iRecomendYou: A design proposal for the development of a pervasive recommendation system based on student's profile for Ecuador's students' candidature to a scholarship

Filipe Mota Pinto[1,2]; Mireya Estefania[1]; Natalia Cerón[1]; Ramiro Andrade[1]; Mauricio Campaña[1];
[1] Universidad de las Fuerzas Armadas ESPE, Computer Science Department, Sangolqui – Quito, Ecuador
[2] Polytechnic Institute of Leiria, Computer Science Department, Leiria, Portugal
filipe.pinto@ipleiria.pt; {mechillan; npceron; reandrade5; emcampania}@espe.edu.ec

Abstract. All recognized successful Ecuador's students have the opportunity to apply for a scholarship abroad within a set of relevant world's universities listed on-line on SENESCYT's website. Students are invited to choose from a list with more than 1500 universities. From those, they only have no more information than each university general URL address. Considering student's limitation to compare and analyze all available courses it is frequent to exist students that complaint about their selection due their capacity to understand all available possibilities. Along this work we develop a proposal design for a pervasive recommendation system based on students' profile for scholarship's application based on their profile and universities programs' main characteristics.

Keywords: Pervasive; Recommendation systems; Profiling.

1 Introduction

Actually almost all countries and almost all developed societies had adopted technologies to support their activity. Nowadays is often to have innumerous examples whereas all the process (application, processing and results) is based on technologies. Countless an individual has little or no personal experience to make choices among the various alternatives presented to it. This happens by questions of time (normally on-line procedures don´t have too much time to be executed) or by the large alternatives presented – normally impossible to evaluate and consider all of them.

Pervasive systems consists of a large set of networked devices, seemingly invisibly embedded in the environment [12] Pervasive systems research was introduce in '80s at Xerox PARC. From de beginning a diversity of application domains have been proposed for pervasive systems, e.g., education [13], [15], public spaces [16] or health care [17][18].

In this work the proposed design solution it will focuses on the availability and manageability issues of pervasive systems. Here the goal is to find dependability

Á. Rocha et al. (eds.), *New Advances in Information Systems and Technologies*,
Advances in Intelligent Systems and Computing 445,
DOI 10.1007/978-3-319-31307-8_56

mechanisms and structures inside websites contents databases that (1) maximize the information required for support user decision, while (2) minimizing user's involvement cost of operation (basically time and effort). Informally, we it be used the term pervasive to refer to these two requirements [22]. Regarding pervasive computing concept, it admits that computing resources might be perceptible or imperceptible distributed related to the user environment (context-awareness). Context-aware applications promise richer and easier interaction, but the current state of research in this field is still far removed from that vision [14]. That is, users´ don't need to understand where system takes place or where to which devices is connected.

Considering options to handle with decision making support, it is possible to identify some scenarios, such as: trusting and using recommendations being passed by others, which can get directly (word of mouth) [2] or given by recommendation texts, reviews of movies and books reviewers, printed newspapers, social networks, among others.

The recommendation systems help to increase the capacity and effectiveness. This nominating process is already well known and established in the social relationship between humans [3].

In a typical recommendation system, there are sources of information (valid and accredited) recognized as recommendations input data, to the system that aggregates and directs (system processing) for individuals potentially viewed interested in this type of advise (target or user). One of the great challenges for this type of systems is to achieve the right mix between the expectations of users and the products, services or people to be recommended to them, that is, define and discover these interests' relationships.

This work focuses the development of design proposal for a recommendation system to solve a specific problem presented by Ecuador's national superior education secretary (SENESCYT): which program should a student select from a list with more than 1500 universities and 3000 possibilities (estimated student options by each scientific area).

From the beginning some conceptual definitions about recommendation and pervasive systems and will be presented (section two and three). Thereafter it will be discussed different sources of data (section four) and the strategy for the recommendation method (section five). Then the architecture is presented (section six) followed by results and conclusion at the closing section.

2 Recommendation systems

The proponents of the first recommendation system called Tapestry [3],[4] coined the term "*filtering collaborative*", aiming to designate a specific type of system in which the filter information was performed with human aid, e.g. the collaboration between stakeholder groups. Thereafter, same authors considered that could have another type of recommendation based on the content, that is, one element just might be considered as recommender if it was familiar, updated and contextualized with the subject: *context-based filtering*. Therefore they assumed that *collaborative filtering*

and *content-based filtering* systems are types of recommendation systems applying different approaches, but have the sole purpose of the recommendation.

Others authors [5] highlighted and proposed that there is a third type of filter referred to as demographic information filtering. *Demographic filtering* uses the description of an individual (profile determination) to learn the relationship between a particular item and the type of person that would come to be interested. This approach uses descriptions of people how to get learned the relationship between an item and the type of person who would like this. The user profile is created for classifying users in stereotypes that represent the characteristics of a class of users. Personal data is requested to the user, usually in registration forms, and used as characterization of users and their interests.

Later another's authors [6],[7] introduced two other techniques. In the first, entitled *filtering based on knowledge*, the recommendation of the items is done based on user preferences of inferences and their needs through structured functionally knowledge; the second technique, entitled *utility-based filtering*, the recommendation is made considering the utility of items for a given user.

Currently e-commerce websites are the main players recommendation systems use and exploration. They use different techniques to find the most suitable products to their customers and thus increase profitability. Introduced in July 1996 MyYahoo was the first website to use recommender systems in large proportions, using the customization strategy [8]. Currently, a large number of websites employ recommendation systems to bring the different user types of suggestions, as cross related offerings (something like, "customers who bought X also bought Y item"), or selling items on each customer´s favorite categories, among others.

This work focuses the development of a recommendation system to solve a specific problem: which program should a student select from a list with more than 1500 universities

2.1 User profile

The *identity* it is one of the most important related to recommendation objectives where's individuals' objectives, subjectivities and features, emerges. Recommendation systems use individuals' identity in order to provide clues on future behavior and needs of users such as in a given environment where customization becomes effective. In computer science, algorithms used to formalize individuals' identity on a given computing environment are based on *User Profile* determination.

On the web, there are many types of user profiles with different degrees of complexity. They are developed in the context of e-commerce, e-learning and e-community, among many others. Nevertheless there are some researchers [9][10] that had developed shell's for users models pointing, as example, categories of information about users in order to better customize the web applications [9] or to model specific users, such as, model of students' profile for learning activities [10].

The user profile might be used to predict user´s needs and behaviors in a computing environment.

To determine a user profile it is necessary to use strategies. On the web, two of the most common forms of user identification are:

- On server: systems' usually provides users with a registration area where's beyond username/password, with personal data such as name, date of birth, sex, address and others. These data is stored in a database on the server. Whenever the user accesses the system, he may make his identification / authentication updating their previously registered login. This mechanism allows the website identify more accurately the user that it connects;
- On the client device: normally use cookies, a mechanism by which a website can identify that particular computer is connecting once again it. This method assumes that the connected machine it is always used by the same person. So to identify the machine, the website is in reality identifying its user. This is a simpler mechanism than the identification through the server, but less reliable, especially if the identified computer is used by more than one person.

Thereafter user identification process, it is possible to collect data about him, implicitly or explicitly, thus allowing the generation and maintenance of their profile. In the explicit collection mode (also known as customization), the user spontaneously indicates what is important to him, improving his profile determination and explicit

3 Web Crawler

Given the current volume of data on the Web, an automatic, methodical process of content indexing becomes extremely necessary. It is in this scenario that the web crawler is present searching, filtering and often persisting content. One of the most powerful known web crawlers is the googlebot, Google's web crawler, he is responsible for scrub the web, looking for new pages to index and analyze whether the existing pages were updated. A Web crawler, which is also known as spider or web robot web, is a software agent used to automate research, data gathering and extraction. Usually trafficked in HTTP along with the TCP/IP model, the most developed robots are meant for the traditional web where *html* documents connected by *hyperlinks, tags* and *keywords* are used to track and meet specific needs as download images and files, mining of e-mail addresses, collection of specific contents for price quote for a particular product or service and can also be developed to aggregate data from various internet sources, among many others conveniences.

These agents are not limited to a specific language actually, since it is possible to extract and apply regular expressions obtained over the contents, any language can be used for the implementation of web crawlers.

The basic structure of running a crawler is not very complex, though the same cannot be said for its implementation. See a model of the basic structure of a crawler in Figure 1.

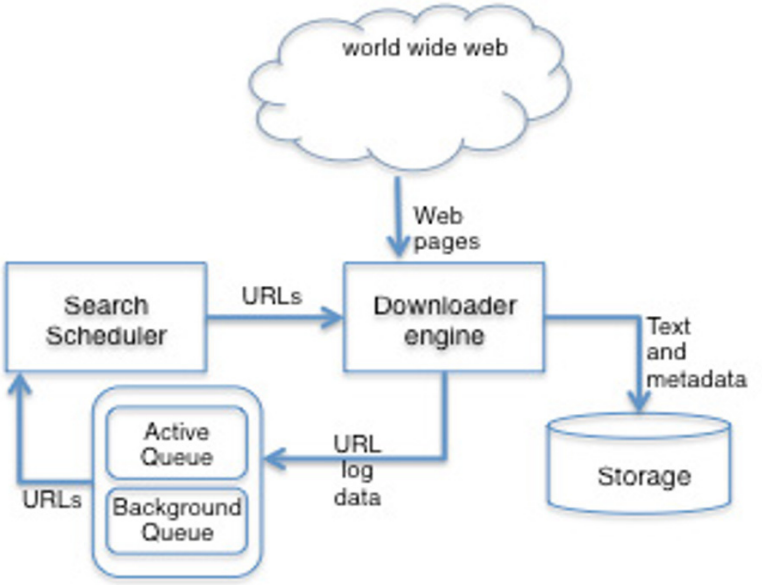

Fig. 1. web crawler's basic model structure. Adapted from [23]

General web crawler's activity steps goes around a content request, which is set in a scheduler. From the scheduler, a list of URLs is passed to the downloader engine, which acts as a multi-threaded download sniffer. This component is responsible to search over through in order to find out the requested webpages referenced on scheduler. All successful web pages found are saved on a specific storage (text and metadata for post processing). Unsuccessful analyzed URLs are posted in a queue (active queue) in order to be included in new late scheduler. The crawler queue storage also includes a background queue, in order to support background updates, to be programmed in order to update some specific URLs related data. The scheduler alignment and downloader engine (extraction process) are inherent in any process of crawling, regardless its objective or technology used.

As noted previously, the crawling is not tied to a web contents, indeed it is possible to act on various types of sources. Thus it can be used over different protocols such as HTTP SMTP, FTP, and others. Once selected the protocol, the agent take a request to the data source, which ultimately return the desired content, which will be working for the extraction of information. The protocol used in this article will be explained more technically below.

3.1 Web crawler at work

The crawler starts with a list of URLs seeds to visit. This is constructed and introduced at user/system request. During crawling work, the application visits these URLs, it identifies all the possible hyperlinks in each page that corresponds to the overall search objectives. Not all visited URLs revels positive results. Here it was considered three different types of possible outcomes: not correspondent contents;

positive contents; unavailable page. Whenever a URL is inaccessible, the application put that address in a queue list. On other side, there is the possibility to select some URLs by their importance to be included on background queue. Each URLs in this queue list is visited periodically (programmer defined) in order to have updated information.

When there is large volume of URLs to explore implies that the crawler can only download a limited number of the web pages within a given time, so it requires to prioritize its downloads. To that end all returned retrieved data (text, metadata, images, among any other type) from analyzed web pages are stored in table, to be delivered and used by the user and also to act as cache for next visit (avoiding all retrieval process over previously visited pages). Generally web crawler's behavior is the outcome of a combination of policies:

- *selection policy*, to determine and define the URLs to analyze and download;
- *re-visit policy,* to consider the set of URLs that might be considered relevant and therefore to be monitored;
- *politeness policy*, despite this doesn´t have direct influence on application results, it should be considered in order to avoid targeted web sites' traffic overload;
- *parallelization policy*, regarding web crawler running multiple process.

As described in the next section, considering the objectives for the current design proposal these policies were considered in terms of search scheduler, download engine and queue list definition.

4 Design Proposal

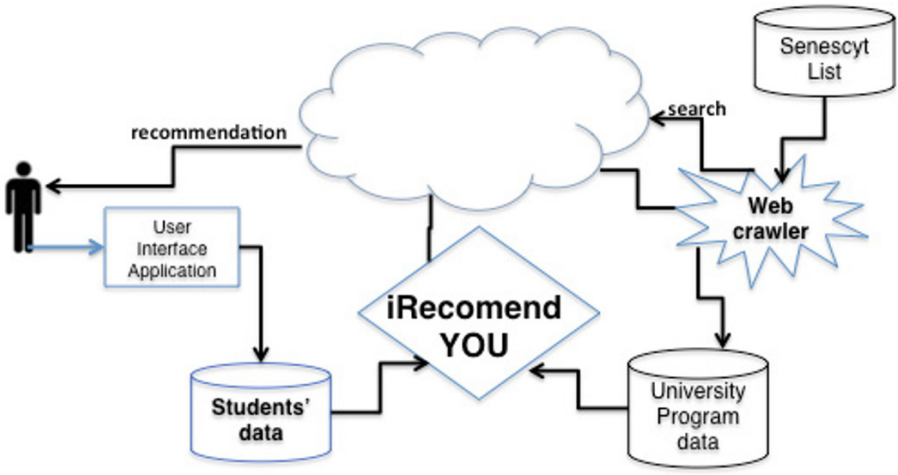

Fig. 2. Proposed iRecommendYOU framework design's structure.

The design proposal, as depicted in figure 2 involves three main components: from user side (student's objective fulfillment); web exploration (data gathering from web) and recommendation system (iRecommendYOU). System's workflow begins with

user's profile determination that will determine the set of URLs to be analyzed. Next, using the web crawler application, the system will collect data from related web pages, that will be stored on a data structure. At the end, using a recommendation algorithm, the system will provide the user with the results that better fits with his profile.

4.1 User profiling and objectives fulfillment

User profiles are generally represented as sets of weighted keywords, semantic networks, or weighted concepts, or association rules. Keyword profiles are the simplest to build, but because they fundamentally have to capture and represent all (or most) words. This technique requires a large amount of user feedback in order to learn the terminology by which a topic might be discussed [23].

This proposed user profile system incorporates a registration section where's user's needs introduce some personal preferences (questionnaire) and personal data (curriculum vitae). Moreover students would be are required to fulfill a questionnaire, that will. As illustrated on figure 3, user's profile is determined through two sets of data: the first considering academic preferences and curriculum; the second, related to a questionnaire previously fulfilled by the student. This questionnaire it is presented by SENESCYT in order to have a standard source of data and covers a variety of subjects' information such as students' preferences, professional options or research investigation interests. The profile is constructed using the most keyword descriptors on user's data. The profiler module uses a knowledge base in order to search, evaluate and define user's profile

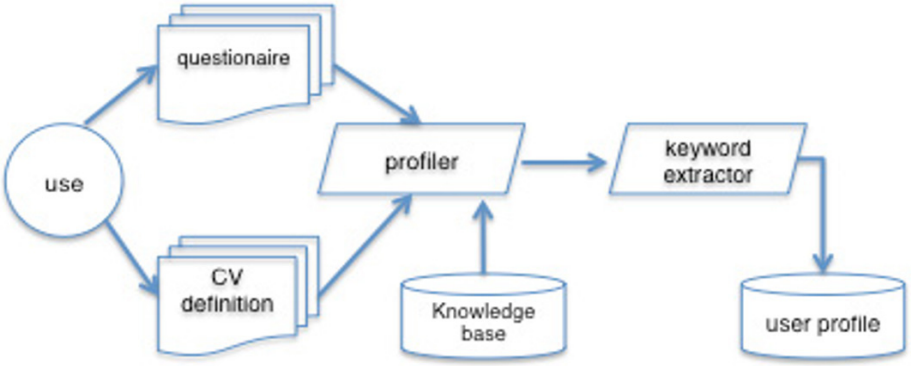

Fig. 3. Users' profile system.

Keyword-based profiles are initially created by extracting keywords from Web pages collected from some information source, e.g., the user's browsing history or Bookmarks[23]. To calculate user's profile it is used the keyword's weighting technic to identify the most important keywords from user's curriculum and questionnaire data. Often the number of words extracted from a single document is capped so that only the top N most highly weighted terms from any page contribute to user's profile [23]. The keyword extractor module will determine final user profile using all keywords extracted and registered on user's profile storage.

4.2 Data gathering

This accomplishment is performed by the web crawler application in order to explore and retrieve data from SENESCYT's universities URLs list (a set of pre selected universities across the world to which students may apply for a scholarship grant).

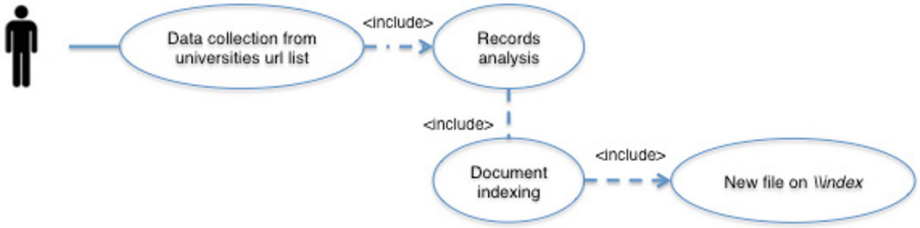

Fig. 4. Data gathering.

As depicted above on figure 4, the web crawler starts with a single URL, collected from universities url's list, downloads that page, retrieves the links from that page to others, and repeats the process with each of those pages – records analysis. Before long, web crawler discovers links to most of the pages on the Web, although it takes some time to actually visit each of those pages – document indexing.

The implementation of this algorithmic might be developed based on Apache LuceneTM, which is a high-performance, full-featured text search engine library. It is a technology suitable for nearly any application that requires full-text search, especially cross-platform. [24]

4.3 Decision support supported by recommendation systems

The recommendation is performed based on match work between the user profile and the programs that have a set of descriptors and keywords similar or on the same scientific area.

This recommendation system allows on this way a customization, over the short list of programs that meet the user's needs or expectation's [1]. From the computational point of view, one system capable of treating each user individually requires a set of specific functions, as example, through constant selection of related program to user interests, a personalized system can reduce the time they take to find relevant information. The user will have a list of possible programs that match his profile. That list has in each record, the program name, the university, the origin country, the set of keywords used to select the program and a matching estimator between user profile and program descriptor. Over the list the user may perform some actions such as select, download related program and create wish list for future analysis.

The system additionally, may identify relationships between items (e.g., "those who view the program X also analyzed the program Y").

5 Discussion and Conclusions

Considering the lack of information from both sides of the problem (candidates and financed programs) it was necessary to propose a valid solution that allows both, users' profile (student) determination from questionnaire and personal data, and data gathering (programs) work from a single url information. The research work was developed towards a solution that covers data collection (candidates and programs) for automatic filtering and analysis in order to support the recommendation action. The recommendation work would be supported by a set of rules that would be matching users' profile main characteristics with main programs' descriptive (scientific area, research lines and objectives) keywords.

Therefore this paper, based on developed work, presents a possible solution for system development in order to solve a problem presented in a national secretary for superior education and investigation.

All proposed development system and solution are based on related works and also on related bibliography.

For future word it is planed the development of this solution and make it available for the large community of Ecuadorian students that every year looks for scholarship opportunity.

7 References

1. Gyara, F., Sachdev, T. Win in the flat world. White paper - Infosys Technologies. Available on-line at: http://www.infosys.com/offerings/it-services/informationmanagement/white papers/documents/personalizing-portals.pdf (2008).
2. Maes, P.; Shardanand, U. "Social information filtering: Algorithms for automating "word of mouth", In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp: 210-217 (1995).
3. Resnick, P. e Varian, H. R. Recommender Systems. Communications of the ACM, New York, 40(3), pp. 55-58, Mar. (1997)
4. Goldberg, D., Nichols, D., Oki, B. M., Terry, D. Using collaborative filtering to weave an information Tapestry. Communications of the ACM, New York, 35 (12), pp. 61-70.(1992)
5. Montaner, M., López, B., de La Rosa, J.L. A Taxonomy of Recommender Agents on the Internet. Artificial Intelligence Review. Netherlands : Kluwer Academic Publishers, pp. 285-330, Aug. (2003)
6. Burke, R. Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction, 12(4) pp:331-370. (2002)
7. Guttman, R. H., Moukas, A. G. and Maes, Pattie. Agent-mediated electronic commerce: a survey. Knowl. Eng. Rev., 13(2) pp:147-159 (1998)
8. Manber, U.; Patel, A.; Robison, J. Experience with Personalization on Yahoo! Communication of the ACM, New York. (2000)
9. Kobsa, A. Generic user modeling systems. In P. Brusilovsky, A. Kobsa, and W. Nejdl, editors, The Adaptive Web, volume 4321 of Lecture Notes in Computer Science, chapter 4, pp:136–154. Springer Verlag. (2007)
10. Paiva, A. and Self, J.A. Tagus - a user and leamer modelling workbench. User Model. User-Adapt. Interact., 4(3) pp:197–226. (1995)

11.Herlocker, J. L., Konstan, J. A., Terveen, L. G. and Riedl ,J. T. Evaluating collaborative Filtering recommender systems. ACM Trans.Inf. Syst., 22(1) pp:5-53. (2004)
12.Weiser M., Gold R., and Brown J. S. The origins of ubiquitous computing research at PARC in the late 1980s. IBM Systems Journal, 38(4) pp:693–693. (1999)
13. Abowd, G. D. Classroom 2000: An experiment with the instrumentation of a living educational environment. IBM Systems Journal, 38(4) pp:508–530, 1999.
14 Dey, Anind, Abowd, Gregory and Salber, Daniel. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. Journal Human-Computer Interaction vol.16, 2, pp 97-166. (2001)
15. Chen A., Muntz R.R., Yuen S., Locher I., Park S.I., and Srivastava M.B.. A support infrastructure for the smart kindergarten. IEEE Pervasive Computing, 1(2):49–57 (2002).
16. Fleck M., Frid M., Kindberg T., O'Brien-Strain E., Rajani R., and Spasojevic M. From informing to remembering: ubiquitous systems in interactive museums. IEEE Pervasive Computing, 1(2) pp:13 – 21 (2002).
17.Stanford V. Using pervasive computing to deliver elder care. IEEE Pervasive Computing, 1(1) pp:10 – 13. (2002).
18. Portela F, Santos M. F., Vilas-Boas M., A Pervasive Approach to a Real-Time Intelligent Decision Support System in Intensive Medicine in Knowledge Discovery, Knowledge Engineering and Knowledge Management V. 272 of Communications in Computer and Information Science pp 368-381 (2013)
19. Edwards, J., McCurley, K. S., and Tomlin, J. A.An adaptive model for optimizing performance of an incremental web crawler. In Proceedings of 10th Conference on World Wide Web (Hong Kong: Elsevier Science):106–113. doi:10.1145/371920.371960. (2001)
20. Bidoki, Y., Yazdani, N. ; Ghodsnia, P., "FICA: A fast intelligent crawling algorithm", Web Intelligence, IEEE/ACM/WIC International conference on Intelligent agent technology pp: 635-641, (2007)
21. Gauch, S., Speretta, M., Chandramouli, A., Micarelli, A.: User Profiles for Personalized Information Access. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) The Adaptive Web: Methods and Strategies of Web Personalization. LNCS, vol. 4321, pp. 54–89. Springer, Heidelberg (2007)
22. Portela F., Santos M. F., Machado J., Abelha A., Álvaro Silva, Rua F. Pervasive and Intelligent Decision Support in Intensive Medicine in Information Technology in Bio and Medical Informatics – The Complete Picture, Lecture Notes in Computer Science V. 8649 pp 87-102 (2014)
23. Shkapenyuk, V. and Suel, T. Design and implementation of a high performance distributed web crawler. In Proceedings of the 18th International Conference on Data Engineering (ICDE), pages 357-368, San Jose, California. IEEE CS Press. (2002)
24. Foundation, T. A. (2011-2012). Apace Lucene Core. http://lucene.apache.org/core/