, G., Kuhl, H., and Weibelzahl, S. (2002). Developing Adaptative Internet Based Courses with the Authoring System NetCoach. In *Proc. of the Int. Work. OHS-7, SC-3, and AH-3 on Hypermedia: Openness, Structural Awareness, and Adaptivity. Revised Papers*, volume 2266 of *LNCS*, pages 226–238. Springer.