Design of a Recommender System for Intelligent Classrooms Based on Multiagent Systems

Dulce Rivero-Albarrán[1,2](✉), Francklin Rivas-Echeverria[1,2], Laura Guerra[1], Brian Arellano[1], and Stalin Arciniegas[1]

[1] Pontificia Universidad Católica del Ecuador,
Sede Ibarra. Escuela de Ingeniería, Ibarra, Ecuador
{dmrivero,firivas,lrguerra,bgarellano,
smarciniegas}@pucesi.edu.ec
[2] Facultad de Ingeniería, Escuela de Ingeniería de Sistemas,
Universidad de Los Andes, Mérida, Venezuela

Abstract. In this work it's presented the description of a Recommender Systems to be used in a learning environment. For this, it's defined the diverse recommendation methods that can be used: Content-based Recommendations, Collaboration recommendations and Hybrid Systems. It is proposed a Teaching agent design for obtaining the most appropriate learning objects based on Multiagent Systems framework composed of a Planning Agent, a Recommendation System and a Learning Object Management System; all of them collaborating on an intelligent platform. This design can be used in intelligent classrooms because it can adapt the environment and the teaching contents according to the needs of each student.

Keywords: Recommender System · Intelligent classrooms
Multiagent systems · Learning objects · Learning environments

1 Introduction

The new technologies have changed the way that people perform their daily processes, these are already ubiquitous in the daily work. For this reason, organizations have incorporated these new technologies in their computer systems, in order to guarantee their permanence in an increasingly competitive market. These systems provide the set of services that employees and users require, improving the quality of their products and increasing the efficiency of their processes.

New technologies have had a positive impact on the way that systems perform their functions. This is the case of online shopping systems and some search engines, which have incorporated in their applications the ability to guide users in a personalized way, presenting objects of interest, where objects are in a wide space of options possible [1]. These applications are known as Recommender Systems (RSs).

Educational environments have not been out of these changes, technologies have improved the educational process. In technology-enhanced learning (TEL), its objective

Á. Rocha and T. Guarda (eds.), *Proceedings of the International Conference on Information Technology & Systems (ICITS 2018)*, Advances in Intelligent Systems and Computing 721,
https://doi.org/10.1007/978-3-319-73450-7_92

is to design, develop and test socio-technical innovations that support and improve the learning practices of both individuals and organizations. It can be found a variety of applications in this domain, some of these systems have been developed as virtual platforms for teaching, this is the case of Blackboard [2], Moodle [3], Schoology [4], Mahara [5]. Other applications were oriented to the development of Intelligent Tutorial Systems (ITS), among this category are Aspire [6], ITSpoke [7], AGT (Advanced Geometry Tutor) [8]. ITSs are systems of mass teaching, so do not contemplate modes of learning that adequately adapt to the previous knowledge and ability of evolution of each student. In recent years, as a branch of research in the field of intelligent environments (IE) has been working on the development of Intelligent Classrooms (IC).

Intelligent Classrooms (IC) are spaces enriched with electronic devices that perceive their environment, because they are sensitive environments, that respond and are adapted to the needs of the present users, they are transparent, ubiquitous and intelligent. In summary, it can be said that the IC are spaces equipped with devices and sensors that communicate, with the purpose of adapting both the environment and the teaching contents according to the needs of each person in the classroom. It creates more comfortable and pleasant spaces, improving productivity and satisfaction, all in a transparent and non-intrusive way. Among this area is ClassMate, an open framework for Intelligent Environments in Education proposed by Leonidis et al. [9]; the conceptual models for designing multiagent systems (MAS) architectures for intelligent rooms can be found in [10–12], and works oriented to the perception of the environment in [13].

A MAS is a system consisting of a set of intelligent agents, with autonomy and proactivity that collaborate with each other using a communication protocol to reach some global objective, these systems are used in various fields (robotics, simulation, mathematics, among others). In the educational environments are aimed at the control of their environment and the management of the teaching-learning process.

The search for learning resources is a fundamental element in TEL, so RSs have become an important element in teaching systems, allow more personalized systems, because identifying the characteristics of each student they can suggest courses and learning activities according to their profile [14, 15]. There are also RSs that suggest changes to teachers in order to improve the effectiveness of education systems using web environments [9]. An improvement to RSs in education environments is found in those, which suggest learning objects (LO) based on the student profile [16].

The literature in this area is broad, different methods and recommendation techniques are proposed. Some authors use multi-attribute collaborative filters with fuzzy matching methods for selecting learning objects [17]; other authors focus on the use of collaborative filters [18, 19] or use content-based method implemented using case-based reasoning (CBR) [20, 21] or using the attribute-based technique [22]. Each work offers a solution to the classic RS problems, which are the “Cold Strat”, and the scattered LO.

This paper presents a MAS that emulates activities in the teaching process in intelligent educational environments. The MAS integrates two systems: (1) a course planning system, whose function is to schedule each course and, if it does not reach an objective in the estimated approving rate, re-schedule its programming; (2) an RS in charge of suggesting to the students, based on their profile and the established plan, the learning objects to be selected that are most suitable to that profile. The RS was designed as a hybrid architecture, in which teachers and students can rate the quality of LO.

The presented system gives a solution to the "Cold Start" problem of each method: For the search method based on content, it starts from the tags found in the analysis of the contents and the LO/teacher matrix that contains the score of each LO suggested by the teacher in the planning phase. In the collaborative method, student profiles are established and the LO/profile matrix is used, which is debugged with the allocated LO utility defined value.

The work is organized as follows, section two presents the objective of this work and the basic concepts. In section three, it's presented the design of the teaching agent with its two components: the planning and recommendation system. Finally, section four presents the conclusions and future work.

2 Objectives of the Work and Basic Concepts

This work aims to develop a more personalized teaching system, therefore, the learning agent is defined as a system that, knowing the students in classroom, find their profile and based on their level of knowledge, learning characteristics, course syllabus and level of learning achieved, suggest the most appropriate learning objects. The design of the agent required, for its design, the concepts and foundations associated with the recommending systems, planning systems and learning object management systems.

2.1 Recommender Systems (RSs)

The RSs are information systems that provide suggestions of objects that are more likely to be of interest to a particular user [23]. Suggestions require a decision-making process such as: what type of learning objects are most appropriate according to the characteristics of the individual since the individual does not have enough knowledge to decide or evaluate the objects that could be of interest.

The RSs distinguish three important elements: the items that are the elements to be recommended, the users who are the individuals who are identified their needs and preferences and the transactions that are the interactions between the user and the system and can be collected implicitly or explicitly. A RS should be able to predict the expected utility of each item in order to select those that are of interest and classify them based on some mechanism of comparison.

According the way the recommendations are classified, there are three methods [24]:

- Content-based Recommendations.
- Collaboration recommendations.
- Hybrid approaches.

2.1.1 Content-Based Method

In the content-based RSs, it's calculated a utility function u (c_i, s_j) of item "c_i" for the user "s_j" which is calculated on the basis of the utility $u(c, s_k)$ assigned by user "s_k" to items $c \in C$ that are "similar" to the item c_i.

This approach emerges from information recuperation **systems** and information filtering. The main differences from recovery systems is that it uses user profiles that contain information about tastes, preferences and needs. Profile information can be obtained from users explicitly, through questionnaires, or implicitly learned from their transactional behavior over time. It focuses mainly on recommending items that contain textual information, such as documents.

It basically uses two vectors: one that contains the set of attributes that characterize an item and is used to determine the suitability of the item for recommendation purposes; and a second to store the preferences or tastes of the user - The contents are usually keywords (k) and indicate by weight the importance of k_i in document j (d_j).

2.1.2 Collaborative Method

The collaborative method tries to predict the usefulness of the elements for a given user based on elements previously qualified by other users. In this method the utility u(c, s) of the item s for user c is estimated based on the utilities $u(c_j, s)$ assigned to the element s by those users $c_j \in C$ that are "similar" to the user c.

The algorithms for collaborative recommendations can be grouped into two categories: (1) memory-based or heuristic-based making predictions based on the complete collection of previously-qualified elements by users; (2) based on models that use the collection of scores to learn a model, which is then used to make score predictions.

2.1.3 Hybrid Method

Combine the content-based method and the collaborative method, there are four ways to combine these methods: (1) implement collaborative and content-based methods separately and combine their predictions, (2) incorporate some content-based features into a collaborative approach, (3) incorporate some collaborative features into a content-based approach and (4) build a general unifying model that incorporates content-based and collaborative features.

2.2 Planning

Planning is a searching and articulating process for a sequence of actions (tasks, activities, processes) that achieve an objective [25]. A plan is generally described by the set of states, the goal to achieve and a set of actions. Each action contains the name of the action that moves from one state to another, the preconditions that must be met to reach the state, the effect that describes how the state changes when the action is executed.

One method to perform task planning is through graphs, this method provides better heuristic estimates. It consists of a sequence of levels corresponding to time steps in the plan, and where the level 0 is the initial state. Each level contains a set of literals that indicates the state that is reached if the actions were executed and a set of actions for the next level. A plan selects "the best sequence of actions" in order to reach the final state (target or goal).

2.3 Learning Objects (LO)

A LO is any digital or non-digital entity that can be used, reused or referenced for technology-supported learning [26]. An LO is characterized because they are small units (low granularity) and reusable. Other features are their customization, the sequence of how to present the LO can vary, their flexibility adaptable to different learning styles and because they are modular. There are several standards to define the structure of LO, some of them are NET Learnactivity and SCORM [27, 28], the latter is the most used model in the creation of learning objects, due to its ease of interchange between platforms or teaching environments and offers the possibility of dynamic content.

3 Teaching Agent Design

For the design of the teaching agent, it was necessary to perform an analysis. First, the processes, activities and objects involved in the teaching process were identified. It was worked with a group of teachers from different university (Pontificia Universidad Católica del Ecuador, Universidad de Los Andes), it was identified which processes were executed and how each was performed. From this study we identified the activities that can be assumed by an agent. Then, in a second phase, it was analyzed the collaboration between this teaching agent with the rest of the agents specified in the platform conceptual model [12].

As a result of the first phase, a process model was obtained, this model allows to specify the processes that are performed, as well as the objective, the inputs (data or events), the products that are generated at the output and the resources and information required in each process. Figure 1 presents the teaching process model using the model proposed by Eriksson and Penker [29]. In the model three processes are distinguished,

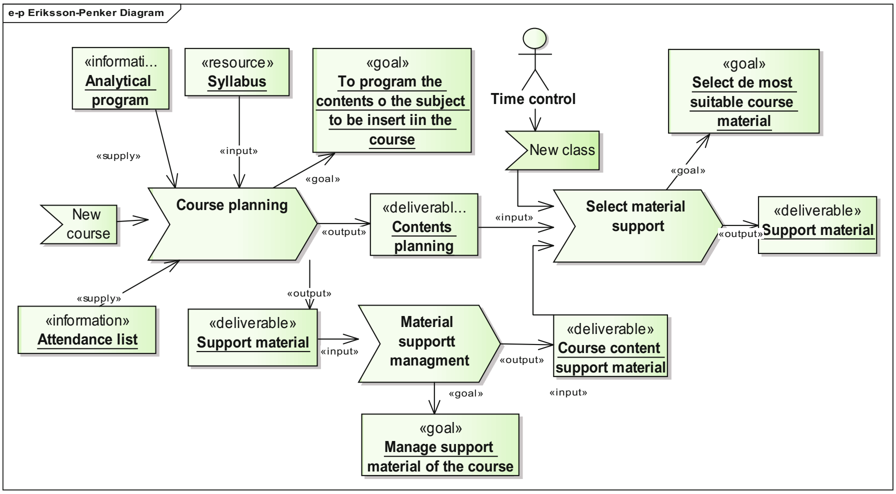
e-p Eriksson-Penker Diagram
«informati... Analytical program
«resource» Syllabus
«goal» To program the contents o the subject to be insert iin the course
Time control
New class
«goal» Select de most suitable course material
«supply»
«input»
«goal»
«goal»
New course
Course planning
«output»
«deliverabl... Contents planning
«input»
Select material support
«output»
«deliverable» Support material
«supply»
«output»
«information» Attendance list
«deliverable» Support material
«input»
Material supportt managment
«output»
«deliverable» Course content support material
«goal»
«input»
«goal» Manage support material of the course

Fig. 1. Teaching process model using UML business.

two of them are fundamental processes (planning of the content of the course and selection of course support material), the other is a support process (management of support material). The support process is transversal to the two previous processes, it is responsible for maintaining the register of books, guides, videos, among others, related to each course content and to support students.

After knowing that activities that are carried out in each process, the services that the intelligent platform had to support were identified. Finally, the services were grouped into three subsystems [30]: A Planning Agent, a Recommendation System and a LO Management System. Figure 2 presents the component model of the teaching agent, which shows the components and functions of each of them as well as their collaboration on the intelligent platform.

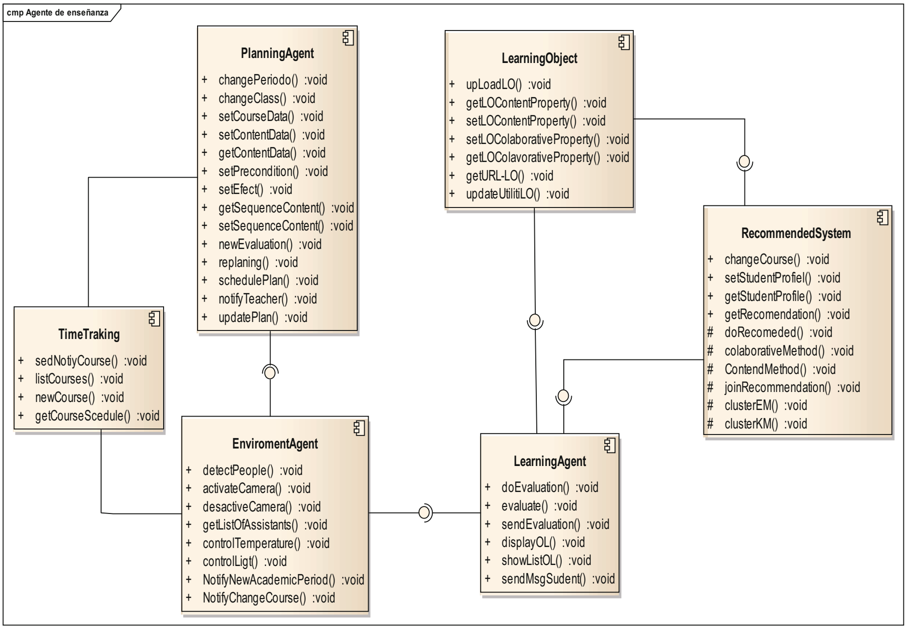

Fig. 2. Components diagram of the teaching agent.

3.1 Planning Agent

The planning agent is responsible for the planning, re-planning and follow-up of the course. The agent is activated when it's received the message changePeriodo() (see Fig. 2), this message notifies the teachers of each subject of the new period that they must update or create the course planning. The planning is done in three stages: first the teacher, using the analytical program of the subject, establishes the contents of the course (setContentData()). These data are: the content identifier, the specific content to be taught (derived from the analytical plan and/or competences to be developed), time dedicated to content (indicated in hours), hours taught (0 by default), type of activity (theory or practice, evaluation), descriptor 1 of content, descriptor 2 of content, content priority over the entire course, content status (not taught by default), minimum approval percentage.

Subsequently, the teacher enters the data sequence in which the contents will be taught (setSequenceContent ()), this is the plan itself. The data describing the sequence of contents are: **the state** indicated by the content to be taught (content identifier), **the preconditions** to be fulfilled to teach a certain content, that is, the contents previously seen and the **effects** caused at the end of the content, these are the contents that can be taken later. These data represent an unguided graph where the nodes represent the state and the arcs represent the preconditions and effects.

Finally, the teacher will be able to upload the LOs, scoring each according to the LO properties. Each learning object is scored in terms of its relation to the characteristics of the content of the subjects to be imparted, these data (LO Attributes), is stored in a matrix (LO x ContentPropierty). Table 1 shows the structure of this table.

Table 1. Score of the characteristics of the LO vs the characteristics of a learning content

Id LO	Complex level	Type	Time duration	Quality	Type of material	Related topics 1	..related topics n	tag1	…..	tag n
1	5	3	30	5	video	math	algebra	principle	….	Invt. opera
2	1	2	15	3	lecture	phisycal	termo	energy c.	….	principle
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
n	3	3	120	3	practice	design	material	…	.	…

The process of monitoring the plan is activated with the arrival of the event that indicates the start of a class session (ChangeClass()), this event activates the schdulePlan() method and causes the agent to send a message to the recommending system to proceed to search for the appropriate LOs for the content being sent (ChangeCourse ()). In addition, the system updates the course plan (updatePlan()), modifying the total number of hours given for the content that is taught, which allows to know the progress and status of a course.

Finally, the planning system, based on the learning results of each content (sent by teaching agent newEvaluation()), calculates the percentage of students who have approved the content and based on this value makes the decision to re-plan the course. If so, modify the time dedicated to that content (replanning()). To do this, the agent uses the graph storing the plan and for calculating the possible sequences contained, it applies a variant of algorithm Graphplan [25], these scenarios are presented to the teacher (getSequenceContent ()) so that the new plan should be selected.

3.2 Recommender System (RSs)

This system is responsible for finding the LOs that best fit the characteristics of each student. The RSs designed uses the hybrid method with a combined approach, where the recommendations generated by the collaborative method and the content-based method are combined into a single recommendation [23], as shown in Fig. 3.

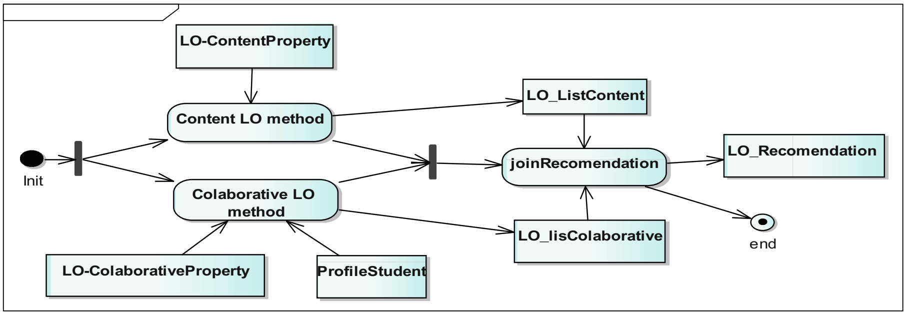

Fig. 3. Activity diagram of the recommendation hybrid strategy for the Recommender System

The recommendation system is made in two phases, in the first one the students' learning profiles are constructed and all the LOs that are associated to the set of characteristics of the content to be taught are searched, this search is done using the algorithm that calculates the similarity of cosines [31] that indicates the percentage of similarity that exists between the descriptors of the subjects and a LO. It's selected the LO with value greater than 50%; in the second, the recommendation process is carried out.

To identify the learning profiles, an array of learning characteristics (reflective, intuitive, active, logical, visual, reading, acoustic, prior knowledge, among others) vs students is defined, which stores the score of those characteristics for each student test results. This matrix is used to find student groups. It's used the EM clustering algorithm, this algorithm belongs to a family of models known as Finite Mixture Models [32]. For each profile a selection of LO is done.

The content-based method uses the characterization of Table 1, the characteristics of the course content and the characteristics of each group of students. First it's combined the last two tables and then is used this result to find the LO groups that are close to the profile.

For the collaborative method it's used the characteristics of each group of students and the matrix of LO valued by the students in previous sessions. There are selected the groups of LO most appropriate. For both methods, the cosine similarity algorithm is used.

There are combined the recommendations resulting from each method in order to obtain the final list of LO suggested by profile and students. In this version of the system only the LOs found in both recommendations are selected.

Finally, an attribute is attached to the LOs scored by the students, this attribute measures the usefulness of each LO, it is calculated as the ratio between the numbers of times that has been suggested among the number of suggestions. If the value tends to 1 LO has a high probability of being suggested.

4 Conclusions and Future Work

In this work, it has been presented a mutiagent systems-based design for a Recommender System to be used in an intelligent classroom.

This design aims to obtain learning objects according to the environment and to the knowledge and skills of each student, obtaining customized learning activities that can be used by the teacher for each of the students.

The designed recommender systems based on multiagent system framework is composed of three services: Planning Agent, a Recommendation System and a Learning Object Management System. All this services have to collaborate in order to guarantee the appropriate selection of the customized learning objects.

Acknowledgments. The authors of this paper thank the support given by PUCESI to the project: Intelligent platform in educational environments case: PUCESI Student Office.

References

1. Lops, P., de Gemmis, M., Semeraro, G., Handbook, R.S.: Content-based recommender systems: state of the Art and Trends. In: Ricci, F., Rokach, L., Shapira, B. Kantor, P.B. (eds.) Recommender Systems Handbook, p. 96. Springer, Heidelberg (2011)
2. Virtual Classroom Software: Blackboard Collaborate. http://www.blackboard.com/online-collaborative-learning/
3. Moodle. https://moodle.org
4. Schoology, Learning Management System: LMS. https://www.schoology.com
5. Mahara. https://mahara.org/
6. Mitrovic, A., Suraweera, P., Martin, B., Zakharov, K., Milik, N., Holland, J.: Authoring constraint-based tutors in ASPIRE. In: Ikeda, M., Ashley, K., Chan, T.-W. (eds.) ITS 2006, LNCS, vol. 4053, pp. 41–50 (2006)
7. Litman, D.J., Silliman, S.: Itspoke: an intelligent tutoring spoken dialogue system. In: Proceedings of the Human HLT/NAACL, Boston, MA, May 2004
8. Matsuda, N., Vanlehn, K.: Advanced geometry tutor: an intelligent tutor that teaches proof-writing with construction. In: Looi, C.-K., McCalla, G., Bredeweg, B., Breuker, J. (eds.) Proceedings of the 12th International Conference on Artificial Intelligence in Education AIED 2005, pp. 443–450. IOS Press, Amsterdam (2005)
9. Antona, M., Margetis, G., Ntoa, S., Leonidis, A., Korozi, M., Paparoulis, G., Stephanidis, C.: Ambient Intelligence in the classroom: an augmented school desk. World Acad. Sci. Eng. Technol. **66**, 594–598 (2010)
10. José-Guillermo, H.-C., Benítez-Guerrero, E., Mezura-Godoy, C.: Ambientes inteligentes en contextos educativos: modelo y arquitectura. Res. Comput. Sci. **77**, 55–65 (2014)
11. Arturo, O.D., Alberto, J.J.: Ambiente inteligente distribuido de aprendizaje: integracion de ITS y CLSC por medio de agente pedagogicos. In: Revista de la Escuela de Ingeniería de Antoquia (EIA 2006), pp. 89–104. Antoquia (2006)
12. Rivero, D., Arciniega, S., Narváez, L., Puetate, G.C.: Ambientes Inteligente para la Educación: Un modelo conceptual. In: Avances y aplicaciones de sistemas inteligentes y nuevas tecnologías, pp. 117–130. Universidad de Los Andes, Ecuador (2016)
13. Rabie, A., Hagras, R.H., Nawito, M., El Faham, A.: The intelligent classroom: towards an educational ambient intelligence testbed. In: Sixth International Conference on Intelligent Environments (IE), pp. 344–349. IEEE (2010)
14. Farzan, R., Brusilovsly, P.: Social navigation support in a course recommender system. In: Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, pp. 91–100. Springer, Berlin (2006)

15. Hummel, H., Van den Berg, B., Berlanga, A., Drachsler, H., Jansenn, J., Nadolski, R., Koper, R.: Combining social-based and information-based approaches for personalised recommendation on sequencing learning activities. Int. J. Learn. Technol. **3**(2), 152–168 (2007)
16. García-Salcines, E., Romero-Morales, C., Ventura-Soto, S., Castro-Lozano, C.: Sistema recomendador colaborativo usando minería de datos distribuida para la mejora continua de cursos e-learning. IEEE-RITA **3**(1), 19–30 (2008)
17. Lu, J.: Personalized e-learning material recommender system. In: International Conference on Information Technology for Application, pp. 374–379 (2004)
18. Koper, R.: Increasing learner retention in a simulated learning network using indirect social Interaction. J. Artif. Soc. Soc. Simul. **8**(2) (2005)
19. Janssen, J., den Berg, B., Tattersall, C., Hummel, H., Koper, R.: Navigational support in lifelong learning: enhancing effectiveness through indirect social navigation. Interact. Learn. Environ. **15**(2), 127–136 (2007)
20. Heraud, J.-M., France, L., Mille, A.: Pixed: an ITS that guides students with the help of learners' interaction log. In: International Conference on Intelligent Tutoring Systems, Workshop Analyzing Student Tutor Interaction Logs to Improve Educational Outcomes, Maceio, Brazil, pp. 57–64 (2004)
21. Sørmo, F., Aamodt, A.: Knowledge communication and CBR. In: ECCBR Workshops, pp. 47–60 (2002)
22. Drachsler, H., Hummel, H.G.K., Koper, R.: Personal recommender systems for learners In lifelong learning networks: the requirements, techniques and model. Int. J. Learn. Technol. **3** (4), 404–423 (2008)
23. Ricci, F., Rokach, L., Shapira, B.: Recommender Systems Handbook. Springer, Berlin (2015)
24. Klasna-Milicevic, A., Ivanovic, M., Nanopoulos, A.: Recommender systems in e-learning environments: a survey of the state-of-the-art and possible extensions. Artif. Intell. Rev. **44** (4), 571–604 (2015)
25. Russell, S., Norvig, P.: Artificial Intelligence: A Modern Approach. Prentice-Hall, Egnlewood Cliffs 25 (1995)
26. IEEE: Learning Object Metadata. http://ltsc.ieee.org/wg12/
27. L'Allier, J.: Frame of Reference: NETg's Map to Its Products, Their Structures and Core Beliefs. http://web.archive.org/web/20020615192443/www.netg.com/research
28. ADL. http://www.adlnet.org/
29. Eriksson, H.-E., Penker, M.: Business Modeling with UML, New York (2000)
30. Rivero-Albarrán, D., Barrios-Albornóz, J., Rivas-Echeverría, F.: An evolutionary strategy for transforming the Business Object Model in a Web Services Model. Int. J. Internet Things Web Serv. **1**(1), 8–21 (2016).
31. Cohen, W.W.: Integration of heterogeneous databases without common domains using queries based on textual similarity. In: Proceedings of the SIGMOD International Conference Management of Data SIGMOD 1998, pp. 201–212 (1998)
32. Garre, M., Charro, M.: Estimación del esfuerzo de un proyecto software utilizando el criterio MDLEM y componentes normales NDimensionales. Aplicación a un caso práctico. In: Revista de Procesos y Métricas de las Tecnologías de la Información (RPM), 13-24 March 2005, vol. 2, no. 1. Asociación Española de Sistemas de Informáticos (AEMES) (2005). ISSN: 1698 2029